Tight Bounds for Keyed Sponges and Truncated CBC

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

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

Abstract. We prove (nearly) *tight* bounds on the concrete PRF-security of two constructions of message-authentication codes (MACs):

(1) The truncated CBC-MAC construction, which operates as plain CBC-MAC (*without* prefix-free encoding of messages), but only returns a subset of the output bits.

(2) The MAC derived from the sponge hash-function family by pre-pending a key to the message, which is the de-facto standard method for SHA-3-based message authentication.

The tight analysis of keyed sponges is our main result and we see this as an important step in validating SHA-3-based authentication before its deployment. Still, our analysis crucially relies on the one for truncated CBC as an intermediate step of independent interest. Indeed, no previous security analysis of truncated CBC was known, whereas only significantly weaker bounds have been proved for keyed sponges following different approaches.

Our 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 $q \geqslant \ell$ queries. Our proofs rely on a novel application of Patarin's H-coefficient method to iterated MAC constructions.

Keywords. Message-authentication, sponges, CBC-MAC, H-coefficient method, concrete security.

1 Introduction

Message-authentication codes. Message authentication codes (or MACs, for short) are central components of secure communication protocols like TLS. Secure MACs are required to be *unpredictable*, meaning that it is hard for an attacker to predict the MAC output (usually called the *tag*) under a secret key on a message, even given the tags of a number of (different) messages.

Practical MAC constructions have been based on block ciphers – like *cipher block chaining* (CBC) and its variants (first analyzed in [4]) – and on hash functions – typically, using HMAC [3]. Security analyses often show that MAC constructions achieve a property even stronger than unpredictability, namely that of being a *pseudorandom function* (PRF) [22], i.e., the outputs under a secret key are indistinguishable from random, except with a (small) distinguishing gap ε.

Contributions, in a nutshell. This paper studies the concrete PRF security of MAC constructions (i.e., how small can ε be?) by solving two technically connected open problems whose solutions require techniques that are substantially different from those in previous MAC analyses: *First*, we prove bounds for the concrete security of a variant of CBC-mode – *truncated* CBC – which remained unanalyzed to date. *Second*, we show improved bounds for the security of MAC constructions based on the sponge hash-function construction [11] which underlies the SHA-3 standard [1].

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 most envisioned application scenarios.[3] The following paragraphs will elaborate on our contributions in detail.

CBC-MACs. The *cipher block-chaining* mode (or CBC, for short) is arguably the most natural block-cipher based MAC. It has been standardized already three decades ago [15,28], and its security was first analyzed by Bellare, Kilian, and Rogaway [4]. In its *basic* form, CBC is very simple: Given a block cipher E with n-bit block size, an input $M \in \{0,1\}^*$ is 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

$$Y_0 \leftarrow \mathsf{IV}\ , \quad Y_i \leftarrow E_K(Y_{i-1} \oplus M[i])\ , \tag{1}$$

where IV is an appropriate initialization value, e.g., $\mathsf{IV} = 0^n$. Unfortunately, the basic CBC construction is only secure for messages of *equal* length ℓ, as proved in [4]. Otherwise, one can easily mount an extension attack, i.e., obtaining $\mathsf{CBC}_K(M) = Y$, and $\mathsf{CBC}_K(Y \oplus \mathsf{IV} \oplus M') = Y'$ (for n-bit values M, M') reveals us that $\mathsf{CBC}_K(M \,\|\, M') = Y'$ *without* having to query $M \,\|\, M'$.

Three (variants of) solutions have been considered to prevent extension attacks: The first one is to consider prefix-free encoding of messages [36]. The second, known as *encrypted CBC*, 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 [39]). 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 [4,36,40,29,25,5,7,37,34,32], we are not aware of any formal analysis of truncated CBC having ever been published, let alone a tight one, even though truncation did appear in standards [15,28], mostly to increase flexibility.

Hash-function MACs. It appears simpler to derive hash-based MACs: It is not hard to prove that $\mathsf{MAC}_K(M) = \mathsf{H}(K \,\|\, M)$ is a secure MAC (and PRF) for an *ideal* hash function $\mathsf{H} : \{0,1\}^* \to \{0,1\}^n$,

[3] In particular, for $r = 64$, the restriction induced by 2^r blocks still accommodates messages of length up to (and beyond) 1.47×10^8 TB – around 10 times the predicted storage capacity of NSA's "Utah Data Center". For SHA-3, we even have $r \geqslant 224$ and $n = 1600$.

i.e., which behaves as a random oracle [6]. Unfortunately, legacy hash functions like MD5, SHA-1, and SHA-256 based on the Merkle-Damgård construction [17,31] are far from ideal. In particular, they allow for extension attacks – given $\mathsf{H}(K \parallel M)$ (and without knowing K and M), one can compute $\mathsf{H}(K \parallel M \parallel M')$ for any M'. HMAC [3] was the first hash-function based MAC construction preventing such extension attacks, and has been the object of several security analyses [2,26,18,21].

Sponges. In sharp contrast with legacy hash functions, the *sponge* construction [11] was designed with the goal of behaving as a random oracle (in the sense of indifferentiability [30]). The construction relies on an invertible permutation π on n-bit strings.[4] For a parameter $b < n$, it then pads the message M into b-bit blocks $M[1], \ldots, M[\ell]$, and keeps a state $S_i \parallel T_i$, where $S_i \in \{0,1\}^b$ and $T_i \in \{0,1\}^{n-b}$, and outputs $S_\ell[1..r]$ for some parameter[5] $r \leqslant b$ after the following iteration:

$$S_0 \parallel T_0 \leftarrow 0^n \ , \quad S_i \parallel T_i \leftarrow \pi((S_{i-1} \oplus M[i]) \parallel T_{i-1}) \ .$$

A variant of KECCAK [10], using the sponge paradigm, was selected as the new hash function standard SHA-3 [1] by NIST.

In view of the lack of extension attacks, it is suggested (e.g. in [11]) that sponge-based message-authentication should simply occur by prepending the key to the message, with no need of using the HMAC construction. This mode was analyzed in [8] and a similar bound can also be inferred from the indifferentiability analysis of the sponge construction [11]. However, as we show below, these bounds are far from tight and are substantially improved by our work.

Our contributions, in detail. We present two technically related results:

Security of truncated CBC. 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, i.e., a function returning random outputs for each distinct message, except with distinguishing gap

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

This in turn implies security when the random permutation is replaced by a secure block cipher which is a good PRP. The first term matches the one from the best known analysis of prefix-free CBC [5]. Moreover, we show that the second term is *tight* for $q \geqslant \ell$, i.e., we exhibit a generic distinguishing attack with advantage $\Omega(\frac{q^2}{2^{n-r}})$.

Security of sponge-based MAC. We prove that no attacker making q_C queries of length at most $\ell < 2^{n/4}$ to the keyed Sponge construction using a *random* permutation π, and q_π queries to π itself, can distinguish it from a random function, except with distinguishing gap roughly

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

for sufficient key length. This model – where π is ideal – is the traditional model for studying sponge-like constructions and their security against generic attacks. The previously best known bound for sponge-based authentication in this model [8] was dominated by a term of much larger magnitude $O(\ell^2 q_C^2/2^{n-r})$. We also show tightness of the second term for $\ell < q_\pi$ via an attack achieving distinguishing advantage $\Omega\left(\frac{q_C \cdot q_\pi}{2^{n-r}}\right)$.

[4] Naming consistency with the TCBC setting forces us to deviate from the usual naming in the literature on sponges, where our parameters n, b, r are usually denoted b, r, d; respectively. Hopefully this does not cause any confusion.

[5] The sponge paradigm also allows for output 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.

We stress that the salient feature of these bounds is that the dependence on the length ℓ only affects terms with denominator 2^n, or appears in *linear* terms $\ell q/2^{n-r}$ (where here and below, for sponges q naturally represents q_C). This makes our bounds tight, as long as $\ell \leqslant \min\{2^r, 2^{n/4}\}$ and $q \geqslant \ell$ – which is a very common scenario. We leave the question of proving tightness of the remaining terms (or, alternatively, of improving our bounds) as an open problem which we believe to be quite challenging.

The truncated CBC result makes it evident that security requires either r fairly small (this is the case when using AES with $n = 128$), or a restriction on the maximum number of queries q, or the usage of a block cipher with larger block size n, such as Rijndael.[6] A small r is acceptable in settings where we use TCBC to obtain pseudorandom bits, or where it is used as a MAC, but only security against few ($< 2^r$) verification queries is needed. Either way, this is by far not an issue in the setting of sponges, where n is usually much larger than r. (For example, SHA-3-224 has $r = 224$, $b = 1152$ and $n = 1600$.) In fact, our results show that the parameters used in SHA-3 are more than generous for usage as a MAC, and setting e.g. $r = 64$ and $n = 192$ would already imply comfortable levels of security against generic attacks.

Another interesting consequence of our results is that *with respect to pseudorandomness (and MAC)* security, we are not constrained to any block length $b < n$ when evaluating the sponge construction – we could well XOR n-bit message blocks to the *whole* state. Indeed, our proof considers this generalized variant that pads the message into n-bit blocks that are XORed to the state during the absorption phase; this highlights the connection to the TCBC construction. Shorter block lengths can then be enforced by the padding function setting some of the bits to be 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 $\mathsf{donkeySponge}$ [9], and our result implicitly covers this construction as well.

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.[7] 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 the adversary from issuing a later query M', 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 [35]: In this framework, one fixes 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 compares the probabilities that such a transcript would indeed occur with $\mathcal{A}$ in the real and

[6] Also note that the domain of a block cipher can be extended using e.g. EME [23], or even better suited to preserving tightness, recent beyond-birthday secure constructions by Shrimpton and Terashima [38].

[7] This notwithstanding, proving bounds on the collision probability is far from trivial [5,37].

in the ideal worlds, 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, and far from trivial. While an upper bound on δ follows by using techniques from [5,37], upper-bounding ε will require substantially new techniques.

Our security proof for sponges is very similar, and will essentially rely on the argument that with good probability (roughly $\ell q_{\pi} q/2^n$), 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 significantly more involved. We also give a security proof for this more complex setting using techniques inspired by [14]. Although our analysis assumes that the (padded) keys and the actual message occupy separate blocks, our results can be extended to the completely general case at the cost of additional notational overhead.

Comparison with previous results on sponges. As already mentioned, the work [8] gives a bound for PRF-security of the sponge construction in the random permutation model. Their bound is dominated by a term which (with respect to our naming conventions) is roughly $O(\ell^2 q^2/2^{n-r})$, significantly worse than the terms $O(q(q+\ell)/2^{n-r})$ and $O(\ell q^2/2^n)$ from our 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 $\mathsf{GSponge}$ below) to the security of TCBC for a random permutation *and* the PRP security against ℓq queries of a carefully crafted block cipher E^{π}. The latter is built from the permutation π inside the sponge construction as $E^{\pi}_K(X) = (0^b \,\|\, K) \oplus \pi(X \oplus (0^b \,\|\, K))$ for $X \in \{0,1\}^n$ and $K \in \{0,1\}^{n-b}$, where b is the block length. This construction is essentially a low-entropy single-key version of the Even-Mansour cipher [20,19], and one can apply the same analysis (with a lower-entropy key) in the setting where the attacker makes q_{π} queries to π, this results in an additive term of $O(\ell q q_{\pi}/2^{n-b})$. In contrast, our analysis only incurs into an extra term of $O(\ell q q_{\pi}/2^n)$.

More on truncation. There is a folklore belief that given a secure MAC, truncating its output may actually *increase* its security by hindering collision detection. This has never been verified formally, and providing an answer is an interesting open question. Nonetheless, our $\Omega(q^2/2^{n-r})$ lower bound does not contradict this belief, as we are applying truncation to a construction which (by itself, without truncation) is not a secure MAC, as our attacks query for non-prefix free messages.

2 Preliminaries

Basic notation. 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 $S[i \dots j]$, for every $1 \leqslant i \leqslant j \leqslant n$, the sub-sequence consisting of $S[i], S[i+1], \dots, S[j]$, with the convention that $S[i \dots i] = S[i]$. Moreover, we denote by $S \parallel S'$ the concatenation of two sequences in $\mathcal{S}^*$, and also, we let $S \mid T$ be the usual prefix-of relation: $S \mid T :\Leftrightarrow (\exists S' \in \mathcal{S}^* : S \parallel S' = T)$.

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 and permutations. 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 going to consider the security of F as a *pseudorandom function* (or PRF, for short) [22]. This is defined via the following advantage measure, involving an adversary $\mathcal{A}$, such that

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

Similarly, a block cipher is a keyed function $\mathsf{E} : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ such that $\mathsf{E}_K \in \mathsf{Perm}(n)$, i.e., it is a permutation, for all κ-bit K. The traditional security of E is that of being a *pseudorandom permutation* (or PRP, for short), defined via the advantage measure

$$\mathsf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(\mathcal{A}) = \left|\mathsf{Pr}\left[K \xleftarrow{\$} \{0,1\}^\kappa : \mathcal{A}^{\mathsf{E}_K} \Rightarrow 1\right] - \mathsf{Pr}\left[\pi \xleftarrow{\$} \mathsf{Perm}(n) : \mathcal{A}^{\pi} \Rightarrow 1\right]\right| .$$

Informally, we say that F is a PRF, or E is a PRP, if the corresponding advantage is "negligible" for all "efficient" $\mathcal{A}$'s.

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). Moreover, we can consider the natural instantiation of π via a block cipher E, and denote by $\mathsf{C}[\mathsf{E}]$ the function which, for key K and input M, returns $\mathsf{C}[\mathsf{E}_K](M)$. Then, the following relates the prf advantages for $\mathsf{C}[\mathsf{E}]$ and for C.

Proposition 1. *For every adversary $\mathcal{A}$ with running time t and making q queries to its oracle, where each query results in at most ℓ invocations of the underlying π when input to $\mathsf{C}[\pi]$, there exists an adversary $\mathcal{B}$ such that*

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{C}[\mathsf{E}]}(\mathcal{A}) \leqslant \mathsf{Adv}^{\mathsf{prp}}_{\mathsf{E}}(\mathcal{B}) + \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{C}}(\mathcal{A}) ,$$

where the adversary $\mathcal{B}$ makes $q \cdot \ell$ queries, and runs in time $t + \tilde{O}(q \cdot \ell)$.

In other words, if we assume that E is a good PRP (for example, E is AES), then we can focus on upper bounding the distinguishing advantage when C is instantiated with a randomly chosen permutation, which is a truly information-theoretic problem.

Pseudorandom functions in the ideal permutation model. For our analysis of sponges below, we are going to consider constructions F^π which make queries to a randomly chosen permutation $\pi \xleftarrow{\$} \mathsf{Perm}(n)$ which can be evaluated by the adversary in both directions. For this case, we use the following notation to express the PRF advantage of $\mathcal{A}$:

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

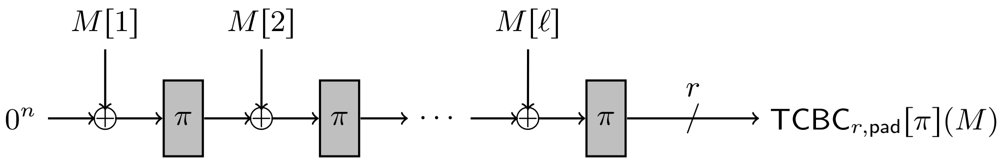

Fig. 1. Truncated CBC. Representation of $\mathsf{TCBC}_{r,\mathsf{pad}}[\pi]$. Here, $M[1], \ldots, M[\ell]$ are n-bit blocks resulting from applying the padding scheme pad to the input message $M \in \{0,1\}^*$.

MACs and Unpredictability. It is appropriate to note that our actual target is that of a message-authentication code (MAC). The requirement on a keyed function $\mathsf{F} : \{0,1\}^\kappa \times \{0,1\}^* \to \{0,1\}^r$ to be a secure MAC is that of *unpredictability under a chosen-message attack*, i.e., an attacker $\mathcal{A}$, given adaptive access to $\mathsf{F}_K(\cdot)$, cannot output a *valid* pair (M, τ) such that $\mathsf{F}_K(M) = \tau$ and M was not queried to F_K. We note that if $\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{F}}(\mathcal{A}) \leqslant \varepsilon$ for any time t attacker $\mathcal{A}$ making $q + q_V$ queries, then no time t attacker making q queries to $\mathsf{F}_K(\cdot)$ can output such a valid pair (M, τ) within q_V attempts, except with probability at most $\varepsilon + q_V/2^r$.

3 Truncated CBC and its Security

This first part of the paper deals with the concrete security of truncated CBC, which we first review.

Truncated CBC. We fix 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 stress that we are *not* requiring the padding to be prefix-free. The canonical padding scheme 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. In particular, a message M is encoded into $\ell = \lceil \frac{|M|+1}{n} \rceil$ n-bit blocks.

We first introduce the CBC construction for padding scheme pad, based on $\pi \in \mathsf{Perm}(n)$:

```
Construction CBC_{r,pad}[π](M):                          // $M \in \{0,1\}^*$

(1) Compute $\mathsf{pad}(M) = M[1] \ldots M[\ell]$ (for some $\ell$).
(2) $S_0 \leftarrow \mathsf{IV}$. For all $i \in [\ell]$, compute $S_i \leftarrow \pi(M[i] \oplus S_{i-1})$.
(3) Output $S_\ell$
```

Then, for any $\pi \in \mathsf{Perm}(n)$, truncated CBC (or TCBC, for short) behaves as follows (also cf. Figure 1 for a pictorial representation) on input $M \in \{0,1\}^*$,

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

i.e., it outputs the first $r < n$ bits of $\mathsf{CBC}_{\mathsf{pad}}[\pi](M)$.

Security analysis. We prove the following theorem about the security of the TCBC construction in the case where π is randomly sampled from $\mathsf{Perm}(n)$. By Proposition 1, this in particular implies security when the permutation is instantiated with a block cipher which is a good PRP.

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 randomly sampled permutation $\pi \in \mathsf{Perm}(n)$. Then, for any $t \geqslant 1$,*

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{TCBC}}(\mathcal{A}) \leqslant (6t + 17)\,\frac{\ell q^2}{2^n} + \frac{7n \cdot q^2}{2^{n-r}} + \frac{8q\ell}{2^{n-r}} + \frac{2q}{2^n} + \frac{136\ell^4 q^2}{2^{2n}} + \frac{2q^{t+1}\ell^{t+1}}{2^{nt}} \ . \qquad (2)$$

The proof of Theorem 1 is found below in Section 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. The above bound requires some discussion. 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 crucial ones, with the remaining terms being high order terms: 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 $t \geqslant 4$.

We now show that this bound is essentially tight for the case where $\ell < 2^r$ and $q \geqslant \ell$. Indeed, we 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.

Matching attack. For a parameter Q and $t := \lceil n/r \rceil$, the attacker $\mathcal{A}_{Q,t}$ proceeds as follows, given access to an oracle O, which we assume without loss of generality takes inputs $M \in (\{0,1\}^n)^+$.[8]

```
Adversary $\mathcal{A}_{Q,t}$:

1. Query random $M_i \in \{0,1\}^n$ for all $i \in [Q]$ to $\mathsf{O}$, obtaining output $Y_i \in \{0,1\}^r$.
2. For all $i \in [Q]$ and $j \in [t]$, query $\mathsf{O}(M_i \parallel Y_i \parallel 0^{n-r} \parallel 0^{n(j-1)})$, obtaining values $Y_{i,j}$.
3. If there exist distinct $i, i' \in [Q]$ with $Y_{i,j} = Y_{i',j}$ for all $j \in [t]$ then output 1, else
   output 0.
```

Note that the attacker makes $q = (t+1)Q$ queries. We are going to show that

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{TCBC}}(\mathcal{A}) \geqslant \Omega\left(\frac{r^2 q^2}{n^2 2^{n-r}}\right) ,$$

independently of how the permutation π used by TCBC is instantiated.

Analysis. We first analyze what happens in the real world when $\mathsf{O} = \mathsf{TCBC}^\pi$ for some permutation $\pi \in \mathsf{Perm}(n)$. Let COLL be the event that for some M_i and $M_{i'}$, we have $\pi(M_i)[r+1 \ldots n] = \pi(M_{i'})[r+1 \ldots n]$. Note that since the messages are chosen uniformly at random and independently, by the Birthday bound we have $\mathsf{Pr}\left[\mathsf{COLL}\right] = \Omega(Q^2/2^{n-r})$. Moreover, given COLL occurs due to message M_i and $M_{i'}$, then by construction $Y_{i,j} = Y_{i',j}$ for all $j \in [t]$. Therefore,

$$\mathsf{Pr}\left[\mathcal{A}^{\mathsf{TCBC}^\pi} \Rightarrow 1\right] \geqslant \mathsf{Pr}\left[\mathsf{COLL}\right] = \Omega(Q^2/2^{n-r}) .$$

However, if $\mathsf{O} = \mathsf{R}$ for a truly random function $\mathsf{R} : \{0,1\}^* \to \{0,1\}^r$ then, unless there is a collision among the values M_i (which occurs with probability $O(Q^2 2^{-n})$), all values $Y_{i,j}$'s are independent random r-bit strings, and thus the probability that there are suitable i and i' is at most (again, by the Birthday bound) $Q^2 2^{-rt} \leqslant Q^2 2^{-n}$. Altogether, this gives us $\mathsf{Pr}\left[\mathcal{A}^{\mathsf{R}} \Rightarrow 1\right] \leqslant 2Q^2 2^{-n}$, for which the advantage bound follows.

Forging attack. Note that it is very easy to turn the above attack into a forging attack. Indeed, given access to TCBC^π, once we have found appropriate collisions $Y_{i,j} = Y_{i',j}$ for all $j \in [t]$, it is very easy to create a forgery, since $M_i \parallel Y_i \parallel 0^{n-r} \parallel 0^{nt}$ and $M_{i'} \parallel Y_{i'} \parallel 0^{n-r} \parallel 0^{nt}$ are also colliding – we can forge a tag of the latter by learning the tag of the former.

4 Proof of Theorem 1

We will start with the high level overview of the proof of Theorem 1, which relies on Patarin's H-coefficient technique [35], for which we give a self-contained introduction. In particular, Section 4.1

[8] Should we only be able to query TCBC via padded inputs, it is not hard to relabel messages in the attack to obtain an equivalent attack.

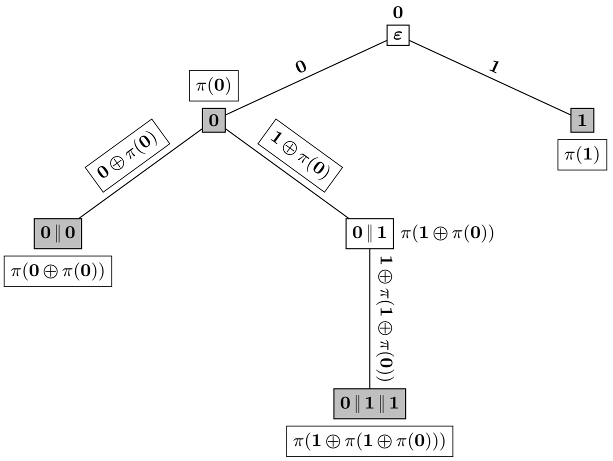
$\mathbf{0}$
ε
$\mathbf{0}$
$\mathbf{1}$
$\pi(\mathbf{0})$
$\mathbf{0}$
$\mathbf{1}$
$\pi(\mathbf{1})$
$\mathbf{0} \oplus \pi(\mathbf{0})$
$\mathbf{1} \oplus \pi(\mathbf{0})$
$\mathbf{0} \,\|\, \mathbf{0}$
$\pi(\mathbf{0} \oplus \pi(\mathbf{0}))$
$\mathbf{0} \,\|\, \mathbf{1}$
$\pi(\mathbf{1} \oplus \pi(\mathbf{0}))$
$\mathbf{1} \oplus \pi(\mathbf{1} \oplus \pi(\mathbf{0}))$
$\mathbf{0} \,\|\, \mathbf{1} \,\|\, \mathbf{1}$
$\pi(\mathbf{1} \oplus \pi(\mathbf{1} \oplus \pi(\mathbf{0})))$

Fig. 2. (Reduced) message tree. Example of 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 $\lambda(\varepsilon) = \mathbf{0} = \mathsf{IV}$. The *boxed* labels are omitted in the reduced message tree.

first introduces the notational framework allowing us to precisely describe interactions between $\mathcal{A}$ and the given system – i.e., either $\mathsf{TCBC}[\pi]$ for a randomly chosen permutation $\pi \overset{\$}{\leftarrow} \mathsf{Perm}(n)$ or a truly random function $f \overset{\$}{\leftarrow} \mathsf{Fcs}(*, n)$. Then, Section 4.2 will review the H-coefficient method, and how it will be applied to the concrete case of our proof. Finally, Section 4.3 will state the main concrete bounds we prove and how they are combined into our main theorem.

Simplifying assumption. Throughout the proof, we fix an adversary $\mathcal{A}$. 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 an arbitrary (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.

4.1 Message trees and transcripts

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 $\mathcal{A}$ in its interaction: This information will be useful for us to make the proof simpler, and it will not help substantially in distinguishing TCBC from random.

The message tree. Let $q \geqslant 1$, $\pi \in \mathsf{Perm}(n)$, and let $M_1, \ldots, M_q$ be arbitrary strings (which we refer to as the *messages*), where M_i is padded into n-bit blocks by the pad function. In particular, we will identify M_i with its padded version, i.e., assume that $M_i \in (\{0, 1\}^n)^+$, without loss of generality. 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 – which we define as follows:

- The set V of vertices of the tree is $V := \{M' \in (\{0,1\}^n)^* : \exists i \in [q] : M' \mid M_i\}$, where $\mid$ is the prefix-of partial ordering of strings. In particular, note that the empty string ε is a vertex.
- The set $E \subseteq V \times V$ of (directed) 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 $\lambda(\varepsilon) = \mathsf{IV}$. 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 Figure 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. We also remark that the labeling of the edges is redundant given the vertex labels as from the vertex-labels and V, it is possible to uniquely reconstruct the edge labels. However, defining the edge labels explicitly will be convenient for the proof.

For convenience, we define for every vertex $M \in V$ (where possibly $M \notin \{M_1, \ldots, M_q\}$) the set $\mathcal{M}_M$ of n-bit blocks m such that $(M, M \,\|\, m) \in E$ and we let $D_M = |\mathcal{M}_M|$ 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 $i \in [q]$. Note 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 messages M_j such that $M_i \,\|\, m$ is a prefix of M_j.

The reduced message tree. We define an abridged version of the above tree, called the *reduced* message tree, which will be used in the definition of transcripts below, and which we denote by $\overline{T}^{\pi}(M_1, \ldots, M_q)$. The intuition is that in an interaction with $\mathsf{TCBC}[\pi]$, even given the *reduced* message tree, the outputs obtained by the adversary will look random and independent of the tree labels. This is far from simple to prove, and will be one of our main steps below.

To compute the reduced message tree, we first compute the whole message tree $T^{\pi}(M_1, \ldots, M_q) = (V, E, \lambda, \gamma)$ with resulting labels λ and γ, and we are going to check whether the following event has occurred (this will correspond to a degenerate labeling case that we will show to be quite unlikely):

- 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 $\lambda(M_i \,\|\, m) = \lambda(M)$.

If so, then we let $\overline{T} = (V, E, \bot, \bot)$, i.e., we set the tree to have the empty labeling function. Note that this corresponds to the case where a label of an actual message in $\{M_1, \ldots, M_q\}$, or of one of its sucessor vertices, collide with some other labels.

Otherwise, if the above event does not occur, 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 $i \in [q]$.[9]

In general, we are allowing labels of vertices to possibly collide with each other. The first check however, potentially setting $(\lambda, \gamma) = (\bot, \bot)$, ensures that no "bad collisions" have occurred, i.e., no labels of actual messages (or their children vertices) collide with labels of other vertices, and this will be instrumental below.

[9] 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 will implicitly see below, this information is irrelevant.

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. (Recall that 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..) In particular, 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 $\mathcal{A}$-valid, and $\pi : \{0,1\}^n \to \{0,1\}^n$ is a permutation. 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, $\mathsf{T}_{\mathsf{real}}$ 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, but here we sample both a random permutation π and q independent random values $Y_1, \ldots, Y_q \in \{0,1\}^r$

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

where we execute $\mathcal{A}$, which asks queries $M_1, \ldots, M_q$ answered with Y_i for all $i \in [q]$, and we let $\overline{T} = \overline{T}^{\pi}(M_1, \ldots, M_q)$. We stress that here we are *augmenting* the ideal world with an additional random permutation π which does not actually exists in the original prf distinguishing game in order to make real- and ideal-world transcripts alike. Like the actual permutation π, the resulting (reduced) message tree is completely independent of the randomness $Y_1, \ldots, Y_q$ used to reply the adversary's queries.

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.

4.2 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 in terms of the statistical distance of the transcripts, i.e.,

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

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 [35]. This just 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, whose proof is given for completeness in Appendix A.

Lemma 1 (The H-Coefficient Method [35]). *Let $\delta, \varepsilon \in [0,1]$ be such that:*

(a) $\mathsf{Pr}[\mathsf{T}_{\mathsf{ideal}} \in \mathsf{BT}] \leqslant \delta$.

(b) *For all $\tau \in \mathsf{GT}$,*

$$\frac{\Pr[\mathsf{T}_{\mathsf{real}} = \tau]}{\Pr[\mathsf{T}_{\mathsf{ideal}} = \tau]} \geqslant 1 - \varepsilon \ .$$

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 want a set of good transcripts GT such that with very high probability (i.e., $1 - \delta$) a generated transcript *in the ideal world* is going to be in this set, 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 $\tau = ((M_1, Y_1), \ldots, (M_q, Y_q), \overline{T} = (V, E, \gamma, \lambda))$ will be defined as "good" if the associated reduced message tree is not "too degenerate". This requires introducing two relevant quantities. To this end, we first note that $\overline{T}$ defines a partial permutation $\overline{\pi}$: Concretely, we define $\overline{\pi}$ such that $\overline{\pi}(\gamma(e)) = \lambda(v)$ for every edge e with end-node v such that $\gamma(e), \lambda(v) \neq \bot$, and $\overline{\pi}(x) = \bot$ 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, we define:*

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

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

Moreover, for $a \in \{1, 2\}$, let $N^{(a)} = \sum_{i=1}^{q} N_i^{(a)}$.

Let us give some intuition on the above quantities. Note that $\overline{\pi}$ is defined on at most $q \cdot \ell$ values, and the values $\overline{\pi}(x)$, 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-$\bot$) 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 are now ready to state the definition of a good transcript. Informally, what we require is that the actual values of $N^{(1)}$ and $N^{(2)}$ for the transcript τ 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 $(\lambda, \gamma) = (\bot, \bot)$ by the definition of the reduced message tree.

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

(1) $(\lambda, \gamma) \neq (\bot, \bot)$.

(2) $N^{(1)} \leqslant 3q\left(qt\ell/2^r + n\right)$.
(3) $N^{(2)} \leqslant (2n+1)q^2 + (3t+1)q^2\ell/2^r + 8q^2\ell^4/2^{n+r}$.

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 $\mathsf{BT} = \bigcup_{i=1}^{3} \mathsf{BT}_i$.

4.3 High-level lemmas and putting pieces together

Bounding the ratio. In Section 4.4 below, we are going to prove the following lemma.

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

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

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. The first is proved in Appendix C, and the last two are proved in Appendix D.

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

Lemma 4 (Bad-Transcript Analysis for BT_2). *For all $t \geqslant 1$,*

$$\mathsf{Pr}\left[N^{(1)}(\mathsf{T}_{\mathsf{ideal}}) \geqslant 3q\left(qt\ell/2^r + n\right)\right] \leqslant \frac{q}{2^n} + \frac{(q\cdot\ell)^{t+1}}{2^{nt}} .$$

Lemma 5 (Bad-Transcript Analysis for BT_3). *For all $t \geqslant 1$,*

$$\mathsf{Pr}\left[N^{(2)}(\mathsf{T}_{\mathsf{ideal}}) \geqslant (2n+1)q^2 + (3t+1)q^2\ell/2^r + 8q^2\ell^4/2^{n+r}\right] \leqslant \frac{q}{2^n} + \frac{8q\ell}{2^{n-r}} + \frac{(q\cdot\ell)^{t+1}}{2^{nt}} .$$

The proof of Lemma 3 above uses and extends techniques inherited from the work of [5] 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 Lemma 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,

$$\varepsilon = \frac{N^{(1)} + N^{(2)}}{2^{n-r}} + \frac{2q^2}{2^{n-r}} \leqslant \frac{(6t+1)\ell q^2}{2^n} + \frac{7nq^2}{2^{n-r}} + \frac{8q^2\ell^4}{2^{2n}} ,$$

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

$$\varepsilon + \delta \leqslant (6t+17)\frac{\ell q^2}{2^n} + \frac{7n\cdot q^2}{2^{n-r}} + \frac{8q\ell}{2^{n-r}} + \frac{2q}{2^n} + \frac{136\ell^4 q^2}{2^{2n}} + \frac{2q^{t+1}\ell^{t+1}}{2^{nt}} .$$

4.4 Lower-bounding the probability ratio (Proof of Lemma 2)

Recall that we need to lower bound $\mathsf{Pr}\left[\mathsf{T}_{\mathsf{real}} = \tau\right] / \mathsf{Pr}\left[\mathsf{T}_{\mathsf{ideal}} = \tau\right]$ for some $\tau = ((M_1, Y_1), \ldots, (M_q, Y_q), \overline{T}) \in \mathsf{GT}$, which we fix from now on. 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, if $\tau = ((M_1, Y_1), \ldots, (M_q, Y_q), \overline{T})$, let $\Omega'[\tau]$ be the set of permutations π which are consistent with the labels of the reduced message tree $\overline{T}$ (i.e., reducing $T^{\pi}(M_1, \ldots, M_q)$ yields $\overline{T}$), 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

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

and this will be a convenient quantity to work with. In particular, $\overline{p}(\tau)$ is the probability that when sampling a random permutation π which is consistent with the constraints on the *reduced* message tree, we also have $\mathsf{TCBC}^{\pi}(M_i) = Y_i$ for all $i \in [q]$.[10] The following claim will reduce computing the probability ratio to computing $\overline{p}(\tau)$ for $\tau \in \mathsf{GT}$.

Claim. For all good transcripts $\tau \in \mathsf{GT}$,

$$\frac{\mathsf{Pr}\left[\mathsf{T}_{\mathsf{real}} = \tau\right]}{\mathsf{Pr}\left[\mathsf{T}_{\mathsf{ideal}} = \tau\right]} = 2^{r \cdot q} \cdot \overline{p}(\tau) \ . \tag{6}$$

Proof (Of Claim). We first note that $\mathsf{Pr}\left[\mathsf{T}_{\mathsf{ideal}} = \tau\right]$ can be rewritten as the probability that a randomly sampled permutation $\pi \xleftarrow{\$} \mathsf{Perm}(n)$ satisfies $\pi \in \Omega'[\tau]$, and independently $Y_1, \ldots, Y_q$ are the selected outputs, i.e.,

$$\mathsf{Pr}\left[\mathsf{T}_{\mathsf{ideal}} = \tau\right] = 2^{-r \cdot q} \cdot \mathsf{Pr}\left[\pi \xleftarrow{\$} \mathsf{Perm}(n) :\ \pi \in \Omega'[\tau]\right] \ ,$$

whereas

$$\mathsf{Pr}\left[\mathsf{T}_{\mathsf{real}} = \tau\right] = \mathsf{Pr}\left[\pi \xleftarrow{\$} \mathsf{Perm}(n) : \mathsf{T}_{\mathsf{real}}[\pi] = \tau\right] = \mathsf{Pr}\left[\pi \xleftarrow{\$} \mathsf{Perm}(n) : \pi \in \Omega'[\tau]\right] \cdot \overline{p}(\tau) \ ,$$

and the claim follows by dividing both probabilities. □

We are going to lower bound $\overline{p}(\tau) \geqslant (1 - \varepsilon)2^{-rq}$ for ε as in the statement of the lemma, which clearly implies the lemma by the above claim. It is easy to see that the ordering of the message-output pairs $(M_1, Y_1), \ldots, (M_q, Y_q)$ is irrelevant, and we therefore assume without loss of generality that the ordering is prefix-preserving, i.e., if $M_i \mid M_j$, then $i < j$.

We consider an *iterative process* where we start with $\overline{\pi}$ defined by $\overline{T}$ as above, and then set the values of $\overline{\pi}(\gamma(e_i)) = \lambda(M_i)$ for $i = 1, \ldots, q$ one after the other in this order. Moreover, when 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 $\gamma(M_i, M_i \,\|\, m) \leftarrow Z_i \oplus m$

[10] 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 $\overline{T}$.

- If we know the value $\lambda(M_i \,\|\, m)$, we set $\overline{\pi}(Z_i \oplus m) \leftarrow \lambda(M_i \,\|\, m)$.[11]

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. In particular, we consider the random experiment where we sample $\pi \overset{\$}{\leftarrow} \Omega'[\tau]$ (which is in particular consistent with the initial $\overline{\pi}$), and then set $Z_i \leftarrow \pi(\gamma(e_i))$. Then,

$$\overline{p}(\tau) = \mathsf{Pr}\left[\pi \overset{\$}{\leftarrow} \Omega'[\tau] : \forall i \in [q] : Z_i[1\ldots r] = Y_i\right] = \sum_{\substack{(z_1,\ldots,z_q) \\ z_i[1\ldots r]=Y_i}} \mathsf{Pr}\left[\pi \overset{\$}{\leftarrow} \Omega'[\tau] : \forall i \in [q] : Z_i = z_i\right] .$$

Let $\mathcal{L} = \mathcal{L}(\overline{T}, (M_1, Y_1), \ldots, (M_q, Y_q)))$ be the set of possible 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. (Note that since the transcript is good, and $(\lambda, \gamma) \neq (\perp, \perp)$, these are exactly the sequences which are possible, even though an exact match is really not necessary for a lower bound.) Below we are going to show that $|\mathcal{L}|$ is sufficiently large, and hence not empty. For now, we observe the following claim, which will allow us to lower bound $\overline{p}(\tau)$ via $|\mathcal{L}|$.

Claim. If $\mathcal{L} \neq \varnothing$, for all $(z_1, \ldots, z_q) \in \mathcal{L}$,

$$\mathsf{Pr}\left[\pi \overset{\$}{\leftarrow} \Omega'[\tau] : \forall i \in [q] : Z_i = z_i\right] \geqslant \frac{1}{2^{nq}} .$$

Proof (Of Claim). Fix $(z_1, \ldots, z_q) \in \mathcal{L}$, and define $x_i := \gamma(e_i)$ for all $i \in [q]$ as the label $\gamma(e_i)$ we encounter if we were to answer with the sequence $(z_1, \ldots, z_q)$ in the iterative process. Then

$$\mathsf{Pr}\left[\pi \overset{\$}{\leftarrow} \Omega'[\tau] : \forall i \in [q] : Z_i = z_i\right] = \frac{|\{\pi \in \Omega'[\tau] \ : \ \forall i \in [q] : \pi(x_i) = z_i\}|}{|\Omega'[\tau]|} .$$

We let $E' \subseteq E$ be the set of edges $(M, M \,\|\, m)$ such that at the beginning of the process, $\lambda(M)$ or $\lambda(M \,\|\, m)$ (or possibly both) are not defined, and let $D := E \backslash E'$. Clearly we have $\{e_1, \ldots, e_q\} \subseteq E'$ and $|D| + |E'| = |E|$.

Note that because $\tau \in \mathsf{GT}$ and the definition of $\mathcal{L}$, at the end of the process $\overline{\pi}(x)$ is defined for *exactly* $|D| + |E'| = |E|$ values. This is because τ is good, and in particular this means that all values $\overline{\pi}(x)$ we set are for *distinct* x (this follows from the fact that $(\lambda, \gamma) \neq \perp$), and there is one such value being set for every edge in E' (during the process) and for every edge in D (from the beginning). Hence there are exactly $(2^n - |D| - |E'|)!$ ways to complete $\overline{\pi}$ into a permutation $\pi \in \Omega'[\tau]$. Thus, this is exactly the number of permutations in $\Omega'[\tau]$ with $\pi(x_i) = z_i$ for all $i \in [q]$.

On the other hand, the claim follows from the fact, which we show next, that the number of permutations in $\Omega'[\tau]$ satisfies

$$\left|\Omega'[\tau]\right| \leqslant 2^{nq}(2^n - |D| - |E'|)! .$$

This can be seen by encoding $\pi \in \Omega'[\tau]$ as follows (given τ, and in particular the initial value of $\overline{\pi}$): We start with an empty list $\mathcal{H}$ of n-bit strings. We run the above process using π, and every time we set a value $\lambda(M_i) = \pi(\gamma(e_i)) = z_i$, we append z_i to $\mathcal{H}$. Note that since τ is good (and thus

[11] 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.

the initial (λ, γ) are not $(\bot, \bot)$), the $\gamma(e_i)$'s are all distinct. Also, all derived values we set (again because τ is good) will be on different input. Therefore, the q values $z_1, \ldots, z_q$ define the behavior of π on $|E'|$ values, while it was defined on $|D|$ values already at the beginning of the process. Hence, the encoding is completed by adding to $\mathcal{H}$ all $\pi(x)$ (wrt the lexicographic ordering of the inputs x) for all x such that no edge in the graph is labeled with x at the end of the iterative process. It is not hard to see that this encoding is unique, and that there are at most $2^{nq}(2^n - |D| - |E'|)!$ such sequences $\mathcal{H}$. □

To conclude, it is easy to verify that

$$\overline{p}(\tau) \geqslant \sum_{(z_1,\ldots,z_q)\in\mathcal{L}} \mathsf{Pr}\left[\pi \xleftarrow{\$} \Omega'[\tau] : \forall i \in [q] : Z_i = z_i\right] \geqslant \frac{|\mathcal{L}|}{2^{nq}} \,. \tag{7}$$

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 we satisfy this invariant. 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 $\mathcal{L}$.

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 $\lambda(M)[1 \ldots r] = Y_i$, 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

$$\begin{aligned}|\mathcal{L}| &\geqslant \prod_{i=1}^{q}(2^{n-r} - N_i^{(1)} - N_i^{(2)} - q - q \cdot D_i)\\ &= 2^{q\cdot(n-r)} \cdot \prod_{i=1}^{q}\left(1 - \frac{N_i^{(1)} + N_i^{(2)} + q + q \cdot D_i}{2^{n-r}}\right)\\ &\geqslant 2^{q\cdot(n-r)} \cdot \left(1 - \frac{N^{(1)} + N^{(2)}}{2^{n-r}} - \frac{2q^2}{2^{n-r}}\right) \,,\end{aligned} \tag{8}$$

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

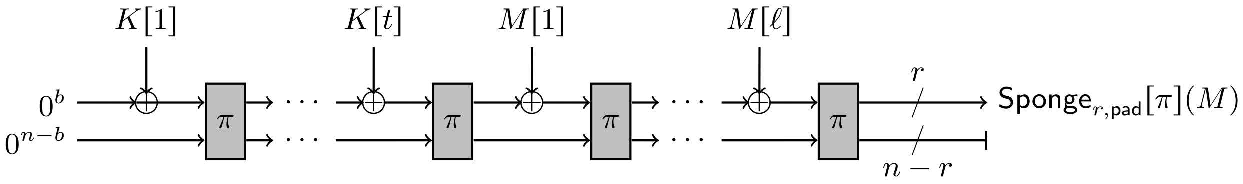

Fig. 3. Sponge construction. Representation of $\mathsf{Sponge}_{r,\mathsf{pad}_b}[\pi]$ used with a padding scheme pad_b that enforces b-bit blocks.

5 Security Analysis of Sponge-Based MACs

Sponge-Based MAC. We first briefly review the usage of the sponge hash-function [11] as a MAC via key-prepending. As in the TCBC case above, we fix parameters n, r and a padding scheme $\mathsf{pad} : \{0,1\}^* \to (\{0,1\}^n)^+$, uniquely encoding arbitrary strings into non-empty sequences of n-bit blocks, not necessarily in a prefix-free fashion. Then, the construction $\mathsf{Sponge} = \mathsf{Sponge}_{r,\mathsf{pad}}[\pi] : \{0,1\}^\kappa \times \{0,1\}^* \to \{0,1\}^r$ operates as follows on input $M \in \{0,1\}^*$ and key $K \in \{0,1\}^\kappa$, for a permutation $\pi \in \mathsf{Perm}(n)$:

Construction $\mathsf{Sponge}_{r,\mathsf{pad}}[\pi]_K(M)$:

(1) Compute $\mathsf{pad}(K \| M) = K[1] \dots K[w]\, M[1] \dots M[\ell]$ (for some ℓ and w).
(2) Then, starting with $V_0 := 0^n$, compute $V_i = \pi(K[i] \oplus V_{i-1})$ for all $i \in [w]$ and let $K' := V_t$.
(3) Next, starting with $S_0 := K'$, compute $S_i = \pi(M[i] \oplus S_{i-1})$ for all $i \in [\ell]$.
(4) Finally, output $S_\ell[1 \dots r]$, i.e., the first r bits of S_ℓ.

Note that we are silently assuming (for simplicity) that the (padded) keys and the actual message end up in different blocks, and hence our naming conventions. Our results can be extended to the more general case, but we avoid the notational overhead in this version of the paper.

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 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 Figure 3.

Finally, we also consider a variant of the construction – called $\mathsf{GSponge}$ – that takes an n-bit key K and differs from Sponge in step **(2)** where it directly sets $K' := K$ instead of absorbing the key. The construction is similar to some other MAC designs such as $\mathsf{donkeySponge}$ [9] and $\mathsf{Pelican}$ [16]. This natural variant will be simpler to analyze – and will be indeed analyzed first. The bound for the Sponge construction will be derived from the one for $\mathsf{GSponge}$ via a high-level lemma of independent interest proving the soundness of the key extraction method.

Security analysis of $\mathsf{GSponge}$. We prove the following theorem about the $\mathsf{GSponge}$ construction.

Theorem 2 (Security of $\mathsf{GSponge}$). *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* $t \geqslant 1$,

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{GSponge}_{r,\mathsf{pad}},\pi}(\mathcal{A}) \leqslant \frac{(6t+17)\ell q_C^2 + 7\ell q_\pi q_C + 2q_C}{2^n} + \frac{6nq_C^2 + 8\ell q_C + q_\pi q_C}{2^{n-r}} + \\ + \frac{136\ell^4 q_C^2}{2^{2n}} + \frac{2(\ell q_C)^{t+1}}{2^{nt}} \ . \quad (9)$$

We note that is in the case of TCBC, for sufficiently large t and for $\ell < 2^{n/4}$, the first two terms are the leading terms. We will prove below tightness of the bound when $\ell < 2^r$ and $q_\pi \geqslant \ell$.

The proof is an adaptation of the proof strategy of Theorem 1 to the setting of sponges. Hence, we start by observing that the current setting with GSponge is very similar to the setting considered in Theorem 1 involving the TCBC construction, with two important differences:

- The processing of a message in GSponge starts from the key K, as opposed to using a fixed initialization vector IV in TCBC.
- We now allow the adversary to also query the random permutation π that was secret before. We will indeed show that as we start with a random IV, the probability that internal queries and direct queries to π will not intersect, except with probability $O(\ell \cdot q_C q_\pi / 2^n)$

In Appendix F we describe the modifications that have to be applied to the proof of Theorem 1 in order to account for these differences.

Tightness. One can trivially adapt the attack given in Section 3 to the setting of sponges, obtaining a prf-adversary that asks q_C construction queries and *no* π-queries, and achieves advantage at least $\Omega(q_C^2/2^{n-r})$. Here we present a different generic attack on sponges that needs q_C construction queries and q_π queries to π, and achieves advantage roughly $\Omega(q_C q_\pi / 2^{n-r})$.

For simplicity, we again assume that the attacker can query the construction with unpadded messages. For parameters Q_1, Q_2 and $t := \lceil n/r \rceil$, the attacker $\mathcal{A}_{Q_1,Q_2,t}$ proceeds as follows, given access to the construction oracle O (which is either GSponge[π] under a random key K or a random function) and the permutation oracle $\pi(\cdot)$.

Adversary $\mathcal{A}_{Q_1,Q_2,t}$:

1. For all $i \in [Q_1]$ query distinct random one-block messages $M_i \in \{0,1\}^n$ to O, obtaining output $Y_i \in \{0,1\}^r$.
2. For all $i \in [Q_1]$ and $j \in [t]$ query $M_i \,\|\, Y_i \,\|\, 0^{n-r} \,\|\, 0^{n(j-1)}$ to O, obtaining output $Y_{i,j} \in \{0,1\}^r$.
3. For all $i \in [Q_2]$, choose distinct random $B_{i,0} \in 0^r \,\|\, \{0,1\}^{n-r}$ and for each $j \in [t]$ query $\pi(\cdot)$ to compute $B_{i,j} = \pi(B_{i,j-1})$.
4. If there exist $i \in [Q_1]$, $i' \in [Q_2]$ with $Y_{i,j} = B_{i',j}[1..r]$ for all $j \in [t]$ then output 1, else output 0.

The attacker $\mathcal{A}_{Q_1,Q_2,t}$ makes $q_C = (t+1)Q_1$ construction queries and $q_\pi = tQ_2$ queries to π. As we sketch in Appendix E, it achieves the advantage $\Omega\left(r^2 q_C q_\pi / n^2 2^{n-r}\right)$.

Just as in the case of TCBC in Section 3, the attack above can trivially be turned into a forging attack. The same attack also works for the construction Sponge.

From GSponge to Sponge: Replacing the Uniform Key. Our final result proves the 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 Figure 3. The proof of the following theorem is given in Appendix G, 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*

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

where A_t denotes the expression on the right-hand side of inequality (9). If $w = 1$, one can replace the whole min*-term by $\frac{q}{2^{bw}}$.*

We remark that our proof is highly non-trivial 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 inspired by [14] to a slightly more general setting. Roughly, the extra term is obtained by upper bounding the probability that all queries necessary for absorbing the actual sampled key are contained among the q permutation queries made by the attacker or by the sponge construction (*after* key absorption).

We note that the additional terms are smaller than $A(q_c, q_\pi)$ when the key length is $bw \approx 2n$ and $q > 2^{(n-b)/2}$ or $bw \approx n$ and $q < 2^{(n-b)/2}$. (Note that the latter case is in the same query regime as the indifferentiability proof [11].) Also, in SHA-3, where e.g. we could have $b = 1152$, the case $w = 1$ is largely sufficient, as security would hold as long as $q < 2^b$.

Acknowledgments

We thank Mihir Bellare for insightful feedback and comments on this manuscript.

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, and by a gift from the Gareatis Foundation. 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, Apr. 2014.
2. M. Bellare, "New proofs for NMAC and HMAC: Security without collision-resistance," in *CRYPTO 2006*, vol. 4117 of *LNCS*, pp. 602–619, Springer, Aug. 2006.
3. M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in *CRYPTO'96*, vol. 1109 of *LNCS*, pp. 1–15, Springer, Aug. 1996.
4. M. Bellare, J. Kilian, and P. Rogaway, "The security of cipher block chaining," in *CRYPTO'94*, vol. 839 of *LNCS*, pp. 341–358, Springer, Aug. 1994.
5. M. Bellare, K. Pietrzak, and P. Rogaway, "Improved security analyses for CBC MACs," in *CRYPTO 2005*, vol. 3621 of *LNCS*, pp. 527–545, Springer, Aug. 2005.
6. M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *ACM CCS 93*, pp. 62–73, ACM Press, Nov. 1993.
7. D. J. Bernstein, "A short proof of the unpredictability of cipher block chaining.." Available at `http://cr.yp.to/antiforgery/easycbc-20050109.pdf`, 2005.
8. G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche, "On the security of the keyed sponge construction." Symmetric Key Encryption Workshop (SKEW), February 2011.
9. G. Bertoni, J. Daemen, and M. Peeters, "Permutation-based encryption, authentication and authenticated encryption," in *Directions in Authenticated Ciphers*, 2012.

10. G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche, "Keccak," in *EUROCRYPT 2013*, vol. 7881 of *LNCS*, pp. 313–314, Springer, May 2013.
11. G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, "On the indifferentiability of the sponge construction," in *EUROCRYPT 2008*, vol. 4965 of *LNCS*, pp. 181–197, Springer, Apr. 2008.
12. J. Black and P. Rogaway, "CBC MACs for arbitrary-length messages: The three-key constructions," in *CRYPTO 2000*, vol. 1880 of *LNCS*, pp. 197–215, Springer, Aug. 2000.
13. D. Chang, M. Dworkin, S. Hong, J. Kelsey, and M. Nandi, "A keyed sponge construction with pseudorandomness in the standard model," in *Proceedings of the Third SHA-3 Candidate Conference*, 2012.
14. S. Chen and J. P. Steinberger, "Tight security bounds for key-alternating ciphers," in *EUROCRYPT 2014*, vol. 8441 of *LNCS*, pp. 327–350, Springer, May 2014.
15. "Computer data authentication." National Bureau of Standards, NBS FIPS PUB 113, U.S. Department of Commerce, May 1985.
16. J. Daemen and V. Rijmen, "The mac function pelican 2.0." Cryptology ePrint Archive, Report 2005/088, 2005. http://eprint.iacr.org/.
17. I. Damgård, "A design principle for hash functions," in *CRYPTO'89*, vol. 435 of *LNCS*, pp. 416–427, Springer, Aug. 1989.
18. Y. Dodis, T. Ristenpart, J. P. Steinberger, and S. Tessaro, "To hash or not to hash again? (in)differentiability results for h^2 and HMAC," in *CRYPTO 2012*, vol. 7417 of *LNCS*, pp. 348–366, Springer, Aug. 2012.
19. O. Dunkelman, N. Keller, and A. Shamir, "Minimalism in cryptography: The Even-Mansour scheme revisited," in *EUROCRYPT 2012*, vol. 7237 of *LNCS*, pp. 336–354, Springer, Apr. 2012.
20. S. Even and Y. Mansour, "A construction of a cipher from a single pseudorandom permutation," *Journal of Cryptology*, vol. 10, no. 3, pp. 151–162, 1997.
21. P. Gaži, K. Pietrzak, and M. Rybár, "The exact PRF-security of NMAC and HMAC," in *CRYPTO 2014, Part I*, vol. 8616 of *LNCS*, pp. 113–130, Springer, Aug. 2014.
22. O. Goldreich, S. Goldwasser, and S. Micali, "On the cryptographic applications of random functions," in *CRYPTO'84*, vol. 196 of *LNCS*, pp. 276–288, Springer, Aug. 1984.
23. S. Halevi and P. Rogaway, "A parallelizable enciphering mode," in *CT-RSA 2004*, vol. 2964 of *LNCS*, pp. 292–304, Springer, Feb. 2004.
24. T. Iwata and K. Kurosawa, "OMAC: One-key CBC MAC," in *FSE 2003*, vol. 2887 of *LNCS*, pp. 129–153, Springer, Feb. 2003.
25. T. Iwata and K. Kurosawa, "Stronger security bounds for OMAC, TMAC, and XCBC," in *INDOCRYPT 2003*, vol. 2904 of *LNCS*, pp. 402–415, Springer, Dec. 2003.
26. N. Koblitz and A. Menezes, "Another look at HMAC." Cryptology ePrint Archive, Report 2012/074, 2012. http://eprint.iacr.org/2012/074.
27. K. Kurosawa and T. Iwata, "TMAC: Two-key CBC MAC," in *CT-RSA 2003*, vol. 2612 of *LNCS*, pp. 33–49, Springer, Apr. 2003.
28. "Information technology security techniques message authentication codes (macs) part 1: Mechanisms using a block cipher." ISO/IEC 9797-1, 1999.
29. U. M. Maurer, "Indistinguishability of random systems," in *EUROCRYPT 2002*, vol. 2332 of *LNCS*, pp. 110–132, Springer, Apr. / May 2002.
30. U. M. Maurer, R. Renner, and C. Holenstein, "Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology," in *TCC 2004*, vol. 2951 of *LNCS*, pp. 21–39, Springer, Feb. 2004.
31. R. C. Merkle, "One way hash functions and DES," in *CRYPTO'89*, vol. 435 of *LNCS*, pp. 428–446, Springer, Aug. 1989.
32. K. Minematsu and T. Matsushima, "New bounds for PMAC, TMAC, and XCBC," in *FSE 2007*, vol. 4593 of *LNCS*, pp. 434–451, Springer, Mar. 2007.

33. R. Motwani and P. Raghavan, *Randomized Algorithms.* New York, NY, USA: Cambridge University Press, 1995.
34. M. Nandi, "A simple and unified method of proving indistinguishability," in *INDOCRYPT 2006*, vol. 4329 of *LNCS*, pp. 317–334, Springer, Dec. 2006.
35. J. Patarin, "The "coefficients H" technique (invited talk)," in *SAC 2008*, vol. 5381 of *LNCS*, pp. 328–345, Springer, Aug. 2008.
36. E. Petrank and C. Rackoff, "CBC MAC for real-time data sources," *Journal of Cryptology*, vol. 13, no. 3, pp. 315–338, 2000.
37. K. Pietrzak, "A tight bound for EMAC," in *ICALP 2006, Part II*, vol. 4052 of *LNCS*, pp. 168–179, Springer, July 2006.
38. T. Shrimpton and R. S. Terashima, "A modular framework for building variable-input-length tweakable ciphers," in *ASIACRYPT 2013, Part I*, vol. 8269 of *LNCS*, pp. 405–423, Springer, Dec. 2013.
39. J. Vandewalle, D. Chaum, W. Fumy, C. J. A. Jansen, P. Landrock, and G. Roelofsen, "A european call for cryptographic algorithms: Ripe; race integrity primitives evaluation," in *EUROCRYPT'89*, vol. 434 of *LNCS*, pp. 267–271, Springer, Apr. 1989.
40. S. Vaudenay, "Decorrelation over infinite domains: The encrypted CBC-MAC case," in *SAC 2000*, vol. 2012 of *LNCS*, pp. 189–201, Springer, Aug. 2000.

A The *H*-Coefficient Method

In this section we prove the basic lemma underlying Patarin's H-Coefficient method [35].

Lemma 1 (restated). *Let* $\delta, \varepsilon \in [0,1]$ *be such that:*

(a) $\Pr[\mathsf{T}_{\mathsf{ideal}} \in \mathsf{BT}] \leqslant \delta$.
(b) *For all* $\tau \in \mathsf{GT}$,

$$\frac{\Pr[\mathsf{T}_{\mathsf{real}} = \tau]}{\Pr[\mathsf{T}_{\mathsf{ideal}} = \tau]} \geqslant 1 - \varepsilon \ .$$

Then,

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

Proof. Let $\mathcal{T}$ be the set of valid transcripts such that $\Pr[\mathsf{T}_{\mathsf{ideal}} = \tau] \geqslant \Pr[\mathsf{T}_{\mathsf{real}} = \tau]$. Then,

$$\mathsf{SD}(\mathsf{T}_{\mathsf{real}}, \mathsf{T}_{\mathsf{ideal}}) = \sum_{\tau \in \mathcal{T}} (\Pr[\mathsf{T}_{\mathsf{ideal}} = \tau] - \Pr[\mathsf{T}_{\mathsf{real}} = \tau])$$

by the fundamental properties of the statistical distance. Then, note that $\mathcal{T}$ can be partitioned into two blocks $\mathcal{T} \cap \mathsf{BT}$ and $\mathcal{T} \cap \mathsf{GT}$. On the one hand, we can use **(a)** to upper bound

$$\sum_{\tau \in \mathcal{T} \cap \mathsf{BT}} (\Pr[\mathsf{T}_{\mathsf{ideal}} = \tau] - \Pr[\mathsf{T}_{\mathsf{real}} = \tau]) \leqslant \sum_{\tau \in \mathcal{T} \cap \mathsf{BT}} \Pr[\mathsf{T}_{\mathsf{ideal}} = \tau] \leqslant \sum_{\tau \in \mathsf{BT}} \Pr[\mathsf{T}_{\mathsf{ideal}} = \tau] \leqslant \delta \ .$$

On the other hand, **(b)** implies

$$\sum_{\tau \in \mathcal{T} \cap \mathsf{GT}} (\Pr[\mathsf{T}_{\mathsf{ideal}} = \tau] - \Pr[\mathsf{T}_{\mathsf{real}} = \tau]) \leqslant \varepsilon \cdot \sum_{\tau \in \mathcal{T} \cap \mathsf{GT}} \Pr[\mathsf{T}_{\mathsf{ideal}} = \tau] \leqslant \varepsilon \ .$$

Therefore, $\mathsf{SD}(\mathsf{T}_{\mathsf{real}}, \mathsf{T}_{\mathsf{ideal}}) \leqslant \varepsilon + \delta$. Moreover, every adversary $\mathcal{A}$ can be turned into a distinguisher $\mathcal{A}'$ for $\mathsf{T}_{\mathsf{real}}$ and $\mathsf{T}_{\mathsf{ideal}}$, which looks at the first part of the transcript (i.e., the one containing the q message-output pairs $(M_1, Y_1), \ldots, (M_q, Y_q)$), and outputs the corresponding decision bit $\mathcal{A}$ would output (this bit is uniquely defined by the fact that $\mathcal{A}$ is deterministic). Then, we clearly have

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{TCBC}}(\mathcal{A}) = \Pr[\mathcal{A}'(\mathsf{T}_{\mathsf{real}}) \Rightarrow 1] - \Pr[\mathcal{A}'(\mathsf{T}_{\mathsf{ideal}}) \Rightarrow 1] \leqslant \mathsf{SD}(\mathsf{T}_{\mathsf{real}}, \mathsf{T}_{\mathsf{ideal}}) \leqslant \varepsilon + \delta \ ,$$

as the statistical distance is the quantity corresponding to the advantage of the best $\mathcal{A}$'. □

B Chernoff bounds

Below, we are going to use the following standard variant of the Chernoff bound. See e.g. [33] for a proof.

Theorem 4 (Chernoff bound). *Let $X_1, \ldots, X_T$ be independent random variables with $\mathsf{E}[X_i] = p_i$ and $X_i \in [0,1]$. Let $X = \sum_{i=1}^T X_i$ and $\mu = \sum_{i=1}^T p_i = \mathsf{E}[X]$. Then, for all $\delta \geqslant 0$,*

- $\mathsf{Pr}\left[X \geqslant (1+\delta)\mu\right] \leqslant e^{-\frac{\delta^2}{2+\delta}\mu}$.
- $\mathsf{Pr}\left[X \leqslant (1-\delta)\mu\right] \leqslant e^{-\frac{\delta^2}{2+\delta}\mu}$.

C Bad-Transcript Analysis: The Collision Events

To bound the probability of the ideal transcript being bad, in this section we start by bounding $\mathsf{Pr}\left[\mathsf{T}_{\mathsf{ideal}} \in \mathsf{BT}_1\right]$. We take a combinatorial approach inspired by [5,21] and represent the computation of TCBC on various inputs by directed graphs; the following useful notation closely follows [5].

Graph-Based Representation of TCBC. Let $\mathcal{M} = (M_1, M_2)$ be two distinct messages that can be parsed into n-bit blocks as $M_i = M_i^1 \parallel \cdots \parallel M_i^{\ell_i}$ for some $\ell_1, \ell_2 \leqslant \ell$, and let $\varLambda := \ell_1 + \ell_2$. For convenience, we use the notation $M^{(i)}$ as a reference to the block M_1^i if $i \leqslant \ell_1$, otherwise it denotes the block $M_2^{i-\ell_1}$. For any fixed permutation $\pi \in \mathsf{Perm}(n)$ and a pair of such messages $\mathcal{M}$ we define the *structure graph*[12] $G_\pi^{\mathcal{M}}$, which is a directed graph (V, E) where $V \subseteq \{0, \ldots, \varLambda\}$ together with a edge-labeling function $L : E \to \{M^{(1)}, \ldots, M^{(\varLambda)}\}$. The structure graph $G_\pi^{\mathcal{M}} = G = (V, E, L)$ is defined as follows: We set $C_0 = 0^n$ and for $i = 1, \ldots, \varLambda$ we define

$$C_i = \begin{cases} \pi(C_{i-1} \oplus M_i) & \text{for } i \neq \ell_1 + 1 \\ \pi(M_i) & \text{for } i = \ell_1 + 1 \end{cases}$$

From these values C_i we define the mapping $[.]_G : \{0, \ldots, \varLambda\} \to \{0, \ldots, \varLambda\}$ as $[i]_G = \min\{j : C_j = C_i\}$. It is convenient to also define a mapping $[.]'_G$ as $[i]'_G = 0$ if $i = \ell_1$ and $[i]'_G = [i]_G$ otherwise. Now the structure graph $G_\pi^{\mathcal{M}} = G = (V, E, L)$ is given by

$$V = \{[i]_G : 1 \leqslant i \leqslant \varLambda\}\ , \qquad E = \{([i-1]'_G, [i]_G) : 1 \leqslant i \leqslant \varLambda\}\ , \qquad L(([i-1]'_G, [i]_G)) = M^{(i)}\ .$$

Let $\mathcal{G}(\mathcal{M}) = \{G_\pi^{\mathcal{M}} :\ \pi \in \mathsf{Perm}(n)\}$ denote the set of all structure graphs associated to the message pair $\mathcal{M}$. Note that sampling the permutation π uniformly at random also induces a probability distribution on the set $\mathcal{G}(\mathcal{M})$. For $G = (V, E, L) \in \mathcal{G}(\mathcal{M})$ we denote with $G_i = (V_i, E_i, L_i)$ the subgraph of G given by the i first edges, i.e., we let $V_i = \{v \in V : v \leqslant i\}, E_i = \{(u, v) \in E : u, v \in V_i\}$ and L_i is L with the domain restricted to E_i. We will refer to $G \in \mathcal{G}(\mathcal{M})$ as consisting of two paths, the "M_1-path" which passes through the vertices $0, [1]_G, \ldots, [\ell_1]_G$ and the "M_2-path" $0, [\ell_1+1]_G, \ldots, [\ell_1+\ell_2]_G$. We denote by $V_j^i(G)$ the i-th vertex on the M_j-path, hence for $1 \leqslant i \leqslant \ell_1$ we have $V_1^i(G) = [i]_G$, while for $1 \leqslant i \leqslant \ell_2$ we get $V_2^i(G) = [i + \ell_1]_G$; and $V_1^0(G) = V_2^0(G) = 0$.

Collisions. Suppose a structure graph $G = G_\pi^{\mathcal{M}} \in \mathcal{G}(\mathcal{M})$ is exposed edge by edge (i.e. in step i the value $[i]_G$ is shown to us). We say that G has a *collision* is step i if the edge exposed in step i points to a vertex which is already in the graph. With $\mathsf{Col}(G)$ we denote all collisions, i.e. all pairs (i, j) where in step i there was a collision which hit the vertex computed in step $j < i$:

$$\mathsf{Col}(G) = \{(i, [i]_G) : [i]_G \neq i\}\ .$$

[12] Note that the structure graph differs from the message tree considered before.

Induced Collisions and Accidents. We distinguish two types of collisions, *induced collisions* and *accidents*. Informally, an induced collision in step i is a collision which is implied by the collisions in the first $i-1$ steps, whereas an accident is a "surprising" collision.

Assume that after step $i-1$ we see that for some $a < i$ the a-th edge $([a-1]'_G, [a]_G)$ has the same label ($M^{(a)} = M^{(i)}$) and the same starting point $(([a-1]'_G = [i-1]'_G)$ as the next (i-th) edge to be exposed. Then we know that the endpoint of the i-th edge must also be $[a]_G$ as $[a-1]_G = [i-1]_G$ means $C_{[a-1]_G} = C_{[i-1]_G}$, and as π must produce the same output on the same input, we also get $C_{[a]_G} = \pi(C_{[a-1]'_G} \oplus M^{(a)}) = \pi(C_{[i-1]'_G} \oplus M^{(i)}) = C_{[i]_G}$. More generally, it was shown in [5] that G has an induced collision is step i if the edge added in step i (or, that would be added if it was not already there) closes a cycle with alternating edge directions, moreover then the XOR of all labels of all the edges of that cycle is 0^n. (Note that the case of two parallel edges considered before form exactly such a cycle of length two and also $0^n = M^{(i)} \oplus M^{(a)}$ as we saw that $M^{(i)} = M^{(a)}$.)

Formally, we define a function AltCyc which takes as input a partial structure graph $G_i = (V_i, E_i, L_i)$, a vertex v and a label X as follows

$$\mathsf{AltCyc}(G_i = (V_i, E_i, L_i), v, X) = \begin{cases} j = v_{2k} & \text{if } \exists k \geqslant 1, \{v_1, \ldots, v_{2k}\} \in V_i, \{e_1, \ldots, e_{2k}\} \in E_i \text{ where} \\ & e_i = (v_i, v_{i+1}) \text{ for odd, and } e_i = (v_i, v_{i+1}) \text{ for even } i, \\ & \text{and } v_1 = v, \\ & \text{and } X \oplus L_i((u_1, v_1)) \oplus \ldots L_i((u_{2k}, v_{2k})) = 0^n. \\ \perp & \text{otherwise} \end{cases}$$

Now the induced collisions are the collisions (i, j) where the i-th edge $([i-1]'_G, j)$ can (and thus must) be added to G_{i-1} such that we close a cycle with alternating edge directions where the labels on the cycle XOR to 0^n, i.e.,

$$\mathsf{IndCol}(G) = \{(i, j) : 1 \leqslant i \leqslant m, j = \mathsf{AltCyc}(G_{i-1}, [i-1]'_G, M^{(i)}) \text{ and } j \neq \perp\} ,$$

and *accidents* are all the non-induced collisions: $\mathsf{Acc}(G) := \mathsf{Col}(G) \backslash \mathsf{IndCol}(G)$.

Let $\mathcal{G}^a(\mathcal{M}) = \{G : G \in \mathcal{G}(\mathcal{M}), |\mathsf{Acc}(G)| = a)$ denote all structure graphs with exactly a accidents. For any predicate P on structure graphs, let $\phi_{\mathcal{M}}[P]$ denote the set of all structure graphs G having exactly one accident and satisfying the predicate P, i.e.,

$$\phi_{\mathcal{M}}[P] = \left\{G \in \mathcal{G}^1(\mathcal{M}) \colon \; G \text{ satisfies } P\right\} .$$

As an example, consider the predicate P defined as $V_1^{\ell_1}(\cdot) = V_2^{\ell_2}(\cdot)$, in that case we obtain $\phi_{\mathcal{M}}[V_1^{\ell_1} = V_2^{\ell_2}] = \left\{G \in \mathcal{G}^1(\mathcal{M}) \colon \; V_1^{\ell_1}(G) = V_2^{\ell_2}(G)\right\}$.

Finally, following [5], for two messages $M_1, M_2 \in B_n^+$ we let $\mathsf{FCP}_n(M_1, M_2)$ (the *full collision probability*) be the probability, over $\pi \leftarrow \mathsf{Perm}(n)$, that

$$\mathsf{CBC}^\pi(M_2) \in \left\{\mathsf{CBC}^\pi(M') \, : \, (M' \mid M_1 \vee M' \mid M_2) \wedge M' \neq M_2\right\} .$$

Note that if $M_2 \nmid M_1$ then our definition matches the definition of full collision probability in [5]. On the other hand, if $M_2 \mid M_1$ the original definition becomes void (the probability is equal to 1), while our variant will prove to be useful also in this case.

We will make use of the following results proven in [5].

Proposition 2. *Let $\mathcal{M} = (M_1, M_2)$ be a pair of messages such that $M_i \in B_n^{\ell_i}$ and $\ell_i \leqslant \ell$ for both $i \in \{1, 2\}$.*

(i) *[5, Lemma 2] If $M_1 \nmid M_2$ and $M_2 \nmid M_1$ then we have*

$$\mathsf{FCP}_n(M_1, M_2) \leqslant \frac{8\ell}{2^n} + \frac{64\ell^4}{2^{2n}} .$$

(ii) *[5, Lemma 8] For any structure graph $H \in \mathcal{G}(\mathcal{M})$ we have*

$$\mathsf{Pr}[G \xleftarrow{\$} \mathcal{G}(\mathcal{M}) \colon \ G = H] \leqslant (2^n - 2\ell)^{-|\mathsf{Acc}(H)|} .$$

(iii) *[5, Lemma 9] We have*

$$\mathsf{Pr}[G \xleftarrow{\$} \mathcal{G}(\mathcal{M}) \colon \ |\mathsf{Acc}(G)| \geqslant 2] \leqslant \frac{4(\ell_1 + \ell_2)^4}{2^{2n}} .$$

(iv) *[5, Lemma 19] For any $b \in \{1, 2\}$ and $r \in [0, \ldots, \ell_b]$ we have*

$$\left|\phi_{\mathcal{M}}\left[V_b^r \in \{V_b^0, \ldots, V_b^{r-1}, V_b^{r+1}, \ldots, V_b^{\ell}\}\right]\right| \leqslant \ell_b .$$

We can now proceed to upper-bounding the probability $\mathsf{Pr}[\mathsf{T}_{\mathsf{ideal}} \in \mathsf{BT}_1]$.

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

Proof. We will denote by $\mathsf{MsgQ}(\tau) = \{M_1, \ldots, M_q\}$ the set of all message queries present in the transcript τ. By definition of BT_1, we have $\mathsf{T}_{\mathsf{ideal}} \in \mathsf{BT}_1$ if $\overline{T} = (V, E, \bot, \bot)$ and this occurs only if at least one of the following two events happens in the ideal experiment (in their description, λ refers to the non-restricted labelling):

(1) There exists $i \in [q]$ and $M \in V \backslash \{M_i\}$ such that $\lambda(M_i) = \lambda(M)$.
(2) There exists $i \in [q]$, $m \in \mathcal{M}_i$, and $M \in V \backslash \{M_i \,\|\, m\}$ such that $\lambda(M_i \,\|\, m) = \lambda(M)$.

Let us denote these two events as $\mathcal{B}_1$ and $\mathcal{B}_2$ respectively, and let us first consider $\mathcal{B}_1$. Since every vertex in $T^\pi(M_1, \ldots, M_q)$ lies on some path from the root to some leaf (and all leaves belong to $\mathsf{MsgQ}(\tau)$), by union bound we have

$$\mathsf{Pr}[\mathcal{B}_1] \leqslant \sum_{M_i} \mathsf{Pr}[\exists M \in V \backslash \{M_i\} \colon \mathsf{CBC}^\pi(M_i) = \mathsf{CBC}^\pi(M)] \leqslant \sum_{M_i, M_j} \mathsf{FCP}_n(M_j, M_i) \qquad (10)$$

summing over all $M_i \in \mathsf{MsgQ}(\tau)$, and all $M_j \in \mathsf{MsgQ}(\tau)$ that correspond to leaves in $T^\pi(M_1, \ldots, M_q)$. The probability is taken over the choice of a uniformly random permutation π.

If the message pair $\{M_j, M_i\}$ is prefix-free (i.e., $M_i \nmid M_j$ and $M_j \nmid M_i$), then we can apply statement (i) in Proposition 2 to conclude that $\mathsf{FCP}_n(M_j, M_i) \leqslant 8\ell/2^n + 64\ell^4/2^{2n}$. Since M_j is a leaf in $T^\pi(M_1, \ldots, M_q)$, clearly it cannot be a nontrivial prefix of M_i and hence the last case that needs to be considered is if $M_i \mid M_j$. In this case the structure graph for $\{M_j, M_i\}$ consists of the M_j-path and M_i does not introduce any additional vertices. Let us denote by ℓ_j and ℓ_i the lengths of M_j and M_i, respectively; and let V_1^i and V_2^i denote the i-th vertex on the M_j-path and M_i-path, respectively (starting from 0). Then we have $V_1^{\ell_i} = V_2^{\ell_i}$, and the equality $V_1^{\ell_i} = V_1^i$ for some $i \neq \ell_i$ can only occur if at least one accident happens. We know from Proposition 2 (iii) that two or more accidents can only occur with probability at most $64\ell^4/2^{2n}$. For the case of exactly one accident, Proposition 2 (iv) shows that there are at most ℓ structure graphs satisfying $V_1^{\ell_i} = V_1^i$ for some $i \neq \ell_i$, and by Proposition 2 (ii) each of these graphs can occur with probability at most $2/2^n$ as long as $\ell \leqslant 2^{n-2}$.

Putting it all together, the values $\mathsf{FCP}_n(M_j, M_i)$ in (10) are upper-bounded by $8\ell/2^n + 64\ell^4/2^{2n}$ for all pairs (M_j, M_i) such that $M_i \nmid M_j$, and by $2\ell/2^n + 64\ell^4/2^{2n}$ for all pairs (M_j, M_i) such that $M_i \mid M_j$. We are summing over at most q^2 pairs (M_j, M_i) in total, hence from (10) we obtain $\mathsf{Pr}[\mathcal{B}_1] \leqslant 8\ell q^2/2^n + 64\ell^4 q^2/2^{2n}$.

While for upper-bounding $\mathsf{Pr}[\mathcal{B}_1]$ we considered the probability that one of the vertices M_i will obtain the same label as some other vertex in the message tree, for $\mathsf{Pr}[\mathcal{B}_2]$ we need to consider the same probability for child vertices $M_i \,\|\, m$ for some $m \in \mathcal{M}_i$. However, note that in the analysis above, we did not use any property of the inspected vertices M_i beyond the fact that there are q of them. Since by (3) there are at most q child vertices $M_i \,\|\, m$ for some $i \in [q]$ and $m \in \mathcal{M}_i$, we can apply the same analysis as above to conclude that also $\mathsf{Pr}[\mathcal{B}_2] \leqslant 8\ell q^2/2^n + 64\ell^4 q^2/2^{2n}$. □

D Bad-Transcript Analysis: The $N^{(i)}$ quantities

We now turn to upper-bounding the probabilities $\mathsf{Pr}\left[\mathsf{T}_{\mathsf{ideal}} \in \mathsf{BT}_i\right]$ for $i \in \{2, 3\}$, i.e., proving Lemmas 4 and 5. While the proofs may appear at first to be a simple application of usual conventional Chernoff-like concentration techniques, they will require extra care due to the process of defining the trees. In particular, we will have to tackle two main challenges:

- The vertex labels are outputs of a permutation, and not of a function.
- Multiple vertices M and M' can be assigned the same label $\lambda(M)$ and $\lambda(M')$, however, whether this is the case depends on the value of labels assigned earlier in the process of computing the labels of the reduced message tree $\overline{T}^{\pi}$.

An alternative sampling process. As the common denominator between the proofs of Lemmas 4 and 5, it will be convenient to think of an alternative process to compute the transcript (and in particular $T^{\pi}(M_1, \dots, M_q)$, and its reduced version), parameterized by the value $t \geqslant 1$ from the theorem statement. This will allow us to use conventional Chernoff bounds for independent random variables at the cost of considering t times more samples.

Process SimulateIdealTranscript$(\mathcal{A}, t)$:

1. We first sample independent r-bit strings $Y_1, \dots, Y_q$
2. The attacker $\mathcal{A}$ gives queries M_i, which are replied with Y_i, for all $i \in [q]$.
3. Then, we compute the message tree $T = (V, E, \lambda, \gamma)$ by defining V and E with respect to $M_1, \dots, M_q$, and setting the labels λ, γ as follows. We initially set $\lambda(\varepsilon) \leftarrow \mathsf{IV}$, and set the labels $\gamma(e)$ of the edges leaving the root.

   Then, for every vertex $M \neq \varepsilon$ (traversed in some prefix-preserving order, i.e., if $M_1 \mid M_2$, then we visit M_1 before M_2) with parent vertex M' and $e := (M', M)$, sample t independent uniform n-bit strings $U_{M,1}, \dots, U_{M,t}$, and proceed as follows.
   - If $\pi(\gamma(e)) \neq \bot$, we set $\lambda(M) \leftarrow \pi(e)$.
   - Otherwise, we set $\lambda(M) = \pi(\gamma(e)) \leftarrow U_{M,j}$ for the smallest j such that $U_{M,j}$ was not used earlier as a vertex label yet. If all values have been used, then we set a bad flag to 1, and set $\pi(\gamma(e)) = \lambda(M) \leftarrow U_{M,t}$.
   - Also, for all $m \in \mathcal{M}_M$, set the edge labels $\gamma((M, M \,\|\, m)) \leftarrow \lambda(M) \oplus m$.

The resulting transcript $\mathsf{T}'_{\mathsf{ideal}}[t]$ is defined as $((M_1, Y_1), \dots, (M_q, Y_q), \overline{T})$, where $\overline{T}$ is the reduced version of the message tree $T = (V, E, \lambda, \gamma)$ defined by the above process. Moreover, let $\widehat{\mathsf{T}}'_{\mathsf{ideal}}[t]$ be the same as $\mathsf{T}'_{\mathsf{ideal}}[t]$, except that the tree is *not reduced*.

Let $\widehat{\mathsf{T}}_{\mathsf{ideal}}$ denote the ideal transcript with the *non-reduced* version of the message tree. The following lemma shows that for sufficiently large parameter $t \geqslant 1$, $\widehat{\mathsf{T}}_{\mathsf{ideal}}$ and $\widehat{\mathsf{T}}'_{\mathsf{ideal}}[t]$ are very close in statistical distance.

Lemma 6. *For all $t \geqslant 1$,*

$$\mathsf{SD}(\widehat{\mathsf{T}}_{\mathsf{ideal}}, \widehat{\mathsf{T}}'_{\mathsf{ideal}}[t]) \leqslant \frac{(q \cdot \ell)^{t+1}}{2^{nt}} \ . \tag{11}$$

Proof. Let p be the probability that bad is set to 1 at some point during the generation of $\widehat{\mathsf{T}}'_{\mathsf{ideal}}[t]$. Then, by a standard argument $\mathsf{SD}(\widehat{\mathsf{T}}_{\mathsf{ideal}}, \widehat{\mathsf{T}}'_{\mathsf{ideal}}[t]) \leqslant p$, since as long as bad is not set to 1, the process simulates the exact distribution as if we were using a random permutation. In every step i where we set the output value of π, we can easily show (as the values $U_{M,1}, \ldots, U_{M,t}$ are independent) that $\mathsf{Pr}\left[\mathsf{bad} \text{ is set in Step } i\right] \leqslant \left(\frac{q\ell}{2^n}\right)^t$, as π has been defined for at most $i \leqslant q\ell$ input values so far. A bound on p follows by the union bound. □

Proof of Lemma 4. Fix $Y_1, \ldots, Y_q$. We are going to upper-bound the probability that $N^{(1)} > B$, where

$$B := 3t \cdot q^2 \cdot \ell/2^r + 2qn \ . \tag{12}$$

Define $\widetilde{N}^{(1)} = \sum_{i=1}^q \widetilde{N}_i^{(1)}$ as $N_i^{(1)} = \sum_{i=1}^q N_i^{(1)}$, however all values $\widetilde{N}_i^{(1)}$ are defined with respect to a transcript sampled from the alternative transcript distribution $\mathsf{T}'_{\mathsf{ideal}}[t]$. Then, by Lemma 6,

$$\mathsf{Pr}\left[N^{(1)} > B\right] \leqslant \mathsf{Pr}\left[\widetilde{N}^{(1)} > B\right] + \mathsf{SD}(\widehat{\mathsf{T}}_{\mathsf{ideal}}, \widehat{\mathsf{T}}'_{\mathsf{ideal}}[t]) \leqslant \mathsf{Pr}\left[\widetilde{N}^{(1)} > B\right] + \frac{(q\ell)^{t+1}}{2^{nt}} \ .$$

For all $i \in [q]$, define $\bar{N}_i$ to be the number of $U_{M,j}$'s sampled in the process of generating $\mathsf{T}'_{\mathsf{ideal}}[t]$ such that $U_{M,j}[1\ldots,r] = Y_i$. (Independently of whether these values come to use or not.) Note that we sample $T := (|V| - 1) \cdot t \leqslant (q\ell)t$ independent n-bit values in the process (i.e., t values for every non-root vertex), and that clearly $\widetilde{N}_i^{(1)} \leqslant \bar{N}_i$. We expect $T/2^r$ of these values to equal Y_i. Consider now the following two cases:

Case 1: $T/2^r \geqslant n$. Here, by applying the Chernoff bound (Theorem 4)

$$\mathsf{Pr}\left[\bar{N}_i \geqslant 3 \cdot T/2^r + 2n\right] \leqslant \mathsf{Pr}\left[\bar{N}_i \geqslant 3 \cdot T/2^r\right] \leqslant e^{-T/2^r} \leqslant e^{-n} \ .$$

Case 2: $T/2^r < n$. Here, again by the Chernoff bound,

$$\mathsf{Pr}\left[\bar{N}_i \geqslant 2T/2^r + 2n\right] \leqslant \mathsf{Pr}\left[\bar{N}_i \geqslant T/2^r(1 + 2n2^r/T)\right] \leqslant e^{-\frac{4n^2 2^{2r}}{T^2(2+2n2^r/T)}\frac{T}{2^r}} \leqslant e^{-n}$$

because

$$\frac{4n^2 2^{2r}}{T^2(2 + 2n2^r/T)}\frac{T}{2^r} = \frac{2n^2 2^r}{T(1 + n2^r/T)} = \frac{2n^2 2^r}{T + n2^r} \geqslant \frac{2n^2 2^r}{2n2^r} = n \ .$$

We can wrap up using the union bound: The probability that there exists some $i \in [q]$ such that $\overline{N}_i > 3T/2^r + 2n$ is at most $q \cdot e^{-n}$. Therefore, except with probability $q \cdot e^{-n}$, $\widetilde{N} \leqslant \overline{N} \leqslant 3qT/2^r + 2qn \leqslant 3tq^2\ell/2^r + 2qn$, as we wanted to show.

Proof of Lemma 5. Recall that we are interested in studying the quantity

$$N^{(2)} = \sum_{i=1}^{q} N_i^{(2)}$$

for a given transcript τ sampled according to $\mathsf{T}_{\mathsf{ideal}}$. In particular, we assume now that the first part of the transcript $(M_1, Y_1), \ldots, (M_q, Y_q)$ has been fixed arbitrarily, and we are computing the tree $T^{\pi}(M_1, \ldots, M_q)$, for an independent permutation $\pi \overset{\$}{\leftarrow} \mathsf{Perm}(n)$. For every edge $e \in E$ and every $i \in [q]$, define $\widetilde{N}_{e,i}$ to be the number of $m \in \mathcal{M}_i$ such that $(\gamma(e) \oplus m)[1 \ldots r] = Y_i$, and we also let

$$\widetilde{N} = \sum_{e \in E} \sum_{i=1}^{q} \widetilde{N}_{e,i} = \sum_{e \in E} \widetilde{N}_e = \sum_{i=1}^{q} \widetilde{N}_i \ ,$$

where we have used the conventions $\widetilde{N}_e := \sum_{i=1}^{q} \widetilde{N}_{e,i}$ and $\widetilde{N}_i := \sum_{e \in E} \widetilde{N}_{e,i}$. Similarly, let $\widetilde{N}_{e,i,m} = 1$ if $(\gamma(e) \oplus m)[1 \ldots r] = Y_i$, and 0 otherwise, and thus clearly $\widetilde{N}_{e,i} = \sum_{m \in \mathcal{M}_i} \widetilde{N}_{e,i,m}$.

We first observe that $N_i^{(2)} \leqslant \widetilde{N}_i$, and thus $N^{(2)} \leqslant \widetilde{N}$. This is because for every $z \in \{0,1\}^n$ such that $z[1 \ldots r] = Y_i$ and there exists $m \in \mathcal{M}_i$ and $e \in E$ such that $\gamma(e) = z \oplus m$, we have $\gamma(e) \oplus m = z$ and thus $(\gamma(e) \oplus m)[1 \ldots r] = Y_i$, or in other words, $\widetilde{N}_{e,i,m} = 1$. Moreover, we are also possibly overcounting, since the new random variables are defined with respect to the whole tree T^{π}, and not just its reduced version.

The splitting trick. In the following, we are going to show that $\widetilde{N}$ (and hence $N^{(2)}$) cannot be too large. To this end, we will split every random variable $\widetilde{N}_{e,i,m}$ as

$$\widetilde{N}_{e,i,m} = \widetilde{N}_{e,i,m}^{\top} + \widetilde{N}_{e,i,m}^{\perp} \ ,$$

where $\widetilde{N}_{e,i}^{\top}$ is defined as $\widetilde{N}_{e,i}$ above, but only if the tail of the edge e was assigned as a label a fresh permutation value, and is 0 otherwise.[13] Conversely, $\widetilde{N}_{e,i,m}^{\perp}$ is defined as $\widetilde{N}_{e,i}$ above, but only if the tail node of e was re-assigned a previously used permutation value, or if the tail node is the root. For $b \in \{\top, \perp\}$, we define analogously $\widetilde{N}_{e,i}^{n}$, $\widetilde{N}^{b}$, $\widetilde{N}_{e}^{b}$ and $\widetilde{N}_{i}^{b}$ by only taking partial sums over the corresponding $\widetilde{N}_{e,i,m}^{b}$'s.

In the following, we are going to show, via two different analyses, that $\widetilde{N}^{\top}$ and $\widetilde{N}^{\perp}$ are not too large, except with small probability. The two analyses use very different techniques.

Lemma 7. *For all $t \geqslant 1$, we have*

$$\mathsf{Pr}\left[\widetilde{N}^{\top} \geqslant 3t \cdot q^2 \ell / 2^r + 2nq^2\right] \leqslant \frac{q}{2^n} + \frac{(q\ell)^{t+1}}{2^{nt}} \ .$$

Lemma 8. *We have,*

$$\mathsf{Pr}\left[\widetilde{N}^{\perp} \geqslant q^2 + q^2 \ell / 2^r + 8 q^2 \ell^4 / 2^{n+r}\right] \leqslant \frac{8q\ell}{2^{n-r}} \ .$$

Therefore, we conclude the proof of Lemma 5 by adding the two bounds.

[13] Note that whether was the case is easy to identify by looking at T^{π}.

Analysis of $\widetilde{N}^{\top}$ (Proof of Lemma 7). Let $V' \subset V$ be the set of non-root inner vertices of T, and let E' be the set of edges with their tail node in V'. (These are all edges, except those leaving the root.) Now, we consider the random experiment defining $\mathsf{T}'_{\mathsf{ideal}}[t]$. There, for every $j \in [t]$ and $e = (M', M' \,\|\, m')$, we are going to define similar random variables $\widehat{N}^{\top}_{e,i,j,m}$ as $\widehat{N}^{\top}_{e,i,m}$, except that $\widehat{N}^{\top}_{e,i,j,m}$ is simply one whenever the j-th value $U_{M',j}$ (out of t of them) generated when visiting M' is such that $(U_{M',j} \oplus m' \oplus m)[1 \ldots r] = Y_i$, regardless of whether $\lambda(M')$ is assigned $U_{M',j}$ or not. Then, we let

$$\widehat{N}^{\top} = \sum_{i=1}^{q} \sum_{m \in \mathcal{M}_i} \sum_{e \in E'} \sum_{j=1}^{t} \widehat{N}^{\top}_{e,i,j,m} \,.$$

Then, by Lemma 6, for $B^{\top} := 3t \cdot q^2\ell/2^r + 2nq^2$,

$$\mathsf{Pr}\left[\widetilde{N}^{\top} \geqslant B^{\top}\right] \leqslant \mathsf{Pr}\left[\widehat{N}^{\top} \geqslant B^{\top}\right] + \frac{(q\ell)^{t+1}}{2^{nt}} \,. \tag{13}$$

In particular, this is true if we only consider $\widetilde{N}^{\top}$ in the experiment where $\mathsf{T}'_{\mathsf{ideal}}[t]$ is sampled, and the inequality holds, because $\widehat{N}^{\top}$ can not be smaller then $\widetilde{N}^{\top}$.

For every $M' \in V'$, $i \in [q]$, $j \in [t]$, and $m \in \mathcal{M}_i$, we now introduce the shorthand

$$\widehat{N}^{\top}_{M',i,j,m} = \sum_{m' \in \mathcal{M}_{M'}} \widehat{N}^{\top}_{(M', M' \,\|\, m'),i,j,m} \,,$$

i.e., the number of edges e outgoing from M' for which $(U_{M',j} \oplus m' \oplus m)[1 \ldots r] = Y_i$ for the j-th value $U_{M',j}$ generated when visiting M'. Then,

$$\widehat{N}^{\top} = \sum_{i=1}^{q} \sum_{m \in \mathcal{M}_i} \widehat{N}^{\top}_{i,m}$$

where

$$\widehat{N}^{\top}_{i,m} = \sum_{j=1}^{t} \sum_{M' \in V'} \widehat{N}^{\top}_{M',i,j,m} \,.$$

In particular, $\widehat{N}^{\top}_{i,m}$ is the sum of $T := t \cdot |V'|$ independent random variables, where we note that $\mathsf{E}[\widehat{N}^{\top}_{M',i,j,m} = 1] = D_{M'}/2^r$, and $\widehat{N}^{\top}_{M',i,j,m} \in [0, D_{M'}]$. Thus, by linearity, its expected value is $\mu_{m,i} = \mathsf{E}[\widehat{N}^{\top}_{i,m}] = t \cdot |E'|/2^r \leqslant tq\ell/2^r$.

To apply the Chernoff bound, it is worth it to *scale* the random variables $\widehat{N}^{\top}_{i,m}$, dividing them by D to make them into $[0,1]$-values, where $D := \max_{M' \in V'} D_{M'}$. Note that $D \leqslant q$. That is, define $\widehat{L}^{\top}_{M',i,j,m} := \widehat{N}^{\top}_{M',i,j,m}/D$ and $\widehat{L}^{\top}_{i,m} = \sum_{j=1}^{t} \sum_{M' \in V'} \widehat{L}^{\top}_{M',i,j,m}$. Clearly, $\mathsf{E}[\widehat{L}^{\top}_{i,m}] = \mu_{i,m}/D$.

To conclude, we are going to show that this probability is smaller than $e^{-n} \leqslant 2^{-n}$. Therefore, by the union bound, it follows that the probability that $\mathsf{Pr}\left[\widehat{N}^{\top} \geqslant B^{\top}\right]$ is bounded by $q \cdot 2^{-n}$, as there are at most q pairs $i \in [q], m \in \mathcal{M}_i$. Together with (13), this implies Lemma 7.

Case 1: $\mu_{i,m} = t|E'|/2^r \leqslant q \cdot n$**.**

$$\begin{aligned}\mathsf{Pr}\left[\widehat{N}^{\top}_{i,m} \geqslant B^{\top}/q\right] &\leqslant \mathsf{Pr}\left[\widehat{N}^{\top}_{i,m} > 3\mu_{i,m} + 2nq\right]\\ &\leqslant \mathsf{Pr}\left[\widehat{N}^{\top}_{i,m} > \mu_{i,m} + 2nq\right]\\ &= \mathsf{Pr}\left[\widehat{L}^{\top}_{i,m} > \frac{\mu_{i,m}}{D} + \frac{2nq}{D}\right]\\ &= \mathsf{Pr}\left[\widehat{L}^{\top}_{i,m} > \frac{\mu_{i,m}}{D}\left(1 + \frac{2nq}{\mu_{i,m}}\right)\right] \leqslant e^{-\varDelta}\end{aligned}$$

where, using $D \leqslant q$ and $\mu_{i,m} \leqslant q \cdot n$,

$$\varDelta = \left(\frac{2n \cdot q}{\mu_{i,m}}\right)^2 \frac{1}{2 + \left(\frac{2n \cdot q}{\mu_{i,m}}\right)} \cdot \frac{\mu_{i,m}}{D} = \frac{4n^2 \cdot q^2}{2\mu_{i,m} + 2n \cdot q} \cdot \frac{1}{D} \geqslant \frac{n^2 \cdot q}{n \cdot q} = n\ .$$

Case 2. $\mu_{i,m} > q \cdot n$**.** Here, we compute

$$\begin{aligned}\mathsf{Pr}\left[\widehat{N}^{\top}_{i,m} \geqslant B^{\top}/q\right] &\leqslant \mathsf{Pr}\left[\widehat{N}^{\top}_{i,m} > 3\mu_{i,m} + 2nq\right]\\ &\leqslant \mathsf{Pr}\left[\widehat{N}^{\top}_{i,m} > 3\mu_{i,m}\right]\\ &= \mathsf{Pr}\left[\widehat{L}^{\top}_{i,m} > 3\mu_{i,m}/D\right]\\ &= \mathsf{Pr}\left[\widehat{L}^{\top}_{i,m} > \frac{\mu_{i,m}}{D}(1+2)\right] \leqslant e^{-\varDelta'}\ ,\end{aligned}$$

where, using again $D \leqslant q$,

$$\varDelta' = \frac{4}{4}\mu_{i,m}/D \geqslant q \cdot n/D \geqslant n\ .$$

Analysis of $\widetilde{N}^{\perp}$ (Proof of Lemma 8). First, let $E' \subseteq E$ be the set of edges which are not outgoing from the root. Then, we are going to split $\widetilde{N}^{\perp}$

$$\widetilde{N}^{\perp} = \sum_{e \in E \setminus E'} \sum_{i=1}^{q} \widetilde{N}^{\perp}_{e,i} + \sum_{e \in E'} \sum_{i=1}^{q} \widetilde{N}^{\perp}_{e,i} \leqslant q^2 + \sum_{e \in E'} \sum_{i=1}^{q} \widetilde{N}^{\perp}_{e,i}\ ,$$

as there are at most q edges in $E \setminus E'$. We are going to bound the sum by using Markov inequality. Therefore, we start by computing the expectation $\mu := \sum_{e \in E'} \sum_{i=1}^{q} \mathsf{E}\left[\widetilde{N}^{\perp}_{e,i}\right]$.

Assume now without loss of generality that T^{π} and its labels are defined iteratively by traversing it in some order preserving the prefix-of order. We focus now on computing, for a particular e and i, the value $\mathsf{E}[\widetilde{N}^{\perp}_{e,i}]$. To this end, define $\widetilde{N}^{\perp}_{M',e,m,i}$ for every $M' \leqslant e$ (which means that M' was traversed before reaching edge e) and every $m \in \mathcal{M}_i$: It equals one if $(\gamma(e) \oplus m)[1 \ldots r] = Y_i$, $\lambda(M) = \lambda(M')$, where M is the tail node of e, *and* M' was the first node to be set to this label. Otherwise, $\widetilde{N}^{\perp}_{M',e,m,i}$ is 0. Then, we clearly have

$$\widetilde{N}^{\perp}_{e,i} = \sum_{m \in \mathcal{M}_i} \sum_{M' \leqslant e} \widetilde{N}^{\perp}_{M',e,m,i} = \sum_{m \in \mathcal{M}_i} \widetilde{N}^{\perp}_{e,i,m}\ ,$$

where

$$\widetilde{N}^{\perp}_{e,i,m} = \sum_{M' \leqslant e} \widetilde{N}^{\perp}_{M',e,m,i}\ ,$$

is a binary variable taking the value one if and only if $(\gamma(e) \oplus m)[1 \ldots r] = Y_i$ and there exists some $M' \leqslant e$ such that $\lambda(M') = \lambda(M)$.

Therefore, once again by linearity, it is enough to compute $\mathsf{E}[\widetilde{N}^{\perp}_{e,m,i}] = \mathsf{Pr}\left[\widetilde{N}^{\perp}_{e,m,i} = 1\right]$ for all e. Note that if this is the case, this means that there exists some message M_i (whose corresponding node was traversed before getting to e and M), such that $\lambda(M)$ collides with one of the internal values in the computation of $\mathsf{CBC}(M_i)$, or in other words (using the language of [5]), this would imply a *full* collision in computing $\mathsf{CBC}(M)$ and $\mathsf{CBC}(M_i)$: Formally, a full collision means that $\mathsf{CBC}(M)$ collides with one of the other state values generated in the computation of $\mathsf{CBC}(M_i)$ and $\mathsf{CBC}(M)$, also cf. Appendix C above for a definition, and the associated full collision probability $\mathsf{FCP}_n(M_i, M)$. Therefore,

$$\mathsf{E}[\widetilde{N}^{\perp}_{e,m,i}] = \mathsf{Pr}\left[\widetilde{N}^{\perp}_{e,m,i} = 1\right] \leqslant \sum_{M_i < M} \mathsf{FCP}_n(M_i, M) \leqslant \frac{8q\ell}{2^n} + \frac{64q\ell^4}{2^{2n}} \; .$$

Note that [5] only gives bounds for $\mathsf{FCP}_n(M_i, M)$ when M_i is *not* a prefix of M, but the bound trivially extends, since if M_i is a prefix of M, then a full collision implies a full collision for M and any other non-prefix message M'. Thus, to conclude, we observe that

$$\mu = \sum_{i=1}^{q} \sum_{m \in \mathcal{M}_i} \sum_{e \in E'} \mathsf{E}[\widetilde{N}^{\perp}_{e,m,i}] \leqslant \frac{8q^3\ell^2}{2^n} + \frac{64q^3\ell^5}{2^{2n}} \; ,$$

where we have used the fact that $\sum_{i=1}^{q} D_i \leqslant q$. Thus, by Markov's inequality,

$$\mathsf{Pr}\left[\sum_{e \in E'} \sum_{i=1}^{q} \widetilde{N}^{\perp}_{e,i} \geqslant q^2\ell/2^r + 8q^2\ell^4/2^{n+r}\right] \leqslant \frac{8q\ell}{2^{n-r}} \; .$$

This concludes the proof of Lemma 8.

Remark. In the above proof, one would expect that $\mathsf{E}[\widetilde{N}^{\perp}_{e,m,i}]$ is much smaller (perhaps even by a multiplicative factor 2^r). In fact, we are essentially assuming that as soon as collision occur, then this collision implies $\widetilde{N}^{\perp}_{e,m,i} = 1$. While this may not always be the case, it is very hard to argue about the distribution of $\mathsf{CBC}(M)$ conditioned on the computation $\mathsf{CBC}(M_i)$ and $\mathsf{CBC}(M)$ provoking a full collision. In particular, one can build examples showing that it is *not* uniform, and in general, very badly understood.

E Tightness of bound for GSponge – Missing Analysis

For the analysis, in the real world we consider the event COLL that for some $i \in [Q_1]$ and $i' \in [Q_2]$ we have $\pi(K' \oplus M_i)[r+1 \ldots n] = B_{i',0}[r+1 \ldots n]$ and observe that $\mathsf{Pr}[COLL] \geqslant \Omega(Q_1Q_2/2^{n-r})$ (assuming $Q_1 \ll 2^{n-r}$). Again, if COLL occurs due to some indices i and i' then by construction $Y_{i,j} = B_{i',j}$ for all $j \in [t]$ and $\mathcal{A}$ outputs 1. Therefore,

$$\mathsf{Pr}\left[\mathcal{A}^{\mathsf{GSponge}[\pi],\pi,\pi^{-1}} \Rightarrow 1\right] \geqslant \mathsf{Pr}\left[\mathsf{COLL}\right] = \Omega(Q_1Q_2/2^{n-r}) \; .$$

However, if $\mathsf{O} = \mathsf{R}$ for a truly random function $\mathsf{R} : \{0,1\}^* \rightarrow \{0,1\}^r$, all values $Y_{i,j}$'s are independent random r-bit strings, since the messages M_i are distinct. Hence, by union bound, suitable i and i' exist with probability at most

$$\mathsf{Pr}\left[\mathcal{A}^{\mathsf{R},\pi,\pi^{-1}} \Rightarrow 1\right] \leqslant Q_1Q_22^{-rt} \leqslant Q_1Q_22^{-n} \; .$$

F Proof of Theorem 2 (sketch)

In this Appendix, we discuss the changes that are necessary to adapt the proof approach we used to analyze truncated CBC to the setting of sponges-based MACs.

The transcripts. First, since the adversary is now allowed to query π, we will also include its π-queries and the respective answers in the transcript. Hence, if (A_i, B_i) for $i \in \{1, \ldots, q_\pi\}$ represents the π-queries asked by $\mathcal{A}$ and the respective responses (i.e., for each i we have $\pi(A_i) = B_i$ and $\mathcal{A}$ either asked a forward π-query A_i or a backward π-query B_i) then the transcript in both the real and ideal world will also contain the list $(A_1, B_1), \ldots, (A_{q_\pi}, B_{q_\pi})$.

Moreover, we also modify the definitions of the full and reduced message tree. The full message tree changes in one aspect: the label of the root vertex $\lambda(\varepsilon)$ will be set to the key $K = K'$ instead of the initialization vector IV, while the rest of the tree labeling is computed from this root label identically as in the proof of Theorem 1 and the same process is applied to obtain its reduced version. Since the full and reduced message trees now depend on K, we will denote them as $T^{\pi,K}(M_1, \ldots, M_{q_C})$ and $\overline{T}^{\pi,K}(M_1, \ldots, M_{q_C})$, respectively.

Finally, we add one additional bit into the transcript, carrying the information whether an overlap has occurred between the π-queries asked directly by the adversary, and the π-queries that were needed to compute the labelling of the (full) message tree. Formally, we define PiHit as

$$(\mathsf{PiHit} = 1) :\Leftrightarrow (\exists M \in V : \lambda(M) \in \{B_1, \ldots, B_{q_\pi}\}) ,$$

where λ denotes the non-reduced labelling function. Since the bit PiHit is determined by giving a labelled message tree T and a list of π-queries $\{(A_i, B_i)\}_{i=1}^{q_\pi}$, we sometimes use the notation $\mathsf{PiHit}(T, \{(A_i, B_i)\}_{i=1}^{q_\pi})$ to give these explicitly.

Hence, both the real and the ideal transcript will have the form

$$\tau = ((A_1, B_1), \ldots, (A_{q_\pi}, B_{q_\pi}), (M_1, Y_1), \ldots, (M_{q_C}, Y_{q_C}), \overline{T}^{\pi,K}(M_1, \ldots, M_{q_C}), \mathsf{PiHit}) , \quad (14)$$

where the pairs (M_i, Y_i) represent the adversary's construction queries, just like before. The real-world (resp. ideal-world) experiment generating the real-world transcript $\mathsf{T}_{\mathsf{real}}$ (resp. the ideal-world transcript $\mathsf{T}_{\mathsf{ideal}}$) is defined as follows:

Real world. The transcript $\mathsf{T}_{\mathsf{real}}$ for the adversary $\mathcal{A}$ is obtained by sampling π uniformly at random from the set of permutations on $\{0,1\}^n$ and a uniform key $K \leftarrow \{0,1\}^n$, and letting

$$\mathsf{T}_{\mathsf{real}} = ((A_1, B_1), \ldots, (A_{q_\pi}, B_{q_\pi}), (M_1, Y_1), \ldots, (M_{q_C}, Y_{q_C}), \overline{T}^{\pi,K}(M_1, \ldots, M_{q_C}), \mathsf{PiHit}) , \quad (15)$$

where we execute $\mathcal{A}$, which adaptively asks construction queries $M_1, \ldots, M_{q_C}$ answered with $Y_i = \mathsf{GSponge}[\pi]_K(M_i)$ for all $i \in [q_C]$, and q_π π-queries that result in input-output pairs $(A_1, B_1), \ldots, (A_{q_\pi}, B_{q_\pi})$, and then we compute $\overline{T} = \overline{T}^{\pi,K}(M_1, \ldots, M_q)$ and use its non-reduced version $T^{\pi,K}(M_1, \ldots, M_q)$ and $\{(A_i, B_i)\}_{i=1}^{q_\pi}$ to determine PiHit. This time $\mathsf{T}_{\mathsf{real}}$ only depends on π and K, hence we sometimes write $\mathsf{T}_{\mathsf{real}}(\pi, K)$ for the corresponding map.

Ideal world. The transcript $\mathsf{T}_{\mathsf{ideal}}$ for the adversary $\mathcal{A}$ is obtained similarly to the above, but here we sample both a random permutation π and q_C independent random values $Y_1, \ldots, Y_{q_C} \in \{0,1\}^r$ and we let

$$\mathsf{T}_{\mathsf{ideal}} = ((A_1, B_1), \ldots, (A_{q_\pi}, B_{q_\pi}), (M_1, Y_1), \ldots, (M_q, Y_q), \overline{T}^{\pi,K}(M_1, \ldots, M_{q_C}), \mathsf{PiHit}) ,$$

where we execute $\mathcal{A}$, which adaptively asks construction queries $M_1, \ldots, M_q$ answered with Y_i for all $i \in [q]$, and q_π π-queries that result in input-output pairs $(A_1, B_1), \ldots, (A_{q_\pi}, B_{q_\pi})$, and then we compute $\overline{T} = \overline{T}^{\pi,K}(M_1, \ldots, M_q)$ for some independent, randomly chosen K; and use its non-reduced version $T^{\pi,K}(M_1, \ldots, M_q)$ and $\{(A_i, B_i)\}_{i=1}^{q_\pi}$ to determine PiHit.

The proof again proceeds using the H-coefficient method (cf. Lemma 1), hence we need to define good and bad transcripts. A transcript (14) is good if it fulfills the requirements **(1)** - **(3)** from Definition 2 (with the natural substitution of q_C instead of q) and also the following additional requirement:

(4) We have $\mathsf{PiHit} = 0$.

We again denote the sets of good and bad transcripts by GT and BT respectively, this time we have $\mathsf{BT} = \bigcup_{i=1}^{4} \mathsf{BT}_i$.

Bounding bad transcripts. We have:

$$\begin{aligned}\mathsf{Pr}[\mathsf{T}_{\mathsf{ideal}} \in \mathsf{BT}] &= \mathsf{Pr}[\mathsf{T}_{\mathsf{ideal}} \in \bigcup_{i=1}^{4} \mathsf{BT}_i] \\ &= \mathsf{Pr}[\mathsf{T}_{\mathsf{ideal}} \in \mathsf{BT}_4] + \sum_{i=1}^{3} \mathsf{Pr}[\mathsf{T}_{\mathsf{ideal}} \in \mathsf{BT}_i \mid \mathsf{T}_{\mathsf{ideal}} \notin \mathsf{BT}_4] \\ &= \mathsf{Pr}[\mathsf{PiHit} = 1] + \sum_{i=1}^{3} \mathsf{Pr}[\mathsf{T}_{\mathsf{ideal}} \in \mathsf{BT}_i \mid \mathsf{PiHit} = 0] \,.\end{aligned}$$

The probability $\mathsf{Pr}[\mathsf{PiHit} = 1]$ can be bounded easily.

Lemma 9. *In the ideal world we have* $\mathsf{Pr}[\mathsf{PiHit} = 1] \leqslant \ell q_\pi q_C / 2^n$.

Proof. For each $M \in V$ and $i \in \{1, \ldots, q_\pi\}$ let $\mathsf{PiHit}_{M,i}$ denote the event that $\lambda(M) = B_i$ for the non-reduced labelling λ. By definition of $\mathsf{GSponge}[\pi]_K$ we know that there exists a permutation ρ independent of K such that $\lambda(M) = \rho(K)$, namely ρ corresponds to the evaluation of the sponge on a fixed input M, from the initial state K. Since the key K is chosen uniformly at random and independently of the view of the adversary, the value $\lambda(M)$ will also be uniformly random and independent of the adversarial π-queries. Hence $\mathsf{Pr}[\mathsf{PiHit}_{M,i}] = 2^{-n}$ and by union bound over all (M, i) we have $\mathsf{Pr}[\mathsf{PiHit} = 1] \leqslant \ell q_\pi q_C / 2^n$. □

Next we need to upper-bound the probabilities $\mathsf{Pr}[\mathsf{T}_{\mathsf{ideal}} \in \mathsf{BT}_i \mid \mathsf{PiHit} = 0]$ for $i \in \{1, \ldots, 3\}$.

Lemma 10. *We have:*

1. $\mathsf{Pr}[\mathsf{T}_{\mathsf{ideal}} \in \mathsf{BT}_1 \mid \mathsf{PiHit} = 0] \leqslant 16\ell q_C^2/2^n + 128\ell^4 q_C^2/2^{2n} + 2\ell q_C q_\pi/2^n$.
2. *For any parameter* $t \geqslant 1$ *defining* BT_2,
$$\mathsf{Pr}[\mathsf{T}_{\mathsf{ideal}} \in \mathsf{BT}_2 \mid \mathsf{PiHit} = 0] \leqslant \frac{q_C}{2^n} + \frac{(q_C \cdot \ell)^{t+1}}{2^{nt}} + \frac{2\ell q_C q_\pi}{2^n} \,.$$
3. *For any parameter* $t \geqslant 1$ *defining* BT_3,
$$\mathsf{Pr}[\mathsf{T}_{\mathsf{ideal}} \in \mathsf{BT}_3 \mid \mathsf{PiHit} = 0] \leqslant \frac{q_C}{2^n} + \frac{8\ell q_C}{2^{n-r}} + \frac{(q_C \cdot \ell)^{t+1}}{2^{nt}} + \frac{2\ell q_C q_\pi}{2^n} \,.$$

Proof (sketch). The bounds can be established from Lemmas 3–5 in a generic way. Namely, since we are conditioning on $\mathsf{PiHit} = 0$ we know that there will be no overlap between the π-queries asked by the adversary and those that are needed to label the full message tree T. Second, to get closer to the TCBC setting, we consider the key K to be fixed and only average over its choice in the end. This means that the argument can follow the same path as in the case with the all-zero initial state and no adversarial π-queries, with a small difference: now, whenever π is lazy-sampled at a fresh point, instead of returning a uniform element from the set of all $2^n - i$ unused values, it will instead sample from the smaller set of $2^n - q_\pi - i$ values, since the query-answer pairs asked by the adversary also have to be avoided to maintain $\mathsf{PiHit} = 0$. The statistical distance between the old and new distribution at every such sampling is hence upper-bounded by

$$\frac{q_\pi}{2^n - i} < \frac{q_\pi}{2^n - q_C} < \frac{2q_\pi}{2^n} ,$$

assuming $q_C \leqslant 2^{n-1}$ as otherwise the bounds are trivial. As π is always invoked at most ℓq_C times during the labeling of T, the probability of a bad transcript ocurring in our setting differs from the probability of its ocurrence in the settings considered in Lemmas 3–5 by at most $2\ell q_C q_\pi / 2^n$. □

Lower-bounding the probability ratio. It now remains to bound the ratio

$$\Pr\left[\mathsf{T}_{\mathsf{real}} = \tau\right] / \Pr\left[\mathsf{T}_{\mathsf{ideal}} = \tau\right]$$

for all good transcripts τ. Towards this, we proceed as in Section 4.4, describing the necessary modifications along the way.

We start by observing that since the padding function $\mathsf{pad}(\cdot)$ only outputs non-empty strings, we have $M_i \neq \varepsilon$ for all $i \in [q_C]$. Hence, by the construction of the reduced message tree $\overline{T}$, we have $\lambda(\varepsilon) \neq \bot$ and therefore $\lambda(\varepsilon) = K$ for every good transcript. Each good transcript τ hence determines the key K and we will sometimes refer to this key by writing $K(\tau)$. Now, similarly as before we can define

$$\Omega[\tau] = \{\pi \,:\, \mathsf{T}_{\mathsf{real}}(\pi, K(\tau)) = \tau\}$$

and

$$\Omega'[\tau] := \left\{\pi \,:\, \overline{T}^{\pi, K(\tau)}(M_1, \ldots, M_{q_C}) = \overline{T}(\tau)\right\} \cap \{\pi \,:\, \forall i \in [q_\pi] : \pi(A_i) = B_i\} \cap$$
$$\cap \left\{\pi \,:\, \mathsf{PiHit}\left(T^{\pi, K(\tau)}(M_1, \ldots, M_{q_C}), \{(A_i, B_i)\}_{i=1}^{q_\pi}\right) = 0\right\}$$

where $\overline{T}(\tau)$ denotes the reduced message tree given in the transcript τ. Intuitively, Ω' contains all permutations π that are consistent with the transcript τ on all its parts except possibly the outputs $Y_1, \ldots, Y_{q_C}$ (note that we have $\mathsf{PiHit} = 0$ in τ since $\tau \in \mathsf{GT}$).

We again denote by $\overline{p}(\tau)$ the ratio

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

and observe that

$$\frac{\Pr\left[\mathsf{T}_{\mathsf{real}} = \tau\right]}{\Pr\left[\mathsf{T}_{\mathsf{ideal}} = \tau\right]} = 2^{r \cdot q_C} \cdot \overline{p}(\tau) ,$$

remains satisfied since this time we have

$$\begin{aligned}\Pr\left[\mathsf{T}_{\mathsf{real}} = \tau\right] &= \Pr\left[K \xleftarrow{\$} \{0,1\}^n : K = K(\tau)\right] \cdot \Pr\left[\pi \xleftarrow{\$} \mathsf{Perm}(n) : \mathsf{T}_{\mathsf{real}}(\pi, K(\tau)) = \tau\right] \\ &= 2^{-n} \cdot \Pr\left[\pi \xleftarrow{\$} \mathsf{Perm}(n) : \pi \in \Omega'[\tau]\right] \cdot \Pr\left[\pi \xleftarrow{\$} \mathsf{Perm}(n) : \pi \in \Omega[\tau] \,\middle|\, \pi \in \Omega'[\tau]\right]\end{aligned}$$

and

$$\Pr\left[\mathsf{T}_{\mathsf{ideal}} = \tau\right] = 2^{-n} \cdot 2^{-r \cdot q_C} \cdot \Pr\left[\pi \xleftarrow{\$} \mathsf{Perm}(n) :\; \pi \in \Omega'[\tau]\right] ,$$

where the first two factors in the latter equation capture the (independent, uniformly random) choice of the key K and the outputs Y_i, respectively.

To lower-bound $\overline{p}(\tau)$, we will again use the fact that

$$\overline{p}(\tau) = \sum_{(z_1,\ldots,z_q):z_i[1\ldots r]=Y_i} \mathsf{Pr}\left[\pi \xleftarrow{\$} \Omega'[\tau] : \forall i \in [q_C] : Z_i = z_i\right]$$

and consider the same process for setting labels $\lambda(M_i)$, $\gamma(M_i, M_i \parallel m)$ and extending the partial permutation $\overline{\pi}$ as in the proof of Theorem 1. We also consider the set $\mathcal{L}$ of all tuples $(z_1, \ldots, z_{q_C})$ of q_C distinct values such that:

- $z_i[1..r] = Y_i$ for all $i \in [q_C]$,
- when assigning $\lambda(M_i) \leftarrow z_i$ for all $i \in [q]$ in the above process, at the end of the process all the labels $\lambda(M_i) = z_i$ are unique within the set of all vertex labels; for all $i \in [q]$ and all $m \in \mathcal{M}_i$ the labels $\lambda(M_i \parallel m)$ are unique in the same sense; and $\mathsf{PiHit} = 0$.

The lower bound on $\overline{p}(\tau)$ is then obtained by giving a lower bound on:

- the probability that any particular tuple from $\mathcal{L}$ appears as the labels for vertices $(M_1, \ldots, M_{q_C})$ when π is chosen at random from $\Omega'[\tau]$; and
- the size of the set $\mathcal{L}$.

For the former, one can again for any $(z_1, \ldots, z_{q_C}) \in \mathcal{L}$ establish the bound

$$\mathsf{Pr}\left[\pi \xleftarrow{\$} \Omega'[\tau] : \forall i \in [q_C] : \pi(\gamma(e_i)) = z_i\right] \geqslant \frac{1}{2^{nq_C}}$$

using an analogous argument as in the case considered for Theorem 1. Namely, we again have

$$\mathsf{Pr}\left[\pi \xleftarrow{\$} \Omega'[\tau] : \forall i \in [q_C] : \pi(\gamma(e_i)) = z_i\right] = \frac{|\{\pi \in \Omega'[\tau] \ : \ \forall i \in [q_C] : \pi(x_i) = z_i\}|}{|\Omega'[\tau]|}$$

and it remains to bound the numbers of permutations in the sets in both the numerator and the denominator. Here we additionally (in contrast to the proof of Theorem 1) have to take into account that the permutations we are counting are always defined on additional q_π points to comply with the π-queries listed in τ. However, this affects both counts equally and hence cancels out.

To lower-bound $|\mathcal{L}|$, we will again be choosing the values z_i one by one and for every i give a lower bound on the number of admissible values z_i. Namely, we will exclude a candidate value for z_i if it satisfies any of the properties **(1)**–**(5)** given on page 15 or the additional properties **(6)**–**(7)** defined as follows:

(6) There exists an adversarial π-query (A_j, B_j) such that $z_i = B_j$.
(7) There exists an adversarial π-query (A_j, B_j) and an $m \in \mathcal{M}_i$ such that $z_i \oplus m = A_j$.

The numbers of excluded values due to conditions **(1)**–**(5)** are estimated in the same way as in the proof of Theorem 1, while for conditions **(6)**–**(7)** we need to exclude additional at most q_π and $D_i q_\pi$ values, respectively. Redoing the computation (8), we arrive at

$$|\mathcal{L}| \geqslant 2^{q_C \cdot (n-r)} \cdot \left(1 - \frac{N^{(1)} + N^{(2)}}{2^{n-r}} - \frac{2q_C^2}{2^{n-r}} - \frac{2q_\pi q_C}{2^{n-r}}\right) .$$

Putting pieces together. The above gives us

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

and by plugging this into Lemma 1 together with the results of Lemma 9 and Lemma 10, we obtain

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

for any parameter $t \geqslant 1$. □

G Proof of Theorem 3

In Theorem 2 we proved the PRF security of the $\mathsf{GSponge}$ construction, here we will explain how to extend the proof to the case where the key K' is not uniform, but generated from w short b-bit keys $K[1], \ldots, K[w]$ as illustrated in Figure 3, i.e,

$$\text{set } V_0 := 0^n \text{ then compute } V_i = \pi((K[i] \,\|\, 0^{n-b}) \oplus V_{i-1}) \text{ for } i = 1, \ldots, w \text{ and set } K' := V_w. \quad (16)$$

For this we define the transcript $\mathsf{T}'_{\mathsf{real}}$ for Sponge in almost the same way as we defined the transcript $\mathsf{T}_{\mathsf{real}}$ for $\mathsf{GSponge}$ (cf eq.(15) and the paragraphs above and below it), but where instead of sampling a random n-bit string K and then setting $Y_i = \mathsf{GSponge}[\pi]_K(M_i)$, we now sample a key $K = (K[1], \ldots, K[w])$ that consists of a w-tuple of random b-bit strings, and compute K' from K and π as in (16). Note that we do not include the inputs/outputs of the w extra invocation of π required to compute K' into the transcript $\mathsf{T}'_{\mathsf{real}}$, so the transcripts for Sponge and $\mathsf{GSponge}$ are defined over the same domain. By the lemma below, these transcripts are also statistically close:

Lemma 11. *The statistical distance of the real transcript* $\mathsf{T}_{\mathsf{real}}$ *for* $\mathsf{GSponge}$ *as defined in (15) and* $\mathsf{T}'_{\mathsf{real}}$ *for* Sponge *as defined above is (below $q = q_\pi + \ell q_C$ denotes an upper bound on the total number of invocations of π in the attack on* Sponge*)*

$$\mathsf{SD}(\mathsf{T}'_{\mathsf{real}}, \mathsf{T}_{\mathsf{real}}) \leqslant \frac{q}{2^{bw}} + \frac{wq}{2^n} + \frac{q^2}{2^{n-b}}$$

If $w = 1$, one can remove the $q^2/2^{n-b}$ term above. Assuming $q \leqslant 2^{n-b}$ we can remove this term at the cost of increasing the first

$$\mathsf{SD}(\mathsf{T}'_{\mathsf{real}}, \mathsf{T}_{\mathsf{real}}) \leqslant \frac{q}{2^{\frac{b-\log(3n)-1}{2}w}} + \frac{wq}{2^n}$$

Before we prove this lemma, we observe that together with Theorem 2, this implies Theorem 3. We have

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{Sponge}_{r,\mathsf{pad}_b},\pi}(\mathcal{A}) \;\leqslant \mathsf{SD}(\mathsf{T}'_{\mathsf{real}}, \mathsf{T}_{\mathsf{ideal}}) \leqslant \mathsf{SD}(\mathsf{T}'_{\mathsf{real}}, \mathsf{T}_{\mathsf{real}}) + \mathsf{SD}(\mathsf{T}_{\mathsf{real}}, \mathsf{T}_{\mathsf{ideal}})$$

and we can bound the last two terms with the bounds from Lemma 11 and Theorem 2, respectively. Note that we didn't state the $q \leqslant 2^{n-b}$ condition from Lemma 11 in the theorem because if this condition is not satisfied, the theorem is void anyway.

Proof (of Lemma 11). Below we define a way of sampling a joint distribution $(\mathsf{T}, \mathsf{aux})$. There are two distinguished values $\perp_1, \perp_2$ in the support of aux, and we will sometimes write $\mathsf{aux} = \perp$ for $\mathsf{aux} \in \{\perp_1, \perp_2\}$. The distribution $(\mathsf{T}, \mathsf{aux})$ will satisfy the following three conditions:

(1) $\mathsf{SD}(\mathsf{T}, \mathsf{T}_{\mathsf{real}}) = 0$, so the marginal distribution T is the same as real $\mathsf{GSponge}$ transcripts.

(2) For any τ^* we have

$$\Pr[(\tau, \alpha) \leftarrow (\mathsf{T}, \mathsf{aux}) \;:\; (\tau = \tau^*) \wedge (\mathsf{aux} \neq \perp)] \leqslant \Pr[\mathsf{T}'_{\mathsf{real}} \;:\; (\tau = \tau^*)]\,. \quad (17)$$

(3) The probability that the auxiliary information is $\perp$ is at most

$$\Pr[(\tau, \alpha) \leftarrow (\mathsf{T}, \mathsf{aux}) \;:\; \mathsf{aux} = \perp] \leqslant q^2/2^{n-b} + q/2^{bw} + wq/2^n\;,$$

or alternatively

$$\Pr[(\tau, \alpha) \leftarrow (\mathsf{T}, \mathsf{aux}) \;:\; \mathsf{aux} = \perp] \leqslant q/2^{\frac{b-\log(3n)-1}{2}w} + wq/2^n\;.$$

Note that the three points above imply the Lemma with

$$\mathsf{SD}(\mathsf{T}_{\mathsf{real}}, \mathsf{T}'_{\mathsf{real}}) = \mathsf{SD}(\mathsf{T}, \mathsf{T}'_{\mathsf{real}}) \leqslant \Pr[(\tau, \alpha) \leftarrow (\mathsf{T}, \mathsf{aux}) \;:\; \mathsf{aux} = \perp]$$

where the last equality above follows as for any random variables X, Y and any event E we have that if for all v in the domain $\Pr[X = v \wedge E] \leqslant \Pr[Y = v]$ then $\mathsf{SD}(X, Y) \leqslant \Pr[\neg E]$.

Sampling $(\mathsf{T}, \mathsf{aux})$. The augmented transcript distribution $(\mathsf{T}, \mathsf{aux})$ is defined as follows. We first sample a transcript $\mathsf{T}_{\mathsf{real}}$ and set $\mathsf{T} = \mathsf{T}_{\mathsf{real}}$ (note that this already implies condition (1) above).

We now define how to sample the second part aux of the distribution. Note that to sample $\mathsf{T} = \mathsf{T}_{\mathsf{real}}$, we have sampled a uniformly random key $K' \in \{0,1\}^n$ and a random permutation π. It will be convenient to think of π being "lazy" sampled, so at this point we only have a partially defined permutation $\overline{\pi}$ that is defined on at most $q_\pi + \ell q_C$ inputs.

We now sample a key $K = K[1], \ldots, K[w]$ consisting of w random b-bit blocks. Next, we check if the partially defined $\overline{\pi}$ allows to compute the key: Starting with $V_0 := 0^n$, compute $V_i = \overline{\pi}((K[i] \,\|\, 0^{n-b}) \oplus V_{i-1})$ for all $i = 1, 2, \ldots$ until either $i = w$ or we get an input on which $\overline{\pi}$ is not yet defined. If $i = w$, i.e., we have computed the key, we set $\mathsf{aux} = \bot_1$ (the reason is that now almost certainly $V_w \neq K'$, and thus we we have a transcript which we cannot make look as if it was generated by Sponge).

Proving Condition (2). Intuitively, if in the process just described we stopped at $i < w$, then the next V_i will be the output of $\overline{\pi}$ on a "fresh" input, and thus close to uniform. At this point, we define the $V_{i+1}, \ldots, V_{w-1}$ uniformly at random and set $V_w = K'$. If any of the $V_{i+1}, \ldots, V_w$ is not "fresh" in the sense that it appears anywhere else in the transcript, we set $\mathsf{aux} = \bot_2$. Otherwise, we set aux to contain the input/output pairs of the w queries to $\overline{\pi}$ made while computing K' (note that this implicitly also defines the key $K = K[1], \ldots, K[w]$).

To prove condition (2) we now also augment the transcripts of $\mathsf{GSponge}$ to get a distribution $(\mathsf{T}'_{\mathsf{real}}, \mathsf{aux}')$, where aux' contains the w queries made to π while computing K'. To prove (2) we'll show that for any transcript (τ^*, α^*) in the support of $(\mathsf{T}'_{\mathsf{real}}, \mathsf{aux}')$ we have

$$\mathsf{Pr}[(\tau, \alpha) \leftarrow (\mathsf{T}, \mathsf{aux}) \;:\; (\tau^*, \alpha^*) = (\tau, \alpha)] \leqslant \mathsf{Pr}[(\tau, \alpha) \leftarrow (\mathsf{T}'_{\mathsf{real}}, \mathsf{aux}') \;:\; (\tau^*, \alpha^*) = (\tau, \alpha)]\,. \qquad (18)$$

With this, (17) follows by taking the sum over all α^* on both sides of the above equation.

Note that we only must consider transcripts (τ^*, α^*) where during the computation of K' from K we made a "fresh" query (otherwise $\mathsf{aux} = \bot$, but aux' is never $\bot$, so it's not in the support of $(\mathsf{T}'_{\mathsf{real}}, \mathsf{aux}')$). Consider any such fixed transcript (τ^*, α^*), and assume we sample $(\mathsf{T}'_{\mathsf{real}}, \mathsf{aux}')$ or $(\mathsf{T}, \mathsf{aux})$ such that we immediately abort as soon as we're inconsistent with (τ^*, α^*), and we sample the "fresh" outputs $V_{i+1}, \ldots, V_{w-1}$ at the very end. Here, the sampling of $(\mathsf{T}'_{\mathsf{real}}, \mathsf{aux}')$ and $(\mathsf{T}, \mathsf{aux})$ is identical (and thus has the same probability of being consistent with (τ^*, α^*)) up to the point where we must sample the $V_{i+1}, \ldots, V_{w-1}$. In $(\mathsf{T}, \mathsf{aux})$ these are sampled uniformly at random, whereas when sampling $(\mathsf{T}'_{\mathsf{real}}, \mathsf{aux}')$, this values are implicitly defined as we sample the outputs of $\overline{\pi}$ on the fresh inputs $V_j = \overline{\pi}((K[j] \| 0^{n-b}) \oplus V_{j-1}$ for $j = i+1, \ldots, w-1$. The probability of sampling consistently in the latter case is at least as high as sampling the V_j's uniformly, as now the sampling space is smaller (because $\overline{\pi}$ is already defined on some outputs, and these can now be excluded), this proves condition (2).

Proving Condition (3). It remains to upper bound the probability that $\mathsf{aux} \in \{\bot_1, \bot_2\}$ to prove condition (3). For $\bot_2$ we get the following simple claim

Claim. Let $q = q_\pi + \ell q_C$ be an upper bound on the total number of invocations of π. Then

$$\mathsf{Pr}[\mathsf{aux} = \bot_2] \leqslant \frac{wq}{2^n}$$

Proof. The probability that any particular of the uniformly sampled V_i's "hits" any of the at most q values on which $\overline{\pi}$ is already defined is $q/2^n$. As we sample at most w of them, we get the claimed bound by the union bound. □

The bound on the probability of $\mathsf{aux} = \bot_1$ is a bit more tedious, and we outsourced it to Lemma 12 below. Note that the above claim with Lemma 12 prove condition (3). □

Lemma 12. *Let $q = q_\pi + \ell q_C$, then*

$$\mathsf{Pr}[\mathsf{aux} = \bot_1] \leqslant \frac{q^2}{2^{n-b}} + \frac{q}{2^{bw}} \tag{19}$$

if $w = 1$ (i.e., the key is just one b-bit block) then we can ignore the $\frac{q^2}{2^{n-b}}$ term above. If $q \leqslant 2^{n-b}$ we can ignore this term, at the prize of increasing the 2nd

$$\mathsf{Pr}[\mathsf{aux} = \bot_1] \leqslant 2^{-n} + \frac{q \cdot n^w}{2^{bw}} = 2^{-n} + \frac{q}{2^{\frac{b-\log(3n)-1}{2}w}} \; . \tag{20}$$

Proof (of Lemma 12). Consider the sampling of $(\mathsf{T}, \mathsf{aux})$ right after we sampled the key $K = (K[1], \ldots, K[w])$. We will say that K is *fixed* if we can compute $K' = V_w$ without having to define $\overline{\pi}$ on new points (i.e., with $V_0 := 0^n$, we can compute $V_i = \overline{\pi}((K[i] \,\|\, 0^{n-b}) \oplus V_{i-1})$ for $i = 1, \ldots, w$). Let $\#K$ denote the number of fixed keys, note that with this we can express the probability of $\mathsf{aux} = \bot_1$ as

$$\mathsf{Pr}[\mathsf{aux} = \bot_1] = \frac{\#K}{\#\text{ of keys}} = \frac{\#K}{2^{bw}} \; .$$

So it remains to bound $\#K$. Below, we will first define an event γ_0 (think of γ_0 as a boolean variable where $\gamma_0 = 0$ which means the condition holds, and $\gamma_0 = 1$ means it failed), where conditioned on γ_0 holding, the number of fixed keys $\#K$ can be upper bounded with q. We'll show that γ_0 fails with probability at most $q^2/2^{n-b}$, which then gives us the bound in (19).

Unfortunately, the $q^2/2^{n-b}$ term is quite large, in particular, it would dominate our bounds for Sponge. To get rid of this term, we generalise the event γ_0 to γ_m for any $m \in \mathbb{N}$. For γ_{3n-1} we can show that it fails with only extremely small probability $< 2^{-n}$, and conditioned on γ_{3n-1} we can still upper bound the number of fixed keys $\#K$ with qn^w (as opposed to q under γ_0). This will then give us the bound (20).

The event γ_0. The event γ_0 fails if during the experiment (during which we made at most q invocations to $\overline{\pi}$) we made a forward or backward query, where the output collided with some previous value on the last $n - b$ bits. Concretely, $\gamma_0 = 1$ if at some point we made either a fresh forward query X and got the output $Y \leftarrow \overline{\pi}(X)$ where $\overline{\pi}$ was already defined on some (X', Y') satisfying $Y[b+1 \ldots n] = Y'[b+1 \ldots n]$, or an inverse query Y and got the output $X \leftarrow \overline{\pi}^{-1}(Y)$ where some (X', Y') satisfies $X[b+1 \ldots n] = X'[b+1 \ldots n]$. The probability of γ_0 failing can be bounded by a standard birthday bound

$$\mathsf{Pr}[\gamma_0 = 1] \leqslant q^2/2^{n-b} \; . \tag{21}$$

Moreover, we claim that

$$\mathsf{Pr}[\mathsf{aux} = \bot_1 \mid \gamma_0 = 0] \leqslant \#K/2^{bw} \leqslant q/2^{bw} \; . \tag{22}$$

We postpone the proof of (22), and note that now using that

$$\begin{aligned} \mathsf{Pr}[\mathsf{aux} = \bot_1] &= \mathsf{Pr}[\mathsf{aux} = \bot_1 \wedge \gamma_m = 1] + \mathsf{Pr}[\mathsf{aux} = \bot_1 \wedge \gamma_m = 0] \\ &\leqslant \mathsf{Pr}[\gamma_m = 1] + \mathsf{Pr}[\mathsf{aux} = \bot_1 \mid \gamma_m = 0] \end{aligned} \tag{23}$$

implies the first bound (19) in the statement of the claim.

The event γ_m. For $m \in \mathbb{N}^+$, the event γ_m fails if we made $m+1$ or more forward queries (or $m+1$ backward queries) that collide on the last $n-b$ bits. More precisely, $\gamma_m = 1$ if during the experiment the permutation $\overline{\pi}$ was invoked on $m+1$ forward queries $X_0 \dots X_m$ which resulted in answers $Y_i \leftarrow \overline{\pi}(X_i)$ that all had the same last $n-b$ bits, similarly for backward queries. We claim the following:

Claim.

$$\mathsf{Pr}[\gamma_{3n-1} = 1] \leqslant 2^{-n} \ . \tag{24}$$

Proof (of Claim). This follows by a Chernoff bound: Assume we made a forward query $Y_0 \leftarrow \overline{\pi}(X_0)$ (backwards queries are proven similarly) to $\overline{\pi}$. Subsequently, we make at most q other forward queries, and for any of those queries, the probability that their output collides with Y_0 on the last $n-b$ bits is $1/2^{n-b}$.[14] The expected number of such collisions μ is thus $\leqslant q/2^{n-b} \leqslant 1$ (as we assume $q \leqslant 2^{n-b}$). Using Theorem 4 with $\delta = 3n-1$ and $\mu \leqslant 1$,

$$\mathsf{Pr}[\# \text{ of collisions} \geqslant 3n] \leqslant e^{-(3n-1)^2/(3n+1)} \leqslant 2^{-2n} \ .$$

The above bounds the probability of a particular query being "hit" by $3n-1$ or more subsequent queries. Taking the union bound over all $q \leqslant 2^{n-b} < 2^n$ such queries proves (24) as $2^{-2n}2^n = 2^{-n}$. □

It remains to prove that conditioned on $\gamma_{3n-1} = 0$, the number of fixed keys can be bounded as

$$\mathsf{Pr}[\mathsf{aux} = \bot_1 \mid \gamma_{3n-1} = 0] \leqslant \frac{\#K}{2^{bw}} \leqslant \frac{q2^w(2^{\frac{bw}{2}} \cdot (3n-1)^{\frac{w}{2}})}{2^{bw}} \leqslant \frac{q}{2^{\frac{(b-\log(3n)w}{2}-1}} \ . \tag{25}$$

Before proving this, note that plugging in the bounds (25) and (24) to (23) implies statement (20) of the claim.

Bounding $\#T$ assuming $\gamma_m = 0$. It remains to prove (22) and (25). For this we build a tree from the partially defined $\overline{\pi}$ which will capture the possible computations of K' from $K = (K[1], \dots, K[w])$ as follows:

- The root of the tree is v and we assign the label 0^n to it (we will write v' to denote the label of v and e' to denote the label of an edge e).
- For every $k_1 \in \{0,1\}^b$ on which $\overline{\pi}((k_1 \| 0^{n-b}) \oplus v)$ is defined, we add a node v_{k_1} with label $v'_{k_1} = \overline{\pi}((k_1\|0^{n-b}) \oplus v)$ and a directed edge $e_{k_1} = (v, v_{k_1})$ with label $e'_{k_1} = (k_1\|0^{n-b} \oplus v)$.
- We continue to build this tree level by level. Assume we build level $i < w$ (the root is level 0, and we just described how to build the nodes on level 1).
  The nodes on level i are named $v_{k_1,\dots,k_i}$ (each $k_j \in \{0,1\}^b$), consider any such node. If there exists $k_{i+1} \in \{0,1\}^b$ s.t. $\overline{\pi}((k_{i+1}\|0^{n-b}) \oplus v'_{k_1,\dots,k_i})$ is defined, then add a node $v_{k_1,\dots,k_i,k_{i+1}}$ with label $\overline{\pi}((k_{i+1}\|0^{n-b}) \oplus v'_{k_1,\dots,k_{i+1}})$ and connect $v_{k_1,\dots,k_i}$ and $v_{k_1,\dots,k_{i+1}}$ with an edge $e_{k_1,\dots,k_{i+1}}$ with label $(k_{i+1}\|0^{n-b}) \oplus v'_{k_1,\dots,k_{i+1}}$.

Note that the computation of a key K' from a fixed key $(K[1], \dots, K[w])$ corresponds to a path from the root to the node $v_{K[1],\dots,K[w]}$, and the label of this node is K'. So, every leave of this tree at the last level w correspond to a fixed key.

[14] This is not exactly true, as π is a permutation not a random function, but this doesn't matter as it will finally bound the probability in the right direction.

The collapsed Tree Graph $G_{\overline{\pi}}$. Consider the tree we just defined, then we denote with $G_{\overline{\pi}}$ the (directed, loop-free) graph we get when merging all nodes with the same label in the same layer of the tree.

The γ_0 Case. To prove (22), we will show that $\#K \leqslant q$ if $\gamma_0 = 0$. This follows from the following two claims:

Claim. If $G_{\overline{\pi}}$ is a tree (i.e., we haven't collapsed any labels), then there are at most q fixed keys.

Proof (of Claim). In $G_{\overline{\pi}}$ every node at level w (recall that there are at most q nodes in total) corresponds to exactly one fixed key, as in a tree there can be only one path from any node to the root. □

Claim. If $\gamma_0 = 0$ then $G_{\overline{\pi}}$ is a tree.

Proof (of Claim). It will be convenient to assume that the very first query made by the adversary was a forward query 0^n to $\overline{\pi}$. Assume $G_{\overline{\pi}}$ is not a tree, we must show that then $\gamma_0 = 1$. As $G_{\overline{\pi}}$ is not a tree, there is at least one vertex v^* with in-degree two. Let v_0 and v_1 be the two nodes that point to v^*, then $v'_0[b+1\ldots n] = v'_1[b+1\ldots n]$.[15] If both outputs v'_0, v'_1 resulted originally from forward queries to $\overline{\pi}$ then by definition $\gamma_0 = 1$ and we're done. So, assume that at least one of them, say v'_0, resulted from an inverse query. Let v_2 be a parent of v_0 and e_0 the edge $v_2 \rightarrow v_0$. Note that $e'_0 \leftarrow \overline{\pi}^{-1}(v'_0)$ is the label of e_0, and that $v'_2[b+1\ldots n] = e'_0[b+1\ldots n]$ (as the label of a vertex and any edge leaving it are always identical on the last $n-b$ bits). We consider 3 possible cases:

- If v_2 is the root, then we made an inverse query whose output ended with 0^{n-b}, and thus we have a collision with the very first forward query and thus $\gamma_0 = 1$ (recall that we assumed the first query is the forward query 0^n).
- If v'_2 was the output of a forward query, then this forward query collides on the last $n-b$ bits with the inverse query on input v'_0 (as explained above) and thus $\gamma_0 = 1$.
- If v_2 is the output of an inverse query, we repeat the above argument with v_2 taking the role of v_0.

In the first two cases above we have $\gamma_0 = 1$, in the last case we walked down one layer in in $G_{\overline{\pi}}$, and we can do this at most w times before hitting the root (and thus land in the first case), so ultimately $\gamma_0 = 1$ will hold. □

The γ_m Case. We just bounded $\#K$ assuming γ_0, we will now show a bound for $\#K$ assuming only the weaker condition γ_m for some $m > 0$ holds.

Recall that with any fixed key $(K[1], \ldots, K[w])$ we can associate a path of length w from the root v to $v_{K[1],\ldots,K[w]}$ in $G_{\overline{\pi}}$, and each edge in $G_{\overline{\pi}}$ corresponds to a query (either a forward or a backward query) that was made to $\overline{\pi}$. We define the signature $s \in \{0,1\}^w$ of K as follows: For $j \in \{1\ldots n\}$, $s[j] = 0$ if the query corresponding to the j-th edge on the path was a forward query, and $s[j] = 1$ otherwise.

With $G^s_{\overline{\pi}}$ we denote the subgraph of $G_{\overline{\pi}}$ where we delete every edge e at level i if this edge corresponds to a forward query but $s[i] = 1$ or to a backward query but $s[i] = 0$. Note that any path corresponding to a fixed key K with signature s in $G_{\overline{\pi}}$ is also contained in $G^s_{\overline{\pi}}$. Thus, to upper bound $\#K$ it suffices to upper bound the number of paths in each $G^s_{\overline{\pi}}$ separately, which we do in the following claim.

[15] This holds as for some $x, y \in \{0,1\}^b$ we have $v'_0 \oplus (x \| 0^{n-b}) = v'_1 \oplus (y \| 0^{n-b})$.

Claim. If $\gamma_m = 0$, then for any signature s, any node u (with label u') at level w in $G^s_{\overline{\pi}}$, the number of paths from the root to u (or equivalently, the number of fixed keys with signature s that result in a key $V_w = u'$ using the rule (16)) is at most

$$2^{\frac{bw}{2}} m^{\frac{w}{2}} \ .$$

Proof (of Claim). We will assume that $m < 2^b$ as otherwise the claim is trivial (as the bound is larger than the total number of paths in $G_{\overline{\pi}}$ which is at most 2^{bw}).

As a warm-up, consider the case where the signature is $s = 0^w$, i.e., all forward queries. As $\gamma_m = 0$, we never had a $m+1$-wise collision in forward queries, which implies that the in-degree of any node in $G^s_{\overline{\pi}}$ is at most m. So, the single node u at level w is connected to at most m nodes at level $w-1$, each node in level $w-1$ is connected to at most m nodes at level $w-2$, etc. With every level the number of paths increases by a factor m, and thus when we reach the root (at level 0) we have at most $m^w \leqslant 2^{\frac{bw}{2}} m^{\frac{w}{2}}$ paths.

Let $|s|$ denote the Hamming weight (i.e., the number of 1's) of s. Now consider any s where $|s| \leqslant w/2$. We can make the above argument, that going from level i down to $i-1$ increases the number of paths by a factor at most m for any level i where $s[i] = 0$. For the remaining $|s|$ steps we can't say anything, except the trivial fact that the outdegree of any node is bounded by 2^b, and thus going from level i down to level $i-1$ increases the number of paths by a factor at most 2^b even if $s[i] = 1$. The total number of paths is thus at most

$$2^{b|s|} m^{w-|s|} \leqslant 2^{\frac{bw}{2}} m^{\frac{w}{2}}$$

where we used that $|s| \leqslant w/2$ and $m \leqslant 2^b$.

For the remaining cases where $|s| > w/2$, can do a similar argument, but now we upper bound the number of paths in the other direction, starting at the root going towards u. By definition, $\gamma_m = 0$ implies that for any $i \in [n]$ with $s[i] = 1$, the nodes in $G^s_{\overline{\pi}}$ at level $i-1$ have out-degree at most m (as otherwise we had at least $m+1$ collisions on inverse queries, and thus $\gamma_m = 1$). By a counting argument as before, the number of paths from the root to u can be now upper bounded by

$$2^{b(w-|s|)} m^{|s|} \leqslant 2^{\frac{bw}{2}} m^{\frac{w}{2}}$$

where we used that $|s| > w/2$ and $m \leqslant 2^b$. □

As there are at most q nodes in $G_{\overline{\pi}}$ (and thus also in $G^s_{\overline{\pi}}$ for any s), and 2^w possible signatures, using the above claim we get for $m = 3n - 1$

$$\frac{\#K}{2^{bw}} \leqslant \frac{q \cdot 2^w \cdot 2^{\frac{bw}{2}} (3n-1)^{\frac{w}{2}}}{2^{bw}} \leqslant \frac{q}{2^{\frac{b-\log(3n)-1}{2}w}} \ .$$

This proves (25). □