Improving Tightness Gap of GGM Construction

Mridul Nandi

Indian Statistical Institute, Kolkata
mridul@isical.ac.in

Abstract. The prefix-free PRF (pseudorandom function) security of a cascade function based on a compression function f against a q-query distinguisher is reduced to a q-query PRF security of f with a tightness gap ℓq where ℓ represents the length of the longest query among all q queries. In this paper, we have shown a reduction which is also applicable to multiuser setup and improves the tightness gap for both adaptive and non-adaptive distinguishers. As an immediate application of our result, we have shown multiuser security of NMAC, HMAC and many other MACs for the first time. Moreover, the tightness gap is improved in comparison with known single-user analysis. We also have shown a similar tightness gap for single-keyed NMAC. As a result, the constants ipad and opad used in HMAC and existing PRB (pseudorandom bit) assumption on the underlying compression function become redundant.

Keywords: PRF, HMAC, NMAC, cascade, non-adaptive security

1 Introduction

Brief History of Hash-based MAC. Verifying the integrity and authenticity of data is a prime necessity in computer systems and networks. Two parties communicating over an insecure channel use a message authentication code or MAC (or a stronger notion called a pseudorandom function or PRF) with which, the receiver validates data as being sent by the sender. MACs and PRFs are commonly constructed out of block ciphers (e.g. CBC-MAC [6,5], PMAC [9], the NIST-recommended CMAC etc. [18,12]). Popular hash functions were earlier faster than block ciphers in software. Moreover, since hash functions are not usually subject to the export restriction rules of the USA and other countries, there had been a surge of interest in constructing MACs from cryptographic hash functions. However, hash functions were not originally designed for MACs or PRFs and do not accommodate a secret key naturally.

One of the earliest ideas for converting a hash function (mainly the Merkle-Damgård hash [27,11]) into a MAC was simply to prepend the message with the secret key. However, it was soon found that those hash functions suffer from the *length extension attack* (see Example 9.64 of [26]) However, it can be shown to be secure for prefix-free message spaces (no two messages are prefix to each other) as shown in [4] for the cascade construction.

Envelope MAC, HMAC and NMAC. Tsudik in [32] proposed the envelope MAC where was shown to be insecure (even for other variants [19,28]) by

Preneel and van Oorschot [29]. The attacks were possible only because of poor formatting of the key block processed at the end. Later Yasuda [35] and Koblitz and Menezes [22] proved the PRF security of Envelope MAC when appropriate formatting of message is applied. In CRYPTO 1996, authors of [3] proposed NMAC and HMAC and proved them secure under certain assumptions on the underlying compression function.

1.1 Definitions of Cascade, HMAC, NMAC and Envelope MAC

NOTATIONS. In this paper, we fix two positive integers b and c and we write $\{0,1\}^b$ and $\{0,1\}^c$ as B (called set of blocks) and C respectively. We also fix a function Let λ denote the empty string and B^* (or B^+) denote the set of all block tuples (or block tuples with at least one block respectively). For $m := (m[1], \ldots, m[r]) \in \mathsf{B}^+$ and $1 \leq i \leq j \leq r$, we write (i) the number of blocks $\|m\| = r$, (ii) sub-tuple $m[i..j] := (m[i], \ldots, m[j])$, (iii) suffix $m[i..] = m[i..r]$ and (iv) prefix $m[..j] = m[1..j]$. We follow the same conventions when the index of m starts with 0 (i.e. $m = (m[0], m[1], \ldots, m[r])$). For a function $f : \mathsf{C} \times \mathsf{B} \to \mathsf{C}$, we define the *cascade function* $f_h^*(m[..i]) := f^*(h, m[..i]) = f^*\big(f(h, m[..i-1]), m[i]\big)$ and $f^*(h, \lambda) = h$ for all $h \in \mathsf{C}, m[..i] \in \mathsf{B}^+$. One can further extend the domain of f^* to the set of all arbitrary bit strings by applying an appropriate injective padding rule as a preprocessor of the above cascade function. As there is no loss in security we, throughout the paper, assume message space as B^*. .

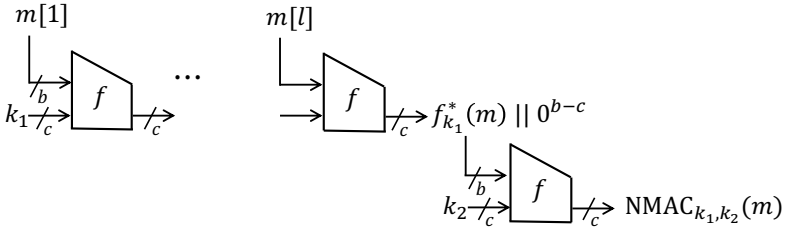

Fig. 1: $\mathsf{NMAC}_{k_1,k_2}(m)$: The top layer represents the cascade output and the bottom layer represents the finalization process applied to the output of the cascade.

NMAC AND HMAC. For keys $k, k_1, k_2 \in \{0,1\}^c$, b-bit constants $\mathsf{ipad}, \mathsf{opad}$ specified in [3], a c-bit initial value IV and message $m \in \mathsf{B}^*$

$$\mathsf{NMAC}_{k_1,k_2}(m) = f_{k_2}(f_{k_1}^*(m) \| 0^{b-c}),$$
$$\mathsf{HMAC}_k(m) = \mathsf{NMAC}_{\mathsf{KDF}(k)}(m),$$

where $\mathsf{KDF}(k) = \big(k_1 := f(\mathrm{IV}, k^{\oplus \mathsf{ipad}}), k_2 := f(\mathrm{IV}, k^{\oplus \mathsf{opad}})\big)$ and $k^{\oplus \alpha} = (k \| 0^{b-c}) \oplus \alpha$. Here, we must assume that $c \leq b$, which used to hold for the earlier compression functions. [1]

[1] Later in RFC 2014 [23] and the special publication FIPS PUB 198-1 [13] by NIST, the MD hash was replaced by any recommended hash function H while defining

ENVELOPE MAC. Finally, we define another old cascade-based MAC construction, called the Envelope MAC or EvMAC. Let pad map a k-bit string to B. For example, if $k \le b$, we consider $\mathsf{pad}(K) = K \| 0^{b-k}$. We define a dual keyed function (interchanging the position of the input and key)

$$f_K^{\downarrow}(x) := f^{\downarrow}(K, x) := f(x, \mathsf{pad}(K)).$$

For any $m \in \mathsf{B}^+$, $K \in \{0,1\}^k$, we define $\mathsf{EvMAC}(K, m) = f^*(\mathrm{IV}, (K', m, K'))$ where $K' = \mathsf{pad}(K)$ and $\mathrm{IV} \in \{0,1\}^c$ is a fixed constant specified by the MD hash function based on f. Using the dual function notation, we can equivalently write

$$\mathsf{EvMAC}(K, m) = f_K^{\downarrow}\big(f_{K'}^*(m)\big)$$

where $K' = f_K^{\downarrow}(\mathrm{IV})$.

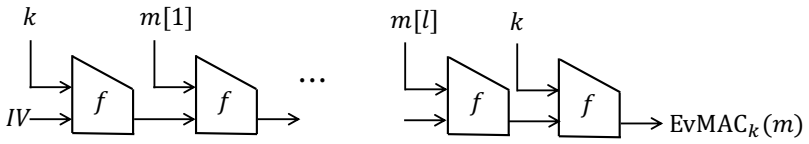

Fig. 2: $\mathsf{EvMAC}_k(m)$: Envelope MAC or Sandwiched MAC.

1.2 Known Results of Hash-based MAC

For a keyed function F, we denote the maximum PRF advantage of F as $\mathbf{Adv}_F^{\mathrm{prf}}(q, \ell, \sigma, T)$ where the maximum is taken over all q-query distinguishers D running in time T such that the total number of blocks and maximum size of the query is at most σ and ℓ respectively. We use the superscripts (i) nprf, (ii) pf_prf and (iii) pf_nprf when we restrict to distinguishers that are (i) non-adaptive (all queries are made before observing any responses), (ii) prefix-free (query tuples are prefix-free) and (iii) prefix-free non-adaptive, respectively. The multiuser advantage for at most u users is similarly denoted as $\mathbf{Adv}_F^{\mathrm{mu\_prf}}(u, q, q_{\max}, \ell, \sigma, T)$ where $q_{\max}$ denotes the maximum number of queries for all users. So, we use mu to denote multiuser distinguisher. When $F = f$ (so $\ell = 1$, $\sigma = q$) we ignore the parameters ℓ and σ. We use the notation θ to denote the tuple $(u, q, q_{\max}, \ell, \sigma)$.

PRF Analysis on Cascade. The security of a fixed-length cascade construction (a special case of a cascade with a prefix-free domain, known as GGM construction) was first implicitly shown in 1984 [15] (and later published in 1986 [16]). The authors have proved asymptotically that a c-bit to $2c$-bit PRBG

HMAC. Similar to the original definition of HMAC, the new definition assumed the hash size to be less than the block size.

(pseudorandom bit generator) can be extended to a fixed-length PRF. Note that such a PRBG is equivalent to a PRF with a one-bit domain (i.e., $b = 1$). In 1996 [4], the GGM results were extended for a general value of b and with an arbitrary prefix-free domain and showed the following:

$$\mathbf{Adv}^{\text{pf\_prf}}_{f^*}(q, \ell, \sigma, T) \leq \ell q \cdot \mathbf{Adv}^{\text{prf}}_{f}(q, T')$$

where $T' := T + O(\sigma)$ (throughout the paper we use this notation). The GGM construction is exactly the fixed-length domain cascade construction after viewing the PRBG as a one-bit PRF. In 2014 [14], Gazi et al. proved the above reduction for non-adaptive PRF security.

PRF Analysis on NMAC and HMAC. Bellare [1] proved that

$$\mathbf{Adv}^{\text{prf}}_{\mathsf{NMAC}^f}(q, \ell, \sigma, T) \leq \ell q^2 \cdot \mathbf{Adv}^{\text{prf}}_{f}(2, O(\ell)) + \mathbf{Adv}^{\text{prf}}_{f}(q, T').$$

Bellare assumed that a good compression function f must satisfy $\mathbf{Adv}^{\text{prf}}_{f}(2, \ell) \approx \ell/2^c$ (presuming that key-guessing is the best strategy for distinguishing f from a random function) and hence the security of NMAC is dominated by the bound $\ell^2 q^2/2^c$. Koblitz and Menezes (KM) observed that the reduction used in the preceding proof is non-constructive or existential (see [30,21] for details about different types of reductions). KM and later Bernstein and Lange [8] showed that for almost all functions $f' : \{0,1\}^c \times \mathsf{B} \to \{0,1\}$, there exists a 1-query distinguisher $\mathcal{A}$ running in $O(1)$ time such that

$$\mathbf{Adv}^{\text{prf}}_{f'}(\mathcal{A}) \geq \frac{1}{2^{c/2}}. \tag{1}$$

This means that Bellare's result cannot guarantee security better than $\ell q^2/2^{c/2}$. This violates the tightness claim of Bellare (see [20] for a detailed discussion). Later in [2], the above tightness claim was withdrawn and revised. In 2013, Koblitz-Menezes [20] also provided a constructive reduction and proved the following result (ignoring a dominated term):

$$\mathbf{Adv}^{\text{prf}}_{\mathsf{NMAC}^f}(q, \ell, \sigma, T) \leq \ell q \cdot \mathbf{Adv}^{\text{prf}}_{f}(q, T'). \tag{2}$$

One year later in 2014 [14], Gazi et al. proved the following security of NMAC through a constructive reduction:

$$\mathbf{Adv}^{\text{prf}}_{\mathsf{NMAC}^f}(q, \ell, \sigma, T) \leq \ell q \cdot \mathbf{Adv}^{\text{nprf}}_{f}(q, T') + \mathbf{Adv}^{\text{prf}}_{f}(q, T') + q^2/2^c. \tag{3}$$

From the definition of HMAC, one can easily see that PRF security of HMAC can be reduced to PRF security of HMAC and PRBG security of KDF.

1.3 Our Contributions

1. multiuser PRF Security of Cascade. In this paper, we provide two reductions for the multiuser PRF security of the cascade construction:

$$\mathbf{Adv}^{\text{mu\_pf\_prf}}_{f^*}(\theta, T) \leq \sigma \cdot \mathbf{Adv}^{\text{prf}}_{f}(q_{\max}, T') \tag{4}$$

$$\mathbf{Adv}^{\text{mu\_pf\_nprf}}_{f^*}(\theta, T) \leq \sigma \cdot \mathbf{Adv}^{\text{nprf}}_{f}(1, T') + u \cdot \mathbf{Adv}^{\text{nprf}}_{f}(q_{\max}, T'). \tag{5}$$

The first reduction improves the tightness gap from ℓq to σ. The second reduction further improves the query complexity by bringing down it to 1.

2. Non-adaptive PRF security under weak f. Due to the key guessing attack, the above bounds cannot guarantee a security better than $\sigma T/2^c$ for any function f. Suppose f is a keyed function with a higher non-adaptive PRF advantage, such as $\mathbf{Adv}^{\mathrm{nprf}}_{f}(D,T) \approx DT/2^c + 2^{-c/2}$. We still prove a similar advantage (up to a logarithmic factor) against a non-adaptive distinguisher:

$$\mathbf{Adv}^{\mathrm{mu\_pf\_nprf}}_{f^\star}(\theta,T) \leq \frac{(\sigma' T + \sigma'^2)\cdot \log_2 q_{\max}}{2^c}$$

Applications to HMAC NMAC and others. It can be shown that the generic reduction from NMAC to cascade (see Eq. 3) and HMAC to NMAC can be extended for a multiuser set-up. Hence, our results for non-adaptive PRF security of cascade can be directly applied for multiuser security of HMAC and NMAC. Following similar approach, simpler and improved analysis for boosted MD (Asiacrypt 2007 [34]) and MDP (JoC 2007 [17]) are given.

3. Security of Single Keyed NMAC, constant-free HMAC and EvMAC. We prove the security of the single-keyed NMAC construction $\mathsf{1k\_NMAC}_K = \mathsf{NMAC}_{K,K}$. This helps not only to eliminate the two constants used in HMACbut also to weaken the PRBG assumption on f. To prove this, we first establish a reduction for single-keyed composition. This single-keyed composition also helps us to prove the multiuser PRF security of Envelope MAC. Our result on Envelope MAC does not require any related-key type assumption appearing in [22].[2] In particular, we show the following three results:

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{mu_prf}}_{\mathsf{1k_NMAC}^f}(\theta') &\leq \mathbf{Adv}^{\mathrm{mu_pf_nprf}}_{f^*}(q',q',q_{\max},\ell+1,\sigma',\sigma_{\max},T') + \\
&\quad + u'\cdot \mathbf{Adv}^{\mathrm{prf}}_{f}(2q_{\max},T') + \frac{2q'^2}{2^c}, \\
\mathbf{Adv}^{\mathrm{mu_prf}}_{\mathsf{EvMAC}}(\theta') &\leq \mathbf{Adv}^{\mathrm{mu_pf_nprf}}_{f^*}(q',q',q_{\max},\ell,\sigma',\sigma_{\max},T') + \\
&\quad + u'\cdot \mathbf{Adv}^{\mathrm{prf}}_{f}(q_{\max}+1,T') + \frac{q'^2}{2^c} \text{ and} \\
\mathbf{Adv}^{\mathrm{mu_prf}}_{\mathsf{HMAC}'}(\theta') &\leq \mathbf{Adv}^{\mathrm{mu_prf}}_{\mathsf{1k_NMAC}}(\theta'),
\end{aligned}$$

where HMAC' is the same as HMAC when the two constants opad and ipad are replaced by the zero-bit string. Moreover, the security of the modified HMAC does not require the PRBG property (it only needs the regular property as KDF does not expand the output size in the modified definition, assuming key size to be as large as the chain size c).

[2] Yasuda proved PRF security of Envelope MAC (also called "Sandwich MAC," see [35]), along the lines of Bellare's NMAC security proof in [2]. Thus, the issues for NMAC are also present in his analysis. Koblitz and Menezes [22] proved the constructive reduction, but relies on some related-key security.

We note that all of our non-adaptive multiuser PRF securities for the cascade construction are applicable to these variants. We finally note that all reductions in our analysis are constructive and so the bounds apply to a uniform setting when we naturally extend the result in an asymptotic set-up.

2 Preliminaries

NOTATIONS. We follow the notation as described in Sect. 1.1. We write x^q to denote a q tuple $(x_1, \ldots, x_q)$. For the sake of notational simplicity we write a tuple of q pairs $((x_1, y_1), \ldots, (x_q, y_q))$ as (x^q, y^q) (and similarly for more than two tuples). When x is chosen uniformly from S and independent with all random variables defined so far, we simply denote it as $x \xleftarrow{\$} S$. We write $\mathsf{B}^+_{\mathrm{mu}} = \mathcal{I} \times \mathsf{B}^+$ and $\mathsf{B}^*_{\mathrm{mu}} = \mathcal{I} \times \mathsf{B}^*$ for some set $\mathcal{I}$ (representing user index space).

PREFIX-FREE. For $a \leq b$, we call $m[..a]$ a *prefix* of $m[..b]$ and denote it as $m[..a] \preceq m[..b]$. If $m[..a] \neq m[..b]$ then we also denote it as $m[..a] \prec m[..b]$. In this case, we write $m[..b] \setminus m[..a] = m[a+1..b]$. A tuple of messages m^q is called prefix-free if for all $i \neq j$, $m_i \not\preceq m_j$.

JOINT QUERY SPACE Let $\mathcal{Q}(u, q, q_{\max}, \ell, \sigma)$, called join query space, represent the set of all q tuples m^q of messages of at most ℓ blocks with altogether at most σ blocks such that

- the number of distinct elements present in $m_1[0], \ldots, m_q[0]$ is at most u and
- for all $\gamma \in \mathcal{I}$, the set $\mathcal{Q}_\gamma := \{i : m_i[0] = \gamma\}$ has at most $q_{\max}$ elements,

When $\ell = 1$ or $u = 1$ (single user) we simply write the set as $\mathcal{Q}^{\mathrm{mu}}(u, q, q_{\max})$ or $\mathcal{Q}^{\mathrm{su}}(q, \ell, \sigma)$ respectively. When $u = 1$, we simply skip the user index space and consider m^q with $m_i \in \mathsf{B}^*$.

For any joint query space $\mathcal{Q}$, we write $\mathcal{Q}_{\mathrm{pf}} = \mathcal{Q} \cap \mathcal{P}$ where $\mathcal{P}$ is the set of tuples of prefix-free messages.

2.1 Prefix-Tree

Let us fix a parameter tuple $\theta = (u, q, q_{\max}, \ell, \sigma)$ and $m^q \in \mathcal{Q}_{\mathrm{pf}}(\theta)$. We now associate the tuple m^q with a tree $\mathcal{T}_{m^q}$, called *prefix tree*, over the vertex set $V \cup \{\lambda\}$ where

$$V = \mathsf{Prefix}(m^q) = \{x \in \mathsf{B}^*_{\mathrm{mu}} : x \preceq m_i,\ i \in [q]\}.$$

It consists of all directed edges of the form $\mathsf{chop}(y) \to y$ for $y \in V$ where $\mathsf{chop}(y)$ represents the tuple after removing the last block from y. It is a rooted tree with λ as the root (it is the only vertex with in-degree zero). For every $v \in V \cup \{\lambda\}$, we define the set of outwards nodes, or children nodes, as $\mathsf{ch}(v) = \{u : v \to u\}$. For a set $\mathsf{ch}(S) = \cup_{v \in S} \mathsf{ch}(v)$. Let L denote the set of leaf nodes (having zero out-degree) which is same as the set $\{m_1, \ldots, m_q\}$ (as m^q is a prefix-free). Let $\mathsf{Prefix}'(m^q) := V \setminus L$ be the set of all intermediate nodes. We denote $d := |V \setminus L| - 1$ and so $d \leq \sigma' := \sigma - q$.

Definition 1 (leave-cut). *A subset $U \subseteq V$ is called leave-cut if it is prefix-free and for every leaf m_i in the prefix tree T_{m^q} there exists exactly one node $u \in U$ such that $u \preceq m_i$.*

It is easy to see that whenever U is a leave-cut and $S \subseteq U \setminus L$ then $U' = (U \setminus S) \cup \mathsf{ch}(S)$ is also a leave-cut as we cut all leaf nodes m_i with a member of S as a prefix by one of its children nodes.

2.2 Distinguisher and Distinguishing Advantage

Oracle and Keyed Function. An $(\mathcal{X}, \mathcal{Y})$-oracle $\mathcal{O}$ is an interactive probabilistic algorithm that takes inputs from the set $\mathcal{X}$ and returns elements from the set $\mathcal{Y}$. A *random function* $\mathsf{RF}_{\mathcal{X}\to\mathcal{Y}}$ is a $(\mathcal{X}, \mathcal{Y})$-oracle which returns $\mathsf{RF}(x)$ on an input $x \in \mathcal{X}$, where $\mathsf{RF} \leftarrow_{\$} \mathsf{Func}(\mathcal{X}, \mathcal{Y})$, the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$. We write $\mathsf{RF}_{*\to\mathcal{Y}}$ to denote a random function from some domain $\mathcal{X}$ to $\mathcal{Y}$. A *keyed function* $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ can be viewed as an $(\mathcal{X}, \mathcal{Y})$-oracle where the key $K \leftarrow_{\$} \mathcal{K}$ (key space), and then for every query x it returns $F_K(x) := F(K, x)$. Note that the key is sampled once and is used for every query. We also call it $(\mathcal{X}, \mathcal{Y})$ keyed function (with a key space $\mathcal{K}$). A random function $\mathsf{RF}_{\mathcal{X}\to\mathcal{Y}}$ is an example of $(\mathcal{X}, \mathcal{Y})$ keyed function.

Definition 2 (multiuser keyed function). *Let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a keyed function. An* $\mathcal{I}$-folded multiuser *$F^{\otimes\mathcal{I}}$ (or simply $F^{\otimes}$) is an $(\mathcal{I} \times \mathcal{X}, \mathcal{Y})$ keyed function which returns*

$$F^{\otimes}(\gamma\ ;\ x) := F(\mathsf{RF}_{\mathcal{I}\to\mathcal{K}}(\gamma), x)$$

on an input (γ, x).

In other words, a multiuser keyed function samples keys independently for all user index γ from the user index space $\mathcal{I}$ (i.e. $K_\gamma \leftarrow_{\$} \mathcal{K}$ for all $\gamma \in \mathcal{I}$) and then it behaves as the original keyed function F_{K_γ} for any query with the user input γ.

Oracle Algorithm. A q-query t-time $(\mathcal{X}, \mathcal{Y})$-oracle algorithm $\mathcal{A}$ is an interactive algorithm that can interact with any $(\mathcal{X}, \mathcal{Y})$-oracle $\mathcal{O}$ (called a compatible oracle with $\mathcal{A}$), that makes at most q queries to its oracle, runs for time t, and finally returns some output z, denoted as $\mathcal{A}^{\mathcal{O}} \to z$, (if $z \in \{0, 1\}$, we also call it a *distinguisher*). When $\mathcal{O}$ is a multiuser oracle, $\mathcal{A}$ is called a u-user oracle algorithm if the number of distinct user indices queried is at most u.

Transcript. The transcript of interaction between $\mathcal{A}$ and $\mathcal{O}$ is denoted as

$$\tau(\mathcal{A}^{\mathcal{O}}) := (\tau_{\text{query}}(\mathcal{A}^{\mathcal{O}}) := x^{q'},\ \tau_{\text{resp}}(\mathcal{A}^{\mathcal{O}}) := y^{q'})$$

where x_i denotes the ith query (which includes the user index in case of multiuser oracle algorithm) and y_i denotes the response of the query, $1 \le i \le q'$.

Definition 3. *A distinguisher* D *is called a* (θ, t)*-complexity distinguisher (or prefix-free distinguisher) if* D *runs for time* t *and for all compatible oracles* $\mathcal{O}$*, the transcript* $\tau_{\text{query}}(\mathsf{D}^{\mathcal{O}}) \in \mathcal{Q}(\theta)$ *(or* $\tau_{\text{query}}(\mathsf{D}^{\mathcal{O}}) \in \mathcal{Q}_{\text{pf}}(\theta)$ *respectively).*

We sometimes ignore the time parameter t. Sometimes, we adjoin a post-processing oracle $\mathcal{O}_{\text{pp}}$ (may be an internal state shared oracle with $\mathcal{O}$) which returns an additional response S after all queries have been made. In this case, we write the response transcript as $\tau_{\text{resp}}(\mathcal{A}^{\mathcal{O},\mathcal{O}_{\text{pp}}}) := (y^{q'}, S)$.

Non-adaptive Oracle Algorithm. We call $\mathcal{A}$ *non-adaptive* if the number of queries and all the queries do not depend on responses. We denote a *deterministic non-adaptive oracle as* D_{x^q} *which makes* $x_1, \ldots, x_q$ *as all queries.*

Definition 4 (distinguishing advantage). *Let* F *and* G *be* $(\mathcal{X}, \mathcal{Y})$*-oracles. We define the* distinguishing advantage *of a distinguisher* D *as* $\Delta_{\mathsf{D}}(F\ ;\ G) := \left|\Delta^{*}_{\mathsf{D}}(F\ ;\ G)\right|$ *where* $\Delta^{*}_{\mathsf{D}}(F\ ;\ G) = \Pr(\mathsf{D}^{F} = 1) - \Pr(\mathsf{D}^{G} = 1)$ *is called signed distinguishing advantage.*

In the above definition when G is a random function, we call the distinguishing advantages the PRF-advantages of D against F. More precisely, the (multiuser) PRF-advantage of D against F are $\mathbf{Adv}^{\text{prf}}_{F}(\mathsf{D}) := \Delta_{\mathsf{D}}(F\ ;\ \mathsf{RF}_{\mathcal{X}\to\mathcal{Y}})$ and $\mathbf{Adv}^{\text{mu\_prf}}_{F}(\mathsf{D}) := \Delta_{\mathsf{D}}(F^{\otimes}\ ;\ \mathsf{RF}^{\otimes}_{\mathcal{X}\to\mathcal{Y}})$. We use the superscript nprf or mu_nprf when we consider non-adaptive or multiuser non-adaptive distinguishers respectively. When distinguishers make only prefix-free queries only, we use superscript pf_nprf or pf_mu_nprf. For any one of the superscripts xxx, we now define $\mathbf{Adv}^{\text{xxx}}_{F}(\theta, t) = \max_{\mathsf{D}} \mathbf{Adv}^{\text{prf}}_{F}(\mathsf{D})$ where the maximum is taken over (θ, t)-distinguishers corresponding to xxx security notion.

2.3 Tools for Security Proofs

Convention. Without loss of generality, we can assume that any q-query oracle makes exactly q queries (since otherwise, some dummy queries can be added with no loss in distinguishing advantage).

Definition 5. *Two* $(\mathcal{X}, \mathcal{Y})$*-oracles* $\mathcal{O}$ *and* $\mathcal{O}'$ *are called* equivalent *on* $\mathcal{T} \subseteq \mathcal{X}^q \times \mathcal{Y}^q$*, denoted as* $\mathcal{O} \cong_{\mathcal{T}} \mathcal{O}'$*, if for all* $(x^q, y^q) \in \mathcal{T}$

$$\Pr(\tau_{\text{resp}}(\mathsf{D}^{\mathcal{O}}_{x^q}) = y^q) = \Pr(\tau_{\text{resp}}(\mathsf{D}^{\mathcal{O}'}_{x^q}) = y^q)$$

When $\mathcal{T} = \mathcal{Q} \times \mathcal{Y}^q$ *we simply write* $\mathcal{O} \cong_{\mathcal{Q}} \mathcal{O}'$ *and we call* $\mathcal{O}$ *and* $\mathcal{O}'$ *are equivalent on a joint query space* $\mathcal{Q}$*.*

For $\mathcal{T} \subseteq \mathcal{X}^q \times \mathcal{Y}^q \times \mathcal{S}$*, we say that* $(\mathcal{O}, \mathcal{O}_{\text{pp}}) \cong_{\mathcal{T}} (\mathcal{O}', \mathcal{O}'_{\text{pp}})$ *if for all* $(x^q, y^q, S) \in \mathcal{T}$

$$\Pr(\tau_{\text{resp}}(\mathsf{D}^{\mathcal{O},\mathcal{O}_{\text{pp}}}_{x^q}) = (y^q, S)) = \Pr(\tau_{\text{resp}}(\mathsf{D}^{\mathcal{O}',\mathcal{O}'_{\text{pp}}}_{x^q}) = (y^q, S))$$

Observation. When the user index is also considered as a part of the input, the *two random functions* $\mathsf{RF}^{\otimes\mathcal{I}}_{\mathcal{X}\to\mathcal{Y}}$ *and* $\mathsf{RF}_{\mathcal{I}\times\mathcal{X}\to\mathcal{Y}}$ *are equivalent and so we use the notations interchangeably in the paper.*

Lemma 1 (identical until bad). *(1) Let* $\mathcal{O} \cong_{\mathcal{T}} \mathcal{O}'$ *for a collection of transcript* $\mathcal{T}$ *then for any distinguisher* D*,*

$$\Delta_{\mathsf{D}}(\mathcal{O}\ ;\ \mathcal{O}') \leq \Pr(\tau(\mathsf{D}^{\mathcal{O}}) \notin \mathcal{T}).$$

(2) Suppose $(\mathcal{O}, \mathcal{O}_{\mathrm{pp}}) \cong_{\mathcal{T}} (\mathcal{O}', \mathcal{O}'_{\mathrm{pp}})$ *for a collection of transcript* $\mathcal{T}$ *then for any distinguisher* D*,* $\Delta_{\mathsf{D}}(\mathcal{O}; \mathcal{O}') \leq \Pr(\tau(\mathsf{D}^{\mathcal{O},\mathcal{O}_{\mathrm{pp}}}) \notin \mathcal{T})$*.*

This is a classical result that is widely used in game-playing technique proofs [7,31]. The proof simply follows from the definition of distinguishing advantage and equivalence of oracles.

For the sake of simplified presentation, we also denote a transcript in a different order namely (x^q, S, y^q). A similar statement of the following lemma can be found in [24] in the language of a random system. This says that under certain assumptions probability of realizing a collection of transcripts

Lemma 2 (adaptive to non-adaptive). *Suppose* $\mathcal{O}$ *and* $\mathcal{O}_{\mathrm{pp}}$ *use independent random coins. Let* $\mathcal{T} = \mathcal{T}' \times \mathcal{Y}^q$ *for some* $\mathcal{T}' \subseteq \mathcal{X}^q \times \mathcal{S}$*. Then, for every adaptive distinguisher* D *there is a non-adaptive distinguisher* D_0 *such that*

$$\Pr(\tau(\mathsf{D}_0^{\mathcal{O},\mathcal{O}_{\mathrm{pp}}}) \notin \mathcal{T}) = \Pr(\tau(\mathsf{D}_0^{\mathcal{O},\mathcal{O}_{\mathrm{pp}}}) \notin \mathcal{T}).$$

Proof. The non-adaptive distinguisher runs D and samples a random coin R for $\mathcal{O}$. Thus, all queries made by D is simulated by D_0 using its random coin. Let x^q be all queries of D and y^q be responses of D_0 of these queries. Then, D_0 returns all queries to its oracle $\mathcal{O}$ and obtain responses (z^q, S) where z^q is the responses of $\mathcal{O}$ but using another independent random coin and $\mathcal{O}_{\mathrm{pp}} \to S$. Now

$$\begin{aligned}\Pr(\tau(\mathsf{D}_0^{\mathcal{O},\mathcal{O}_{\mathrm{pp}}}) \notin \mathcal{T}) &= \Pr((x^q, S, z^q) \notin \mathcal{T}) \\ &= \Pr((x^q, S, y^q) \notin \mathcal{T}) = \Pr(\tau(\mathsf{D}^{\mathcal{O},\mathcal{O}_{\mathrm{pp}}}) \notin \mathcal{T}). \quad \square\end{aligned}$$

3 Formalizing Hybrid Reduction Proof

A reduction algorithm is an essential object in every reduction proof. We now formalize the notion of reduction algorithm and hybrid reduction algorithm.

Definition 6. *A* **reduction algorithm** *or* **simulator** Sim *is an interactive algorithm such that*

- *for any compatible oracle* $\mathcal{O}$*,* $\mathsf{Sim}^{\mathcal{O}}$ *behaves as an oracle and*
- *for any oracle algorithm* $\mathcal{A}$ *(so that* $\mathsf{Sim}^{\mathcal{O}}$ *is a compatible oracle of* $\mathcal{A}$*),* $\mathcal{A}^{\mathsf{Sim}}$ *behaves as an oracle algorithm.*

The joint interaction among all three is denoted as

$$\mathcal{A}^{\mathsf{Sim}^{\mathcal{O}}}.$$

If for every θ*-complexity algorithm* $\mathcal{A}$*,* $\mathcal{A}^{\mathsf{Sim}}$ *is an* θ'*-complexity algorithm, we write* $\theta^{\mathsf{Sim}} = \theta'$*. Note that the run time for* $\mathcal{A}^{\mathsf{Sim}}$ *is* $t_{\mathsf{Sim}} + t_{\mathcal{A}}$ *where* t_X *represents the time for algorithm* X*.*

Note that a reduction algorithm is neither an oracle nor an oracle algorithm. However, it can be placed in between an oracle algorithm $\mathcal{A}$ and an oracle $\mathcal{O}$. Moreover, $\mathcal{A}^{\mathsf{Sim}}$ is an oracle algorithm and $\mathsf{Sim}^{\mathcal{O}}$ is an oracle.

A set I is called hybrid-index space of Sim if the random coin $\mathcal{C} = I \times \mathcal{C}'$ for some $\mathcal{C}'$. So for every $h \in I$, $\mathsf{Sim}(h)$ represents a reduction algorithm with coin space $\mathcal{C}'$ which behaves like Sim conditioned on the hybrid index h. Let $I \subseteq J$ be a set and $\perp$ represents an arbitrary but a fixed deterministic oracle. Given a reduction algorithm Sim with coin space $I \times \mathcal{C}$, we consider the following reduction algorithm (abusing notation, we also write Sim to denote it) with the coin space $J \times \mathcal{C}'$:

- $h \leftarrow_{\$} J$.
- If $h \notin I$ then it behaves as $\perp$ (irrespective of its oracle).
- else, runs $\mathsf{Sim}(h)$.

Substitution Reduction. In many proofs, we simply substitute one oracle with another (e.g., a PRF by a random function and a pseudorandom permutation or PRP by a random permutation). This substitution reduction can be described formally as follows.

Definition 7 (substitution reduction). *Let F, G, F', G' be four oracles. A simulator Sim is called an 1-step reduction (or a substitution reduction) from the pair of oracles (F', G') to the pair (F, G) on a joint query space $\mathcal{Q}$, if $\mathsf{Sim}^{F} \cong_{\mathcal{Q}} F'$ and $\mathsf{Sim}^{G} \cong_{\mathcal{Q}} G'$.*

We simply denote the above 1-step reduction as $F' \overset{\mathsf{Sim}^{F/G}}{\longrightarrow} G'$. In this case for any $\mathcal{Q}$-query distinguisher D,

$$\Delta^{*}_{\mathsf{D}}(F'; G') = \Delta^{*}_{\mathsf{D}_0}(F; G), \qquad \Delta_{\mathsf{D}}(F'; G') = \Delta_{\mathsf{D}_0}(F; G)$$

where $\mathsf{D}_0 := \mathsf{D}^{\mathsf{Sim}}$. The reason we call it substitution reduction as follows. Let C be a construction which uses a primitive F (or G) as an oracle and let G' be any oracle. Then, distinguishing advantage between C^F and G' can be bounded as

$$\begin{aligned}\Delta_{\mathsf{D}}(C^F, G') &\le \Delta_{\mathsf{D}}(C^F; C^G) + \Delta_{\mathsf{D}}(C^G; G') \\ &\le \Delta_{\mathsf{D}_0}(F; G) + \Delta_{\mathsf{D}}(C^G; G')\end{aligned}$$

where $\mathsf{D}_0 = \mathsf{D}^{\mathsf{Sim}}$ and Sim is the substitution reduction which simply simulates the construction $C^{\mathcal{O}}$ with the help of its oracle $\mathcal{O}$ (which is either F or G). When G and G' are random functions, the above relation can be written as

$$\begin{aligned}\mathbf{Adv}^{\mathrm{prf}}_{C^F}(\mathsf{D}) &\le \mathbf{Adv}^{\mathrm{prf}}_{F}(\mathsf{D}_0) + \mathbf{Adv}^{\mathrm{prf}}_{C^{\mathsf{RF}}}(\mathsf{D}) \\ \mathbf{Adv}^{\mathrm{prf}}_{C^F}(\theta) &\le \mathbf{Adv}^{\mathrm{prf}}_{F}(\theta_0) + \mathbf{Adv}^{\mathrm{prf}}_{C^{\mathsf{RF}}}(\theta)\end{aligned}$$

where $\theta_0 = \theta^{\mathsf{Sim}}$. So, it would be sufficient to bound the PRF advantage of C^{RF}. This approach has been used in several security proofs.

3.1 Hybrid Reduction Algorithm

We now extend the notion of substitution reduction or 1-step reduction to a d-step reduction in the following way.

Definition 8 (d-step reduction). *A simulator* Sim *with a hybrid index space* $[d]$ *is called a* d*-step substitution reduction from a pair of oracles* (F', G') *to a pair of oracles* (F, G) *on a joint query space* $\mathcal{Q}$ *if there are* $d-1$ *intermediate oracles* $\mathcal{O}_1, \ldots, \mathcal{O}_{d-1}$ *such that*

$$\mathcal{O}_0 := F' \overset{\mathsf{Sim}(1)^{F/G}}{\longrightarrow} \mathcal{O}_1 \overset{\mathsf{Sim}(2)^{F/G}}{\longrightarrow} \mathcal{O}_2 \cdots \overset{\mathsf{Sim}(d-1)^{F/G}}{\longrightarrow} \mathcal{O}_{d-1} \overset{\mathsf{Sim}(d)^{F/G}}{\longrightarrow} G' := \mathcal{O}_d$$

In other words, $\mathsf{Sim}(j)$ *is a substitution reduction from* $(\mathcal{O}_{j-1}, \mathcal{O}_j)$ *to* (F, G) *on a joint query space* $\mathcal{Q}$*. When the simulator* Sim *and the oracles* F, G *are understood, we simply ignore the notation. When* G *and* G' *are random functions, we call* Sim *a* d*-step PRF-reduction from* F' *to* F*.*

The above definition can be easily extended for any arbitrary hybrid index space I (not necessarily of the form $[d]$). Note that in the above definition, it is not required to define the intermediate oracles explicitly. It is sufficient to show the following two conditions :

- (boundary condition): $\mathsf{Sim}^F(1) \cong_{\mathcal{Q}} F'$, $\mathsf{Sim}^G(d) \cong_{\mathcal{Q}} G'$ and
- (transition equivalence): $\mathsf{Sim}^G(j) \cong_{\mathcal{Q}} \mathsf{Sim}^F(j+1)$ for all $j \in [d-1]$.

The reduction algorithm Sim is a d-step reduction as we set $\mathcal{O}_j = \mathsf{Sim}^F(j+1)$ for all $1 \le j \le d-1$. However, sometimes it is easier to first describe the intermediate oracles $\mathcal{O}_j$ in a stand-alone way and then we show the equivalence between oracles. Now we state an abstraction of hybrid proof. Some known applications are given in Appendix A.

Lemma 3 (hybrid reduction). *Let* Sim *be a* d*-step substitution reduction from* (F', G') *to* (F, G)*. Then for any* (θ, t)*-complexity distinguisher* D*, we have a* $(\theta^{\mathsf{Sim}}, t + t_{\mathsf{Sim}})$*-complexity distinguisher* $\mathsf{D}' := \mathsf{D}^{\mathsf{Sim}}$ *such that*

$$\Delta_{\mathsf{D}'}(F \ ; \ G) = \frac{1}{d} \cdot \Delta_{\mathsf{D}}(F' \ ; \ G').$$

So, if Sim *is a* d*-step PRF-reduction from* F' *to* F*, we have*

$$\mathbf{Adv}^{\mathrm{prf}}_{F'}(\theta, t) \le d \cdot \mathbf{Adv}^{\mathrm{prf}}_{F}(\theta', t + t_{\mathsf{Sim}}).$$

Proof. Let the hybrid index space of Sim be I with $|I| = d$. Note that $\Pr(\mathsf{D}'^{\mathcal{O}} \to 1) = \frac{1}{d} \cdot \sum_{i \in I} \Pr(\mathsf{D}^{\mathsf{Sim}^{\mathcal{O}}(i)} \to 1)$ for any compatible oracle $\mathcal{O}$. Hence,

$$\begin{aligned}
\Delta^*_{\mathsf{D}'}(F; G) &= \frac{1}{d} \cdot \sum_{i \in I} \Pr(\mathsf{D}^{\mathsf{Sim}^F(i)} \to 1) - \frac{1}{d} \cdot \sum_{i \in I} \Pr(\mathsf{D}^{\mathsf{Sim}^G(i)} \to 1) \\
&\overset{(1)}{=} \frac{1}{d} \cdot \Pr(\mathsf{D}^{F'} \to 1) - \frac{1}{d} \cdot \Pr(\mathsf{D}^{G'} \to 1) \\
&= \frac{1}{d} \cdot \Delta^*_{\mathsf{D}}(F'; G')
\end{aligned}$$

Note that the sum in the first equation is a telescoping sum due to the definition of d-step reduction of Sim. This justifies equality (1). Now the result follows by taking absolute value in both sides. □

4 A Generalized Adaptive Reduction for Cascade

In this section, we provide a general method of reduction proof for the multiuser cascade against both adaptive and non-adaptive distinguishers. Let $\mathcal{Q} := \mathcal{Q}_{\mathrm{pf}}(\theta)$ be a prefix-closed (i.e., for all $m^q \in \mathcal{Q}$, and $i \in [q]$, $m^i \in \mathcal{Q}$) joint-query space of prefix-free tuples. Suppose an adversary makes queries $m_1, m_2, \ldots, m_q$ adaptively. On ith query, we represent the state as m^i which captures all queries till the ith query (including the ith query). We write $m_i = m_i[0..\ell_i] \in \mathcal{I} \times \mathsf{B}^{\ell_i}$. Before we define our simulator for cascade we define a structure τ which uniquely associates a simulator Sim_τ.

Definition 9 (structure). *A structure for a joint query space $\mathcal{Q}_{\mathrm{pf}}(\theta)$ is a pair $\tau := (R, \rho)$ of functions $(R, \rho) : \mathcal{Q} \to \mathsf{B}^*_{\mathrm{mu}} \times \{\mathsf{orc}, \mathsf{sim}\}$ satisfying the following conditions:*

- *the set $\{R(m^i) : i \in [q]\}$ is a leave-cut for $\mathcal{T}_{m^q}$,*
- *$R(m^i) = R(m^j) \Rightarrow \rho(m^i) = \rho(m^j)$ and*
- *$\rho(m^i) = \mathsf{orc} \Rightarrow R(m^i) \neq m_i$.*

We define a $\mathcal{Q}$-oracle $\mathcal{O}_R$ which returns responses $z_1, \ldots, z_q$ on queries $m_1, \ldots, m_q$ respectively where

$$z_i = f^*_{\mathsf{RF}(R(m^i))}(m_i \setminus R(m^i)) \quad \forall i \in [q].$$

The function R puts an independent key at the node $R(m^i)$ on ith query m_i and follows the definition of f^* starting from the key on the node applied to the rest of the message blocks. We can imagine this as a traversal of the path from the root till the leaf node m_i. Two extreme and trivial examples are $R_{\mathrm{root}}(m^i) = m_i[0]$ and $R_{\mathrm{full}}(m^i) = m_i$ for all m^i. As $R_{\mathrm{root}}(m^i) = m_i[0]$, we actually compute $f^{*\otimes}$. Whereas in case of $R_{\mathrm{full}}(m^i) = m_i$, we assign independent keys at each leaf node and all these keys are eventually outputs (so, behaves like a random function). Thus, $\mathcal{O}_{R_{\mathrm{root}}} \cong f^{*\otimes}$ and $\mathcal{O}_{R_{\mathrm{full}}} = \mathsf{RF}$. Any other R induces an immediate oracle in between these two extreme examples. The function ρ suggests where to make oracle query and where it would be simulated by the simulator itself. More precisely, we define a simulator Sim_τ below.

Definition 10 (Simulator associated with a structure τ). *Let $\mathcal{C} = \{0,1\}^c$. For every structure $\tau = (R, \rho)$ on a joint query space $\mathcal{Q}$, we associate a simulator Sim_τ defined as follows. Given any multiuser $(\mathsf{B}, \mathcal{C})$-oracle $\mathcal{O}^\otimes$ with user index space $\mathcal{I} \times \mathsf{B}^*$, Sim_τ returns z^q on q adaptive query m^q (from an oracle algorithm) where*

$$z_i = \begin{cases} f^*_{\mathsf{RF}_{*\to\mathcal{C}}(m_i[..s])}(m_i[s+1..]) & \text{if } \rho(m^i) = \mathsf{sim}, \\ f^*_{\mathcal{O}^\otimes(m_i[..s]\ ;\ m_i[s+1])}(m_i[s+2..]) & \text{if } \rho(m^i) = \mathsf{orc}, \end{cases}$$

where $R(m^i) = m_i[..s]$. *Here, the outputs of* $\mathsf{RF}_{*\to\mathcal{C}}(m_i[..s])$ *is simulated by the simulator itself (as a part of the simulator's random coin). We write* $\mathcal{Q}^\tau = \mathcal{Q}(u, q, q_{\max})$ *if for all* $m^q \in \mathcal{Q}$,

1. $Q := \{i : \rho(m^i) = \mathsf{orc}\}$ *has at most* u *elements,*
2. $\sum_{i\in Q} |\mathsf{ch}(R(m^i))| \le q$ *and*
3. $\max_{i\in Q} |\mathsf{ch}(R(m^i))| \le q_{\max}$.

Thus, $\theta^{\mathsf{Sim}_\tau} = (u, q, q_{\max})$.

By definition, whenever $\rho(m^i) = \mathsf{orc}$, $s < \ell_i$ and so $m_i[s+1]$ is defined. Moreover, both $\mathsf{RF}_{*\to\mathcal{C}}(m_i[..s])(m_i[s{+}1..])$ and $\mathcal{O}^{\otimes}(m_i[..s]\,;\, m_i[s{+}1])(m_i[s{+}2..])$ are members of $\mathcal{C}$ and hence f^* outputs are defined. For a structure $\tau = (R < \rho)$, we define

$$\mathrm{next}(\tau)(m^i) = \begin{cases} m_i[..s] & \text{if } \rho(m^i) = \mathsf{sim} \\ m_i[..s+1] & \text{if } \rho(m^i) = \mathsf{orc}, \end{cases}$$

where $R(m^i) = m_i[..s]$. Whenever $\rho(m^i) = \mathsf{orc}$, $s < \ell_i$ and so $m_i[s{+}1]$ is defined. Let $U = \{R(m^i) : i \in [q]\}$ and $S = \{R(m^i) : \rho(m^i) = \mathsf{orc}\}$. It is easy to see that the set $\{\mathrm{next}(\tau)(m^i) : i \in [q]\} = U \setminus S \cup \mathsf{ch}(S)$ and hence it is a leave-cut set for all m^q (see Sect. 2).

Proposition 1 (generalized reduction for multiuser cascade). *Let* θ *be some complexity parameter. Suppose* $\tau_i := (R_i, \rho_i)$, $0 \le i \le d$, *are hybrid structures over* $\mathcal{Q}_{\mathrm{pf}}(\theta)$ *such that*

1. $R_0 = R_{\mathrm{root}}$ *and* $R_d = R_{\mathrm{full}}$ *(boundary condition)*
2. $\mathrm{next}(\tau_{i-1}) = R_i$ *for all* $i \in [d]$ *(transition equivalence)*
3. $\mathcal{Q}_{\mathrm{pf}}(\theta)^{\tau_i} = \mathcal{Q}(u, q, q_{\max})$ *for all* $i \in [d]$ *(complexity reduction)*

Then,

$$\mathbf{Adv}^{\mathrm{mu\_pf\_prf}}_{f^{*\otimes}}(\theta) \le d \cdot \mathbf{Adv}^{\mathrm{mu\_prf}}_{f^{\otimes}}(u, q, q_{\max}). \tag{6}$$

Proof. Let Sim be a reduction which samples $h \leftarrow\!\!\$\, [d]$ and then run $\mathsf{Sim}_{\tau_{h-1}}$. Note that for any prefix-free θ-complexity distinguisher $\mathcal{A}$, $\mathcal{A}^{\mathsf{Sim}}$ is a $(u, q, q_{\max})$-complexity distinguisher. Now we show the following d-step reduction:

$$f^{*\otimes} \xrightarrow{\mathsf{Sim}(1)^{f^\otimes/\mathsf{RF}}} \mathcal{O}_{R_1} \xrightarrow{\mathsf{Sim}(2)^{f^\otimes/\mathsf{RF}}} \mathcal{O}_{R_2} \cdots \xrightarrow{\mathsf{Sim}(d-1)^{f^\otimes/\mathsf{RF}}} \mathcal{O}_{R_{d-1}} \xrightarrow{\mathsf{Sim}(d)^{f^\otimes/\mathsf{RF}}} \mathsf{RF}'$$

From the definition of Sim_τ and oracle $\mathcal{O}_R$, it is straightforward that $\mathsf{Sim}_\tau^{f^\otimes}$ is equivalent to $\mathcal{O}_R$ and $\mathsf{Sim}_\tau^{\mathsf{RF}^\otimes}$ is equivalent to $\mathcal{O}_{\mathrm{next}(\tau)}$. However, we have seen that $\mathcal{O}_{R_0} \cong f^{*\otimes}$ and $\mathcal{O}_{R_d} \cong \mathsf{RF}$. The above d-step reduction follows from the given condition that $\mathrm{next}(\tau_{i-1}) = R_i$. Now, by using hybrid reduction lemma, the result follows. □

4.1 Application: Classical Reduction for Cascade

As a first application, we establish the classical reduction proof for cascade. Let $R_h(m^i) = m_i[..h]$ and

$$\rho(m^i) = \begin{cases} \mathsf{orc} & \text{if } h < \ell_i \\ \mathsf{sim} & \text{if } h \geq \ell_i. \end{cases}$$

Then, it is easy to see that for all $h \in [\ell]$, $\mathrm{next}(R_{i-1}, \rho_{i-1}) = R_i$ for all $i \in [\ell]$. Moreover, $R_0 = R_{\mathrm{root}}$ and $R_\ell = R_{\mathrm{full}}$. So, by Proposition 1.

$$\mathbf{Adv}^{\mathrm{mu\_pf\_prf}}_{f^{*\otimes}}(u, q, q_{\max}, \ell, \sigma) \leq \ell \cdot \mathbf{Adv}^{\mathrm{mu\_prf}}_{f^{\otimes}}(u, q, q_{\max}). \tag{7}$$

Now by using standard multiuser to single user reduction we have

$$\mathbf{Adv}^{\mathrm{mu\_pf\_prf}}_{f^{*\otimes}}(u, q, q_{\max}, \ell, \sigma) \leq \ell q_{\max} \cdot \mathbf{Adv}^{\mathrm{mu\_prf}}_{f}(q_{\max}). \tag{8}$$

4.2 Application: A Depth-First Reduction

For every $m^q \in \mathcal{Q}$, we associate a prefix-tree $\mathcal{T}_{m^q}$. We now define a bijective function $\mathsf{DF} : V' \cup \{\lambda\} \to [0..d]$ mapping λ to 0 where $d = |V'| - 1$. So we can write the elements of $V' \cup \{\lambda\}$ in a sequence $v_0 = \lambda, v_1, \ldots, v_d$ where $\mathsf{DF}(v_i) = i$.

Recursive Definition of DF

---

1. Initialize $ctr = 1$, $\mathsf{DF}(\lambda) = 0$, $v_0 = \lambda$.
2. for $i = 1$ to q
3. for $j = 1$ to $\ell_i - 1$
   if $\mathsf{DF}(m_i[..j])$ is not defined then
   $\mathsf{DF}(m_i[..j]) = ctr$, $v_{ctr} = m_i[..j]$ and $ctr \leftarrow ctr + 1$.

Note that $d \leq \sigma' = \sigma - q$. When d is smaller than σ', we define $v_{d+1} = \cdots = v_{\sigma'} = v_d$. Note, we order the vertices following the depth first principle. For any i, exactly any one of the three conditions will hold:

type-1 $\mathsf{DF}(m_i[\ell_i - 1]) < h$.
type-2 $\mathsf{DF}(m_i[..j]) = h$.
type-3 $\mathsf{DF}(m_i[..j-1]) < h$ and $\mathsf{DF}(m_i[..j]) > h$.

Now we define a structure, called depth-first structure.

$$\tau_h(m^i) := (R_h(m^i), \rho_h(m^i)) = \begin{cases} (m_i, \mathsf{sim}) & \text{if type-1} \\ (m_i[..j], \mathsf{orc}) & \text{if type-2} \\ (m_i[..j], \mathsf{sim}) & \text{if type-3} \end{cases}$$

where j is defined in type-2 and type-3 as before.

τ_h IS A STRUCTURE. We first establish that for all h, τ_h is a structure. Clearly, for all i, $R_h(m^i) \preceq m_i$. Now, suppose for some $i_1 \neq i_2$, $R_h(m^{i_1}) \prec R_h(m^{j_2})$. As m_i's are prefix-free, they cannot satisfy type-1. Let j_1, j_2 denote the values of j for i_1, i_2 respectively. As R values are distinct, both cannot satisfy type-2. Moreover, DF values for $m_{i_1}[..j_1 - 1]$ and $m_{i_1}[..j_1 - 1]$ are less than h. But, $m_{i_1}[..j_1] \preceq m_{i_2}[..j_2 - 1]$. This gives a contradiction.

Clearly, from the definition of ρ_h, the second condition is satisfied. Note that for type-2 and type-3, $m_i[..j]$ is an intermediate node and hence the third condition is also satisfied.

Now we show that $\text{next}(\tau_h) = R_{h+1}$ for all $h < \sigma'$. Suppose $\mathsf{DF}(x) = h$ and $v_1, \ldots, v_r$ be its all children nodes prefixes of $m_{i_1}, \ldots, m_{i_r}$ respectively. Let i_1 be the smallest index and hence $v_1 := m_{i_1}[..\|x\| + 1]$ has DF value $h + 1$ and all other nodes has higher DF values. Thus, $R_{h+1}(m^{i_j}) = v_j$ for all j. For all other i, $R_h(m^i) = R_{h+1}(m^i)$. Thus, the set of outputs of R_{h+1} values is exactly the set $U \setminus \{x\} \cup \mathsf{ch}(x)$ where U is the set of outputs of R_h. This proves that $\text{next}(\tau_h) = R_{h+1}$. So, following our generalized reduction lemma, the following theorem for multiuser cascade construction is established.

Theorem 1. *Let $\theta = (u, q, q_{\max}, \ell, \sigma)$. Every (θ, T)-distinguisher D can be reduced to a $(q_{\max}, T' := T + O(\sigma))$-distinguisher $\mathsf{D}' := \mathsf{D}^{\mathsf{Sim}}$ (where Sim is defined as above) so that*

$$\mathbf{Adv}^{\text{mu\_pf\_prf}}_{f^{*\otimes}}(\mathsf{D}) \leq (\sigma - q) \cdot \mathbf{Adv}^{\text{prf}}_{f}(\mathsf{D}'). \tag{9}$$

Hence, $\mathbf{Adv}^{\text{mu\_pf\_prf}}_{f^{*\otimes}}(\theta, T) \leq (\sigma - q) \cdot \mathbf{Adv}^{\text{prf}}_{f}(q_{\max}, T')$. Moreover, D' is non-adaptive whenever D is non-adaptive.

Remark 1. We apply padding whenever the message space is not prefix-free, e.g. $\{0,1\}^*$. Let $\mathsf{pad} : \{0,1\}^* \to \mathsf{B}^+$ be defined as follows. Given $x \in \{0,1\}^*$ we find smallest non-negative integer d such that $|x| + 1 + d$ is a multiple of $b - 1$. Let $x\|10^d = (x_1, \ldots, x_\ell) \in (\{0,1\}^{b-1})^\ell$. Finally, we define $\mathsf{pad}(x) = (x_1\|0, \ldots, x_{\ell-1}\|0, x_\ell\|1)$. Clearly, for any $x \neq x'$, $\mathsf{pad}(x)$ is not a prefix of $\mathsf{pad}(x')$. So, $f^*_K \circ \mathsf{pad}$ is PRF with same security bound as shown before for arbitrary message space.

4.3 Applications: Simple Proofs for MDP and Boosted MD

Let $\alpha : \mathsf{B}' \times \mathcal{C} \to \mathsf{B} \times \mathcal{C}$ be a function. So, $g := f \circ \alpha : \mathsf{B}' \times \mathcal{C} \to \mathcal{C}$. Let pad be a prefix-free padding rule mapping to B'^+. Then, $g^* \circ \mathsf{pad}$ is PRF secure whenever g_K is PRF (with the bound mentioned in Theorem 1). Now we discuss two examples of α which has been used to design some constructions.

Example 1 (MDP PRF). π be a permutation on $\mathcal{C}$ and $\mathsf{B}' = \mathsf{B} \times \{0,1\}$. Now, we define $\alpha((m,0),K) = (m,K)$ and $\alpha((m,1),K) = (m,\pi(K))$. In JoC 2012 [17] authors defined a special case of related key security of f which is essentially same as the PRF security of $f \circ \alpha$. Authors of [17] provided PRF security of

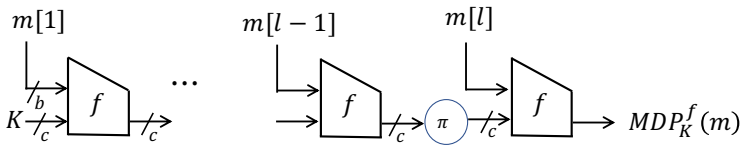

Fig. 3: MDP Keyed PRF.

$g^* \circ \mathsf{pad}$ based on related key PRF security of f (equivalently PRF security of g) with tightness gap $\ell \cdot q$. Clearly this claim is an immediate corollary of existing prefix-free PRF security of cascade. Moreover, using our reduction of this section, we have σ as tightness gap of the MDP construction.

Example 2 (Boosted MD or BMD). In Asiacrypt 2007 [34], author proposed Boosted-MD to provide much faster absorption of message in cascade construction. In particular, we additionally xor c-bit message with chaining values starting from the key (see Fig 4). Let $\mathsf{B}' = \mathsf{B} \times \{0,1\}^c$ and $\alpha(m_1, m_2, K) = (m_1, K \oplus m_2)$. In [34], author considered another variant of related key security of f which is once again same as the PRF security of $g := f \circ \alpha$. Moreover, we have σ as tightness gap of the MDP construction in contrast to original tightness gap ℓq^2.

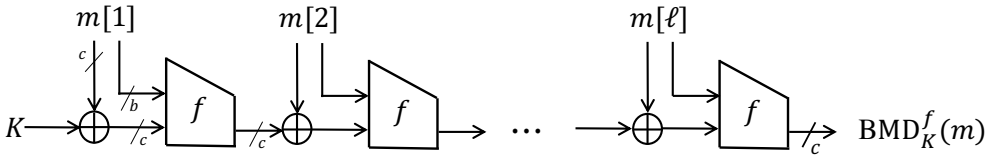

Fig. 4: BMD or Boosted-MD keyed function.

5 Improved Bound for Non-Adaptive Distinguisher

In the last subsection, we have shown a multiplicative gap of σ for the cascade, which is definitely an improvement over the previously known bound of ℓq. Now we show that we can improve the query complexity for the simulator in case of the non-adaptive bound. In Theorem 1, we reduce to a $q_{\max}$-query algorithm. However, it is easy to see that the number of queries can be much less. More precisely, except about q choices, all other hybrid reduction indices reduce to a single-query algorithm. To conclude this, we need the following simple result on a rooted tree.

Lemma 4. *Let $V(T)$ and $L(T)$ denote the set of nodes and the set of leaf nodes of a rooted tree T respectively and $V' = V \setminus L$. Then, $\sum_{v\in V'}(|\mathsf{ch}(v)| - 1) = |L(T)| - 1$. Hence, we have*

$$(1)\ \ |\{v : |\mathsf{ch}(v)| > 1\}| \leq |L(T)| - 1, \qquad (2)\ \ |\{v : |\mathsf{ch}(v)| > i\}| < |L(T)|/i.$$

Proof. The two equations (1) and (2) directly follow from the first part. We prove the first part. Note that for every rooted tree, the sum of the number of all children is $|V(T)| - 1$ (all nodes except the root node are children). As leaf nodes do not have any children it is equivalent to summing over all vertices from V'. So

$$\begin{aligned}\sum_{v\in V'}(|\mathsf{ch}(v)| - 1) &= \sum_{v\in V}|\mathsf{ch}(v)| - |V'| \\ &= |V| - 1 - |V'| = |L| - 1.\end{aligned}$$

As all terms in the L.H.S. are non-negative, we have $|\{v : |\mathsf{ch}(v)| > i\}|\times i \leq |L|-1$ and hence the second part follows. □

Let $\theta = (u, q, q_{\max}, \ell, \sigma)$. In the previous section, we have proved that Sim is a σ-step hybrid reduction from $(f^{*\otimes}, \mathsf{RF}^{\otimes})$ to (f, RF) with hybrid index space $[\sigma]$. Now, for any $m^q \in \mathcal{Q}(\theta)$, we define d as the number of intermediate nodes of the prefix tree and so $d \leq \sigma - q$. Now, for every $h \in [d]$, we associate a unique node v_h. Moreover, $\mathsf{Sim}(h)$ makes $|\mathsf{ch}(v_h)|$ (the number of children) many queries, whenever v_h is defined, otherwise it does not make any query. Note that the number of children (and hence queries) depends on m^q. Let $\mathcal{N}_i$ denote the set of h such that the number of children of v_h is i. As $\mathsf{Sim}(h)$ is a hybrid reduction, we have

$$\begin{aligned}\Pr(\mathsf{D}^{f^{*\otimes}} = 1) - \Pr(\mathsf{D}^{\mathsf{RF}^{\otimes}} = 1) &= \sum_{h\in[\sigma]}(\Pr(\mathsf{D}^{\mathsf{Sim}^f(h)} = 1) - \Pr(\mathsf{D}^{\mathsf{Sim}^{\mathsf{RF}}(h)} = 1)) \\ &= \sum_{i\geq 1}\sum_{h\in\mathcal{N}_i}\Pr(\mathsf{D}^{\mathsf{Sim}^f(h)} = 1) - \Pr(\mathsf{D}^{\mathsf{Sim}^{\mathsf{RF}}(h)} = 1)\end{aligned} \tag{10}$$

Note that the inner sum $\sum_{h\in\mathcal{N}_i}\Pr(\mathsf{D}^{\mathsf{Sim}^f(h)} = 1) - \Pr(\mathsf{D}^{\mathsf{Sim}^{\mathsf{RF}}(h)} = 1)$ the signed advantage for the simulator which makes exactly i queries. Note that the size of $\mathcal{N}_i$ is a random variable depending on m^q. However, we know that for all $i \geq 1$, $\sum_{j>i}|\mathcal{N}_j| \leq q/i$ (due to the above lemma on the tree). Using this bound we can prove our improved reduction for non-adaptive distinguisher. The first one is simple trade-off between the query complexity and tightness gap whereas the second one involves more terms in the trade-off.

Theorem 2 (First Improved Non-adaptive Reduction).

$$\mathbf{Adv}^{\text{mu\_pf\_nprf}}_{f^{*\otimes}}(\theta, T) \leq \sigma \cdot \mathbf{Adv}^{\text{nprf}}_{f}(1, T') + q \cdot \mathbf{Adv}^{\text{nprf}}_{f}(q_{\max}, T') \tag{11}$$

where $T' = T + O(\sigma)$.

Proof. We now define a reduction for a non-adaptive algorithm that uses the simulator $\mathsf{Sim}(h)$ defined as before for the adaptive distinguisher in the previous subsection. For all $h \leq \sigma$, let $\mathsf{q}(h, m^q)$ denote the number of queries $\mathsf{Sim}(h)$ makes to its oracle whenever m^q denotes all non-adaptive queries. It is easy to see that $\mathsf{q}(h, m^q) = |\mathsf{ch}(v_h)|$ (the number of children), whenever v_h is defined. So for all h, m^q, $\mathsf{q}(h, m^q) \leq q_{\max}$. Let

1. $\mathcal{N}_i = \{h : \mathsf{q}(h, m^q) = i\}$ and
2. $\mathcal{N}_{[i,j]} = \{h : i \leq \mathsf{q}(h, m^q) \leq j\}$.

By (1) of Lemma 4, we have $|\mathcal{N}_{[2,q_{\max}]}| \leq q$. Now we define two distinguishers D_1 and $\mathsf{D}_{q_{\max}}$ making at most 1 and $q_{\max}$ queries respectively. Let J_1 and $J_{\max}$ be two sets disjoint from $[\sigma]$ such that $I_1 = \mathcal{N}_1 \sqcup J_1$ and $I_{q_{\max}} = \mathcal{N}_{[2,q_{\max}]} \sqcup J_{q_{\max}}$ have σ and q elements respectively. We define $\mathsf{D}_i := \mathsf{D}^{\mathsf{Sim}_i}$ where $\mathsf{Sim}_i := \mathsf{Sim}(h \leftarrow\!\!\$\, I_i)$, $i = 1, q_{\max}$ elements. So by definition $\mathsf{D}_1, \mathsf{D}_{q_{\max}}$ make at most 1 and $q_{\max}$ queries respectively. Now,

$$\begin{aligned}
\Delta^*_{\mathsf{D}_1}(f\ ;\ \mathsf{RF}) &= \Pr(\mathsf{D}^{\mathsf{Sim}_1^f} = 1) - \Pr(\mathsf{D}^{\mathsf{Sim}_1^{\mathsf{RF}}} = 1) \\
&= \frac{1}{\sigma} \cdot \sum_{h \in I_1} (\Pr(\mathsf{D}^{\mathsf{Sim}^f(h)} = 1) - \Pr(\mathsf{D}^{\mathsf{Sim}^{\mathsf{RF}}(h)} = 1)) \\
&= \frac{1}{\sigma} \cdot \sum_{h \in \mathcal{N}_1} (\Pr(\mathsf{D}^{\mathsf{Sim}^f(h)} = 1) - \Pr(\mathsf{D}^{\mathsf{Sim}^{\mathsf{RF}}(h)} = 1)) \\
&= \frac{1}{\sigma} \cdot \sum_{h \in \mathcal{N}_1} \Delta^*_{\mathsf{D}}(\mathsf{Sim}^f(h); \mathsf{Sim}^{\mathsf{RF}}(h))
\end{aligned}$$

Similarly, $\Delta^*_{\mathsf{D}_{q_{\max}}}(f\ ;\ \mathsf{RF}) = \frac{1}{q} \cdot \sum_{h \in \mathcal{N}_{[2,q_{\max}]}} \Delta^*_{\mathsf{D}}(\mathsf{Sim}^f(h); \mathsf{Sim}^{\mathsf{RF}}(h))$. Now, by using Eq.10, we have

$$\sigma \cdot \Delta^*_{\mathsf{D}_1}(f\ ;\ \Gamma) + q_{\max} \cdot \Delta^*_{\mathsf{D}_{q_{\max}}}(f\ ;\ \Gamma) = \Delta^*_{\mathsf{D}}(f^{*\otimes}\ ;\ \mathsf{RF})$$

The proof follows by taking maximum over all possible D on the absolute value of the above equality. □

We can make further fine-tuned splitting the hybrid index set $[\sigma]$ to obtain a better trade-off between tightness gap and the query complexity. More precisely, let $\mathcal{N}^*_{2^i} = \{h : 2^{i-1} < \mathsf{q}(h, m^q) \leq 2^i\}$ for all $1 \leq i \leq s$ where $2^{s-1} < q_{\max} \leq 2^s$. As before we have $|\mathcal{N}^*_{2^i}| \leq q/2^{i-1}$ and we add dummy sets to $\mathcal{N}^*_{2^i}$ to obtain I_{2^i} such that $|I_{2^i}| = q/2^{i-1}$ for $i \geq 1$. Now, for a θ-distinguisher D and for every $i \geq 1$, we define 2^i-query distinguisher $\mathsf{D}_{2^i} := \mathsf{D}^{\mathsf{Sim}_{2^i}}$ where $\mathsf{Sim}_{2^i} := \mathsf{Sim}(h \leftarrow\!\!\$\, I_{2^i})$. Now following the very similar calculation given in the proof of the previous theorem we have our next improved non-adaptive reduction.

Theorem 3 (Second Improved Non-adaptive Reduction). *For any $\theta := (u, q, q_{\max}, \ell, \sigma, T)$-non-adaptive distinguisher D, we have 1-query non-adaptive distinguisher D'_1 and $(2^{i-1}+1)$-query non-adaptive distinguisher D_i for $1 \leq i \leq$*

$\lceil \log_2 q_{\max} \rceil$ *with run time* $T' = T + O(\sigma)$ *such that*

$$\mathbf{Adv}^{\text{mu\_pf\_nprf}}_{f^{*\otimes}}(\mathsf{D}) \leq \sigma \cdot \mathbf{Adv}^{\text{nprf}}_{f}(\mathsf{D}'_1) + \sum_{i=1}^{\lceil \log_2 q_{\max} \rceil} \frac{q}{2^{i-1}} \cdot \mathbf{Adv}^{\text{nprf}}_{f}(\mathsf{D}_i). \quad (12)$$

Hence,

$$\mathbf{Adv}^{\text{mu\_pf\_nprf}}_{f^{*\otimes}}(\theta) \leq \sigma \cdot \mathbf{Adv}^{\text{nprf}}_{f}(1, T') + \sum_{i=1}^{\lceil \log_2 q_{\max} \rceil} \frac{q}{2^{i-1}} \cdot \mathbf{Adv}^{\text{nprf}}_{f}(2^i, T').$$

Applications to HMAC and NMAC. The generic reduction from NMAC to cascade (see Eq. 3) and HMAC to NMAC (see Eq. **??**) can be easily extended for the multiuser set-up in the following way:

$$\mathbf{Adv}^{\text{mu\_prf}}_{\mathsf{NMAC}^f}(\theta, T) \leq \mathbf{Adv}^{\text{mu\_pf\_nprf}}_{f^*}(\theta, T') + \mathbf{Adv}^{\text{prf}}_{f}(q, T') + \frac{q^2}{2^c}, \quad (13)$$

$$\mathbf{Adv}^{\text{mu\_prf}}_{\mathsf{HMAC}^f}(\theta, T) \leq \mathbf{Adv}^{\text{mu\_prf}}_{\mathsf{NMAC}^f}(\theta, T) + \mathbf{Adv}^{\text{prbg}}_{\mathsf{KDF}}(T'). \quad (14)$$

Hence our results for non-adaptive PRF security of cascade can be directly applied to the multiuser security of HMAC and NMAC.

5.1 Significance of Improvement in the standard model

The known tightness gap ℓq becomes worse when the queries can be both very short as well as large. In that case we can limit $\ell, q, \sigma \leq D$ where D represents the maximum data complexity.[3] With this limit, the known bound for cascade turns out to be

$$\begin{aligned}\mathbf{Adv}^{\text{prf}}_{f^*}(q, \ell, \sigma, T) &\leq \ell q \cdot \mathbf{Adv}^{\text{prf}}_{f}(q, T') \\ &\leq D^2 \cdot \mathbf{Adv}^{\text{prf}}_{f}(q, T')\end{aligned}$$

where $T' = T + O(\sigma)$. In [8,20], authors showed that for almost all designs we can have $\mathbf{Adv}^{\text{prf}}_{f}(q, T') \geq 2^{-c/2}$. Moreover, by key-guessing attack, we also have $\mathbf{Adv}^{\text{prf}}_{f}(q, T') \geq T'/2^c$ (as f^* uses c-bit key). So the known bound can only ensure security as long as

$$D \leq 2^{c/4}, \quad D^2 T \leq 2^c.$$

Using our new bound we have $\mathbf{Adv}^{\text{prf}}_{f^*}(q, \ell, \sigma, T) \leq D \cdot \mathbf{Adv}^{\text{prf}}_{f}(q, T')$ and so our bound can ensure security as long as

$$D \leq 2^{c/2}, \quad DT \leq 2^c.$$

Thus, it improves the data-time trade-off for the cascade construction. A similar improvement works for multiuser setup.

[3] NIST actually considered this in the call for the standardization process of lightweight cipher [25].

5.2 Significance of Improvement in the ideal model

We have defined PRF security in the ideal model. Now we extend the definition in the ideal model. Let Γ be a random function and F_K be a construction which uses Γ as an oracle. A distinguisher D is an oracle algorithm which has access of two oracles. We define PRF distinguishing advantage of D against F as

$$\mathbf{Adv}_F^{\mathrm{prf}}(\mathsf{D}) := |\Pr(\mathsf{D}^{F_K,\Gamma} \to 1) - \Pr(\mathsf{D}^{\mathsf{RF},\Gamma} \to 1|)|$$

where RF is a compatible random function independent with Γ. Complexity parameter of D can be written as (θ, η) where θ and η represent the complexity parameter for all construction queries (i.e., the first oracle which is either F_K or RF) and primitive queries (which is always Γ) respectively. η mostly represents the number of queries to Γ.

Convention. Note that we assume that $\eta = O(T)$ where T is the run time of D (since otherwise, D can make additional queries which increases run time by $O(T)$).

A simple example is the cascade based on a random function (or idealized compression function) $\Gamma : \{0,1\}^{b+c} \to \{0,1\}^c$ as $\Gamma_K(x) := \Gamma(K, x)$ where $K \in \{0,1\}^c$ and $\Gamma_K : \{0,1\}^b \to \{0,1\}^c$ be a keyed function based on an ideal random function Γ. The reduction proved in the paper and in the previous papers for the cascade construction Γ_K^* can be translated to the following relations for the complexity parameters $\theta = (u, q, q_{\max}, \ell, \sigma)$, $\theta_1 = (q, \ell, \sigma)$ and primitive query complexity η:

$$\mathbf{Adv}_{\Gamma^*}^{\mathrm{prf}}(\theta_1, \eta) \le \ell q \cdot \mathbf{Adv}_{\Gamma}^{\mathrm{prf}}(q, \eta + \sigma).$$

Note that reduction algorithm needs to call at most $\eta+\sigma$ many primitive queries to simulate all queries of the distinguisher. It is easy to show that for any (q, η')-query distinguisher D, $\mathbf{Adv}_{\Gamma}^{\mathrm{prf}}(\mathsf{D}) \le \eta'/2^c$ (until we cannot guess the key in the primitive query, all construction oracle queries behave like an independent random function) and the bound is tight. So plugging the bound above we can ensure $\mathbf{Adv}_{\Gamma^*}^{\mathrm{prf}}(\theta_1, \eta) \le \ell q \cdot (\eta + \sigma)/2^c$. The same bounds hold for non-adaptive PRF advantage. Similar bound applies when Γ is replaced by Davis-Meyer compression function based on an c-bit ideal cipher E with key space $\{0,1\}^b$. In particular we define $\mathsf{DM}(K, x) := E_x(K) \oplus K$.

Our adaptive PRF advantage shows that

$$\mathbf{Adv}_{\Gamma^*}^{\text{mu\_prf}}(\theta, \eta) \le \sigma \cdot \mathbf{Adv}_{\Gamma}^{\text{mu\_prf}}(q_{\max}, \eta + \sigma)$$

and hence $\mathbf{Adv}_{\Gamma^*}^{\text{mu\_prf}}(\theta, \eta) \le \sigma(\eta + \sigma)/2^c$. The same relation holds for non-adaptive PRF advantage and also for the Davis-Meyer compression function based cascade construction.

Example 3 (Boosted MD in the ideal model). We define $\Gamma_K^{\oplus}(x_1, x_2) = \Gamma(K \oplus x_1, x_2)$, $(x_1, x_2) \in \{0,1\}^{c+b}$. If a distinguisher cannot make guesses of $K \oplus x_1$ (of construction queries) in any primitive queries, then it cannot distinguish $\Gamma^{\oplus}$ from an independent random function RF. Hence, $\mathbf{Adv}_{\Gamma^{\oplus}}^{\mathrm{prf}}(q, \eta) \le q\eta/2^c$. In fact,

one can construct a distinguisher making q construction queries and η primitive queries with PRF advantage about $\eta q/2^c$ in an ideal random function model. The cascade based on $\mathsf{\Gamma}^{\oplus}$ is called boosted MD which has been studied in Sect.4 in the standard model. A straightforward application of the PRF advantage of $\mathsf{\Gamma}^{\oplus}$ in the ideal model to the cascade construction gives $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{\Gamma}^{\oplus *}}(\theta_1, \eta) \leq \ell q^2(\eta+\sigma)/2^c$. However, our bound provides an improved bound of the form:

$$\mathbf{Adv}^{\mathrm{mu\_nprf}}_{\mathsf{\Gamma}^{\oplus *}}(\theta, \eta) \leq (\sigma + q\lceil \log_2 q_{\max} \rceil)(\eta + \sigma)/2^c.$$

Hence we have shown birthday bound for boosted cascade function in a modular way instead of cubic bound derived from the existing reduction.

Let us now consider some popular examples of compression functions based on a c-bit ideal cipher E with keyspace $\{0,1\}^b$.

Example 4. MMO (Matyas-Meyer-Oseas) compression function based on an ideal cipher E can be defined as $\mathsf{MMO}(K, x) = E_{K\|0^{b-c}}(x) \oplus x$. Note that it can be distinguished from random function with an advantage about $q^2/2^c + \eta/2^c$ where q and η denote the number of construction and primitive queries respectively. The birthday terms arise due to the following reason: In the real construction there cannot be any collision among the values $z_i \oplus x_i$ where z_i's are the outputs of the queries x_i. However, a collision is observed for a random function RF with probability about $q^2/2^{c+1}$ Similarly, we also have key guessing attack for this compression function. Hence, we have $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{MMO}}(q, \eta) \geq \max\{q^2/2^{c+1}, \eta/2^c\}$. Now if we plug in the existing bound we have a cubic bound:

$$\mathbf{Adv}^{\mathrm{nprf}}_{\mathsf{MMO}^*}(\theta, \eta) \leq \frac{\ell q^3}{2^c} + \frac{\ell q \eta}{2^c}.$$

However, if we use our second improved non-adaptive reduction and simplify the sum we have quadratic bound (up to a log factor)

$$\mathbf{Adv}^{\mathrm{nprf}}_{\mathsf{MMO}^*}(\theta, \eta) \leq \lceil \log_2 q_{\max} \rceil \sigma(\eta + \sigma)/2^c + 2qq_{\max}/2^c.$$

Example 5. The similar result like MMO compression function is also applicable for Miyaguchi-Preneel compression function: $\mathsf{MP}(K, x) = E_{K\|0^{b-c}}(x) \oplus x \oplus K$. Once again, we have $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{\Gamma}^{\oplus}}(q, \eta) \geq \max\{q^2/2^{c+1}, \eta/2^c\}$ the same bound as MMO compression function.

6 Single Keyed NMAC, Constant-free HMAC and EvMAC

6.1 PRF Security of Single Keyed Composition

For every key K in a key space, let $g_K : \mathsf{B} \to \mathcal{K}$ and $F : \mathcal{K} \times \mathcal{X} \to \mathsf{B}$. We now define a keyed function $G_K : \mathsf{B} \times \mathcal{X} \to \mathcal{K}$ by combining g and F as follows:

$$G_K(a, x) := g_K\big(F(g_K(a), x)\big).$$

Here, we recall the notation g_K to denote the function $g(K,\cdot)$. Let $\mathcal{I}$ be a user index space for G. For every $\gamma \in \mathcal{I}$, we sample $K_\gamma \leftarrow\!\!\$\, \mathcal{K}$. So, $G^{\otimes}(\gamma,(a,x)) := G_{K_\gamma}(a,x)$. We denote this oracle $\mathcal{G}_0$ (see Fig.5). Now we define an intermediate oracle $\mathcal{G}_1$. We obtain the oracle $\mathcal{G}_1$ replacing g by a (multiuser) random function RF. More precisely, on a user index γ and an input (a,x), it returns

$$\mathcal{G}_1(\gamma,(a,x)) := \mathsf{RF}_\gamma\big(F(\mathsf{RF}_\gamma(a),x)\big)$$

Now, for every θ-complexity distinguisher D,

$$\Delta_{\mathsf{D}}(\mathcal{G}_0,\mathcal{G}_1) = \Delta_{\mathsf{D}_0}(g^{\otimes\mathcal{I}},\mathsf{RF}^{\otimes\mathcal{I}})$$

by using the substitution reduction where D_0 is a $(u,2q,2q_{\max})$-complexity distinguisher (it simply simulates the construction G using its oracle replacing the function g in G).

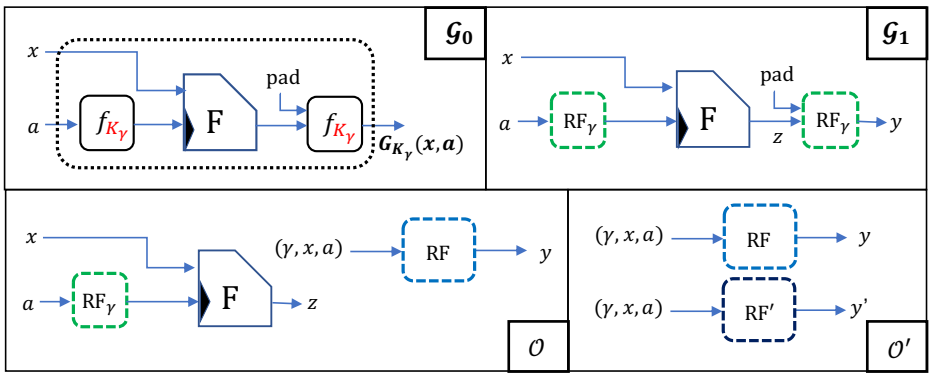

Fig. 5: Games $\mathcal{G}_0$ and $\mathcal{G}_4$ represent the real and ideal world respectively. The games $\mathcal{G}_1,\mathcal{G}_2,\mathcal{G}_3$ are intermediate oracles $\mathcal{G}_1$ is obtained by replacing f by a random function. Game $\mathcal{G}_2$ makes two executions of random functions independent. $\mathcal{G}_3$ replaces F by a random function.

Now we bound the distance between the oracle $\mathcal{G}_1$ and a random function $\mathcal{G}_2$. Let $\mathcal{O}^{\mathrm{pp}}(\gamma,a,x) = F(\mathsf{RF}(\gamma,a),x)$ be a post-processing oracle for both $\mathcal{G}_1$ and $\mathcal{G}_2$. In case of $\mathcal{G}_2$, the random function RF is independent with $\mathcal{G}_2$. Let D be any $(u,q,q_{\max})$-complexity adaptive distinguisher interacting with either $\mathcal{G}_1$ or $\mathcal{G}_2$ and a post-processing oracle which returns responses of all queries made by D after the query-response phase is over. Let $(\gamma_1,(a_1,x_1)),\ldots,(\gamma_q,(a_q,x_q)) \in \mathcal{I}\times\mathsf{B}\times\mathcal{X}$ be all q queries and $y_1,\ldots,y_q \in \mathcal{K}$ be the corresponding responses. Let $z_1,\ldots,z_q \in \mathsf{B}$ be the responses of post-processing oracle, i.e. $F(\mathsf{RF}(\gamma_i,a_i),x_i) = z_i$. We say that bad holds if

1. either $z_i = z_j$ for some $i \neq j$ or
2. $(\gamma_i,z_i) = (\gamma_j,a_j)$ for some i,j.

So, the event bad satisfies conditions of Lemma 2. Hence there is a non-adaptive distinguisher D_1 with same complexity parameter as D_1 such that

$$\Pr(\mathsf{bad}\ \textit{holds in}\ \mathsf{D}^{\mathcal{G}_1,\mathcal{O}_{\mathrm{pp}}}) \le \Pr(\mathsf{bad}\ \textit{holds in}\ \mathsf{D}_1^{\mathcal{G}_2,\mathcal{O}_{\mathrm{pp}}}).$$

Now for any good transcript $(\gamma^q,(a^q,x^q),y^q,z^q)$, it is easy to see that the both worlds realize the transcript with probability

$$\frac{\Pr(F(\mathsf{RF}(\gamma_i,a_i),x_i)=z_i \forall i)}{|\mathcal{K}'|^q}$$

(as the z_i values are all distinct and different from the other inputs of RF). Now by identical until bad lemma, $\Delta_{\mathsf{D}}(\mathcal{G}_1;\mathcal{G}_2) \le \Pr(\tau(\mathsf{D}^{\mathcal{G}_2}) \text{ is } \mathsf{bad})$. So, we have proved

$$\begin{aligned}\Delta_{\mathsf{D}}(\mathcal{G}_0;\mathcal{G}_2) &\le \Delta_{\mathsf{D}}(\mathcal{G}_0;\mathcal{G}_1)+\Delta_{\mathsf{D}}(\mathcal{G}_1;\mathcal{G}_2)\\ &\le \Delta_{\mathsf{D}_0}(g^{\otimes\mathcal{I}},\mathsf{RF}^{\otimes\mathcal{I}})+\Pr(\tau(\mathsf{D}_1^{\mathcal{G}_2,\mathcal{O}_{\mathrm{pp}}}) \text{ is } \mathsf{bad})\end{aligned}$$

Note that the bad does not depend on the response of the oracle $\mathcal{G}_2$. So we can define a bad event $\mathsf{bad}(\mathsf{D}'^{\mathcal{O}})$ for a non-adaptive interaction between a non-adaptive distinguisher D_1 and a $\mathcal{I}$-folded $(\mathsf{B}\times\mathcal{X},\mathsf{B})$-oracle $\mathcal{O}$ whenever

1. either $z_i = z_j$ for some $i \ne j$ or
2. $(\gamma_i,z_i)=(\gamma_j,a_j)$ for some i,j.

hold true where $(\gamma_1,(a_1,x_1)),\ldots,(\gamma_q,(a_q,x_q)) \in \mathcal{I}\times\mathsf{B}\times\mathcal{X}$ be all q non-adaptive queries and $z_1,\ldots,z_q \in \mathsf{B}$ be the corresponding responses. Thus, we have our single-keyed composition theorem.

Theorem 4 (single-keyed composition). *Let g, F and G as defined above. Then, for any $(u,q,q_{\max},\ell,\sigma)$-complexity distinguisher D,*
(i) there is a $(u,2q,2q_{\max})$-complexity distinguisher D_0, and
(ii) $(q,q,q_{\max},\ell,\sigma)$-complexity non-adaptive adversary D_1 such that

$$\Delta_{\mathsf{D}}(G^{\otimes\mathcal{I}},\mathsf{RF}''^{\otimes\mathcal{I}}) \le \Delta_{\mathsf{D}_0}(g^{\otimes\mathcal{I}},\mathsf{RF}^{\otimes\mathcal{I}})+\Pr(\mathsf{bad}(\mathsf{D}_1^{\mathcal{O}})) \tag{15}$$

where $\mathcal{O}(\gamma,a,x)=F(\mathsf{RF}(\gamma,a),x)$.

6.2 Application: Security of Single-Keyed NMAC

Now we show that the single keyed $\mathsf{1k\_NMAC}_K := \mathsf{NMAC}_{K,K}$ based on $f : \{0,1\}^c \times \mathsf{B} \to \{0,1\}^c$ has almost the same security as NMAC_{K_1,K_2}. Note that in Theorem 4 we can consider $g=f$ and $F(K,m)=f_K^*(m)\|0^{b-c}$. Then, the function G is same as $\mathsf{1k\_NMAC}_K$. By our single keyed composition we need to bound the probability of bad event for $\mathsf{D}_1^{F^\otimes}$.

We note that the queries may not be prefix-free and so we cannot replace f^* by a random function. However, we consider another bad event bad' as follows. Let x be a block such that (γ_i,m_i,x)'s are prefix-free. Clearly such a block x exists. Now, we say that bad' holds if

1. either $f(z_i, x) = f(z_j, x)$ for some $i \neq j$ or
2. $(\gamma_i, f(z_i, x)) = (\gamma_j, f(a_j, x))$ for some i, j.

So, $\mathsf{bad} \Rightarrow \mathsf{bad}'$ and hence $\Pr(\mathsf{bad}) \leq \Pr(\mathsf{bad}')$. Now, bad' is actually bad event for queries (γ_i, m_i, x) which are prefix-free. Thus,

$$\begin{aligned}\Pr(\mathsf{bad}'(\mathsf{D}_1^{F^{\otimes}})) &\leq \Pr(\mathsf{bad}'(\mathsf{D}_1^{\mathsf{RF}^{\otimes}})) + \mathbf{Adv}_F^{\text{mu\_nprf}}(q, q, q_{\max}, \ell + 1, \sigma + q) \\ &\leq 3q^2/2^{c+1} + \mathbf{Adv}_F(u, q, q_{\max}, \ell + 1, \sigma + q)\end{aligned}$$

By using randomness of RF, bad' holds with probability at most $3q^2/2^{c+1}$. So, we have proved our PRF analysis for single keyed NMAC.

Theorem 5 (single-keyed NMAC). *For $T' = T + O(\sigma)$,*

$$\begin{aligned}\mathbf{Adv}_{\mathit{1k\_NMAC}^f}^{\text{mu\_prf}}(\theta) \leq \mathbf{Adv}_{f^*}^{\text{mu\_pf\_nprf}}(q, q, q_{\max}, \ell + 1, \sigma + q, T') + \\ + u\mathbf{Adv}_f^{\text{prf}}(2q_{\max}, T') + 1.5q^2/2^c.\end{aligned}$$

The previous result can be plugged into the above expression to get the security of the constant-free variant of HMAC, denoted as HMAC', where

$$\mathsf{HMAC}'(K, m) := \mathsf{1k\_NMAC}(\mathsf{KDF}(K), m)$$

and $\mathsf{KDF}(K) = f(IV, K\|0^*)$. If $f(IV, \cdot, 0^*)$ is (almost) regular then $\mathsf{KDF}(K)$ is uniformly distributed and hence the security of the variant of HMAC' is reduced to $\mathsf{1k\_NMAC}$. So the bound for the single-keyed NMAC can be directly applied to the constant-free variant of HMAC.

6.3 Application: Security Analysis of Enveloped MAC

We now similarly prove an improved analysis for Enveloped MAC (we get a better tightness reduction as well as eliminate the related key advantage in the existing analysis). For any keyed function g, we write

$$\mathsf{EvMAC}_*^g(m) := g\big(f^*(g(IV), m[1..])\big).$$

Note that $\mathsf{EvMAC}_*^{f_K^{\downarrow}}$ is the same as EvMAC. We write $\mathsf{EvMAC}' = \mathsf{EvMAC}_*^{\Gamma}$. By using a reduction similar to $\mathsf{1k\_NMAC}$, we have

$$\Delta_{\mathsf{D}}^*(\mathsf{EvMAC}^{\otimes}\ ;\ \mathsf{RF}) = \Delta_{\mathsf{D}}^*(\mathsf{EvMAC}'^{\otimes}\ ;\ \mathsf{RF}) + \Delta_{\mathsf{D}_0}^*(f^{\downarrow\otimes}\ ;\ \Gamma),$$

where $\mathsf{D}_0^{\mathcal{O}^{\otimes}} = \mathsf{D} \triangleright \mathsf{EvMAC}_*^{\mathcal{O}^{\otimes}}$. So we focus on bounding the multiuser PRF advantage of $\Gamma\big(f^*(\Gamma(IV), m[1..])\big)$. Now we see that if we fix $m_i[1] = IV$ in the proof of NMAC we actually reduce to EvMAC'. In other words, we need to consider the following bad event (for some x to be adjoined at the end as before to make prefix-free queries): Let $\mathsf{bad}(m^q[1], z^q) = 1$ if either $z_i = z_j$ for some $i \neq j$, or $f(z_i, x) = f(IV, x)$ for some i, j. Following a similar argument as $\mathsf{1k\_NMAC}$, we have our result.

Theorem 6 (Envelope MAC).

$$\mathbf{Adv}^{\text{mu\_prf}}_{\mathsf{EvMAC}}(\theta) \leq \mathbf{Adv}^{\text{mu\_pf\_nprf}}_{f^*}(q, q, q_{\max}, \ell, \sigma, \sigma_{\max}, T') + \\ + u' \cdot \mathbf{Adv}^{\text{prf}}_{f}(q_{\max}+1, T') + \frac{q'^2}{2^c}.$$

7 Final Remarks

Let $\epsilon(D,T) := \mathbf{Adv}^{\text{prf}}_{f}(D,T)$. The dominating terms for the best known bound for prefix-free PRF advantage against cascade, NMAC and HMAC before this paper was $\ell q \cdot \epsilon(q, T+\sigma)$. Let D denote the user limit of an application before it re-keys. Some applications can accept a wide range of message sizes (from a single block to very large messages), and hence we must assume $\ell = q = \sigma = O(D)$ (similar guidelines are provided by NIST for lightweight cipher standardization [33]). So the best known bound in this set-up becomes $D^2 \cdot \epsilon(D, T+O(D))$. We already know that $\epsilon(D,T) \geq \max\{T/2^c, 1/2^{c/2}\}$. Both lower bounds weaken the PRF security guarantee of the cascade (and hence of HMAC and NMAC). Our result resolves these issues.

7.1 Open Problems

The following important open problems can be studied in the future.

1. Similar to the reduction for non-adaptive distinguishers, can we have an improved trade-off for adaptive reductions? This problem seems to be challenging as the simulator needs to guess a bound on the number of children of a node adaptively.
2. Having an improved reduction for adaptive PRF distinguisher of AMAC is not yet solved. A different approach to handle prefix queries is needed.
3. All known bounds for cascade construction can be at the best $DT/2^c$. However, there is no such generic matching attack (matching attacks are applicable for some pathological examples). Understanding the right PRF security bound when f behaves like a random function is not yet known.

References

1. Mihir Bellare. New proofs for NMAC and HMAC: security without collision-resistance. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, volume 4117 of *Lecture Notes in Computer Science*, pages 602–619. Springer, 2006. Updated version available at `http://cseweb.ucsd.edu/~mihir/papers/hmac-new.pdf`.
2. Mihir Bellare. New proofs for nmac and hmac: security without collision resistance. *Journal of Cryptology*, 28(4):844–878, 2015.

3. Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 1996. Extended version available at `http://cseweb.ucsd.edu/~mihir/papers/kmd5.pdf`.
4. Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Pseudorandom functions revisited: The cascade construction and its concrete security. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 514–523. IEEE Computer Society, 1996. Extended version available at `http://cseweb.ucsd.edu/~mihir/papers/cascade.pdf`.
5. Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of cipher block chaining. In Yvo Desmedt, editor, *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 341–358. Springer, 1994.
6. Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of the cipher block chaining message authentication code. *J. Comput. Syst. Sci.*, 61(3):362–399, 2000.
7. Mihir Bellare and Phillip Rogaway. The game-playing technique. *International Association for Cryptographic Research (IACR) ePrint Archive: Report*, 331:2004, 2004.
8. Daniel J Bernstein and Tanja Lange. Non-uniform cracks in the concrete: the power of free precomputation. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 321–340. Springer, 2013.
9. J. Black and P. Rogaway. A block-cipher mode of operations for parallelizable message authentication. In *Advances in Cryptology – Eurocrypt 2002*, number 2332 in Lecture Notes in Computer Science, pages 384–397, Berlin, 2002. Springer.
10. Bill Burr. Nist hash function standards status and plans. *National Institute of Standards and Technology, Gaithersburg, Maryland (csrc. nist. gov/groups/SMA/ispab/documents/minutes/2005-12/B_Burr-Dec2005-ISPAB. pdf)*, 2005.
11. Ivan Bjerre Damgård. A design principle for hash functions. In *Conference on the Theory and Application of Cryptology*, pages 416–427. Springer, 1989.
12. Morris Dworkin. Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. `http://csrc.nist.gov/publications/nistpubs/index.html$#$sp800-38B`.
13. PUB FIPS. 198-1. *The keyed-hash message authentication code (HMAC)*, 2008.
14. Peter Gaži, Krzysztof Pietrzak, and Michal Rybár. The exact prf-security of nmac and hmac. In *Annual Cryptology Conference*, pages 113–130. Springer, 2014.
15. Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *25th Annual Symposium on Foundations of Computer Science, West Palm Beach, Florida, USA, 24-26 October 1984*, pages 464–479. IEEE Computer Society, 1984.
16. Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the Association for Computing Machinery*, 33(4):792–807, 1986.
17. Shoichi Hirose, Je Hong Park, and Aaram Yun. A simple variant of the merkle-damgård scheme with a permutation. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 113–129. Springer, 2007.
18. T. Iwata and K. Kurosawa. Omac: One-key cbc mac. In *Fast Software Encryption, 10th International Workshop – FSE 2003*, number 2887 in Lecture Notes in Computer Science, pages 129–153, Berlin, 2003. Springer.

19. B. Kaliski and M. Robshaw. Message authentication with md. *CryptoBytes, 5*, 1(1):5–8, 1995.
20. Neal Koblitz and Alfred Menezes. Another look at HMAC. *J. Math. Cryptol.*, 7(3):225–251, 2013.
21. Neal Koblitz and Alfred Menezes. Another look at non-uniformity. *Groups Complex. Cryptol.*, 5(2):117–139, 2013.
22. Neal Koblitz and Alfred Menezes. Another look at security theorems for 1-key nested macs. In *Open Problems in Mathematics and Computational Science*, pages 69–89. Springer, 2014.
23. Hugo Krawczyk, Mihir Bellare, and Ran Canetti. HMAC: keyed-hashing for message authentication. *RFC*, 2104:1–11, 1997.
24. Ueli Maurer. Indistinguishability of random systems. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 110–132. Springer, 2002.
25. Kerry McKay, Lawrence Bassham, Meltem Sönmez Turan, and Nicky Mouha. Report on lightweight cryptography. Technical report, National Institute of Standards and Technology, 2016.
26. Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.
27. Ralph C. Merkle. One way hash functions and DES. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 428–446. Springer, 1989.
28. P. Piermont and W. Simpson. Ip authentication using keyed md5. *IETF RFC 1828*, August 1995.
29. Bart Preneel and Paul C Van Oorschot. On the security of iterated message authentication codes. *IEEE Transactions on Information theory*, 45(1):188–199, 1999.
30. Phillip Rogaway. Formalizing human ignorance. In Phong Q. Nguyen, editor, *Progressin Cryptology - VIETCRYPT 2006, First International Conference on Cryptology in Vietnam, Hanoi, Vietnam, September 25-28, 2006, Revised Selected Papers*, volume 4341 of *Lecture Notes in Computer Science*, pages 211–228. Springer, 2006.
31. Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. *IACR Cryptol. ePrint Arch.*, 2004:332, 2004.
32. Gene Tsudik. Message authentication with one-way hash functions. *ACM SIGCOMM Computer Communication Review*, 22(5):29–38, 1992.
33. Meltem Sönmez Turan, Kerry A McKay, Çağdaş Çalık, Donghoon Chang, and Larry Bassham. Status report on the first round of the nist lightweight cryptography standardization process. *National Institute of Standards and Technology, Gaithersburg, MD, NIST Interagency/Internal Rep.(NISTIR)*, 2019.
34. Kan Yasuda. Boosting merkle-damgård hashing for message authentication. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 216–231. Springer, 2007.
35. Kan Yasuda. "sandwich" is indeed secure: How to authenticate a message with just one hashing. In Josef Pieprzyk, Hossein Ghodosi, and Ed Dawson, editors, *Information Security and Privacy, 12th Australasian Conference, ACISP 2007, Townsville, Australia, July 2-4, 2007, Proceedings*, volume 4586 of *Lecture Notes in Computer Science*, pages 355–369. Springer, 2007.

A Known Applications of Hybrid Reduction Lemma

A.1 Application: multiuser to Single User Reduction

Let $F^{\otimes}$ be an $\mathcal{I}$-folded keyed function from $\mathcal{X}$ to $\mathcal{Y}$ with a key space $\mathcal{K}$. We now define a u-step reduction algorithm Sim reducing $F^{\otimes}$ to F as follows:

Simulator Sim

---

1. hybrid index $h \leftarrow_{\$} [u]$, keys $K_1, \ldots, K_{h-1}, K_{h+1}, \ldots, K_u \leftarrow_{\$} \mathcal{K}$ and $ctr = 1$.
2. For $i = 1$ to q,
   (a) let ith query be (γ_i, x_i).
   (b) if $\gamma_i \notin \gamma^{i-1}$ then $rank(\gamma_i) \leftarrow ctr$ and $ctr \leftarrow ctr + 1$.
   (c) $r = rank(\gamma_i)$
   (d) response of the query is z_i where

$$z_i = \begin{cases} F_{K_r}(x_i) & \text{if } r > h \\ \leftarrow_{\$} \mathcal{Y} & \text{if } r < h \\ \mathcal{O}(x_i) & \text{if } r = h \end{cases}$$

So when $h = 1$ and $\mathcal{O} = F_K$ we have $z_i = F_{\Gamma(\gamma_i)}(x_i)$ for all $i \in [q]$, where $\Gamma(\gamma_i) = K_i$ if $i \neq 1$ and $\Gamma(\gamma_1) = K$. As K_i's are independent with K, Γ behaves as a random function and so $\mathsf{Sim}^{F_K}(1) \cong F^{\otimes}$. Similarly, $\mathsf{Sim}^{\mathsf{RF}'}(u) \cong_{(u,q,q_{\max})} \mathsf{RF}$ (we restrict all query tuples which makes at most u user queries and so value of rank does not exceed u). This shows the boundary conditions. Now we show the transition equivalence through intermediate oracles $\mathcal{O}_h$ defined below, $0 \leq h \leq u$. It is clear from the description of Sim and the intermediate oracles

Oracle $\mathcal{O}_h$

---

1. keys $K_1, \ldots, K_u \leftarrow_{\$} \mathcal{K}$ and $ctr = 1$.
2. For $i = 1$ to q,
   (a) let ith query be (γ_i, x_i).
   (b) if $\gamma_i \notin \gamma^{i-1}$ then $rank(\gamma_i) \leftarrow ctr$ and $ctr \leftarrow ctr + 1$.
   (c) $r = rank(\gamma_i)$
   (d) response of the query is z_i where

$$z_i = \begin{cases} F_{K_r}(x_i) & \text{if } r > h \\ \leftarrow_{\$} \mathcal{Y} & \text{if } r \leq h \end{cases}$$

---

that $\mathsf{Sim}^{\mathsf{RF}}(h)$ behaves identical to $\mathcal{O}_h$ and $\mathsf{Sim}^{F}(h)$ behaves identical to $\mathcal{O}_{h-1}$.

Moreover, $(u, q, q_{\max})^{\mathsf{Sim}} = q_{\max}$. Thus, by using hybrid reduction proof, we have

$$\mathbf{Adv}_F^{\text{mu\_prf}}(u, q, q_{\max}) \leq u \cdot \mathbf{Adv}_F^{\text{prf}}(q_{\max}).$$

A.2 Application: Known Reduction Proof for Cascade

All known proofs for cascade including GGM proof of PRF to PRB were done in two main steps where each step is a hybrid reduction algorithm. One of the two steps is actually multiuser to single user reduction as discussed before. Now we give details of existing reduction proof. Let us fix a single user complexity parameter $\theta = (q, \ell, \sigma)$. Now we define the reduction algorithm $\mathsf{Sim}(i)$ as follows where $i \in [\ell]$ denotes a hybrid index: On jth query m_j, $\mathsf{Sim}^{\mathsf{O}^{\otimes}}(i)$ returns

$$z_i := f^*(\mathsf{O}^{\otimes}(m_j[..i]\ ;\ m_j[i+1]), m_j[i+2..])$$

(with user index as $m_j[..i]$ in the oracle query). Here, the multiuser oracle $\mathsf{O}^{\otimes}$ has user index space $\mathsf{B}^{\leq \ell} := \cup_{i \leq \ell} \mathsf{B}^i$. For every $0 \leq i \leq \ell$, we define intermediate oracles as

$$\mathcal{O}_i(m) = f^*(\mathsf{RF}(m[..i]), m[i+1..])$$

Clearly, $\mathcal{O}_0 \cong f_K^*$ where $K = \mathsf{RF}(\lambda)$ and $\mathcal{O}_\ell \cong \mathsf{RF}$. Now, we claim that Sim is a ℓ-step reduction. We see that the response z_i can be equivalently written as described below.

1. **Case** $\mathsf{O} = f$: $z_i = f^*(f(\mathsf{RF}(m[..i]), m[i+1]), m[i+2..]) = \mathcal{O}_i(m)$.
2. **Case** $\mathsf{O} = \mathsf{RF}'$:

$$\begin{aligned} z_i &= f^*(\mathsf{RF}'^{\otimes}(m[..i], m[i+1]), m[i+2..]) \\ &\cong f^*(\mathsf{RF}(m[..i+1]), m[i+2..]) = \mathcal{O}_{i+1}(m) \end{aligned}$$

Here we use the observation that $\mathsf{RF}'^{\otimes}(m[..i], m[i+1])$ is equivalent to the oracle $\mathsf{RF}(m[..i+1])$. This proves that $\mathsf{Sim}(\cdot)$ is a ℓ-step reduction algorithm and so by hybrid reduction proof (Lemma 3)

$$\begin{aligned} \mathbf{Adv}_{f^*}^{\text{prf}}(q, \ell, \sigma, t) &\leq \ell \cdot \mathbf{Adv}_f^{\text{mu\_prf}}(u := q, q, q_{\max} := q) \\ &\leq \ell q \cdot \mathbf{Adv}_f^{\text{prf}}(q). \end{aligned}$$

Note that the above reduction algorithm turns a non-adaptive distinguisher to a non-adaptive distinguisher and so the above result also holds for non-adaptive PRF advantage.

Remark 2 (extending proof to multiuser). The above proof can be easily extended to multiuser setup with input space $\mathsf{B}_{\text{mu}}^+ := \mathcal{I} \times \mathsf{B}^+$. The intermediate oracles $\mathcal{O}_i$ is defined as

$$\mathcal{O}_i(m) = f^*(\mathsf{RF}(m[0..i]), m[i+1..]).$$

Exactly same proof as described above holds where $\mathsf{Sim}^{\mathsf{O}^{\otimes}}(h)$ has $\mathcal{I}'$-folded $(\mathsf{B}, \mathcal{K})$-oracle $f^{\otimes}$ or $\mathsf{RF}'^{\otimes}$ where $\mathcal{I}' := \mathcal{I} \times \mathsf{B}^{\leq \ell}$ and it is a substitution reduction from $\mathcal{O}_{h-1}$ to $\mathcal{O}_h$. This would prove

$$\begin{aligned}\mathbf{Adv}^{\text{mu\_pf\_prf}}_{f^{*\otimes}}(u, q, q_{\max}, \ell, \sigma) &\leq \ell \cdot \mathbf{Adv}^{\text{mu\_prf}}_{f^{\otimes}}(q, q, q_{\max}) \\ &\leq \ell q \cdot \mathbf{Adv}^{\text{prf}}_{f}(q_{\max})\end{aligned}$$