Tight Generic PRF Security of HMAC and NMAC

Yaobin Shen[1], Xiangyang Zhang[1], Lei Wang[2], and Dawu Gu[2]

[1] School of Informatics, Xiamen University, Xiamen, China
yaobin.shen@xmu.edu.cn, xiangyang.zhang@xmu.edu.cn
[2] Shanghai Jiao Tong University, Shanghai, China
wanglei_hb@sjtu.edu.cn, dwgu@sjtu.edu.cn

Abstract. HMAC and its variant NMAC are among the most widely used methods for keying a cryptographic hash function to obtain a PRF or a MAC. Yet, even after nearly three decades of research, their generic PRF security still remains poorly understood, where the compression function of the underlying hash function is treated as a black box and accessible to the adversary. Although a series of works have exploited compression function queries to mount generic attacks, proving tight bounds on the generic PRF security of HMAC and NMAC remains a challenging open question until now.

In this paper, we establish tight bounds on the generic PRF security of HMAC and NMAC. Our bounds capture the influence of the number of construction queries, the number of compression function queries, and the maximal block length of a message on their security. The proofs are carried out in the multi-user setting and the bounds hold regardless of the number of users. In addition, we present matching attacks to demonstrate that our bounds are essentially tight. Taken together, our results close a longstanding gap in the generic PRF security analysis of HMAC and NMAC.

Keywords: HMAC · NMAC · Generic security · Provable security

1 Introduction

HMAC [10] is the predominant approach for keying a hash function H to obtain a PRF or a MAC. It has been standardized by ANSI [1], IETF [26], ISO/IEC [2], and NIST [34], and is widely deployed in various Internet security protocols such as SSL/TLS, SSH and IPSec with billions of daily users. It computes the output on a key K and a message M as

$$\mathsf{HMAC}(K, M) = H(IV, K \oplus \mathrm{opad} \parallel H(IV, K \oplus \mathrm{ipad} \parallel M)) \ ,$$

where IV is a public initial vector that is fixed as part of the description of H, opad and ipad are two distinct constants. The hash function H underlying HMAC can be directly instantiated with SHA-1, SHA-2 or MD5 that typically follow the Merkle-Damgård paradigm [29,14]. This paradigm extends a compression function $h : \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$ into a hash function $H : \{0,1\}^c \times \{0,1\}^* \to \{0,1\}^c$ by first padding M into b-bit blocks $m_1, \ldots, m_\ell$, and then generating the output $z_\ell = H(IV, M)$ with $IV \in \{0,1\}^c$, where

$$z_0 \leftarrow IV,\ z_i = h(z_{i-1}, m_i) \text{ for } 1 \le i \le \ell\ .$$

NMAC is a cleaner variant of HMAC that computes the output on two keys K_1, K_2, and a message M as

$$\mathsf{NMAC}(K_1, K_2, M) = H(K_2, H(K_1, M))\ .$$

Compared to HMAC, it is slightly less practical, since it replaces the fixed IV of a hash function with a secret key, thereby preventing the hash function from being used in a black-box manner.

Security of HMAC and NMAC. The security of these two construction has been investigated extensively, through both security proofs and cryptanalytic attacks. On the provable security side, HMAC and NMAC were established as secure pseudorandom functions (PRFs) in the standard model [10], with subsequent work extending the result under weaker assumption [7] and providing a tight bound in the uniform setting [21]. However, as noted by Gaži, Pietrzak, and Rybár [21], the bound in the standard model may be overly pessimistic, since it also accounts for highly unnatural instantiations of the underlying compression function h such as the one crafted in their tightness proof. The authors therefore highlight the importance of analyzing the generic PRF security of HMAC in the ideal compression function model, where the compression function is treated as an ideal random function that is accessible to the adversary. Moreover, proofs in the standard model cannot capture the influence of offline compression function queries that are typically exploited in a series of generic attacks against hash-based MACs [31,27,32,24,16,17,15,6,4]. The cost of a generic attack is typically measured in terms of two types of query: the number of queries to the construction, denoted by q; the number of queries to the underlying compression function, denoted by p. These two types of queries are also commonly described as online and offline queries, respectively.

However, the generic PRF distinguishing attack (and also forgery attack) against HMAC and NMAC that comes from the generic attack against iterated MACs by Preneel and Oorschot [33] requires only $2^{c/2}$ construction queries, but no compression function queries. [3] This immediately raises the question: how does the security of HMAC and NMAC degrade by increasing the number of compression function queries, and especially

> *how to prove a tight bound in terms of compression function queries and construction queries on the generic PRF security of* HMAC *and* NMAC*?*

[3] As noted by Gaži et al. [22], the PRF distinguishing attack against NI construction [21] which is a variant of NMAC by using a keyed compression function, can implicitly work for HMAC and NMAC with advantage roughly $q^2\ell/2^c$. Yet, this attack also assumes that the adversary makes no compression function queries.

This question has been aware of by Gaži, Pietrzak, and Tessaro [22]. Yet, they were only able to prove tight bounds on the generic PRF security of variants of HMAC and NMAC, that are called WHMAC and WNMAC, which use additional key material to whiten message blocks before being processed by the compression function. The above question is explicitly posed as a challenging open problem by Gaži et al. [22, Section 1]:

> "*proving tight bounds in terms of compression function and construction queries on the generic PRF security of* NMAC/HMAC *is a challenging open problem, on which little progress has been made.",*

and also highlighted by Bellare, Bernstein, and Tessaro [8,9].

Our contributions. In this work, we prove the generic PRF security of HMAC and NMAC. We show that the established security bounds are tight via matching attacks. Thus, our results close a longstanding gap in the generic PRF security analysis of HMAC and NMAC. Additionally, our proofs are carried out in the multi-user setting, demonstrating that the generic PRF security of HMAC and NMAC does not degrade as the number of users increases. This contrasts with the typical loss incurred when the hybrid argument is used to lift single-user security to the multi-user setting. The multi-user security result is important, especially for HMAC that is widely deployed in security protocols with billions of daily active users. We elaborate on our results in more detail in the following.

Our main result shows that for any adversary making at most q construction queries of maximal block length ℓ, and p compression function queries, the advantage of distinguishing HMAC from a random function is at most

$$\frac{pq\ell}{2^c} + \frac{q^2\ell}{2^c} \ , \tag{1}$$

by omitting lower order terms and constant factors. Note that this bound holds regardless of the number of users N, which can be adaptively chosen by the adversary, and may be as large as q. Our bound implies that besides the number of construction queries, the number of compression function queries and the maximal block length of a message also have certain impact on the PRF security of HMAC.

On the cryptanalytic side, we present two attacks, each matching one of the two terms in Equation (1). The first attack exploits properties of the functional graph of a random function [19,27,5] and matches the term $pq\ell/2^c$. The second attack is the same as the PRF distinguishing attack against NI construction by Gaži, Pietrzak, and Rybár [21]. We turn it into the PRF distinguishing attack against HMAC that matches the term $q^2\ell/2^c$.

NMAC enjoys almost the same generic PRF security bound as HMAC. Although the main proof idea of NMAC is similar to that of HMAC, to obtain a tight bound, it requires a separate proof for NMAC to handle subtle events due to different keying mechanisms.

Our techniques. We use the H-coefficient technique by Patarin [30,12] in our information-theoretic analysis of HMAC and NMAC. On a high level, the central idea of our proof for HMAC lies in a careful analysis of the event that the last compression function call of a construction query is not fresh in the ideal world, covering the following cases: i) input collision among the last compression function calls of construction queries; ii) input collision between the last compression function calls of construction queries and offline compression function queries; iii) input collision between the last compression function calls and internal compression function calls of construction queries. In particular, the analysis of case i) is much more involved. We further define one bad event and two associated events to guarantee that the chain formed by internal compression function calls and leading to the input to the last compression function call either is partially fresh from offline compression function queries, or the number of these chains that has been fully determined by offline compression function queries can be bounded by a carefully chosen threshold. In the former case, we can properly adapt the result by Gaži et al. [21] that considers no compression function queries. In the latter case, we can rely on the randomness of the output of the first compression function call with a secret key to bound the probability that this output strikes one of these chains. Moreover, as we consider the security of HMAC in the multi-user setting, we also properly handle key collision issues either among two different users, or between construction queries and offline compression function queries. The details of these proofs are deferred to the following sections.

More related work. In practice, HMAC is sometimes deployed in contexts that require security properties beyond standard PRF guarantees. Motivated by this need, the work [18] examines the indifferentiability [28,13] of HMAC from a random oracle. Their result, however, is not directly comparable to ours. Although the stronger notion of indifferentiability captures scenarios where HMAC is used for purposes other than as a PRF, the bound $O(q^2\ell^2/2^c)$ obtained in [18] is considerably weaker and not tight as noted in [22,9]. In addition, recent work [3] studies the dual-PRF security of HMAC. They also investigate the multi-user security of HMAC. Yet, their proofs are in the standard model and do not capture the influence of offline compression function queries on the security, which is the main focus of our work.

Organization. Section 2 introduces notation and definition. Section 3 studies the generic PRF security of HMAC and shows the tightness of the proved bound. Section 4 studies the generic PRF security of NMAC and discuss the tightness of the proved bound.

2 Preliminaries

Notation. Let ε denote the empty string. Let $\{0,1\}^n$ be the set of all n-bit strings and $\{0,1\}^*$ be the set of all finite bit strings including the empty string ε. Let $(\{0,1\}^n)^+$ be the set of all strings of length a positive multiple of n bits. For a finite set $\mathcal{X}$, let $X \leftarrow\!\!\$\ \mathcal{X}$ denote the uniform sampling from $\mathcal{X}$ and assigning the value to X. Let $|X|$ denote the length of string X. For integers $0 \le i \le j$, let $[i, j]$ denote the set of $\{i, i+1, \ldots, j\}$, and $[j]$ the set of $\{1, 2, \ldots, j\}$. If

```
procedure INITIALIZE
$K_1, K_2, \cdots, \leftarrow\!\!\$\ \mathcal{K}$
$f_1, f_2, \cdots, \leftarrow\!\!\$\ \mathrm{Func}(\mathcal{D}, \mathcal{R})$
$b \leftarrow\!\!\$\ \{0,1\}$

procedure PRIM$(u, v)$
return $h(u, v)$

procedure EVAL$(i, M)$
$T_1 \leftarrow F(K_i, M)$
$T_0 \leftarrow f_i(M)$
return $T_b$

procedure FINALIZE$(b')$
return $(b' = b)$
```

Fig. 1: Game $\mathbf{G}_F^{\mathrm{prf}}$ defining multi-user PRF security of the construction F.

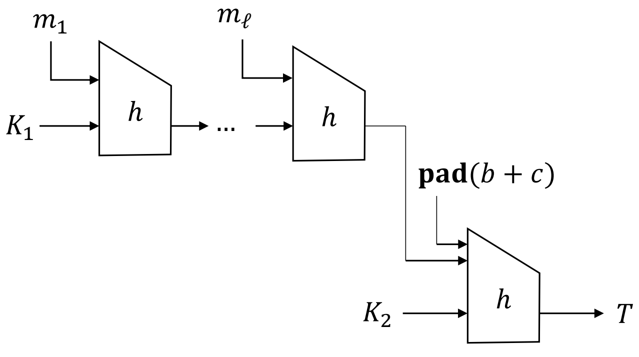

Fig. 2: A pictorial illustration of NMAC.

$i > j$, then $[i, j]$ is an empty set $\emptyset$. Let $Y \leftarrow \mathcal{A}(X_1, \ldots; r)$ denote running algorithm $\mathcal{A}$ with randomness r on inputs $X_1, \ldots$ and assigning the output to Y. We let $Y \leftarrow\!\!\$\ \mathcal{A}(X_1, \ldots)$ be the result of picking r at random and letting $Y \leftarrow \mathcal{A}(X_1, \ldots; r)$. Let $\mathrm{Func}(\mathcal{X}, \mathcal{Y})$ denote the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$.

PRF SECURITY. Let $F : \mathcal{K} \times \mathcal{M} \to \mathcal{R}$ be a keyed function that is instantiated with an ideal compression function $h \leftarrow\!\!\$\ \mathrm{Func}(\{0,1\}^c \times \{0,1\}^b, \{0,1\}^c)$. The advantage of the adversary $\mathcal{A}$ against the multi-user PRF security of F is defined as

$$\mathsf{Adv}_F^{\mathrm{prf}}(\mathcal{A}) = 2\Pr[\mathbf{G}_F^{\mathrm{prf}}(\mathcal{A}) \Rightarrow \mathrm{true}] - 1 \ ,$$

where game $\mathbf{G}_F^{\mathrm{prf}}$ is defined in Fig. 1. In this game, the procedure INITIALIZE will firstly be invoked to initialize the game. Then the adversary $\mathcal{A}$ can have oracle access to both EVAL and PRIM. At the end of interaction with these two oracles, the adversary $\mathcal{A}$ will terminate with an output. This output will be fed to FINALIZE to finish the game.

It is well know that a good PRF is also a good MAC. The security bound for unforgeability can be obtained from the PRF bound via a standard argument.

NMAC AND HMAC. Let $h : \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$ be a compression function with $b > c$. Let pad denote a padding function such that $M^* = M \parallel \mathsf{pad}(|M|) \in (\{0,1\}^b)^+$ for any string $M \in \{0,1\}^*$. The canonical padding method for Merkle-Damgård-based hash functions is $\mathsf{pad}(|M|) = 10^* \langle |M| \rangle$ where

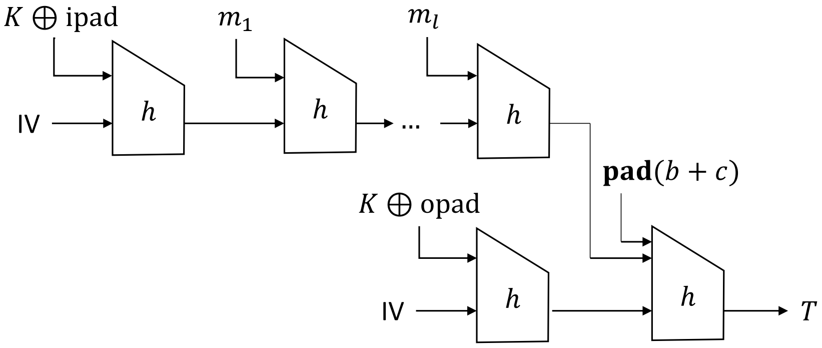

Fig. 3: A pictorial illustration of HMAC.

$\langle |M| \rangle$ is the encoding of length $|M|$ and 0^* denotes the minimum number of zeroes to ensure that the total length is a multiple of b. Let $IV \in \{0,1\}^c$ be an initial vector. For any message $M \in \{0,1\}^*$, we denote by $\mathsf{Casc} : \{0,1\}^c \times \{0,1\}^* \to \{0,1\}^c$ the cascade construction of h as

$$\mathsf{Casc}(IV, M) = z_\ell \ ,$$

where $z_0 = IV$, $z_i = h(z_{i-1}, m_i)$, and $m_1 \parallel m_2 \parallel \ldots \parallel m_\ell \leftarrow M \parallel \mathsf{pad}(|M|)$. This cascade construction is commonly used to design a hash function and also know as Merkle-Damgård transform [29,14].

The construction $\mathsf{NMAC} : (\{0,1\}^c)^2 \times \{0,1\}^* \to \{0,1\}^c$ is obtained from Casc construction by replacing the initial vector IV with a secret key $K_1 \in \{0,1\}^c$ and adding an additional invocation of Casc with another key $K_2 \in \{0,1\}^c$. A pictorial illustration of NMAC is given in Fig. 2. Formally, NMAC is defined as

$$\mathsf{NMAC}((K_1, K_2), M) = \mathsf{Casc}(K_2, \mathsf{Casc}(K_1, M)) \ .$$

The construction $\mathsf{HMAC} : \{0,1\}^c \times \{0,1\}^* \to \{0,1\}^c$ is a non-intrusive version of NMAC such that it allows for the usage of existing implementation of a cryptographic hash function in a black-box manner. The two keys (K_1, K_2) of HMAC are derived from a single key $K \in \{0,1\}^b$ by xoring it with two distinct constant strings $\mathrm{ipad}, \mathrm{opad} \in \{0,1\}^b$. Additionally, these two keys are prepended to the message. A pictorial illustration of HMAC is presented in Fig. 3. Formally, HMAC is defined as

$$\mathsf{HMAC}(K, M) = \mathsf{Casc}(IV, K_2 \parallel \mathsf{Casc}(IV, K_1 \parallel M)) \ ,$$

where $K_1 = K \oplus \mathrm{ipad}$ and $K_2 = K \oplus \mathrm{opad}$. Technically, [26] allows for a key of arbitrary length: a key shorter than b bits is padded with zeros before applying the xor; a longer key is hashed first. Here we mainly focus on the case where the key is of b bits as in [21,22].

The H-coefficient Technique. Following the terminology by Hoang and Tessaro [25], we consider interactions between an adversary $\mathcal{A}$ and an abstract system $\mathbf{S}$ that answers $\mathcal{A}$'s queries. A transcript $\tau = ((x_1, y_1), \ldots, (x_q, y_q))$ is used to record the resulting interaction. Let $\mathsf{p}_{\mathbf{S}}(\tau)$ be the probability that $\mathbf{S}$ generates τ. Note that $\mathsf{p}_{\mathbf{S}}(\tau)$ is the description of $\mathbf{S}$ and independent of the adversary $\mathcal{A}$. A transcript is said to be attainable for the system $\mathbf{S}$ if $\mathsf{p}_{\mathbf{S}}(\tau) > 0$.

We then describe the H-coefficient technique by Patarin [30,12]. Informally speaking, it considers an adversary that aims at distinguishing a "real" system $\mathbf{S}_1$ from an "ideal" system $\mathbf{S}_0$. The interactions of the adversary with those systems produce two transcript distributions X_1 and X_0, respectively. The distinguishing advantage of $\mathcal{A}$ can be upper bounded by the statistical distance $\mathsf{SD}(X_1, X_0)$.

Lemma 1. [30,12] *Suppose that the set of attainable transcripts for the ideal system can be partitioned into good and bad ones. If there exists $\epsilon \geq 0$ such that $\frac{\mathsf{p}_{\mathbf{S}_1}(\tau)}{\mathsf{p}_{\mathbf{S}_0(\tau)}} \geq 1 - \epsilon$ for any good transcript τ, then*

$$\mathsf{SD}(X_1, X_0) \leq \epsilon + \Pr[X_0 \textit{ is bad}] \ .$$

3 Security Analysis of HMAC

In this section, we prove the PRF security of HMAC in the multi-user setting. We also demonstrate that our security bound is tight with matching attacks at the end of this section.

The following theorem shows the PRF security of HMAC. We present the proof of this theorem in Sections 3.1– 3.4, and establish its tightness in Section 3.5.

Theorem 1. *Let $\mathcal{A}$ be an adversary against the multi-user PRF security of HMAC as defined in the game of Fig. 1. Assume that $\mathcal{A}$ makes at most p compression function queries, at most q construction queries of maximal block length ℓ. We have*

$$\begin{aligned}\mathsf{Adv}^{\mathrm{prf}}_{\mathsf{HMAC}}(\mathcal{A}) \leq \frac{pq\ell}{2^c} + \frac{5q^2\ell}{2^c} + \frac{6q^2}{2^c} + \frac{pq}{2^c} + \frac{32q^2\ell^4}{2^{2c}} \\ + \frac{q^2(b+6+\ln p)}{2^{b+1}} + \frac{4pq}{2^b} + \frac{q^2\ell(\ln p + 2)}{2^b} \ .\end{aligned}$$

3.1 Interactions and Transcripts

We consider a computationally unbounded adversary. Without loss of generality, we assume that the adversary is deterministic and does not repeat the same query twice. Assume that the adversary makes at most p compression function queries, at most q construction queries respectively, and each of these construction queries is of maximal block length ℓ. The security game is detailed in Fig. 1. The real system corresponds to the game $\mathbf{G}^{\mathrm{prf}}_{\mathsf{HMAC}}$ with challenge bit $b = 1$ where the adversary has oracle access to HMAC and compression function h. The ideal system corresponds to game $\mathbf{G}^{\mathrm{prf}}_{\mathsf{HMAC}}$ with challenge bit $b = 0$ where the adversary has oracle access to random functions f_i and compression function h.

TRANSCRIPTS. When the adversary is interacting with her oracles, she will obtain the following information in both of the two worlds:

```
**procedure** $\textsc{Dumh}(K_i, M)$
$m_1 \parallel m_2 \parallel \ldots \parallel m_\ell \leftarrow M \parallel \mathsf{pad}(|M|)$
$z_0 \leftarrow h(IV, K_i \oplus \text{ipad})$
**for** $\alpha \leftarrow 1$ **to** $\ell$
    $z_\alpha \leftarrow h(z_{\alpha-1}, m_\alpha)$
$z_{\ell+1} \leftarrow h(IV, K_i \oplus \text{opad})$
```

Fig. 4: Dummy internal compression function calls used in the proof of HMAC.

- Offline compression function queries: for each query $\textsc{Prim}(u, v)$ with answer w, we record this information with an entry (u, v, w). More concretely, we record by (u_i, v_i, w_i) the i-th compression function query with answer w_i. We denote by $\mathcal{Q}_h$ the set of these offline compression function queries.
- Online construction queries: for each query $\textsc{Eval}(i, M)$ with answer T, we record this information with an entry (i, M, T). More specifically, we denote by (i, M_a^i, T_a^i) the a-th query to user i with answer T_a^i.

For any construction query (i, M, T) where $m_1 \parallel m_2 \parallel \ldots \parallel m_\ell \leftarrow M \parallel \mathsf{pad}(|M|)$, let $x_0 = IV$, $y_0 = K_i \oplus \text{ipad}$, $z_0 = h(x_0, y_0)$; for $1 \le \alpha \le \ell$, let $x_\alpha = z_{\alpha-1}$, $y_\alpha = m_\alpha$, $z_\alpha = h(x_\alpha, y_\alpha)$; $x_{\ell+1} = IV$, $y_{\ell+1} = K_i \oplus \text{opad}$, $z_{\ell+1} = h(x_{\ell+1}, y_{\ell+1})$. We denote by $\widetilde{Q}_{\text{in}}$ the set of internal compression function calls before the $(\ell+2)$-th compression function call of each construction query, and by $\widetilde{Q}_{\text{out}}$ the set of all $(\ell + 2)$-th compression function calls of each construction query. We denote by $\widetilde{\mathcal{Q}}_h = \widetilde{Q}_{\text{in}} \cup \widetilde{Q}_{\text{out}}$ the set of these internal compression function calls. In the real world, after the adversary finishes all of her queries, we will grant it the keys $K_1, \ldots, K_u$ of each user and entries of internal compression function calls. In the ideal world, we will instead give the adversary truly random strings $K_i \leftarrow_{\$} \mathcal{K}$ that are independent of her queries. In addition, for each construction query (i, M, T), we will give the adversary dummy internal compression function calls. These dummy calls are generated by the simulation oracle $\textsc{Dumh}(K_i, M)$ as depicted in Fig. 4. Note that these additional information can only increase the adversary's advantage. Overall, a transcript τ consists of offline compression function queries, construction queries, revealed keys K_i, and internal compression function calls.

3.2 Bad and Good Transcripts

We give the definition of bad transcripts. We say a transcript τ is *bad* if one of the following event happens.

1. Key collision among two different users. There exist two different $i, j \in [N]$ such that $K_i = K_j$.
2. Key collision between construction queries and offline compression function queries. There exist $i \in [N]$ and $j \in [p]$ such that either $K_i \oplus \text{ipad} = v_j$ or $K_i \oplus \text{opad} = v_j$.
3. Input collision among the last compression function calls of construction queries to different users. There exist two construction queries (i, M, T) and (j, M', T') such that $(z_{\ell+1}^i, z_\ell^i \parallel \mathsf{pad}(b + c)) = (z_{\ell'+1}^j, z_{\ell'}^j \parallel \mathsf{pad}(b + c))$.

4. Input collision among the last compression function calls of construction queries to the same user. There exist two construction queries (i, M_a, T_a) and (i, M_b, T_b) such that $(z^a_{\ell_a+1}, z^a_{\ell_a} \parallel \mathsf{pad}(b+c)) = (z^b_{\ell_b+1}, z^b_{\ell_b} \parallel \mathsf{pad}(b+c))$.
5. Input collision between the last compression function calls and offline compression function queries. There exists some construction query (i, M, T) such that $(z^i_{\ell+1}, z^i_{\ell} \parallel \mathsf{pad}(b+c)) \in \mathcal{Q}_h$.
6. Input collision between the last compression function calls and internal compression function calls. There exist some construction query (i, M, T) such that $(z^i_{\ell+1}, z^i_{\ell} \parallel \mathsf{pad}(b+c)) \in \widetilde{\mathcal{Q}}_h$.

Denote by bad_i the i-th event. If none of the above events happen, then we say it is a *good* transcript. Let X_0 and X_1 denote the random variables corresponding to the transcript distributions in the ideal world and the real world, respectively.

3.3 Probability of Bad Transcripts

We upper bound the probability that a transcript is bad in the ideal world. By the union bound, we have

$$\begin{aligned}\Pr[\,X_0 \text{ is bad}\,] &= \Pr\left[\bigcup_{i=1}^{6} \mathsf{bad}_i\right] \\ &\le \sum_{i=1}^{2} \Pr[\,\mathsf{bad}_i\,] + \Pr[\,\mathsf{bad}_3 \mid \neg\mathsf{bad}_1\,] + \sum_{i=4}^{6} \Pr[\,\mathsf{bad}_i\,] \ .\end{aligned}$$

For the first bad event bad_1, as each key K_i is selected uniformly at random from the set $\mathcal{K}$, the chance that $K_i = K_j$ is $1/2^b$. Summing over at most $\binom{N}{2}$ pairs of (K_i, K_j), we have

$$\Pr[\,\mathsf{bad}_1\,] \le \frac{N^2}{2^{b+1}} \ .$$

For the second bad event bad_2, as each key K_i is selected uniformly at random from the set $\mathcal{K}$, the probability that either $K_i \oplus \text{ipad} = v_j$ or $K_i \oplus \text{opad} = v_j$ is $2/2^b$. Summing over at most N keys and at most p offline compression function queries, we have

$$\Pr[\,\mathsf{bad}_2\,] \le \frac{2Np}{2^b} \ .$$

Next, we consider the third bad event bad_3. We merely focus on the probability that $z^i_{\ell+1} = z^j_{\ell'+1}$ where $z^i_{\ell+1} = h(IV, K_i \oplus \text{opad})$ and $z^j_{\ell'+1} = h(IV, K_j \oplus \text{opad})$, as this bound is sufficient for our overall security analysis. Conditioned on the event that bad_1 does not occur the probability that $h(IV, K_i \oplus \text{opad}) = h(IV, K_j \oplus \text{opad})$ is $1/2^c$ as h is an ideal compression function and $K_i \ne K_j$. Summing over at most $\binom{q}{2}$ pairs of construction queries, we have

$$\Pr[\,\mathsf{bad}_3 \mid \neg\mathsf{bad}_1\,] \le \frac{q^2}{2^{c+1}} \ .$$

We then analyze the forth bad event bad_4. For any two different construction queries (i, M_a, T_a) and (i, M_b, T_b) to the same user, we always have $z^a_{\ell_a+1} = z^b_{\ell_b+1}$ as both of them equal to $h(IV, K_i \oplus \mathrm{opad})$. Hence, we can only focus on analyzing the probability of the event $z^a_{\ell_a} = z^b_{\ell_b}$ that is the same as $\mathsf{Casc}(IV, K_i \parallel M_a) = \mathsf{Casc}(IV, K_i \parallel M_b)$. To proceed with the analysis of this event, we further define one bad event **EA** and two associated events **EB** and **EC** as follows:

- **EA**: there exists some construction query (i, M, T) such that $(x_\alpha, y_\alpha) \notin \mathcal{Q}_h$ and $(x_{\alpha+1}, y_{\alpha+1}) \in \mathcal{Q}_h$ for some $0 \le \alpha \le \ell - 1$.
- **EB**: either $(x^a_{\ell_a}, y^a_{\ell_a}) \notin \mathcal{Q}_h$, or $(x^b_{\ell_b}, y^b_{\ell_b}) \notin \mathcal{Q}_h$;
- **EC**: $(x^a_\alpha, y^a_\alpha) \in \mathcal{Q}_h$ for all $\alpha \in [1, \ell_a]$, and $(x^b_\alpha, y^b_\alpha) \in \mathcal{Q}_h$ for all $\alpha \in [1, \ell_b]$.

We consider the probability of the event **EA**. As $(x_\alpha, y_\alpha) \notin \mathcal{Q}_h$, the value $x_{\alpha+1} = z_\alpha = h(x_\alpha, y_\alpha)$ is distributed uniformly at random over the set $\{0,1\}^c$. Hence, the probability that $(x_{\alpha+1}, y_{\alpha+1}) \in \mathcal{Q}_h$ is $p/2^c$ via the randomness of $x_{\alpha+1}$. Summing over at most q construction queries whose total length is at most $q\ell$ message blocks, we have

$$\Pr\left[\, \mathbf{EA} \,\right] \le \frac{pq\ell}{2^c} \ .$$

Conditioned on the event that **EA** does not occur, we have

$$\begin{aligned} \Pr\left[\, z^a_{\ell_a} = z^b_{\ell_b} \,\right] &= \Pr\left[\, z^a_{\ell_a} = z^b_{\ell_b} \wedge \mathbf{EB} \,\right] \\ &\quad + \Pr\left[\, z^a_{\ell_a} = z^b_{\ell_b} \wedge \mathbf{EC} \,\right] \\ &\le \Pr\left[\, z^a_{\ell_a} = z^b_{\ell_b} \mid \mathbf{EB} \,\right] \\ &\quad + \Pr\left[\, z^a_{\ell_a} = z^b_{\ell_b} \wedge \mathbf{EC} \,\right] \ . \end{aligned} \tag{2}$$

We first evaluate the first term on the right-hand side of this equation. We partition the event **EB** into the following three sub-events:

- **EB1**: $(x^a_{\ell_a}, y^a_{\ell_a}) \notin \mathcal{Q}_h \wedge (x^b_{\ell_b}, y^b_{\ell_b}) \in \mathcal{Q}_h$;
- **EB2**: $(x^a_{\ell_a}, y^a_{\ell_a}) \in \mathcal{Q}_h \wedge (x^b_{\ell_b}, y^b_{\ell_b}) \notin \mathcal{Q}_h$;
- **EB3**: $(x^a_{\ell_a}, y^a_{\ell_a}) \notin \mathcal{Q}_h \wedge (x^b_{\ell_b}, y^b_{\ell_b}) \notin \mathcal{Q}_h$.

Regarding the sub-event **EB1**, the value $z^a_{\ell_a}$ is distributed uniformly at random in the set $\{0,1\}^c$ as $(x^a_{\ell_a}, y^a_{\ell_a}) \notin \mathcal{Q}_h$, while the value $z^b_{\ell_b}$ has been determined by some offline compression function query as $(x^b_{\ell_b}, y^b_{\ell_b}) \in \mathcal{Q}_h$. Hence, the probability that $z^a_{\ell_a} = z^b_{\ell_b}$ is $1/2^c$. An analogous argument applies to the sub-event **EB2**. For the sub-event **EB3**, let α be the maximal index such that $(x^a_j, y^a_j) \notin \mathcal{Q}_h$ for all $\alpha \le j \le \ell_a$. Similarly, let β be the maximal index such that $(x^b_j, y^b_j) \notin \mathcal{Q}_h$ for all $\beta \le j \le \ell_b$. Hence, the event corresponding to the first term can be written as

$$\mathsf{Casc}(x^a_\alpha, m^a_\alpha \parallel m^a_{\alpha+1} \parallel \ldots \parallel m^a_{\ell_a}) = \mathsf{Casc}(x^b_\beta, m^b_\beta \parallel m^b_{\beta+1} \parallel \ldots \parallel m^b_{\ell_b}) \ . \tag{3}$$

At this stage, since none of compression function calls during the computation of Equation (3) are invoked by the adversary via offline compression function queries, the analysis of the above collision event in the ideal compression function model is equivalent to the one in the standard model where the adversary has no access to the underlying compression function. In the latter case, it was shown by Gaži et al. [21] that the probability of this event can be bounded by[4]

$$\frac{\ell}{2^c} + \frac{64\ell^4}{2^{2c}} \ .$$

Consequently, adding the probabilities of the above three sub-events gives

$$\Pr\left[\, z^a_{\ell_a} = z^b_{\ell_b} \mid \mathbf{EB} \,\right] \le \frac{2}{2^c} + \frac{\ell}{2^c} + \frac{64\ell^4}{2^{2c}} \ .$$

We then evaluate the second term on the right-hand side of Equation (2). Let $\mathcal{S}$ be the set of all possible messages obtained from $\mathcal{Q}_h$, namely for a message $M \in \mathcal{S}$ such that $m_1 \,\|\, m_2 \,\|\, \ldots \,\|\, m_\ell \leftarrow M \,\|\, \mathsf{pad}(|M|)$, there exists a chain of compression function queries $\{(u_1, v_1), (u_2, v_2), \ldots, (u_\ell, v_\ell)\} \subseteq \mathcal{Q}_h$ such that $v_i = m_i$ for $1 \le i \le \ell$ and $u_{i+1} = w_i$ for $1 \le i \le \ell - 1$. Fixing an $IV \in \{0,1\}^c$, for any two distinct messages $M, M' \in \mathcal{S}$, we define a function $g_{M,M'} : \{0,1\}^b \to \{0,1\}^n$ such that for $K \in \{0,1\}^b$,

$$g_{M,M'}(K) = \mathsf{Casc}(IV, K \,\|\, M) \oplus \mathsf{Casc}(IV, K \,\|\, M') \ .$$

Then the event $z^a_{\ell_a} = z^b_{\ell_b} \wedge \mathbf{EC}$ implies that $M_a, M_b \in \mathcal{S}$ and $g_{M_a,M_b}(K_i) = 0^c$. We introduce a threshold γ for the number of tuples $(M, M', \overline{K}_1), \ldots, (M, M', \overline{K}_\gamma)$ that satisfy the following condition. We say the event **mkeys** happens if

- there exist γ distinct keys $\overline{K}_1, \ldots, \overline{K}_\gamma$ and two distinct messages $M, M' \in \mathcal{S}$ such that $g_{M,M'}(\overline{K}_i) = 0^c$ for all $1 \le i \le \gamma$.

Then we have

$$\begin{aligned} \Pr\left[\, z^a_{\ell_a} = z^b_{\ell_b} \wedge \mathbf{EC} \,\right] &\le \Pr\left[\, z^a_{\ell_a} = z^b_{\ell_b} \wedge \mathbf{EC} \mid \neg\mathbf{mkeys} \,\right] + \Pr\left[\, \mathbf{mkeys} \,\right] \\ &\le \frac{\gamma - 1}{2^b} + \Pr\left[\, \mathbf{mkeys} \,\right] \ . \end{aligned}$$

The first term on the right-hand side of this equation comes from the fact that as long as the event **mkeys** does not occur, the number of key candidates $\overline{K}_i$ such that $g_{M,M'}(\overline{K}_i) = 0^n$ is at most $\gamma - 1$ for any two distinct messages $M, M' \in \mathcal{S}$. Hence, the probability that K_i equals to any of these $\gamma - 1$ key candidates is $(\gamma - 1)/2^b$. We proceed to evaluate the probability of the event **mkeys**. Given a pair of $M, M' \in \mathcal{S}$, for each key candidate $\overline{K}_i$, if $g_{M,M'}(\overline{K}_i) = 0^n$, then it holds $h(IV, \overline{K}_i \oplus \text{ipad}) = u_1$ that happens with probability $1/2^c$ where $(u_1, v_1) \in \mathcal{Q}_h$

[4] Note that Gaži et al. [21] consider the case in which the two IVs used by the two cascade queries are fixed and equal. The same argument applies when the IVs are fixed but possibly distinct, since when the two IVs differ, this only eliminates certain potential bad structure graphs that could lead to a collision.

and $m_1 = v_1$. As these key candidates $\overline{K}_1, \overline{K}_2, \dots, \overline{K}_\gamma$ are distinct, the outputs $h(IV, \overline{K}_1), h(IV, \overline{K}_2), \dots, h(IV, \overline{K}_\gamma)$ are γ independent and random strings. Hence, we have

$$\Pr\left[\,\forall i \in [\gamma] : g_{M,M'}(\overline{K}_i) = 0^n\,\right] = \left(\frac{1}{2^c}\right)^\gamma \quad .$$

Note that the size of the set $\mathcal{S}$ is at most $\sum_{i=1}^{\ell} p^i$, as the block length of a message is at most ℓ and each block is determined by a compression function query from $\mathcal{Q}_h$. The number of possible pairs of (M, M') is at most $(\sum_{i=1}^{\ell} p^i)^2 \le p^{2(\ell+1)}$. Hence,

$$\begin{aligned}\Pr\left[\,\mathbf{mkeys}\,\right] &\le p^{2(\ell+1)} \cdot \binom{2^b}{\gamma} \cdot \left(\frac{1}{2^c}\right)^\gamma \\ &\le p^{2(\ell+1)} \cdot \frac{2^{b\gamma}}{(\gamma/e)^\gamma} \cdot \left(\frac{1}{2^c}\right)^\gamma \\ &= p^{2(\ell+1)} \cdot \left(\frac{e2^{b-c}}{\gamma}\right)^\gamma \quad ,\end{aligned}$$

where the second inequality is due to Stirling's approximation: $\gamma! \ge (\gamma/e)^\gamma$ for any $\gamma \ge 1$. Therefore, we have

$$\Pr\left[\,z^a_{\ell_a} = z^b_{\ell_b} \wedge \mathbf{EC}\,\right] \le \frac{\gamma - 1}{2^b} + p^{2(\ell+1)} \cdot \left(\frac{e2^{b-c}}{\gamma}\right)^\gamma \quad .$$

By choosing $\gamma = \left\lceil e \cdot 2^{b-c} + 2(\ell+1)\ln p + b \ln 2 \right\rceil \le 2^b$ so that $p^{2(\ell+1)} \cdot \left(e2^{b-c}/\gamma\right)^\gamma \le 1/2^b$, we have[5]

$$\Pr\left[\,z^a_{\ell_a} = z^b_{\ell_b} \wedge \mathbf{EC}\,\right] \le \frac{e}{2^c} + \frac{2(\ell+1)\ln p + b\ln 2 + 1}{2^b} \quad .$$

By combining the probability of the event **EA**, and summing over at most $\binom{q}{2}$ pairs of construction queries for events **EB** and **EC**, we have

$$\Pr\left[\,\mathsf{bad}_4\,\right] \le \frac{pq\ell}{2^c} + \frac{q^2}{2^c} + \frac{q^2\ell}{2^{c+1}} + \frac{32q^2\ell^4}{2^{2c}} + \frac{eq^2}{2^{c+1}} + \frac{q^2\left(2(\ell+1)\ln p + b\ln 2 + 1\right)}{2^{b+1}} \quad .$$

For the fifth bad event bad_5, we focus on the case when $z^i_{\ell+1} = u$ for some compression function query $(u, v) \in \mathcal{Q}_h$. To guarantee the randomness of $z^i_{\ell+1}$ where $z^i_{\ell+1} = h(IV, K_i \oplus \mathrm{opad})$, we define an associated event **ED** as follows.

[5] The inequality $p^{2(\ell+1)} \cdot \left(e2^{b-c}/\gamma\right)^\gamma \le 1/2^b$ is equivalent to $2(\ell+1)\ln p + \gamma \ln(e2^{b-c}/\gamma) \le -b\ln 2$ by taking logs on the both side. Rearranging this inequality, it is equivalent to $\gamma\ln(\gamma/e2^{b-c}) \ge 2(\ell+1)\ln p + b\ln 2$. Define $g(\gamma) = \gamma\ln(\gamma/e2^{b-c})$. For $\gamma \ge e2^{b-c}$, $g'(\gamma) = \gamma\ln(\gamma/e2^{b-c}) + 1 \ge 1$. Hence, $g(\gamma)$ increases at least linearly and $g(\gamma) \ge \gamma - e2^{b-c}$. By choosing $\gamma = \left\lceil e \cdot 2^{b-c} + 2(\ell+1)\ln p + b\ln 2\right\rceil$, we have $\gamma - e2^{b-c} \ge 2(\ell+1)\ln p + b\ln 2$.

We say the event $\mathbf{ED}$ happens if $K_i \oplus \mathrm{opad} = y$ for some internal compression function query $(x, y) \in \widetilde{Q}_{\mathrm{in}}$, or $K_i \oplus \mathrm{opad} = v$ for some offline compression function query $(u, v) \in \mathcal{Q}_h$. The probability that the event $\mathbf{ED}$ occurs is at most

$$\Pr[\,\mathbf{ED}\,] \le \frac{q(\ell+1)+p}{2^b} \ ,$$

as K_i is a b-bit random string, and there are at most $q(\ell+1)$ elements in the set $\widetilde{Q}_{\mathrm{in}}$ and at most p elements in the set $\mathcal{Q}_h$. Conditioned on the event that $\mathbf{ED}$ does not happen, $z^i_{\ell+1}$ is a random string. Hence, the probability that $z^i_{\ell+1} = u$ for some compression function query $(u, v) \in \mathcal{Q}_h$ is $p/2^c$. Summing over at most q construction queries, we have

$$\Pr[\,\mathsf{bad}_5\,] \le \frac{q^2(\ell+1)+pq}{2^b} + \frac{pq}{2^c} \ .$$

For the sixth bad event bad_6, we can also focus on the case when $z^i_{\ell+1} = x$ for some compression function query $(x, y) \in \widetilde{Q}_h$. Following a similar analysis as that for the event bad_5, we obtain

$$\Pr[\,\mathsf{bad}_6\,] \le \frac{q^2(\ell+1)+pq}{2^b} + \frac{q^2(\ell+2)}{2^c} \ ,$$

since there are at most $q(\ell+2)$ elements in the set $\widetilde{Q}_h$.

By the union bound and wrapping up the probabilities of these six bad events, we have

$$\begin{aligned}
\Pr[\,X_0 \text{ is bad}\,] \le{}& \frac{N^2}{2^{b+1}} + \frac{2Np}{2^b} + \frac{q^2}{2^{c+1}} + \frac{pq\ell}{2^c} + \frac{q^2}{2^c} + \frac{q^2\ell}{2^{c+1}} \\
&+ \frac{32q^2\ell^4}{2^{2c}} + \frac{eq^2}{2^{c+1}} + \frac{q^2\,(2(\ell+1)\ln p + b\ln 2 + 1)}{2^{b+1}} \\
&+ \frac{q^2(\ell+1)+pq}{2^b} + \frac{pq}{2^c} + \frac{q^2(\ell+1)+pq}{2^b} + \frac{q^2(\ell+2)}{2^c} \\
\le{}& \frac{pq\ell}{2^c} + \frac{5q^2\ell}{2^c} + \frac{6q^2}{2^c} + \frac{pq}{2^c} + \frac{32q^2\ell^4}{2^{2c}} \\
&+ \frac{q^2(b+6+\ln p)}{2^{b+1}} + \frac{4pq}{2^b} + \frac{(\ln p + 2)q^2\ell}{2^b} \ ,
\end{aligned}$$

as $N \le q$.

3.4 Transcripts Ratio

We now consider a good transcript τ. Recall that a transcript τ consists of revealed keys K_i, offline compression function queries, internal compression function calls, and construction queries. In the ideal world, these revealed keys are selected independently and uniformly at random from the set $\mathcal{K}$. Hence, the probability that X_0 is compatible with τ regarding these keys is $(1/2^b)^N$ where

N is the number of users. For the offline compression function queries, the probability that X_0 is compatible with τ regarding these offline compression function queries is $(1/2^c)^p$, as h is an ideal compression function. For the internal compression function calls, let $\tilde{p}$ be the size of the set $\mathcal{Q}_h \setminus \widetilde{\mathcal{Q}}_h$. Conditioned on the event that the offline compression function queries are compatible, the probability that X_0 is compatible with τ regarding these internal compression function calls is $(1/2^c)^{\tilde{p}}$. For construction queries, as each output tag T is a random string in the ideal world, the probability that X_0 is compatible with τ regarding these construction queries is $(1/2^c)^q$. Therefore,

$$\Pr[\,X_0 = \tau\,] = (\frac{1}{2^b})^N \cdot (\frac{1}{2^c})^p \cdot (\frac{1}{2^c})^{\tilde{p}} \cdot (\frac{1}{2^c})^q \ .$$

In the real world, the arguments for revealed keys, offline compression function queries, and internal compression function calls are the same as those in the ideal world. For each construction query, as τ is a good transcript, the input to the last compression function call is fresh and distinct from: i) the inputs to the last compression function calls of other construction queries; ii) the inputs to the internal compression function calls of all construction queries; iii) the inputs of offline compression function queries. Hence, each corresponding output tag T is a random string, and the probability that X_1 is compatible with τ regarding these construction queries is $(1/2^c)^q$. Thus,

$$\Pr[\,X_1 = \tau\,] = (\frac{1}{2^b})^N \cdot (\frac{1}{2^c})^p \cdot (\frac{1}{2^c})^{\tilde{p}} \cdot (\frac{1}{2^c})^q \ .$$

Finally, we have

$$\frac{\Pr[\,X_1 = \tau\,]}{\Pr[\,X_0 = \tau\,]} = 1 \ ,$$

and conclude the proof of Theorem 1 via Lemma 1.

3.5 Matching Attacks

For most practical parameters of HMAC with $c < b$, the dominating terms in our bounds are $pq\ell/2^c$ and $q^2\ell/2^c$. We show that these two terms are essentially tight up to some constant factors by giving matching attacks. Hence, our attacks demonstrate that the bound of Theorem 1 is tight for those parameters.

The term $pq\ell/2^c$. Let h_m be the compression function h with a fixed message block m, namely $h_m(x) = h(x, m)$. Then h_m can be regarded as a n-bit to n-bit random function for any fixed message block m. The attack exploits properties of the functional graph of a random function that are recalled in Appendix A. By using the known properties of the functional graph of a random function, we can mount the following PRF distinguishing attack[6] against HMAC.

[6] This attack is almost the same as the distinguish-H attack by Leurent, Peyrin, and Wang [27], which is used to check which cryptographic hash function is embedded in HMAC. However, here we adapt it to distinguish HMAC from a random function. We deem this PRF distinguishing attack against HMAC another interesting application of the functional graph [19,20,5].

1. Search for a cycle in the functional graph of h_{0^n} via offline compression function queries. We repeat this procedure several times to ensure that the found cycle lies in the largest component of the functional graph of h_{0^n}. Denote by L the length of this cycle.
2. Select a message block m randomly, ask two construction queries of messages $M_1 = m \parallel (0^n)^{2^{c/2}} \parallel 1^n \parallel (0^n)^{2^{c/2}+L}$ and $M_2 = m \parallel (0^n)^{2^{c/2}+L} \parallel 1^n \parallel (0^n)^{2^{c/2}}$ for the same user.
3. If the tags of these two messages collide, then output 1. Otherwise, output 0.

We first evaluate the complexity of the above PRF distinguishing attack. The first step requires about $O(2^{c/2})$ offline compression function queries to find the cycle for the compression function h due to the known properties of the functional graph. The second step requires two construction queries, each of length about $3 \cdot 2^{c/2}$ message blocks.

We then evaluate the advantage of the above PRF distinguishing attack. In the ideal world, as the random function f is independent of the compression function h, those compression function queries in the first step have no impact on the construction queries in the second step. Hence, the probability that the third step outputs 1 is $1/2^c$. In the real world, the tags of M_1 and M_2 collide if: i) a randomly chosen message block m sets the online computation of step 2 in the same component as in the step 1. This event happens with probability 0.7582, as the largest component of the functional graph of h_{0^n} has an average size of $0.7582 \cdot 2^c$ elements, according to Theorem 5 in Appendix A. ii) the computation of those $2^{c/2}$ blocks of 0^n after m reaches the cycle of the component. This event happens with probability at least 1/2, since the average tail length is less than $2^{c/2}$ elements, according to Theorem 4 in Appendix A. iii) the message block 1^n sets the online computation back to the same component as in the step 1. The probability of this event is 0.7582; iv) the computation of those $2^{c/2}$ blocks of 0^n after 1^n reaches the cycle of the component. The probability of this event is at least 1/2. Overall, the probability that a collision happens between the tags of M_1 and M_2 in the real world is $(0.7582 \cdot 1/2)^2 \approx 0.14$. Hence, the advantage of this PRF distinguishing attack is $0.14 - 2^{-c} \approx 0.14$. As a conclusion, to make the term $pq\ell/2^c$ non-negligible, it only requires $p \approx 2^{c/2}$, $q = 2$, $\ell \approx 2^{c/2}$, which matches the term. Note that the above PRF distinguishing attack can be easily extended to a forgery attack as follows. After detecting a collision on the tags of messages M_1 and M_2, the adversary can query another message $M_3 = M_1 \parallel x$ to obtain the corresponding tag T_3 by choosing an arbitrary block x and appending it after M_1. Then (M_4, T_3) is a valid forgery against HMAC where $M_4 = M_2 \parallel x$.

The term $q^2\ell/2^c$. We choose q messages $M_1, \ldots, M_q$ in the form of $M_i = x_i \parallel 0^{b(\ell-1)}$ where $x_1, \ldots, x_q$ are q distinct message blocks. Then in the real world for any $1 \le i < j \le q$ and any $0 \le \alpha \le \ell - 1$, if the internal outputs after the computation of the first α blocks of M_i and M_j collide, it will hold $\mathsf{HMAC}(K, M_i) = \mathsf{HMAC}(K, M_j)$ as M_i and M_j share the same suffix and length. For any triple of (i, j, α), the probability that this internal collision happens is roughly $1/2^c$ as long as $\ell \ll 2^{c/2}$. There are $\binom{q}{2}\ell$ possibilities for triples (i, j, α).

Hence, the probability that there is a collision among the tags of these q messages is around $\Theta(q^2\ell/2^c)$. On the other hand in the ideal world, the probability that there is a collision among $f(M_1), \ldots, f(M_q)$ is around $\Theta(q^2/2^c)$ as f is a random function. Hence, the advantage of this distinguishing attack is around $\Theta(q^2\ell/2^c) - \Theta(q^2/2^c) \approx \Theta(q^2\ell/2^c)$ that matching the term of $q^2\ell/2^c$ up to some constant factor. Note that this attack is the same as the PRF distinguishing attack against NI construction by Gaži et al. [21].

4 Security Analysis of NMAC

In this section, we give the PRF security analysis of NMAC in the multi-user setting.

The following theorem shows that NMAC is a good PRF in the multi-user setting by modeling the underlying compression function as an ideal compression function.

Theorem 2. *Let $\mathcal{A}$ be an adversary against the multi-user PRF security of NMAC as defined in the game of Fig. 1. Assume that $\mathcal{A}$ makes at most p compression function queries, at most q construction queries of maximal block length ℓ. We have*

$$\begin{aligned}\mathsf{Adv}^{\mathrm{prf}}_{\mathsf{NMAC}}(\mathcal{A}) \le \frac{pq\ell}{2^c} + \frac{4q^2\ell}{2^c} + \frac{4q^2}{2^c} + \frac{2pq}{2^c} + \frac{32q^2\ell^4}{2^{2c}} \\ + \frac{q^2(b+c+3+\ln p)}{2^{b+1}} + \frac{q^2\ell(\ln p + 1)}{2^b} \quad .\end{aligned}$$

Remark. Although the main proof idea of this theorem is similar to that for HMAC, we found that there are subtle differences between these two analyses as NMAC and HMAC are equipped with different keying mechanisms, which results in essentially different concrete security analysis, especially if we want to prove a tight security bound in the ideal compression model and in the multi-user setting. Hence, it requires a separate proof for NMAC.

4.1 Interactions and Transcripts

We assume that the adversary has unbounded computational power, and is therefore deterministic. We also assume that the adversary does not repeat a prior query as otherwise she will receive the same response. The PRF security game can be found in Fig. 1. For convenience in the following analysis, the game $\mathbf{G}^{\mathrm{prf}}_{\mathsf{NMAC}}$ with challenge bit $b = 1$ is called the real system, while the game $\mathbf{G}^{\mathrm{prf}}_{\mathsf{NMAC}}$ with challenge bit $b = 0$ is called the ideal system. Suppose that the adversary makes at most p compression function queries, q construction queries of maximal block length ℓ.

Transcripts. In both of two systems, the adversary will obtain the following information when interacting with her oracles:

```
procedure $\textsc{Dumn}(K_i, M)$
$(K_{1,i}, K_{2,i}) \leftarrow K_i$
$m_1 \parallel m_2 \parallel \ldots \parallel m_\ell \leftarrow M \parallel \mathsf{pad}(|M|)$
$z_1 \leftarrow h(K_{i,1}, m_1)$
for $\alpha \leftarrow 2$ to $\ell$
    $z_\alpha \leftarrow h(z_{\alpha-1}, m_\alpha)$
```

Fig. 5: Dummy internal compression function calls used in the proof of NMAC.

- Offline compression function queries: for each query $\textsc{Prim}(u, v)$ with response w, we record with an entry (u, v, w). We denote by $\mathcal{Q}_h$ the set of these offline compression function queries.
- Online construction queries: for each query $\textsc{Eval}(i, M)$ with response T, we record with an entry (i, M, T).

For any construction query (i, M, T) where $m_1 \parallel m_2 \parallel \ldots \parallel m_\ell \leftarrow M \parallel \mathsf{pad}(|M|)$, let $x_1 = K_1$, $y_1 = m_1$, $z_1 = h(x_1, y_1)$; for $2 \leq \alpha \leq \ell$, let $x_\alpha = z_{\alpha-1}$, $y_\alpha = m_\alpha$, $z_\alpha = h(x_\alpha, y_\alpha)$. We denote by $\widetilde{Q}_{\text{in}}$ the set of these internal compression function calls. In the real world, once the adversary has completed all of its queries, we reveal to it the user keys $K_1, \ldots, K_u$, where each $K_i = (K_{1,i}, K_{2,i})$, together with the entries of the internal compression function calls. In the ideal world, by contrast, the adversary instead receives independent uniform keys $K_i \leftarrow_{\$} \mathcal{K}$, unrelated to its queries. In addition, for every construction query (i, M, T), the adversary is given dummy entries simulating the internal compression function calls. These dummy entries are generated by the oracle $\textsc{Dumn}(K_i, M)$ as shown in Fig. 5. We stress that this extra information can only increase the adversary's advantage. Consequently, a transcript τ consists of offline compression function queries, construction queries, the revealed keys, and the internal compression function calls.

4.2 Bad and Good Transcripts

Given a transcript τ, we say it is *bad* if one of the following events occurs.

1. Key collision among two different users. There exists two different $i, j \in [N]$ such that $K_{1,i} = K_{1,j}$ or $K_{2,i} = K_{2,j}$. Note that once this event does not happen, it will also imply that input collision among the last compression function calls of construction queries to different users cannot occur as $K_{2,i} \neq K_{2,j}$.
2. Key collision between construction queries and offline compression function queries. There exist $i \in [N]$ and $j \in [p]$ such that either $K_{1,i} = u_j$ or $K_{2,i} = u_j$ for some offline compression function query (u_j, v_j, w_j). Note that once this event does not happen, it will also imply that the input collision between the last compression function calls and offline compression function queries cannot occur as $K_{2,i} \neq u_j$.

3. Input collision among the last compression function calls of construction queries to the same user. There exist two construction queries (i, M_a, T_a) and (i, M_b, T_b) such that $(K_{2,i}, z^a_{\ell_a} \parallel \mathsf{pad}(b+c)) = (K_{2,i}, z^b_{\ell_b} \parallel \mathsf{pad}(b+c))$.
4. Input collision between the last compression function calls and internal compression function calls. There exists some construction query (i, M, T) such that $(K_{2,i}, z^i_\ell \parallel \mathsf{pad}(b+c)) \in \widetilde{\mathcal{Q}}_{\text{in}}$.

Let $\widetilde{\mathsf{bad}}_i$ be the i-th event. We say a transcript τ is *good* if none of the above events occur. Denote by X_0 and X_1 the random variables corresponding to the transcript distributions in the ideal world and the real world, respectively.

4.3 Probability of Bad Transcripts

We upper bound the probability that a transcript is bad in the ideal world. By the union bound, we have

$$\Pr[X_0 \text{ is bad}] = \Pr\left[\bigcup_{i=1}^{4} \widetilde{\mathsf{bad}}_i\right] \leq \sum_{i=1}^{4} \Pr\left[\widetilde{\mathsf{bad}}_i\right] .$$

For the first bad event $\widetilde{\mathsf{bad}}_1$, as each key K_i is selected uniformly at random from the set $\mathcal{K}$, the chance that either $K_{1,i} = K_{1,j}$ or $K_{2,i} = K_{2,j}$ is $2/2^c$. Summing over at most $\binom{N}{2}$ pairs of (K_i, K_j), we have

$$\Pr\left[\widetilde{\mathsf{bad}}_1\right] \leq \frac{N^2}{2^{c+1}} .$$

For the second bad event $\widetilde{\mathsf{bad}}_2$, as each key K_i is selected uniformly at random from the set $\mathcal{K}$, the probability that either $K_{1,i} = u_j$ or $K_{2,i} = u_j$ is $2/2^c$. Summing over at most N keys and at most p offline compression function queries, we have

$$\Pr\left[\widetilde{\mathsf{bad}}_2\right] \leq \frac{2Np}{2^c} .$$

We then analyze the third bad event $\widetilde{\mathsf{bad}}_3$. The collision $z^a_{\ell_a} = z^b_{\ell_b}$ is the same as $\mathsf{Casc}(K_{1,i}, M_a) = \mathsf{Casc}(K_{1,i}, M_b)$. Similar to the analysis of HMAC in Section 3, we define one additional bad event $\widetilde{\mathbf{EA}}$ and two associated events $\widetilde{\mathbf{EB}}$ and $\widetilde{\mathbf{EC}}$ as follows:

– $\widetilde{\mathbf{EA}}$: there exists some construction query (i, M, T) such that $(x_\alpha, y_\alpha) \notin \mathcal{Q}_h$ and $(x_{\alpha+1}, y_{\alpha+1}) \in \mathcal{Q}_h$ for some $1 \leq \alpha \leq \ell - 1$.
– $\widetilde{\mathbf{EB}}$: either $(x^a_{\ell_a}, y^a_{\ell_a}) \notin \mathcal{Q}_h$, or $(x^b_{\ell_b}, y^b_{\ell_b}) \notin \mathcal{Q}_h$;
– $\widetilde{\mathbf{EC}}$: $(x^a_\alpha, y^a_\alpha) \in \mathcal{Q}_h$ for all $\alpha \in [1, \ell_a]$, and $(x^b_\alpha, y^b_\alpha) \in \mathcal{Q}_h$ for all $\alpha \in [1, \ell_b]$.

The analysis of the event $\widetilde{\mathbf{EA}}$ follows exactly the same reasoning as in the case of HMAC. Hence, we have

$$\Pr\left[\widetilde{\mathbf{EA}}\right] \leq \frac{pq\ell}{2^c} .$$

Conditioned on the event that $\widetilde{\mathbf{EA}}$ does not occur, we also have

$$\begin{aligned}\Pr\left[\, z^a_{\ell_a} = z^b_{\ell_b} \,\right] &= \Pr\left[\, z^a_{\ell_a} = z^b_{\ell_b} \wedge \widetilde{\mathbf{EB}} \,\right] + \Pr\left[\, z^a_{\ell_a} = z^b_{\ell_b} \wedge \widetilde{\mathbf{EC}} \,\right] \\ &\le \Pr\left[\, z^a_{\ell_a} = z^b_{\ell_b} \mid \widetilde{\mathbf{EB}} \,\right] + \Pr\left[\, z^a_{\ell_a} = z^b_{\ell_b} \wedge \widetilde{\mathbf{EC}} \,\right] \ . \qquad (4)\end{aligned}$$

The analysis of the event $z^a_{\ell_a} = z^b_{\ell_b} \mid \widetilde{\mathbf{EB}}$ is similar to that for HMAC, and we have

$$\Pr\left[\, z^a_{\ell_a} = z^b_{\ell_b} \mid \widetilde{\mathbf{EB}} \,\right] \le \frac{2}{2^c} + \frac{\ell}{2^c} + \frac{64\ell^4}{2^{2c}} \ .$$

The analysis of the event $z^a_{\ell_a} = z^b_{\ell_b} \wedge \widetilde{\mathbf{EC}}$ follows essentially the same idea as that for HMAC, yet there are some subtle differences: i) the starting point of the chain not only contains the first message block m_1 but also contains the key $K_{1,i}$; ii) there is no fixed IV in NMAC. Fortunately, by fixing the first key and increasing the threshold $\tilde{\gamma}$ by a constant, we can reach almost the same security bound as that of HMAC. We proceed the analysis as follows. Let $\widetilde{\mathcal{S}}$ be the set of all possible messages obtained from $\mathcal{Q}_h$. For any two distinct messages $M, M' \in \widetilde{\mathcal{S}}$ and a $K \in \{0,1\}^c$, we define a function $\tilde{g}_{M,M'} : \{0,1\}^c \to \{0,1\}^c$ as follows:

$$\tilde{g}_{M,M'}(K) = \mathsf{Casc}(K, M) \oplus \mathsf{Casc}(K, M') \ .$$

Obviously, the event $z^a_{\ell_a} = z^b_{\ell_b} \wedge \widetilde{\mathbf{EC}}$ implies that $M_a, M_b \in \widetilde{\mathcal{S}}$ and $\tilde{g}_{M_a,M_b}(K_i) = 0^c$. We also define the event $\widetilde{\mathbf{mkeys}}$ with a threshold $\tilde{\gamma}$:

- there exists $\tilde{\gamma}$ distinct keys $\overline{K}_1, \ldots, \overline{K}_{\tilde{\gamma}}$ and two distinct messages $M, M' \in \widetilde{\mathcal{S}}$ such that $\tilde{g}_{M,M'}(\overline{K}_i) = 0^c$ for all $1 \le i \le \tilde{\gamma}$.

Then we have

$$\begin{aligned}\Pr\left[\, z^a_{\ell_a} = z^b_{\ell_b} \wedge \widetilde{\mathbf{EC}} \,\right] &\le \Pr\left[\, z^a_{\ell_a} = z^b_{\ell_b} \wedge \widetilde{\mathbf{EC}} \mid \neg\widetilde{\mathbf{mkeys}} \,\right] + \Pr\left[\, \widetilde{\mathbf{mkeys}} \,\right] \\ &\le \frac{\tilde{\gamma} - 1}{2^c} + \Pr\left[\, \widetilde{\mathbf{mkeys}} \,\right] \ ,\end{aligned}$$

where the first term follows from a similar argument as that for HMAC. We then analyze the probability of the event $\widetilde{\mathbf{mkeys}}$. Given a pair of $M, M' \in \widetilde{\mathcal{S}}$, we first fix a key candidate $\overline{K}_1 \in \{0,1\}^c$ such that $\tilde{g}_{M,M'}(\overline{K}_1) = 0^n$. Then for the rest of $\tilde{\gamma} - 1$ keys $\overline{K}_i$ with $2 \le i \le \tilde{\gamma}$, the event $\tilde{g}_{M,M'}(\overline{K}_i) = 0^n$ requires that $h(\overline{K}_i, m_1) = u_2$ where $u_2 = h(\overline{K}_1, m_1)$ that has been fixed by $(M, M', \overline{K}_1)$. Note that the equation $h(\overline{K}_i, m_1) = u_2$ holds with probability $1/2^c$ as $\overline{K}_i \neq \overline{K}_1$ and h is an ideal compression function. As these key candidates $\overline{K}_2, \ldots, \overline{K}_{\tilde{\gamma}}$ are distinct, the outputs $h(\overline{K}_2, m_1), h(\overline{K}_3, m_1), \ldots, h(\overline{K}_{\tilde{\gamma}}, m_1)$ are $\tilde{\gamma}-1$ independent and random strings. Thus, we have

$$\Pr\left[\, \forall i \in [\tilde{\gamma}] : \tilde{g}_{M,M'}(\overline{K}_i) = 0^n \,\right] = \left(\frac{1}{2^c}\right)^{\tilde{\gamma}-1} \ .$$

The number of possible pairs of (M, M') is at most $(\sum_{i=1}^{\ell} p^i)^2 \leq p^{2(\ell+1)}$, as the block length of a message is at most ℓ and each block is determined by a compression function query from $\mathcal{Q}_h$. Thus, we have

$$\begin{aligned}\Pr\left[\widetilde{\mathbf{mkeys}}\right] &\leq p^{2(\ell+1)} \cdot \binom{2^b}{\tilde{\gamma}} \cdot \left(\frac{1}{2^c}\right)^{\tilde{\gamma}-1} \\ &\leq p^{2(\ell+1)} \cdot \frac{2^{b\tilde{\gamma}}}{(\tilde{\gamma}/e)^{\tilde{\gamma}}} \cdot \left(\frac{1}{2^c}\right)^{\tilde{\gamma}-1} \\ &= p^{2(\ell+1)} \cdot 2^c \cdot \left(\frac{e2^{b-c}}{\tilde{\gamma}}\right)^{\tilde{\gamma}} \quad ,\end{aligned}$$

where the second inequality is due to Stirling's approximation: $\tilde{\gamma}! \geq (\tilde{\gamma}/e)^{\tilde{\gamma}}$ for any $\tilde{\gamma} \geq 1$. Thus, we have

$$\Pr\left[z^a_{\ell_a} = z^b_{\ell_b} \wedge \widetilde{\mathbf{EC}}\right] \leq \frac{\tilde{\gamma}-1}{2^b} + p^{2(\ell+1)} \cdot 2^c \cdot \left(\frac{e2^{b-c}}{\tilde{\gamma}}\right)^{\tilde{\gamma}} \quad .$$

By choosing $\tilde{\gamma} = \lceil e \cdot 2^{b-c} + 2(\ell+1)\ln p + b\ln 2 + c\ln 2\rceil \leq 2^b$, we have[7]

$$\Pr\left[z^a_{\ell_a} = z^b_{\ell_b} \wedge \widetilde{\mathbf{EC}}\right] \leq \frac{e}{2^c} + \frac{2(\ell+1)\ln p + (b+c)\ln 2 + 1}{2^b} \quad .$$

By adding the probability of the event $\widetilde{\mathbf{EA}}$, and summing over at most $\binom{q}{2}$ pairs of construction queries for events $\widetilde{\mathbf{EB}}$ and $\widetilde{\mathbf{EC}}$, we have

$$\begin{aligned}\Pr\left[\widetilde{\mathsf{bad}}_3\right] \leq \frac{pq\ell}{2^c} + \frac{q^2}{2^c} + \frac{q^2\ell}{2^{c+1}} + \frac{32q^2\ell^4}{2^{2c}} + \frac{eq^2}{2^{c+1}} \\ + \frac{q^2\left(2(\ell+1)\ln p + (b+c)\ln 2 + 1\right)}{2^{b+1}} \quad .\end{aligned}$$

For the forth bad event $\widetilde{\mathsf{bad}}_4$, we can analyze the probability of the equation $K_{2,i} = x$ for some compression function query $(x, y) \in \widetilde{Q}_{\text{in}}$. As $K_{2,i}$ is uniformly distributed at random in the set $\{0,1\}^c$ and there are at most $q\ell$ elements in the set $\widetilde{\mathcal{Q}}_{\text{in}}$, we obtain

$$\Pr\left[\widetilde{\mathsf{bad}}_4\right] \leq \frac{q^2\ell}{2^c} \quad ,$$

by summing over at most q construction queries.

[7] The reasoning to choose the value of $\tilde{\gamma}$ is similar to that of γ in the proof of HMAC, although the exact value of $\tilde{\gamma}$ is different from that of γ as one would expect.

By applying the union bound and summing the probabilities of these bad events, we have

$$\begin{aligned}\Pr[\, X_0 \text{ is bad}\,] &\le \frac{N^2}{2^{c+1}} + \frac{2Np}{2^c} + \frac{pq\ell}{2^c} + \frac{q^2}{2^c} + \frac{q^2\ell}{2^{c+1}} + \frac{32q^2\ell^4}{2^{2c}} + \frac{eq^2}{2^{c+1}} \\ &\quad + \frac{q^2\left(2(\ell+1)\ln p + (b+c)\ln 2 + 1\right)}{2^{b+1}} + \frac{q^2\ell}{2^b} \\ &\le \frac{pq\ell}{2^c} + \frac{4q^2\ell}{2^c} + \frac{4q^2}{2^c} + \frac{2pq}{2^c} + \frac{32q^2\ell^4}{2^{2c}} \\ &\quad + \frac{q^2(b+c+3+\ln p)}{2^{b+1}} + \frac{(\ln p + 1)q^2\ell}{2^b} \ ,\end{aligned}$$

as $N \le q$.

4.4 Transcripts Ratio

We now consider a good transcript τ. Analogous to the analysis of HMAC in Section 3.4, we have

$$\frac{\Pr[\, X_1 = \tau\,]}{\Pr[\, X_0 = \tau\,]} = 1 \ .$$

The proof of Theorem 2 is completed by applying Lemma 1.

4.5 Matching Attacks

The matching attacks that justify the tightness of the dominating terms $pq\ell/2^c$ and $q^2\ell/2^c$ in the security bound of NMAC are similar to those for HMAC, and are therefore omitted here.

A Functional Graph of a Random Function

The properties of the functional graph of a random function has been exploited in a series of generic attacks against hash-based MACs [31,27,32,24,16,17,15,6,4], and also in recent generic attacks against sponge construction [23,11]. Here we recall some of these properties that are useful in our PRF distinguishing attack in Section 3.5.

Let g be random function mapping n bits to n bits. The functional graph is defined by the successive iteration of the random function g. The following theorems by Flajolet and Odlyzko [19], and Flajolet and Sedgewickby [20] capture the structure of this functional graph.

Theorem 3 ([19]). *The expectations of parameters in the functional graph of a random function g, including the number of components, number of cyclic points, number of terminal points, number of image points, and number of α-th iterate image points, have the following asymptotic forms, as $2^n \to \infty$:*

- *The number of components: $\frac{1}{2}\log 2^n = 0.5n$.*

- *The number of cyclic nodes:* $\sqrt{\pi 2^{n-1}} \approx 1.2 \cdot 2^{n/2}$.
- *The number of terminal nodes:* $\frac{1}{e} \cdot 2^n \approx 0.37 \cdot 2^n$.
- *The number of image points:* $(1 - \frac{1}{e}) \cdot 2^n \approx 0.62 \cdot 2^n$.
- *The number of α-th iterate image points:* $(1 - z_\alpha) \cdot 2^n$ *where* z_α *satisfies* $z_0 = 0, z_{\alpha+1} = e^{-1+z_\alpha}$.

Notably, the functional graph of a random function contains only a logarithmic number of distinct components, and the number of cyclic points is on the order of $2^{n/2}$.

By iterating the function g on a random starting point P, it will follow a path in the functional graph starting from P that eventually reaches the cycle of the component to which P belongs. We refer to the number of points in this path as the tail length, the number of nodes in the cycle as the cycle length, the number of points in the non-repeating trajectory from P as the rho length, and the node that connects the tail and the cycle as the Δ-node of the path.

Theorem 4 ([19]). *Iterating the function g on a random starting point, the expectations of parameters, including the tail length, cycle length, rho length, tree size, and predecessors size, have the following asymptotic forms:*

- *Tail length:* $\sqrt{\pi 2^{n-3}} \approx 0.62 \cdot 2^{n/2}$.
- *Cycle length:* $\sqrt{\pi 2^{n-3}} \approx 0.62 \cdot 2^{n/2}$.
- *Rho length:* $\sqrt{\pi 2^{n-1}} \approx 1.2 \cdot 2^{n/2}$.
- *Tree size:* $2^n/3 \approx 0.34 \cdot 2^n$.
- *Component size:* $2^{n+1}/3 \approx 0.67 \cdot 2^n$.
- *Predecessors size:* $\sqrt{\pi 2^{n-3}} \approx 0.62 \cdot 2^{n/2}$.

Theorem 5 ([20]). *In the functional graph of the random function g, the expected size of the largest tree and the expected size of the largest component are asymptotically* $0.48 \cdot 2^n$ *and* $0.7582 \cdot 2^n$*, respectively.*

From the above theorems, one can see that in a random mapping, most of the points tend to be grouped together in a single giant component, which is expected to have a large tree and a large cycle. With these properties, one can ensure that the cycle length of the giant component in the functional graph could be detected by running the cycle search algorithm several times.

References

1. Keyed Hash Message Authentication Code (HMAC), 2000. Specifies HMAC as an ANSI standard.
2. Information technology – Security techniques – Message Authentication Codes (MACs) – Part 2: Mechanisms using a dedicated hash-function, 2021. Defines HMAC as one of the MAC mechanisms.
3. M. Backendal, M. Bellare, F. Günther, and M. Scarlata. When messages are keys: Is HMAC a dual-prf? In *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III*, pages 661–693, 2023.

4. Z. Bao, I. Dinur, J. Guo, G. Leurent, and L. Wang. Generic attacks on hash combiners. *J. Cryptol.*, 33(3):742–823, 2020.
5. Z. Bao, J. Guo, and L. Wang. Functional graphs and their applications in generic attacks on iterated hash constructions. *IACR Trans. Symmetric Cryptol.*, 2018(1):201–253, 2018.
6. Z. Bao, L. Wang, J. Guo, and D. Gu. Functional graph revisited: Updates on (second) preimage attacks on hash combiners. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, pages 404–427, 2017.
7. M. Bellare. New proofs for NMAC and HMAC: security without collision-resistance. In *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, pages 602–619, 2006.
8. M. Bellare, D. J. Bernstein, and S. Tessaro. Hash-function based prfs: AMAC and its multi-user security. *IACR Cryptol. ePrint Arch.*, page 142, 2016.
9. M. Bellare, D. J. Bernstein, and S. Tessaro. Hash-function based prfs: AMAC and its multi-user security. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, pages 566–595, 2016.
10. M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, pages 1–15, 1996.
11. X. Bonnetain, R. H. Boissier, G. Leurent, and A. Schrottenloher. Improving generic attacks using exceptional functions. In *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part IV*, pages 105–138, 2024.
12. S. Chen and J. P. Steinberger. Tight security bounds for key-alternating ciphers. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 327–350, 2014.
13. J. Coron, Y. Dodis, C. Malinaud, and P. Puniya. Merkle-damgård revisited: How to construct a hash function. In *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, pages 430–448, 2005.
14. I. Damgård. A design principle for hash functions. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 416–427, 1989.
15. I. Dinur. New attacks on the concatenation and XOR hash combiners. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, pages 484–508, 2016.
16. I. Dinur and G. Leurent. Improved generic attacks against hash-based macs and HAIFA. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 149–168, 2014.
17. I. Dinur and G. Leurent. Improved generic attacks against hash-based macs and HAIFA. *Algorithmica*, 79(4):1161–1195, 2017.

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 *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, pages 348–366, 2012.
19. P. Flajolet and A. M. Odlyzko. Random mapping statistics. In *Advances in Cryptology - EUROCRYPT '89, Workshop on the Theory and Application of of Cryptographic Techniques, Houthalen, Belgium, April 10-13, 1989, Proceedings*, pages 329–354, 1989.
20. P. Flajolet and R. Sedgewick. *Analytic Combinatorics.* Cambridge University Press, 2009.
21. P. Gazi, K. Pietrzak, and M. Rybár. The exact prf-security of NMAC and HMAC. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 113–130, 2014.
22. P. Gazi, K. Pietrzak, and S. Tessaro. Generic security of NMAC and HMAC with input whitening. In *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, pages 85–109, 2015.
23. H. Gilbert, R. H. Boissier, L. Khati, and Y. Rotella. Generic attack on duplex-based AEAD modes using random function statistics. In *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV*, pages 348–378, 2023.
24. J. Guo, T. Peyrin, Y. Sasaki, and L. Wang. Updates on generic attacks against HMAC and NMAC. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 131–148, 2014.
25. V. T. Hoang and S. Tessaro. Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, pages 3–32, 2016.
26. H. Krawczyk, M. Bellare, and R. Canetti. HMAC: keyed-hashing for message authentication. *RFC*, 2104:1–11, 1997.
27. G. Leurent, T. Peyrin, and L. Wang. New generic attacks against hash-based macs. In *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II*, pages 1–20, 2013.
28. U. M. Maurer, R. Renner, and C. Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, pages 21–39, 2004.
29. R. C. Merkle. A certified digital signature. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 218–238, 1989.
30. J. Patarin. The "coefficients h" technique. In *Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers*, pages 328–345, 2008.
31. T. Peyrin, Y. Sasaki, and L. Wang. Generic related-key attacks for HMAC. In *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the*

Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings, pages 580–597, 2012.

32. T. Peyrin and L. Wang. Generic universal forgery attack on iterative hash-based macs. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 147–164, 2014.
33. B. Preneel and P. C. van Oorschot. Mdx-mac and building fast macs from hash functions. In *Advances in Cryptology - CRYPTO '95, 15th Annual International Cryptology Conference, Santa Barbara, California, USA, August 27-31, 1995, Proceedings*, pages 1–14, 1995.
34. J. M. Turner. The keyed-hash message authentication code (hmac). *Federal Information Processing Standards Publication*, 198(1):1–13, 2008.