IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2025, No. 1, pp. 70–96. DOI:10.46586/tosc.v2025.i1.70-96

Keying Merkle-Damgård at the Suffix

Bart Mennink

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

Abstract. A classical way to turn a cryptographic hash function into a MAC (message authentication code) function is by concatenating key and message and interpreting the result as a tag. For the Merkle-Damgård hash function construction, the approach to prepend the key to the message is known to be insecure, as it is vulnerable to the length extension attack. This observation eventually resulted in the introduction of the HMAC construction. The alternative approach to append the key to the message, even though it already dates back to a work of Tsudik from 1992, has never been investigated in detail. In this work, we perform an in-depth treatment on the possibilities to design a MAC function from the Merkle-Damgård hash function construction by processing the key at the suffix. We formalize two constructions: the suffix keyed Merkle-Damgård construction that simply appends key to message, and the suffix blinded Merkle-Damgård construction that blinds the state before compressing the last message, much like the suffix keyed sponge construction (SuKS). We subsequently prove that both constructions are secure in the standard model under reasonable assumptions on the underlying compression function. We finally investigate the security of these constructions in the leaky setting, and demonstrate that the suffix keyed Merkle-Damgård construction is not leakage resilient, but the suffix blinded Merkle-Damgård construction is leakage resilient as long as an appropriate padding rule is adopted and as long as the underlying building blocks are processing secret data in a leakage resilient manner.

Keywords: suffix keyed Merkle-Damgård · suffix blinded Merkle-Damgård · PRF · leakage resilience · SuKS

1 Introduction

Cryptographic hash functions, functions that map an arbitrarily long message M to a short (e.g., 160 or 256 bits) digest h, are one of the most basic and most prominent building blocks in modern cryptography. They are traditionally expected to satisfy certain security properties such as collision resistance (it should be computationally hard to find two different messages M, M' hashing to the same digest), preimage resistance (it should be computationally hard to find a message M corresponding to a given digest), and second preimage resistance (it should be computationally hard to find a message M that collides with another given message M') [RS04]. The first generic classical hash function construction dates back to the 80s and is attributed to Damgård [Dam89] and Merkle [Mer89]. Internally, it uses a *compression function* F from $n+m$ to n bits, where n denotes the state size and m the amount of message bits that one evaluation of F can absorb into the state. The Merkle-Damgård hash function construction (or, *MD*) then (i) initializes a state using an n-bit initial value IV, (ii) applies an injective padding on M so that its resulting length is a multiple of m bits, and (iii) absorbs the message into the state using the compression function F, m bits at a time. The output of the last compression function evaluation is the digest. Refer to Figure 1, where the message is padded by appending a single one and a minimal but sufficient number of zeros. Exactly

Received: 2024-09-01 Revised: 2024-11-22 Accepted: 2025-01-23 Published: 2025-03-07

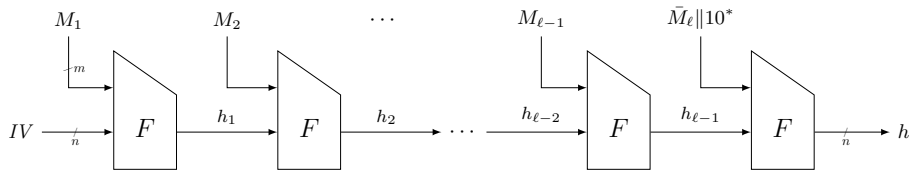

Figure 1: The Merkle-Damgård construction (MD). Here, the message M is split into message blocks $M_1, \ldots, M_{\ell-1}, \bar{M}_\ell$, where the M_is are all m bits long and $\bar{M}_\ell$ is between 0 and $m-1$ bits.

this construction, but with a slightly more involved padding rule that includes the length encoding of M (a concept known as *strengthening*), has been used in standardized hash functions such as SHA-1 [Nat15a] and the SHA-2 family [Nat15a].

Given the prevalence of hash functions back in the old days, they were used for many purposes, one of which being message authentication. The idea of a message authentication code (MAC) function is that it, for a secret key K, maps an arbitrarily long message M to a short tag h. The resemblance with hash functions is clear, and one may be tempted to simply hash key and message:

$$MAC(K, M) = MD(K\|M)\,. \tag{1}$$

Unfortunately, it was quickly acknowledged [Tsu92, KR95][1] that this is an unsafe approach, as it is vulnerable to the length extension attack. In detail, given a tag h corresponding to a message M, one can obtain a tag h' corresponding to the message $M\|10^*\|X$ for $|X| \leq m-1$ without using the key, as

$$h' = MD(K\|M\|10^*\|X) = F(h, X\|10^*)\,.$$

Apart from this construction, Tsudik [Tsu92] described two other ways to key Merkle-Damgård, namely by keying it at the suffix, as in $MD(M\|K)$, and by keying it at both the prefix and the suffix ("enveloping"), as in $MD(K\|M\|K)$. The former one was quickly discarded: Preneel and Van Oorschot [PvO95] demonstrated that this construction is vulnerable to an offline attack, where the adversary finds an inner collision in the keyless part of the Merkle-Damgård construction in around $2^{n/2}$ offline compression function evaluations. The envelope construction eventually evolved into the invention of the *HMAC* design [BCK96, KBC97, Nat08]. In a nutshell, *HMAC* operates on an inner key K_{in} and outer key K_{out}, it processes the message basically as in (1) with the inner key, and subsequently processes the result likewise as in (1) but using the outer key. Bellare et al. [BCK96] and Bellare [Bel06] proved that, security of *HMAC* follows from that of F, or in detail, that if F is a pseudorandom function (PRF), then so is *HMAC*. Yasuda proved that a similar result applies to the envelope construction, under the assumption that F is a PRF both if keyed through the chaining path as well as if keyed through the data path.

Of course, retrospectively, *HMAC* was a solution to a problem that should not have existed in the first place, the length extension attack, but this may in part have been caused by the lack of theoretical knowledge on generic hash function design. It was not until 2004 that a formal treatment of generic hash function design gave evidence as to which designs could be used as in (1) and which not. In detail, in 2004, Maurer et al. [MRH04] introduced the indifferentiability framework, which was tailored to hashing one year later by Coron et al. [CDMP05]. This model says that if a hash function construction is indifferentiable from a random oracle, it basically *behaves* like a random oracle and it can be used as such in many applications.[2] Thus, if a hash function construction H behaves like a random

[1] Tsudik [Tsu92] attributes the observation to Dave Solo and Steve Kent.
[2] This claim is restricted to single-stage games, cf., Ristenpart et al. [RSS11].

oracle, one can among others safely evaluate $H(K\|M)$. This would for instance work for variants of Merkle-Damgård that chop the final output [CDMP05], including some instances of the SHA-2 family (generically), or designs based on the Merkle-Damgård with Permutation (*MDP*) construction [HPY07, Hir21]. More modernly, it also works for sponges [BDPV07, BDPV11a] and (generically) SHA-3 [Nat15b]. In particular, the US NIST standardized MAC function *KMAC* [Joh16] follows this design approach with the Keccak/SHA-3 hash function [BDPV11b] underneath.

As the sponge function is a powerful function in general [BDPV07, BDPV11a, BDPA08, BDPA11, BDPV11b, DMA17, Men23], there has also been dedicated research in keying the sponge to obtain a MAC function. Notably, it was observed that if the key is prepended to the data, one can even authenticate more efficiently by absorbing over the entire state [BDPV12, MRV15] (a similar observation applies to the envelope construction on top of Merkle-Damgård [Yas07a]). For the sponge, there was also particular interest in its security if keyed *at the suffix*. The initial idea was already described by Bertoni et al. [BDPV11a, Section 5.11.2], and a general treatment of this approach, currently known as the suffix keyed sponge (or, *SuKS*), was given by Dobraunig and Mennink [DM19b]. They proved that *SuKS* is a secure PRF as long as the permutation is random. They furthermore demonstrated that, purely due to the keying at the suffix instead of the prefix, this construction is secure even in the case of side-channel attacks. More detailed, they proved that, by keying at the suffix, this construction is not only a black-box secure PRF but also a leakage resilient PRF.

1.1 Suffix Keyed and Suffix Blinded Merkle-Damgård

In general, one may say the plain use of Merkle-Damgård for message authentication has been overtaken by time. Specifically, using cryptographic hash functions whose underlying mode is indifferentiable from a random oracle yields a much simpler MAC design, most notably through $H(K\|M)$. This observation can particularly be seen as argument to step away from *HMAC* on top of SHA-1/SHA-2 and use *KMAC* instead.

That said, inspired by the recent works on sponges, and *SuKS* in particular, it makes sense from a theoretical perspective to investigate what we can achieve with the classical Merkle-Damgård construction when keying at the suffix, both in the black-box setting and in the leakage resilience setting. We do so by formalizing and analyzing two constructions:

- Suffix keyed Merkle-Damgård (or, *sukMD*) as described in Section 3.1 and depicted in Figure 2, that appends the key to the message with wise padding (to avoid the key being split into two data blocks);
- Suffix blinded Merkle-Damgård (or, *subMD*) as described in Section 3.2 and depicted in Figure 3, that blinds the state with the key using a uniform and universal hash function G, right before compressing the last message block.

At first sight, it appears that the analysis of these two constructions is obvious and quickly/immediately follows from several earlier results. However, this is not at all the case, for multiple reasons:

- One possibility to attempt is to rely on indifferentiability of variants of the Merkle-Damgård construction. Indeed, *sukMD* is basically prefix-free Merkle-Damgård *as long as the adversary does not guess the key*. This means that one could potentially reduce security of *sukMD* to the indifferentiability of prefix-free Merkle-Damgård [CDMP05], plus a bad event covering key guessing. A similar reasoning holds for *subMD* in relation to *MDP* [HPY07]. However, this approach inherently requires F to be modeled as a random function, which is a too strong condition for reasoning about PRF security, and in particular earlier works of Bellare et

al. [BCK96], Bellare [Bel06], and Yasuda [Yas07b] suggest that we can avoid that assumption;

- Looking at those earlier proofs [BCK96, Bel06, Yas07b], actually, they assume PRF security on F and then observe that a cascade of PRFs yields PRF security up to a certain bound. This approach cannot be adapted here, as in *sukMD*/*subMD* only the last evaluation of F gets secret input;
- In fact, the *sukMD*/*subMD* constructions remind us more of the hash-then-PRF approach, where a cryptographic hash function is used to turn the message into a digest and a PRF to turn this digest into a tag. Such construction can be proven secure under the assumption that the hash function is a random oracle [BR93], and in addition for composition reasons the hash and PRF are required to be independent primitives. These are two requirements we wish to avoid.

However, there is one earlier result that gets pretty close to what we would need, which is to argue security of the hash-then-PRF approach under the assumption that the hash function is collision resistant and the finalization is a PRF. This approach was followed by Rogaway in the human ignorance model [Rog06]. In detail, he proved that if one has an iterated hash function construction based on compression function F' followed by a PRF F, the resulting construction is PRF secure as long as F' is collision resistant (in the human ignorance model) and F is PRF secure.

1.2 Black-Box Security

Our security proofs for *sukMD* and *subMD* will be inspired by the proof of Rogaway, but tackle several technical issues present in the new constructions. Firstly, we will use the same primitive F for both hashing and finalization. Secondly, in *sukMD* the message is not only processed in the hashing part but possibly also partially alongside the key in the PRF part. Thirdly, in *subMD*, a uniform and universal hash function G blends the key into the state, and this is not supported by earlier proofs. While the first issue is mostly editorial (Rogaway proved his result through composition and for the current type of composition we turn out to be able to derive a single direct result), the other two are more subtle. Nonetheless, we manage to derive tight security bounds in Theorem 1 (for *sukMD*) and Theorem 2 (for *subMD*).

To be precise, we prove that *sukMD* is a secure PRF under the assumption that F is (keyless) collision resistant as well as PRF secure, and we prove that *subMD* is a secure PRF under the assumption that F is collision resistant as well as related-key PRF secure [BK03] under a key derivation function set based on universal hash function G. We remark that these security results do not contradict aforementioned offline attack [PvO95], as it trickles down to the collision resistance assumption of F.

We also demonstrate that these results straightforwardly generalize to the case where the keying mechanisms, suffix keying and suffix blinding, are applied to the Merkle-Damgård with Permutation (*MDP*) construction, yielding *sukMDP* and *subMDP*.

1.3 Leakage Resilience

In addition, we investigate the security of our constructions against side-channel adversaries. Indeed, if our constructions are evaluated in a hostile environment, side-channel attacks [Koc96, KJJ99] where the adversary may obtain certain information about the secret data through for example power consumption, become a serious threat. In the permutation-based setting, *SuKS* has been observed to generically behave quite well against side-channel attackers [DM19b, BM24] (in fact, unlike the plain keyed sponge that prepends the key to the message [DM19a]). In other words, at the mode level, *SuKS*

achieves a quite decent level of security against side-channel attacks and this means that only lighter security measures against side-channel attacks have to be included at the implementation level, most notably against SPA attacks [DM19b]. Not surprisingly, this *SuKS* construction has been used as authentication in the NIST lightweight competition finalist ISAP v2 [DEM+17, DEM+20, DEM+21].

Note that, as mentioned above, *sukMD* and *subMD* resemble a bit of the hash-then-PRF approach, for which leakage resilience analysis has already been performed before (cf., [BKP+16, BPPS17, GSWY19, BGP+20, BGP+23], among others). However, these results assume independent primitives and/or a perfectly protected cryptographic primitive, both of which are assumptions we aim to avoid in this work.

In particular, we derive our own results, and to do so we adopt the bounded leakage model of Dodis and Pietrzak [DP10], where we assume a generous upper bound λ on leakage per evaluation of a secret primitive, and where we consider an adversary that is given access to a leaky oracle and a challenge oracle.[3] The idea of this model is that, even if the adversary gains a certain amount of leakage, new evaluations of the construction should look random. We particularly restrict our focus to non-adaptive bounded leakage [FPS12], where the leakage function is defined prior to the experiment and is meant to capture specifically the attack target (such as a specific byte) of the side-channel adversary. This model of (non-adaptive) bounded leakage was used before for the analysis of various constructions [Pie09, YSPY10, FPS12, SPY+10, DP10, BMOS17] and it was also used by Dobraunig and Mennink [DM19a] for the analysis of the duplex construction and by the same authors [DM19b] for the analysis of *SuKS*.

In this model, we argue that *sukMD* and *subMD* as depicted in Figures 2 and 3, respectively, actually cannot achieve security: for both, there exists a leakage function complying with the model that allows the adversary to obtain the secret key or a secret state and to distinguish the scheme from random. We also argue that the attack against *sukMD* applies to enveloping and *HMAC*, too. That said, for *subMD* we demonstrate that with a slightly different padding rule, i.e., one that appends m zeros, the construction does achieve quite strong leakage resilience, in a similar fashion as how *SuKS* does [DM19b].

1.4 Outline

We discuss security notions and the concept of plain iterated hashing in the preliminary material in Section 2. We describe our two variants to key Merkle-Damgård at the suffix in Section 3: suffix keyed Merkle-Damgård is specified and analyzed in Section 3.1 and suffix blinded Merkle-Damgård is specified and analyzed in Section 3.2. These two sections also describe the generalizations to suffix keyed and suffix blinded Merkle-Damgård with Permutation, *sukMDP* and *subMDP*, respectively. Then, we investigate leakage resilience of the constructions in Section 4. The model we adopt is given in Section 4.1 and the particular assumption on G that we make is given in Section 4.2. We then investigate leakage resilience of *sukMD* in Section 4.3 and *subMD* in Section 4.4. The work is concluded in Section 5.

2 Preliminaries

Let $n \in \mathbb{N}$ and $m \in \mathbb{N} \cup \{*\}$. We denote the set of n-bit strings by $\{0,1\}^n$, the set of arbitrarily long strings by $\{0,1\}^*$, and the set of m-to-n-bit functions by $\mathsf{func}(m,n)$.

We define a padding function $\not\leftarrow_n : \{0,1\}^{n*} \to (\{0,1\}^n)^*$ that takes as input a bitstring $X \in \{0,1\}^{n*}$ of length a multiple of n bits and splits it into n-bit blocks $X_1, \ldots, X_\ell$. We define by $\not\leftarrow_n^{10} : \{0,1\}^* \to (\{0,1\}^n)^*$ the well-known 10^*-padding that takes as input an arbitrarily long string $X \in \{0,1\}^*$, appends a 1 and a minimal but sufficient number of 0s

[3] Alternative bounding approaches exist, cf., [KR19].

so that the length of the resulting string is a multiple of n, and splits this string into n-bit blocks $X_1, \ldots, X_\ell$. In other words, $\mathrel{\not\prec}^{10}_n(X) = \mathrel{\not\prec}_n(X \| 10^{-|X|-1 \bmod n})$. For $X \in \{0,1\}^n$ and if $m \leq n$, the leftmost m bits of X are denoted by $\mathrm{left}_m(X)$. For a finite set $\mathcal{X}$, $X \xleftarrow{\$} \mathcal{X}$ denotes the uniformly random drawing of an element X from $\mathcal{X}$.

2.1 Uniform and Universal Hashing

Let $k, m, n \in \mathbb{N}$ and $\delta, \varepsilon \in [0,1]$. A function $G : \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^n$ is δ-uniform if for any $X \in \{0,1\}^m$ and $Y \in \{0,1\}^n$,

$$\mathbf{Pr}\left(G(K,X) = Y\right) \leq \delta\,,$$

where $K \xleftarrow{\$} \{0,1\}^k$. The function is ε-universal if for any distinct $X, X' \in \{0,1\}^m$,

$$\mathbf{Pr}\left(G(K,X) = G(K,X')\right) \leq \varepsilon\,,$$

where $K \xleftarrow{\$} \{0,1\}^k$.

We note that the simple XOR operator, e.g., $G(K,X) = K \| 0^{m-k} \oplus X$ assuming $k \leq m$, is 2^{-k}-uniform and 0-universal. This is a logical choice for our application in *subMD* in Section 3.2. However, in the context of leakage, we will require G to be easy to protect against leakage, and then we may require a more involved function G.

2.2 Collision Resistance

Let $m, n \in \mathbb{N}$. Consider a function $F : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$. Often, one requires that F has no structural weaknesses and behaves like a random function $R_{n+m,n} \xleftarrow{\$} \mathsf{func}(n+m, n)$. However, in our analysis, we will use F as a building block and merely require collision resistance (and PRF security in a certain form of keying, cf., Sections 2.3 and 2.4). Unfortunately, defining collision resistance of F is a paradoxical task. As $n+m \geq n$, the function *does have collisions*, and these may be hardwired in the algorithm of the adversary. Such an adversary can thus output collisions for F with probability 1 in constant time. Of course, it is hard to actually *find* such an adversary, even though it is known to exist. This gives room for still defining collision resistance of F, adopting the human ignorance formalization of Rogaway [Rog06]. In detail, we define the collision security of F against an adversary $\mathcal{A}$ as follows:

$$\mathbf{Adv}^{\mathrm{col}}_F(\mathcal{A}) = \mathbf{Pr}\left((X,X') \leftarrow \mathcal{A} \;:\; X \neq X' \wedge F(X) = F(X')\right)\,, \tag{2}$$

where the randomness is taken over the random coins of $\mathcal{A}$. As said before, as $n+m \geq n$, there exists an adversary $\mathcal{A}$ such that $\mathbf{Adv}^{\mathrm{col}}_F(\mathcal{A}) = 1$. However, as in our work we consider *explicitly constructed* adversaries, this is not a problem.

We will, however, use a *slightly stronger* notion where $\mathcal{A}$ also wins if it finds a preimage for a dedicated initialization vector IV. This idea appeared before, e.g., in [AMPS12], and allows us to refrain from using length encoding in our constructions of Section 3. In detail, instead of (2), we define the collision security of F against an adversary $\mathcal{A}$ as follows:

$$\mathbf{Adv}^{\mathrm{col}}_F(\mathcal{A}) = \mathbf{Pr}\left((X,X') \leftarrow \mathcal{A} \;:\; X \neq X' \wedge F(X) \in \{F(X'), IV\}\right)\,, \tag{3}$$

where $IV \in \{0,1\}^n$ is a predetermined constant, and where the randomness is taken over the random coins of $\mathcal{A}$. We say that F is *collision resistant* if $\mathbf{Adv}^{\mathrm{col}}_F(\mathcal{A})$ is sufficiently small for all *known* adversaries that operate in a certain predefined time T.

2.3 PRF Security

Let $k, m, n \in \mathbb{N}$ and $m' \in \mathbb{N} \cup \{*\}$. Consider a function $F : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$, and a construction $C[F] : \{0,1\}^k \times \{0,1\}^{m'} \to \{0,1\}^n$ that merely describes how a key goes into F.[4] We write $C[F]$ instantiated with a key $K \in \{0,1\}^k$ as $C[F]_K$. The pseudorandom function (PRF) security of $C[F]$ is defined by how hard it is for an adversary to distinguish $C[F]_K$ with a secret key K from a random oracle [BR93] $R_{m',n} \xleftarrow{\$} \mathsf{func}(m', n)$. In detail, we consider an adversary $\mathcal{A}$ that is given access to either of those and aims to determine which one it communicates with, by outputting a decision bit $b \in \{0,1\}$. We define the pseudorandom function (PRF) security of F against an adversary $\mathcal{A}$ as follows:

$$\mathbf{Adv}^{\mathrm{prf}}_{C[F]}(\mathcal{A}) = \mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{C[F]_K}\right) - \mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{R_{m',n}}\right), \tag{4}$$

where $K \xleftarrow{\$} \{0,1\}^k$ and $R_{m',n} \xleftarrow{\$} \mathsf{func}(m', n)$. We say that $C[F]$ is a *secure pseudorandom function* if $\mathbf{Adv}^{\mathrm{prf}}_{C[F]}(\mathcal{A})$ is sufficiently small for any adversary $\mathcal{A}$ with a certain query complexity Q of total length S bits and time complexity T.

2.4 Related-Key PRF Security

We also require PRF security in the related-key setting [BK03]. Let $\Phi = \{\phi : \{0,1\}^k \to \{0,1\}^k\}$ be a set of key derivation functions, and let $RK : \{0,1\}^k \times \Phi \to \{0,1\}^k$ be defined as $RK(K, \varphi) = \varphi(K)$. The related-key pseudorandom function (RK-PRF) security of $C[F]$ is defined by how hard it is for an adversary to distinguish $C[F]_{RK(K,\cdot)}$ with a secret key K from a family of random oracles [BR93] $(R_{m',n})_{(\cdot)} \xleftarrow{\$} \mathsf{func}(m', n)^{|\Phi|}$ indexed by the set of key deriving functions. In detail, we consider an adversary $\mathcal{A}$ that is given access to either of those and aims to determine which one it communicates with, by outputting a decision bit $b \in \{0,1\}$. We define the related-key pseudorandom function (RK-PRF) security of F against an adversary $\mathcal{A}$ as follows:

$$\mathbf{Adv}^{\text{rk-prf}}_{\Phi,C[F]}(\mathcal{A}) = \mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{C[F]_{RK(K,\cdot)}}\right) - \mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{(R_{m',n})_{(\cdot)}}\right), \tag{5}$$

where $K \xleftarrow{\$} \{0,1\}^k$ and $(R_{m',n})_{(\cdot)} \xleftarrow{\$} \mathsf{func}(m', n)^{|\Phi|}$. We say that $C[F]$ is a *secure related-key pseudorandom function* under key derivation function set Φ if $\mathbf{Adv}^{\text{rk-prf}}_{\Phi,C[F]}(\mathcal{A})$ is sufficiently small for any adversary $\mathcal{A}$ with a certain query complexity Q of total length S bits and time complexity T.

2.5 Iterated Hashing and Merkle-Damgård Hash Function

Let $m, n \in \mathbb{N}$. Let $IV \in \{0,1\}^n$ be a predefined initialization value. Consider a function $F : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$. We define the iterated hash function construction $IH : \{0,1\}^n \times (\{0,1\}^m)^* \to \{0,1\}^n$ as

$$IH(IV, M_1, \ldots, M_i) = \begin{cases} IV\,, & \text{if } i = 0\,, \\ F(IH(IV, M_1, \ldots, M_{i-1}), M_i)\,, & \text{otherwise}\,. \end{cases}$$

The Merkle-Damgård hash function construction $MD : \{0,1\}^* \to \{0,1\}^n$ is simply defined as evaluating IH on a *padded* message and for predefined $IV \in \{0,1\}^n$:

$$MD(M) = IH(IV, \not\leftarrow^{10}_m(M))\,. \tag{6}$$

[4] In our schemes, the key may go into F either through the data or message path.

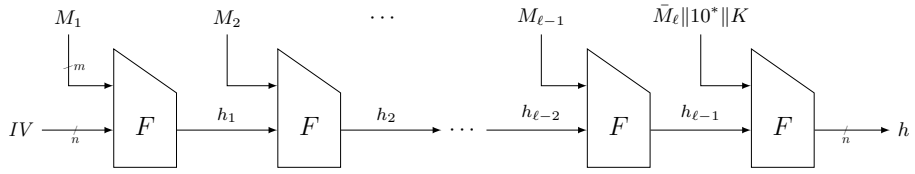

Figure 2: The suffix keyed Merkle-Damgård construction (*sukMD*) for the specific case where $|M| \bmod m \in [0, m-k-1]$. Here, the message M is split into message blocks $M_1, \ldots, M_{\ell-1}, \bar{M}_\ell$, where the M_is are all m bits long and $\bar{M}_\ell$ is between 0 and $m-k-1$ bits. If $|M| \bmod m \in [m-k, m-1]$, $\bar{M}_\ell$ would be compressed separately from K.

3 Keying Merkle-Damgård

Let $k, m, n \in \mathbb{N}$ such that $k \leq \min\{m, n\}$ (we get back to this condition in Section 5). As mentioned in Section 1, keying the Merkle-Damgård construction by simply concatenating $K \in \{0,1\}^k$ and message $M \in \{0,1\}^*$, as $MD(K\|M)$, does not work. In this section, we will explore the possibilities of keying the Merkle-Damgård construction at the end. We first describe suffix keyed Merkle-Damgård (or, *sukMD*) in Section 3.1 and then suffix blinded Merkle-Damgård (or, *subMD*) in Section 3.2.

3.1 Suffix Keyed Merkle-Damgård

The *sukMD* construction is fairly straightforward, and can be dubbed folklore in the sense that one simply appends key to the message. The first appearance of this idea goes back to Tsudik [Tsu92]. In its native form, the plain definition $sukMD(K, M) = MD(M\|K)$ is not ideal: indeed, an adversary could vary the length of the plaintext and this way slide the key K bit-by-bit into the last padded message block, and this will not allow us to prove security under well-established security notions such as PRF security of Section 2.3.[5] Instead, we pad M internally to make sure the padding function does not split K into two message blocks:

$$sukMD(K, M) = IH\Big(IV, \not\prec_m(M\|10^{-|M|-k-1 \bmod m}\|K)\Big)\,. \tag{7}$$

The construction is depicted in Figure 2.

3.1.1 Security

This *sukMD* construction is a secure PRF under the assumption that F is collision resistant and the overlying construction $C_{suk}[F] : \{0,1\}^k \times \{0,1\}^{n+m-k} \to \{0,1\}^n$ defined as

$$C_{suk}[F](K, X) = F(X\|K) \tag{8}$$

is PRF secure. The proof resembles that of Rogaway [Rog06, Theorems 3 and 4], with the difference that (i) the same building block is used for both the keyless and the keyed part and (ii) the last message block may be processed together with the key.[6] In particular issue (ii) makes the proof more subtle.

Theorem 1. *Let $k, m, n \in \mathbb{N}$. Let $F : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$ be a compression function. Consider the suffix keyed Merkle-Damgård construction sukMD of (7). For*

[5] It may be possible to derive security in the random function model, using ideas of Mennink [Men18] on how to bound key prediction in the sponge construction if the key is absorbed in multiple permutation calls.

[6] Note that, typically, m is larger than n. For example, for the SHA-2 family [Nat15a], we have $(m, n) = (512, 256)$ or $(m, n) = (1024, 512)$.

any adversary $\mathcal{A}$ with construction query complexity Q of total length S bits and time complexity T,

$$\mathbf{Adv}^{\mathrm{prf}}_{sukMD}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{col}}_{F}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{prf}}_{C_{suk}[F]}(\mathcal{C})\,,$$

for some adversary $\mathcal{B}$ that runs in time at most $T + O(T_F S)$ and $\mathcal{C}$ that makes at most Q queries and runs in time at most $T + O(T_F S)$, where T_F is the time to evaluate the function F.

Proof. Consider any adversary $\mathcal{A}$ that has access to either $sukMD_K$ or a random function $R_{*,n}$. It makes Q queries of total length S bits and operates in time T.

Collision Adversary $\mathcal{B}$. We construct a collision finding adversary $\mathcal{B}$ for F as follows. Adversary $\mathcal{B}$ itself has no construction oracle, but it runs $\mathcal{A}$ as an oracle, and takes a dummy key $K^\star$. Whenever $\mathcal{A}$ makes a query $M^{(i)}$, $\mathcal{B}$ computes padded message blocks

$$M^{(i)}_1, \ldots, M^{(i)}_{\ell^{(i)}} \leftarrow \bowtie_m (M^{(i)} \| 10^{-|M^{(i)}|-k-1 \bmod m} \| K^\star)\,.$$

It discards the last block $M^{(i)}_{\ell^{(i)}}$ (the key absorption merely serves as decoration to well-define the indices), and computes the intermediate chaining values $h^{(i)}_1, \ldots, h^{(i)}_{\ell^{(i)}-1}$ corresponding to the absorptions of the first $\ell^{(i)} - 1$ blocks, exactly as in Figure 2. The final reply of adversary $\mathcal{B}$ to $\mathcal{A}$ depends on the set

$$\mathcal{J}^{(i)} = \{j < i \mid h^{(i)}_{\ell^{(i)}-1} = h^{(j)}_{\ell^{(j)}-1} \wedge M^{(i)}_{\ell^{(i)}} = M^{(j)}_{\ell^{(j)}}\}\,.$$

In detail, it replies as follows:

- If $|\mathcal{J}^{(i)}| = 0$, $\mathcal{B}$ responds to $\mathcal{A}$ with a uniform random $h \xleftarrow{\$} \{0,1\}^n$;
- If $|\mathcal{J}^{(i)}| > 0$, $\mathcal{B}$ responds to $\mathcal{A}$ with $h = h^{(j)}$ for some $j \in \mathcal{J}^{(i)}$.

At the end of the experiment, $\mathcal{A}$ outputs a decision bit $b \in \{0,1\}$. Adversary $\mathcal{B}$, instead, ignores the decision bit and uses its received data to output a collision, *if possible*. In detail, if for all queries $i = 1, \ldots, Q$, $|\mathcal{J}^{(i)}| = 0$, $\mathcal{B}$ simply *fails*. Otherwise, Let i be the smallest index such that $j \in \mathcal{J}^{(i)}$ exists (which is unique as i is the smallest index). Without loss of generality, $\ell^{(i)} \geq \ell^{(j)}$. Note that $M^{(i)} \neq M^{(j)}$ but $M^{(i)}_{\ell^{(i)}} = M^{(j)}_{\ell^{(j)}}$. We can distinguish two cases:

- Case $M^{(i)}_{\ell^{(i)}-\kappa} \neq M^{(j)}_{\ell^{(j)}-\kappa}$ for some $\kappa \in \{1, \ldots, \ell^{(j)} - 1\}$. Let κ be minimal such that this condition holds. Then, $\mathcal{B}$ has found a compression function collision

  $$F(h^{(i)}_{\ell^{(i)}-\kappa-1}, M^{(i)}_{\ell^{(i)}-\kappa}) = F(h^{(j)}_{\ell^{(j)}-\kappa-1}, M^{(j)}_{\ell^{(j)}-\kappa})\,,$$

  where $h_0 = IV$ by definition;

- Case $M^{(i)}_{\ell^{(i)}-\kappa} = M^{(j)}_{\ell^{(j)}-\kappa}$ for all $\kappa \in \{1, \ldots, \ell^{(j)} - 1\}$. As $M^{(i)} \neq M^{(j)}$ but $M^{(i)}_{\ell^{(i)}} = M^{(j)}_{\ell^{(j)}}$, this necessarily means that $\ell^{(i)} > \ell^{(j)}$, and $\mathcal{B}$ has found a compression function collision

  $$F(h^{(i)}_{\ell^{(i)}-\ell^{(j)}}, M^{(i)}_{\ell^{(i)}-\ell^{(j)}+1}) = F(h^{(j)}_0, M^{(j)}_1)$$

  if $h^{(i)}_{\ell^{(i)}-\ell^{(j)}} \neq IV$, or

  $$F(h^{(i)}_{\ell^{(i)}-\ell^{(j)}-1}, M^{(i)}_{\ell^{(i)}-\ell^{(j)}}) = IV$$

  otherwise.

PRF Adversary $\mathcal{C}$. We construct a PRF adversary $\mathcal{C}$ for $C_{suk}[F]$ as follows. Adversary $\mathcal{C}$ has access to either $C_{suk}[F]_K$ or a random function $R_{n+m-k,n}$, and it runs $\mathcal{A}$ as an oracle. Whenever $\mathcal{A}$ makes a query $M^{(i)}$, $\mathcal{C}$ computes padded message blocks (again, the key absorption of $K^\star$ merely serves as decoration to well-define the indices)

$$M_1^{(i)}, \ldots, M_{\ell^{(i)}}^{(i)} \leftarrow \mathbin{\not\leftarrow}_m(M^{(i)} \| 10^{-|M^{(i)}|-k-1 \bmod m} \| K^\star) .$$

It computes the intermediate chaining values $h_1^{(i)}, \ldots, h_{\ell^{(i)}-1}^{(i)}$ corresponding to the absorptions of the first $\ell^{(i)} - 1$ blocks, exactly as in Figure 2. It queries $(h_{\ell^{(i)}-1}^{(i)} \| \mathrm{left}_{m-k}(M_{\ell^{(i)}}^{(i)}))$ to its own oracle ($C_{suk}[F]_K$ or $R_{n+m-k,n}$) and relays the response. At the end of the experiment, $\mathcal{A}$ outputs a decision bit $b \in \{0,1\}$ and $\mathcal{C}$ relays this decision bit.

Conclusion of Proof. Note that

$$\begin{aligned} \mathbf{Adv}^{\mathrm{prf}}_{sukMD}(\mathcal{A}) - \mathbf{Adv}^{\mathrm{prf}}_{C_{suk}[F]}(\mathcal{C}) = &\left(\mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{sukMD_K}\right) - \mathbf{Pr}\left(1 \leftarrow \mathcal{C}^{C_{suk}[F]_K}\right)\right) \\ &+ \left(\mathbf{Pr}\left(1 \leftarrow \mathcal{C}^{R_{n+m-k,n}}\right) - \mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{R_{*,n}}\right)\right). \end{aligned} \tag{9}$$

By construction of $\mathcal{C}$,

$$\mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{sukMD_K}\right) = \mathbf{Pr}\left(1 \leftarrow \mathcal{C}^{C_{suk}[F]_K}\right) .$$

Furthermore, denote by col the event that in the evaluation of $\mathcal{C}$, there are two different queries i, j such that $(h_{\ell^{(i)}-1}^{(i)} \| \mathrm{left}_{m-k}(M_{\ell^{(i)}}^{(i)})) = (h_{\ell^{(j)}-1}^{(j)} \| \mathrm{left}_{m-k}(M_{\ell^{(j)}}^{(j)}))$. Clearly, as long as col does not happen, also the other two probabilities of (9) are the same, and by the fundamental lemma of game-playing [BR06],

$$\mathbf{Pr}\left(1 \leftarrow \mathcal{C}^{R_{n+m-k,n}}\right) - \mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{R_{*,n}}\right) \leq \mathbf{Pr}\left(\mathsf{col}\right) . \tag{10}$$

However, note that the event that col happens is equivalent to stating that $|\mathcal{J}^{(i)}| > 0$ for any $i = 1, \ldots, Q$. In other words, col happens if and only if $\mathcal{B}$ succeeds in finding a collision:

$$\mathbf{Pr}\left(\mathsf{col}\right) = \mathbf{Adv}^{\mathrm{col}}_{F}(\mathcal{B}) . \tag{11}$$

Combining (9)–(11), we obtain

$$\mathbf{Adv}^{\mathrm{prf}}_{sukMD}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{col}}_{F}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{prf}}_{C_{suk}[F]}(\mathcal{C}) .$$

The complexities of $\mathcal{B}$ and $\mathcal{C}$ are as stated in the theorem environment. □

3.1.2 Extension to Suffix Keyed Merkle-Damgård with Permutation

Given the resemblance between Merkle-Damgård and Merkle-Damgård with Permutation (*MDP*) [HPY07], where a non-cryptographic permutation π is used to transform the state prior to the last compression function call, it makes sense to also consider the extension of *sukMD* to *MDP*. This leads to the following construction:

$$sukMDP(K, M) = IH\Big(\pi\big(IH(IV, M_1, \ldots, M_{\ell-1})\big), M_\ell\Big) , \tag{12}$$

where $M_1, \ldots, M_\ell \leftarrow \mathbin{\not\leftarrow}_m(M \| 10^{-|M|-k-1 \bmod m} \| K)$.

It turns out that Theorem 1 immediately carries over to this construction. This is because the only difference between *sukMD* and *sukMDP* is the non-cryptographic permutation π, but in the proof of *sukMD* of Theorem 1, collision adversary $\mathcal{B}$ "stops" before the last F and thus before π whereas $\mathcal{C}$ computes everything offline until the last F so until and including π. This gives a clean standard model proof of *sukMDP* (noting that a random function model proof would already follow from the indifferentiability of *MDP* [HPY07, Hir21]).

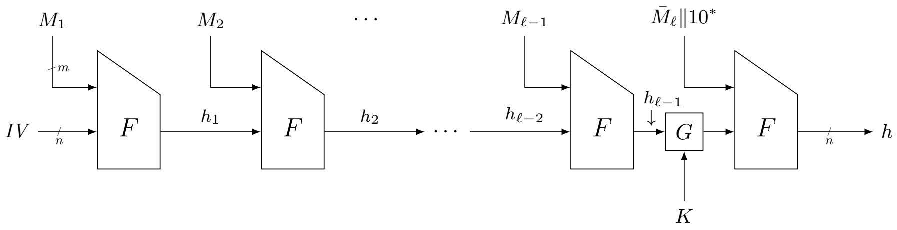

Figure 3: The suffix blinded Merkle-Damgård construction (*subMD*). Here, the message M is split into message blocks $M_1, \ldots, M_{\ell-1}, \bar{M}_\ell$, where the M_is are all m bits long and $\bar{M}_\ell$ is between 0 and $m-1$ bits.

3.2 Suffix Blinded Merkle-Damgård

Although the *sukMD* construction can be seen as the Merkle-Damgård equivalent of *SuKS*, one may actually consider the *subMD* construction to be a closer resemblance. In detail, *subMD updates* the inner part of the state by the key using a function $G : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$. In detail, it is defined as follows:

$$\mathit{subMD}(K, M) = \mathit{IH}\Big(G\big(K, \mathit{IH}(IV, M_1, \ldots, M_{\ell-1})\big), M_\ell\Big)\,, \tag{13}$$

where $M_1, \ldots, M_\ell \leftarrow ✂^{10}_m(M)$. The construction is depicted in Figure 3. Note that this function can be implemented black-box on top of an implementation of IH. Looking ahead, the function G will be required to be uniform and universal, meaning that a simple XOR that *blinds* the state with the key, i.e., $G(K,h) = K\|0^{n-k} \oplus h$, suffices.

3.2.1 Security

This *subMD* construction is a secure PRF under the assumption that F is collision resistant and the overlying construction $C_{sub}[F] : \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^n$ defined as

$$C_{sub}[F](K, X) = F(K\|X) \tag{14}$$

is related-key PRF secure under a key deriving function set based on G. To be precise, we take key derivation function set[7]

$$\Phi_{sub}[G] = \{h : K \mapsto G(K,h) \mid h \in \{0,1\}^n\}\,. \tag{15}$$

The proof resembles a bit that of Theorem 1, but significant additional complexity appears in the fact that the keying is performed differently at the end, namely external of the function F. If we were to analyze our constructions in the ideal function model, the reasoning would be much simpler, but now, the PRF security of the underlying primitive has to account for this difference through related-key security.

Theorem 2. *Let $k, m, n \in \mathbb{N}$ and $\delta, \varepsilon \in [0,1]$. Let $F : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$ be a compression function and $G : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ a function that is δ-uniform and ε-universal. Consider the suffix blinded Merkle-Damgård construction subMD of (13). For any adversary $\mathcal{A}$ with construction query complexity Q of total length S bits and time complexity T,*

$$\mathbf{Adv}^{\mathrm{prf}}_{subMD}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{col}}_{F}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{rk\text{-}prf}}_{\Phi_{sub}[G], C_{sub}[F]}(\mathcal{C})\,,$$

[7] We admit that this definition is slightly abusing notation as the functions in this key derivation function set map from k to n bits. This is not a problem for the formalization.

for some adversary $\mathcal{B}$ that runs in time at most $T + O(T_F S)$ and $\mathcal{C}$ that makes at most Q queries and runs in time at most $T + O(T_F S)$, where T_F is the time to evaluate the function F.

Proof. Consider any adversary $\mathcal{A}$ that has access to either $subMD_K$ or a random function $R_{*,n}$. It makes Q queries of total length S bits and operates in time T.

Collision Adversary $\mathcal{B}$. We construct a collision finding adversary $\mathcal{B}$ for F as follows. Adversary $\mathcal{B}$ itself has no construction oracle, but it runs $\mathcal{A}$ as an oracle, and takes a dummy key $K^\star$. In fact, the adversary $\mathcal{B}$ behaves almost identical to $\mathcal{B}$ of the proof of Theorem 1: it discards (and ignores) the last message block and outputs random responses to $\mathcal{A}$ whenever needed. As the only main difference between *sukMD* and *subMD* is in the processing of this last block, the reasoning almost seamlessly carries over. To be precise, the main difference is that, whenever $\mathcal{A}$ makes a query $M^{(i)}$, $\mathcal{B}$ now computes padded message blocks

$$M_1^{(i)}, \ldots, M_{\ell^{(i)}}^{(i)} \leftarrow \not\ltimes_m(M^{(i)} \| 10^{-|M^{(i)}|-1 \bmod m}) .$$

Then, it discards the last block $M_{\ell^{(i)}}^{(i)}$, and computes the intermediate chaining values $h_1^{(i)}, \ldots, h_{\ell^{(i)}-1}^{(i)}$ corresponding to the absorptions of the first $\ell^{(i)} - 1$ blocks, exactly as in Figure 3, and it proceeds identical to $\mathcal{B}$ of the proof of Theorem 1, using the same decision set $\mathcal{J}^{(i)}$ and the same collision derivation.

PRF Adversary $\mathcal{C}$. We construct a RK-PRF adversary $\mathcal{C}$ for $C_{sub}[F]$ under key deriving function set $\Phi_{sub}[G]$ as follows. Adversary $\mathcal{C}$ has access to either $C_{sub}[F]_{RK(K,\cdot)}$ or a family of random functions $(R_{m,n})_{(\cdot)}$, and it runs $\mathcal{A}$ as an oracle. In fact, the adversary $\mathcal{C}$ behaves very similar to $\mathcal{C}$ of the proof of Theorem 1: the only difference is in how it hides its own oracle into the bigger evaluations of $\mathcal{A}$. Whenever $\mathcal{A}$ makes a query $M^{(i)}$, $\mathcal{C}$ computes padded message blocks

$$M_1^{(i)}, \ldots, M_{\ell^{(i)}}^{(i)} \leftarrow \not\ltimes_m(M^{(i)} \| 10^{-|M^{(i)}|-1 \bmod m}) .$$

It computes the intermediate chaining values $h_1^{(i)}, \ldots, h_{\ell^{(i)}-1}^{(i)}$ corresponding to the absorptions of the first $\ell^{(i)} - 1$ blocks, exactly as in Figure 3. It queries $(h_{\ell^{(i)}-1}^{(i)}, M_{\ell^{(i)}}^{(i)})$ to its own oracle ($C_{sub}[F]_{RK(K,\cdot)}$ or $(R_{m,n})_{(\cdot)}$) and relays the response. At the end of the experiment, $\mathcal{A}$ outputs a decision bit $b \in \{0,1\}$ and $\mathcal{C}$ relays this decision bit.

Conclusion of Proof. The conclusion is fairly identical to that of the proof of Theorem 1, but the change towards relying on RK-PRF security is a bit subtle and we will state that part explicitly here. Note that

$$\begin{aligned}\mathbf{Adv}^{\mathrm{prf}}_{subMD}(\mathcal{A}) - \mathbf{Adv}^{\text{rk-prf}}_{\Phi_{sub}[G],C_{sub}[F]}(\mathcal{C}) = &\left(\mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{subMD_K}\right) - \mathbf{Pr}\left(1 \leftarrow \mathcal{C}^{C_{sub}[F]_{RK(K,\cdot)}}\right)\right) \\ &+ \left(\mathbf{Pr}\left(1 \leftarrow \mathcal{C}^{(R_{m,n})_{(\cdot)}}\right) - \mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{R_{*,n}}\right)\right) .\end{aligned} \tag{16}$$

By construction of $\mathcal{C}$,

$$\mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{subMD_K}\right) = \mathbf{Pr}\left(1 \leftarrow \mathcal{C}^{C_{sub}[F]_{RK(K,\cdot)}}\right) .$$

Furthermore, let the event col be as in the proof of Theorem 1. Clearly, as long as col does not happen, also the other two probabilities are the same, and by the fundamental lemma

of game-playing [BR06],

$$\mathbf{Pr}\left(1 \leftarrow \mathcal{C}^{(R_{m,n})_{(\cdot)}}\right) - \mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{R_{*,n}}\right) \leq \mathbf{Pr}\left(\mathsf{col}\right) . \tag{17}$$

However, as before, col happening is equivalent to stating that $|\mathcal{J}^{(i)}| > 0$ for any $i = 1, \ldots, Q$, and thus:

$$\mathbf{Pr}\left(\mathsf{col}\right) = \mathbf{Adv}_F^{\mathrm{col}}(\mathcal{B}) . \tag{18}$$

Combining (16)–(18), we obtain

$$\mathbf{Adv}_{subMD}^{\mathrm{prf}}(\mathcal{A}) \leq \mathbf{Adv}_F^{\mathrm{col}}(\mathcal{B}) + \mathbf{Adv}_{\Phi_{sub}[G], C_{sub}[F]}^{\text{rk-prf}}(\mathcal{C}) .$$

The complexities of $\mathcal{B}$ and $\mathcal{C}$ are as stated in the theorem environment. □

3.2.2 Extension to Suffix Blinded Merkle-Damgård with Permutation

As in Section 3.1, we can extend *subMD* to the case the underlying hash function construction is Merkle-Damgård with Permutation (*MDP*) [HPY07]. This leads to the following construction:

$$subMDP(K, M) = IH\Big(\pi\big(G\big(K, IH(IV, M_1, \ldots, M_{\ell-1})\big)\big), M_\ell\Big) , \tag{19}$$

where $M_1, \ldots, M_\ell \leftarrow ✂_m^{10}(M)$.

The result of Theorem 2 immediately carries over to this construction, because for *subMDP* the function π can be integrated into G.

3.2.3 Impact of Related-Key PRF Security Term

The impact of the security of G, i.e., the impact of δ and ε, on the bound is implicit in the term $\mathbf{Adv}_{\Phi_{sub}[G], C_{sub}[F]}^{\text{rk-prf}}(\mathcal{C})$. Similar to Bellare and Kohno showed [BK03, Theorem 1], we can make it explicit by considering the ideal model where F is a random function and the adversary can make T queries to it:

$$\mathbf{Adv}_{\Phi, C[F]}^{\text{i-rk-prf}}(\mathcal{A}) = \mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{C[F]_{RK(K,\cdot)}, F}\right) - \mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{(R_{m',n})_{(\cdot)}, F}\right) , \tag{20}$$

where $F \xleftarrow{\$} \mathsf{func}(n+m, n)$, $K \xleftarrow{\$} \{0,1\}^k$, and $(R_{m',n})_{(\cdot)} \xleftarrow{\$} \mathsf{func}(m', n)^{|\Phi|}$.

Proposition 1. *Let $k, m, n \in \mathbb{N}$ and $\delta, \varepsilon \in [0,1]$. Let $G : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a function that is δ-uniform and ε-universal. For any adversary $\mathcal{C}$ with construction query complexity Q of total length S bits and primitive query complexity T,*

$$\mathbf{Adv}_{\Phi_{sub}[G], C_{sub}[F]}^{\text{i-rk-prf}}(\mathcal{C}) \leq QT\delta + \binom{Q}{2}\varepsilon .$$

Proof. The proof is based on [BK03, Theorem 1] but then in the language of G being a hash function family. Consider any adversary $\mathcal{C}$ that has access to either $C_{sub}[F]_{RK(K,\cdot)}$ or a family of random functions $(R_{m,n})_{(\cdot)}$, and it can make a total amount of Q queries. In addition, it has access to F and it can make T queries.

Denote by col the event that there are two evaluations of $C_{sub}[F]_{RK(K,\cdot)}$ for different queries X, X' such that $G(K, X) = G(K, X')$. In addition, denote by guess the event that there is an evaluation of $C_{sub}[F]_{RK(K,\cdot)}$ for a query X and an evaluation of F for a query Y such that $G(K, X) = Y$. Clearly, as long as col and guess do not happen, evaluations

of $C_{sub}[F]_{RK(K,\cdot)}$ are perfectly indistinguishable from $(R_{m,n})_{(\cdot)}$, and by the fundamental lemma of game-playing [BR06],

$$\mathbf{Adv}^{\text{i-rk-prf}}_{\Phi_{sub}[G],C_{sub}[F]}(\mathcal{C}) \leq \mathbf{Pr}\left(\mathsf{col}\right) + \mathbf{Pr}\left(\mathsf{guess}\right) . \tag{21}$$

As the adversary makes Q construction queries and T primitive queries, we have that $\mathbf{Pr}\left(\mathsf{col}\right) \leq \binom{Q}{2}\varepsilon$ and $\mathbf{Pr}\left(\mathsf{guess}\right) \leq QT\delta$. □

A similar bound can be derived in the ideal cipher model where F is assumed to be a blockcipher-based compression function such as Davies-Meyer [MOI90, PGV93]. This can be done by re-performing a security proof of Davies-Meyer [BRS02, BRSS10] with as additional bad events the events col and guess of Proposition 1. These ideal model analyses would, however, give no guarantees in case F is instantiated using, for example, one of the SHA-1 or SHA-2 compression functions [Nat15a]. We would like to stress, however, that this step, i.e., the term $\mathbf{Adv}^{\text{rk-prf}}_{\Phi_{sub}[G],C_{sub}[F]}(\mathcal{C})$ in Theorem 2, is admittedly a weak step in the composition. In fact, the adversary $\mathcal{A}$ against *subMD* cannot freely choose the key relation as it is defined by the last keyless chaining value $h_{\ell-1}$, whereas $\mathcal{C}$ against $C_{sub}[F]$ can.

4 Leakage Resilience of Constructions

We will investigate the security of *sukMD* and *subMD* in the leaky setting. Before doing so, we remark that it is fairly straightforward to prove that the constructions are leakage resilient, *provided a strong enough model is adopted*. In particular, if we assume that the *last* evaluation of F is assumed to be leak-free (a concept known as leveled implementations [PSV15]), the original security result for *sukMD* of Theorem 1 immediately carries over to leakage resilience, and if we additionally assume that G is leak-free, the same observation can be made for the security result for *subMD* of Theorem 2. These assumptions would not be novel. In fact, various research works [BKP+16, BPPS17, BKP+18, GSWY19, GPPS20, BGP+20, BGP+23] prove leakage resilience of cascaded constructions where the first and/or last cryptographic primitive enjoys stronger leakage protection and is thus assumed to be leak-free. We would like to avoid this rather strong restriction if not strictly necessary, and therefore, we will not impose it on F. We will make a comparable assumption on just G, which we believe is reasonable: in our constructions G can be a cryptographically weaker primitive than F, and this would possibly make it cheaper to apply strong protection on. This reasoning was also used in the design of various rekeying schemes [AB00, Bor01, MSGR10, MPR+11, DEMM14, DKM+15, Men20] and in the leakage resilience of *SuKS* [DM19b].

The rest of this section is outlined as follows. First, we discuss the model of leakage resilient PRFs that we adopt in this work in Section 4.1. We elaborate on the assumption of leak-freeness of G in Section 4.2. Then, we investigate the leakage resilience of *sukMD* in Section 4.3 and that of *subMD* in Section 4.4.

4.1 Leakage Resilient PRF Security

We will expand the notion of PRF security of Section 2.3 in the setting of non-adaptive $\mathcal{L}$-resilience of Dodis and Pietrzak [DP10], where the adversary receives leakage under any leakage function $L \in \mathcal{L}$ applied on the data dealt with in the construction. In this security model, the adversary gets access to a leak-free version of the construction that it has to distinguish from random, just as in PRF security of Section 2.3, but it additionally gets access to a leaky version in both worlds, and it may use this oracle to gain additional knowledge [Pie09, YSPY10, FPS12, SPY+10, DP10, BMOS17]. The idea of this model is

that, even with certain leakage already obtained, new evaluations of the construction are indistinguishable from random.

We remark that the same model was also used in the analysis of the duplex construction [DM19a] and *SuKS* by Dobraunig and Mennink [DM19b], with a crucial difference that their analysis was in the ideal permutation model. This model simplifies in the sense that (i) the leakage functions are by default independent of the random primitive and (ii) this primitive itself always generates a sufficient level of randomness so one does not have to rely on the HILL-pseudoentropy approach [HILL99, HLR07] (see also [DM19a, Section 4.1]). These advantages would have also applied in our setting if we had opted to perform security analysis under the assumption that F is a random function. Still, it turns out that for our construction the non-adaptive model of Dodis and Pietrzak [DP10] works. The reason is that, for each evaluation of *subMD*/*sukMD*, the function F is only evaluated *once* per query on secret input (this was in fact the whole point of the design) and this conceptually simplifies the leakage analysis.

In detail, let $k, m, n \in \mathbb{N}$ such that $k \leq \min\{m, n\}$, $m' \in \mathbb{N} \cup \{*\}$, and let $\lambda, \lambda' \in \mathbb{N}$. Let $C[F]$ be either the suffix keyed or suffix blinded Merkle-Damgård construction from Section 3 that internally uses a function $F : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$ and, in the case of *subMD*, additionally a function $G : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$. (The usage of G is not made explicit in "$C[F]$" as we do not require cryptographic properties of G.) Let $\mathcal{L}_F = \{L_F : \{0,1\}^n \times \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^\lambda\}$ be a fixed leakage set that consists of all allowed leakage functions on F and $\mathcal{L}_G = \{L_G : \{0,1\}^k \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{\lambda'}\}$ be a fixed leakage set that consists of all allowed leakage functions on G. Write $\mathcal{L} = \mathcal{L}_F \times \mathcal{L}_G$. For any $L = (L_F, L_G) \in \mathcal{L}$, we define by $[C[F]_K]_L$ an evaluation of $C[F]_K$ that leaks $L_F(h, M, h')$ for any evaluation of F and $L_G(K, h, h')$ for any evaluation G. To make it explicit,

- If $C = \mathit{sukMD}$, any evaluation of $[C[F]_K]_L$ leaks $L_F(h_{\ell-1}, \bar{M}_\ell \| 10^* \| K, h)$ of the last evaluation of F in Figure 2 (the function G is not used and L_G is simply ignored);
- If $C = \mathit{subMD}$, any evaluation of $[C[F]_K]_L$ leaks $L_G(K, h_{\ell-1}, G(K, h_{\ell-1}))$ and $L_F(G(K, h_{\ell-1}), M_\ell, h)$ of the last evaluations of G and F in Figure 3.

Note that these leakage functions L_F and L_G are deterministic, and whenever the same input is given, the same leakage is responded.

Now, the non-adaptive leakage resilient PRF (NALR-PRF) security of $C[F]$ against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\text{nalr-prf}}_{C[F]}(\mathcal{A}) = \max_{L \in \mathcal{L}} \mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{[C[F]_K]_L, C[F]_K}\right) - \mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{[C[F]_K]_L, R_{m',n}}\right), \quad (22)$$

where $K \xleftarrow{\$} \{0,1\}^k$ and $R_{m',n} \xleftarrow{\$} \mathsf{func}(m', n)$. The adversary is never allowed to repeat a leaky oracle query to its leak-free oracle and vice versa. We say that $C[F]$ is a *secure leakage resilient pseudorandom function* if $\mathbf{Adv}^{\text{nalr-prf}}_{C[F]}(\mathcal{A})$ is sufficiently small for any adversary $\mathcal{A}$ with a certain query complexity Q of total length S bits and time complexity T.

Likewise, the non-adaptive leakage resilient related-key PRF (NALR-RK-PRF) security of $C[F]$ against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\text{nalr-rk-prf}}_{\Phi,C[F]}(\mathcal{A}) = \mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{[C[F]_{RK(K,\cdot)}]_L, C[F]_{RK(K,\cdot)}}\right) - \mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{[C[F]_{RK(K,\cdot)}]_L, (R_{m',n})_{(\cdot)}}\right), \quad (23)$$

where $K \xleftarrow{\$} \{0,1\}^k$ and $(R_{m',n})_{(\cdot)} \xleftarrow{\$} \mathsf{func}(m', n)^{|\Phi|}$. The adversary is never allowed to repeat a leaky oracle query to its leak-free oracle and vice versa. We say that $C[F]$ is a *secure leakage resilient related-key pseudorandom function* under key derivation function set Φ if $\mathbf{Adv}^{\text{nalr-rk-prf}}_{\Phi,C[F]}(\mathcal{A})$ is sufficiently small for any adversary $\mathcal{A}$ with a certain query complexity Q of total length S bits and time complexity T.

4.2 Leak-Free Uniform and Universal Hashing

We will assume that G is not only δ-uniform and ε-universal in the native way, but also that it is leak-free, i.e., even under internal leakage. Stated differently, this definition implies that for any leakage function $L_G \in \mathcal{L}_G$, and any $X, X' \in \{0,1\}^m$ and $Y \in \{0,1\}^n$,

$$\begin{aligned}\mathbf{Pr}\left(G(K,X) = Y \mid \{L_G(K,Z,G(K,Z))\}_{Z\in\{0,1\}^m}\right) &\leq \delta\,,\\ \mathbf{Pr}\left(G(K,X) = G(K,X') \mid \{L_G(K,Z,G(K,Z))\}_{Z\in\{0,1\}^m}\right) &\leq \varepsilon\,,\end{aligned}$$

where $K \xleftarrow{\$} \{0,1\}^k$. We remark that the condition that the function must be leak-free over *any possible* early evaluation of G on Z is quite generous.

However, just like in Dobraunig and Mennink [DM19b], the situation is slightly more complex in that a surrounding function (in our case F) may leak information about the output data of G. Indeed, in our evaluations of *subMD* the secret value coming out of G is fed into F, possibly for different values of $\bar{M}_\ell$ (see Figure 3) and this affects the uniformity of G. In detail, we require that for any leakage function $L_G \in \mathcal{L}_G$, and auxiliary leakage function $L_{\mathrm{aux}}\colon \{0,1\}^n \to \{0,1\}^\mu$ (for some μ), and any $X, X' \in \{0,1\}^m$ and $Y \in \{0,1\}^n$,

$$\begin{aligned}\mathbf{Pr}\left(G(K,X) = Y \mid \{L_G(K,Z,G(K,Z))\}_{Z\in\{0,1\}^m} \wedge L_{\mathrm{aux}}(G(K,X))\right) &\leq 2^\mu\delta\,,\\ \mathbf{Pr}\left(G(K,X) = G(K,X') \mid \{L_G(K,Z,G(K,Z))\}_{Z\in\{0,1\}^m}\right) &\leq \varepsilon\,,\end{aligned}$$

where $K \xleftarrow{\$} \{0,1\}^k$. The leakage function L_{aux} is supposed to capture external leakage coming from F.

4.3 Leakage Resilience of Suffix Keyed Merkle-Damgård

It turns out that *sukMD* does not achieve leakage resilience in the security model of Section 4.1. The core reason is that a side-channel adversary can potentially use the message portion in the last padded block (i.e., $\bar{M}_\ell$ of Figure 2) to manipulate the data that L_F leaks about the secret key K. For example, even if $\lambda = 1$ we can define the evaluation

$$L_F(h_{\ell-1}, \bar{M}_\ell\|10^*\|K, h)$$

to interpret the first $\log_2(k)$ bits of $\bar{M}_\ell$ as an encoding of a key index ι and to reveal the ιth bit of K. However, even if we would adapt the padding in such a way that $\bar{M}_\ell$ is always of size 0, the values $h_{\ell-1}$ are sufficiently different and L_F can be constructed in such a way that it leaks the entire key K in $O(k)$ queries. We will describe this attack against the leakage resilience in below proposition.

Proposition 2. *Let $k, m, n \in \mathbb{N}$, and let $\lambda = 1$. Let $F : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$ be a compression function. Consider the suffix keyed Merkle-Damgård construction sukMD of (7). There exists an adversary $\mathcal{A}$ with construction query complexity $Q = \alpha k(\ln(k)+1)+1$ of total length $S \leq mn$ bits and time complexity $T = O(T_F)$, where T_F is the time to evaluate the function F, such that*

$$\mathbf{Adv}^{\text{nalr-prf}}_{sukMD}(\mathcal{A}) \geq (1 - 1/\alpha)\cdot(1 - 1/2^n)\,.$$

Proof. Define the following leakage function:

$$L_F(h_{\ell-1}, 0^{m-k}\|K, h)$$

(explicitly described in the context of *sukMD* with zero-length last message portion $\bar{M}_\ell$) to interpret the first $\log_2(k)$ bits of $h_{\ell-1}$ as an encoding of a key index ι and to reveal the ιth bit of K. The adversary now has to make different evaluations of $[F_K]_L$ for different

messages to obtain hash values $h_{\ell-1}$ to cover all possible encodings. This is the coupon collector problem with k coupons. Denoting by A the required number of attempts, we have that $\mathbf{Ex}(A) = k \cdot H(k)$ [ER61], where $H(k)$ is the kth harmonic number. Due to Markov's bound, $\alpha\mathbf{Ex}(A)$ attempts are sufficient except with probability at most $1/\alpha$. Then, after having obtained key candidate $K^\star$, a single evaluation of the challenge oracle ($sukMD_K$ or $R_{*,n}$) for a new message $M^\star$ can be made and matched with an offline evaluation of $sukMD_{K^\star}(M^\star)$. If the evaluations match, the adversary outputs 1, otherwise it outputs 0. In fact, in the ideal world they only match with probability $1/2^n$ and this yields the result. To finally bound the number of queries that $\mathcal{A}$ makes, we use that $H(k) \leq \ln(k) + 1$. □

The crucial design aspect of *sukMD* that makes the attack work is that the last evaluation of F is on input of a state value $h_{\ell-1}$ concatenated with key. The adversary can know the state value $h_{\ell-1}$ as it can compute it offline, but strictly seen this is not a necessary requirement for the attack. From this observation, we can conclude that the exact same attack would also work against the envelope construction that both prepends and appends the key to the message [Tsu92] and its related *HMAC* (see Section 1).

4.4 Leakage Resilience of Suffix Blinded Merkle-Damgård

Similar to the case of *sukMD*, in *subMD* the adversary can choose a leakage function L_F in a sufficiently smart way so that variance in M_ℓ eventually leaks all information about $G(K, h_{\ell-1})$ in $O(n)$ queries. We will describe this attack against the leakage resilience in below proposition.

Proposition 3. *Let $k, m, n \in \mathbb{N}$ and $\delta, \varepsilon \in [0,1]$, and let $\lambda, \lambda' = 1$. Let $F : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$ be a compression function and $G : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ a function that is δ-uniform and ε-universal. Consider the suffix blinded Merkle-Damgård construction subMD of (13). There exists an adversary $\mathcal{A}$ with construction query complexity $Q = n+1$ of total length $S \leq 2mn$ bits and time complexity $T = O(T_F)$, where T_F is the time to evaluate the function F, such that*

$$\mathbf{Adv}^{\text{nalr-prf}}_{subMD}(\mathcal{A}) \geq 1 - 1/2^n\,.$$

Proof. Define the following leakage function:

$$L_F(G(K, h_{\ell-1}), M_\ell, h)$$

to interpret the first $\log_2(k)$ bits of M_ℓ as an encoding of a secret state index ι and to reveal the ιth bit of $G(K, h_{\ell-1})$. After n evaluations of $[subMD_K]_L$ for messages consisting of the first $\ell - 1$ blocks but a differing M_ℓ, the adversary obtains $G(K, h_{\ell-1})$. Then, after having obtained state candidate $h^\star = G(K, h_{\ell-1})$, a single evaluation of the challenge oracle ($subMD_K$ or $R_{*,n}$) for a new message with the identical first $\ell - 1$ blocks but yet another last padded block M_ℓ can be made and matched with an offline evaluation of $F(h^\star, M_\ell)$. If the evaluations match, the adversary outputs 1, otherwise it outputs 0. In fact, in the ideal world they only match with probability $1/2^n$ and this yields the result. □

On the upside, by adapting the padding such that M_ℓ is a constant (e.g., the zero-string), the attack of Proposition 3 is mitigated and we can actually prove that *subMD* is leakage resilient under the assumption that G is leak-free (cf., Section 4.2). As a matter of fact, this change simplifies the description of (13) to the simpler

$$zsubMD(K, M) = IH\Big(G\big(K, IH(IV, \not<^{10}_m(M))\big), 0^m\Big)\,, \tag{24}$$

where the prefix z refers to the zero-padding of the last block. This is the construction we consider in below leakage resilience result.

Theorem 3. *Let $k, m, n \in \mathbb{N}$ and $\delta, \varepsilon \in [0,1]$, and let $\lambda, \lambda' \in \mathbb{N}$. Let $F : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$ be a compression function and $G : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ a* leak-free *function that is δ-uniform and ε-universal. Consider the zero-padded suffix blinded Merkle-Damgård construction zsubMD of (24). For any adversary $\mathcal{A}$ with construction query complexity Q of total length S bits and time complexity T,*

$$\mathbf{Adv}^{\text{nalr-prf}}_{zsubMD}(\mathcal{A}) \leq \mathbf{Adv}^{\text{col}}_{F}(\mathcal{B}) + \mathbf{Adv}^{\text{nalr-rk-prf}}_{\Phi_{sub}[G],C_{sub}[F]}(\mathcal{C})\,,$$

for some adversary $\mathcal{B}$ that runs in time at most $T + O(T_F S)$ and $\mathcal{C}$ that makes at most Q queries, all for the same data block 0^m, and runs in time at most $T + O(T_F S)$, where T_F is the time to evaluate the function F.

Proof. The proof is identical to that of Theorem 2, as the leakage is encapsulated within the underlying component $\mathbf{Adv}^{\text{nalr-rk-prf}}_{\Phi_{sub}[G],C_{sub}[F]}(\mathcal{C})$. □

We remark that the same reduction applies to *zsubMDP*, which would be defined as *subMDP* of (19) but with zero-padding of the last block.

Just like in the black-box setting in Section 3.2, we can derive a bound in the ideal function model to more precisely describe the impact of δ, ε, and μ. The leakage resilience security model likewise carries over to the ideal model where F is a random function and the adversary can make T queries to it:

$$\mathbf{Adv}^{\text{i-nalr-rk-prf}}_{\Phi,C[F]}(\mathcal{A}) = \mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{\left[C[F]_{RK(K,\cdot)}\right]_L, C[F]_{RK(K,\cdot)}, F}\right) - \mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{\left[C[F]_{RK(K,\cdot)}\right]_L, (R_{m',n})_{(\cdot)}, F}\right)\,, \quad (25)$$

where $F \xleftarrow{\$} \mathsf{func}(n+m, n)$, $K \xleftarrow{\$} \{0,1\}^k$, and $(R_{m',n})_{(\cdot)} \xleftarrow{\$} \mathsf{func}(m', n)^{|\Phi|}$.

Proposition 4. *Let $k, m, n \in \mathbb{N}$ and $\delta, \varepsilon \in [0,1]$, and let $\lambda, \lambda' \in \mathbb{N}$. Let $G : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ a* leak-free *function that is δ-uniform and ε-universal. For any adversary $\mathcal{C}$ with construction query complexity Q, all for the same data block 0^m, and primitive query complexity T,*

$$\mathbf{Adv}^{\text{i-nalr-rk-prf}}_{\Phi_{sub}[G],C_{sub}[F]}(\mathcal{C}) \leq 2^{\lambda} QT\delta + \binom{Q}{2}\varepsilon\,.$$

Proof. The proof is identical to that of Proposition 1, with the difference that leakage coming from F may influence the event guess. In fact, for any evaluation $G(K, X)$ in a construction query, the adversary learns at most λ bits of the resulting value through leakage from F, because $\mathcal{C}$ is restricted to a single data block 0^m, and this scales $\mathbf{Pr}(\mathsf{guess})$ by a factor 2^{λ}. □

The comments after Proposition 1 regarding the meaning of this result for actual instantiations applies here as well.

5 Conclusion

In this work, we provided a – to the best of our knowledge – first in-depth analysis of how to key the Merkle-Damgård construction at the suffix, by introducing and analyzing two constructions: suffix keyed Merkle-Damgård and suffix blinded Merkle-Damgård. In detail, we proved that both constructions are PRF secure if the underlying compression function is collision resistant and (RK-)PRF secure. We also demonstrated how these results generalize to suffix keying or suffix blinding the Merkle-Damgård with Permutation construction. We admit, though, that the constructions are vulnerable to the offline

collision attack [PvO95], something that would not apply to (also) prepending the key, as done in the envelope construction or *HMAC*.

Having said that, we did demonstrate that there is some benefit in simply suffix keying over enveloping or *HMAC*. In particular, inspired by earlier findings in permutation-based authentication [DM19b, BM24], we also investigated the resistance of these constructions against side-channel adversaries, and concluded that with an appropriate padding, suffix blinded Merkle-Damgård achieves leakage resilience, similarly to how *SuKS* did [DM19b] and unlike to suffix keyed Merkle-Damgård, enveloping, or *HMAC*. However, this result on adapted suffix blinded Merkle-Damgård requires the key blinding function G to be leak-free. In their leakage resilience analysis of *SuKS*, the authors suggested [DM19b, Section 6.2] to instantiate the function G with a leakage resilient duplex construction [DM19a], an approach that was also adopted by the NIST lightweight competition finalist ISAP v2 [DEM+17, DEM+20, DEM+21]. This would be an illogical choice for G in the context of the suffix blinded Merkle-Damgård construction, because one could then better resort to sponge-based message authentication at once. A more logical choice for G in suffix blinded Merkle-Damgård would be to base it on *2PRG* [YSPY10], though schemes with weaker security may work as well. Recall that, if leakage resilience is no concern, G can be a simple XOR (cf., Section 2.1).

We remark that our constructions are defined for the case that $k \leq \min\{m, n\}$ only. Note that the assumption $k \leq n$ is reasonable as we typically get birthday bound security in n anyway. Furthermore, for typical hash functions such as the SHA-2 family [Nat15a], $n \leq m$ (see footnote 6). That said, in the theoretical case that the key would be absorbed in multiple rounds (i.e., $m < k$), the subtleties as outlined in the first paragraph of Section 3.1 may apply: a straightforward bounding would result in a PRF security term of the underlying compression function with m-bit key, and one can only get k-bit security by grouping adjacent PRFs or by performing an ideal model proof as described in footnote 5. A comparable issue was seen in the analysis of HMAC as a dual-PRF [BBGS23], where they key is cut into pieces and the subsequent PRFs are treated separately.

We finally conclude by stressing that we *would not advocate* for the use of plain Merkle-Damgård-based designs for message authentication: instead, it is more advisable to use a hash function whose mode is indifferentiable from a random oracle, such as the Merkle-Damgård with permutation (*MDP*) construction [HPY07, Hir21] or the sponge construction [BDPV07, BDPV11a].

Acknowledgments

We thank the reviewers of FSE 2025 for their valuable feedback and suggestions to improve the work. Bart Mennink is supported by the Netherlands Organisation for Scientific Research (NWO) under grant VI.Vidi.203.099.

References

[AB00] Michel Abdalla and Mihir Bellare. Increasing the Lifetime of a Key: A Comparative Analysis of the Security of Re-keying Techniques. In Tatsuaki Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, volume 1976 of *Lecture Notes in Computer Science*, pages 546–559. Springer, 2000.

[AMPS12] Elena Andreeva, Bart Mennink, Bart Preneel, and Marjan Skrobot. Security Analysis and Comparison of the SHA-3 Finalists BLAKE, Grøstl, JH, Keccak, and Skein. In Aikaterini Mitrokotsa and Serge Vaudenay, editors, *Progress*

in Cryptology - AFRICACRYPT 2012 - 5th International Conference on Cryptology in Africa, Ifrance, Morocco, July 10-12, 2012. Proceedings, volume 7374 of *Lecture Notes in Computer Science*, pages 287–305. Springer, 2012.

[BBGS23] Matilda Backendal, Mihir Bellare, Felix Günther, and Matteo Scarlata. When Messages Are Keys: Is HMAC a Dual-PRF? In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 661–693. Springer, 2023.

[BCK96] 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.

[BDPA08] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the Indifferentiability of the Sponge Construction. In Nigel P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 181–197. Springer, 2008.

[BDPA11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers*, volume 7118 of *Lecture Notes in Computer Science*, pages 320–337. Springer, 2011.

[BDPV07] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge functions. Ecrypt Hash Workshop 2007, May 2007.

[BDPV11a] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Cryptographic sponge functions, January 2011.

[BDPV11b] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. The Keccak reference, January 2011.

[BDPV12] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Permutation-based encryption, authentication and authenticated encryption. Directions in Authenticated Ciphers, July 2012.

[Bel06] 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.

[BGP+20] Francesco Berti, Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. TEDT, a Leakage-Resist AEAD Mode for High Physical Security Applications. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(1):256–320, 2020.

[BGP+23] Francesco Berti, Chun Guo, Thomas Peters, Yaobin Shen, and François-Xavier Standaert. Secure Message Authentication in the Presence of Leakage and Faults. *IACR Trans. Symmetric Cryptol.*, 2023(1):288–315, 2023.

[BK03] Mihir Bellare and Tadayoshi Kohno. A Theoretical Treatment of Related-Key Attacks: RKA-PRPs, RKA-PRFs, and Applications. In Eli Biham, editor, *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, volume 2656 of *Lecture Notes in Computer Science*, pages 491–506. Springer, 2003.

[BKP+16] Francesco Berti, François Koeune, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Leakage-Resilient and Misuse-Resistant Authenticated Encryption. Cryptology ePrint Archive, Paper 2016/996, 2016.

[BKP+18] Francesco Berti, François Koeune, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Ciphertext Integrity with Misuse and Leakage: Definition and Efficient Constructions with Symmetric Primitives. In Jong Kim, Gail-Joon Ahn, Seungjoo Kim, Yongdae Kim, Javier López, and Taesoo Kim, editors, *Proceedings of the 2018 on Asia Conference on Computer and Communications Security, AsiaCCS 2018, Incheon, Republic of Korea, June 04-08, 2018*, pages 37–50. ACM, 2018.

[BM24] Henk Berendsen and Bart Mennink. Tightening Leakage Resilience of the Suffix Keyed Sponge. *IACR Trans. Symmetric Cryptol.*, 2024(1):459–496, 2024.

[BMOS17] Guy Barwell, Daniel P. Martin, Elisabeth Oswald, and Martijn Stam. Authenticated Encryption in the Face of Protocol and Side Channel Leakage. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 693–723. Springer, 2017.

[Bor01] Johan Borst. *Block Ciphers: Design, Analysis, and Side-channel Analysis*. PhD thesis, Departement Elektrotechniek — ESAT/COSIC, Katholieke Universiteit Leuven, Leuven, Belgium, September 2001.

[BPPS17] Francesco Berti, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. On Leakage-Resilient Authenticated Encryption with Decryption Leakages. *IACR Trans. Symmetric Cryptol.*, 2017(3):271–293, 2017.

[BR93] Mihir Bellare and Phillip Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993*, pages 62–73. ACM, 1993.

[BR06] Mihir Bellare and Phillip Rogaway. The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer, 2006.

[BRS02] John Black, Phillip Rogaway, and Thomas Shrimpton. Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*, pages 320–335. Springer, 2002.

[BRSS10] John Black, Phillip Rogaway, Thomas Shrimpton, and Martijn Stam. An Analysis of the Blockcipher-Based Hash Functions from PGV. *J. Cryptol.*, 23(4):519–545, 2010.

[CDMP05] Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-Damgård Revisited: How to Construct a Hash Function. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 430–448. Springer, 2005.

[Dam89] Ivan Damgård. A Design Principle for Hash Functions. 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 416–427. Springer, 1989.

[DEM+17] Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, and Thomas Unterluggauer. ISAP - Towards Side-Channel Secure Authenticated Encryption. *IACR Trans. Symmetric Cryptol.*, 2017(1):80–105, 2017.

[DEM+20] Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, Bart Mennink, Robert Primas, and Thomas Unterluggauer. Isap v2.0. *IACR Trans. Symmetric Cryptol.*, 2020(S1):390–416, 2020.

[DEM+21] Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, Bart Mennink, Robert Primas, and Thomas Unterluggauer. ISAP v2. Final Round Submission to NIST Lightweight Cryptography, 2021.

[DEMM14] Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, and Florian Mendel. On the Security of Fresh Re-keying to Counteract Side-Channel and Fault Attacks. In Marc Joye and Amir Moradi, editors, *Smart Card Research and Advanced Applications - 13th International Conference, CARDIS 2014, Paris, France, November 5-7, 2014. Revised Selected Papers*, volume 8968 of *Lecture Notes in Computer Science*, pages 233–244. Springer, 2014.

[DKM+15] Christoph Dobraunig, François Koeune, Stefan Mangard, Florian Mendel, and François-Xavier Standaert. Towards Fresh and Hybrid Re-Keying Schemes with Beyond Birthday Security. In Naofumi Homma and Marcel Medwed, editors, *Smart Card Research and Advanced Applications - 14th International Conference, CARDIS 2015, Bochum, Germany, November 4-6, 2015. Revised Selected Papers*, volume 9514 of *Lecture Notes in Computer Science*, pages 225–241. Springer, 2015.

[DM19a] Christoph Dobraunig and Bart Mennink. Leakage Resilience of the Duplex Construction. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 225–255. Springer, 2019.

[DM19b] Christoph Dobraunig and Bart Mennink. Security of the Suffix Keyed Sponge. *IACR Trans. Symmetric Cryptol.*, 2019(4):223–248, 2019.

[DMA17] Joan Daemen, Bart Mennink, and Gilles Van Assche. Full-State Keyed Duplex with Built-In Multi-user Support. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 606–637. Springer, 2017.

[DP10] Yevgeniy Dodis and Krzysztof Pietrzak. Leakage-Resilient Pseudorandom Functions and Side-Channel Attacks on Feistel Networks. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 21–40. Springer, 2010.

[ER61] Paul Erdös and Alfréd Rényi. On a classical problem of probability theory. *Publ. Math. Inst. Hung. Acad. Sci., Ser. A*, 6:215–220, 1961.

[FPS12] Sebastian Faust, Krzysztof Pietrzak, and Joachim Schipper. Practical Leakage-Resilient Symmetric Cryptography. In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, volume 7428 of *Lecture Notes in Computer Science*, pages 213–232. Springer, 2012.

[GPPS20] Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Towards Low-Energy Leakage-Resistant Authenticated Encryption from the Duplex Sponge Construction. *IACR Trans. Symmetric Cryptol.*, 2020(1):6–42, 2020.

[GSWY19] Chun Guo, François-Xavier Standaert, Weijia Wang, and Yu Yu. Efficient Side-Channel Secure Message Authentication with Better Bounds. *IACR Trans. Symmetric Cryptol.*, 2019(4):23–53, 2019.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A Pseudorandom Generator from any One-way Function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[Hir21] Shoichi Hirose. Collision-Resistant and Pseudorandom Function Based on Merkle-Damgård Hash Function. In Jong Hwan Park and Seung-Hyun Seo, editors, *Information Security and Cryptology - ICISC 2021 - 24th International Conference, Seoul, South Korea, December 1-3, 2021, Revised Selected Papers*, volume 13218 of *Lecture Notes in Computer Science*, pages 325–338. Springer, 2021.

[HLR07] Chun-Yuan Hsiao, Chi-Jen Lu, and Leonid Reyzin. Conditional Computational Entropy, or Toward Separating Pseudoentropy from Compressibility. In Moni Naor, editor, *Advances in Cryptology - EUROCRYPT 2007, 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, Spain, May 20-24, 2007, Proceedings*, volume 4515 of *Lecture Notes in Computer Science*, pages 169–186. Springer, 2007.

[HPY07] Shoichi Hirose, Je Hong Park, and Aaram Yun. A Simple Variant of the Merkle-Damgård Scheme with a Permutation. In Kaoru Kurosawa, editor, *Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on*

the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings, volume 4833 of *Lecture Notes in Computer Science*, pages 113–129. Springer, 2007.

[Joh16] John Kelsey, Shu-jen Chang, Ray Perlner. NIST Special Publication 800-185: SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash, and ParallelHash, December 2016.

[KBC97] Hugo Krawczyk, Mihir Bellare, and Ran Canetti. HMAC: Keyed-Hashing for Message Authentication. Request for Comments (RFC) 2104, February 1997.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[Koc96] Paul C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. 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 104–113. Springer, 1996.

[KR95] Burt Kaliski and Matt Robshaw. Message Authentication with MD5. *CryptoBytes*, 1(1):5–8, 1995.

[KR19] Yael Tauman Kalai and Leonid Reyzin. A Survey of Leakage-Resilient Cryptography. Cryptology ePrint Archive, Paper 2019/302, 2019.

[Men18] Bart Mennink. Key Prediction Security of Keyed Sponges. *IACR Trans. Symmetric Cryptol.*, 2018(4):128–149, 2018.

[Men20] Bart Mennink. Beyond Birthday Bound Secure Fresh Rekeying: Application to Authenticated Encryption. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 630–661. Springer, 2020.

[Men23] Bart Mennink. Understanding the Duplex and Its Security. *IACR Trans. Symmetric Cryptol.*, 2023(2):1–46, 2023.

[Mer89] 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.

[MOI90] Shoji Miyaguchi, Kazuo Ohta, and Masahiko Iwata. Confirmation that Some Hash Functions Are Not Collision Free. In Ivan Damgård, editor, *Advances in Cryptology - EUROCRYPT '90, Workshop on the Theory and Application of of Cryptographic Techniques, Aarhus, Denmark, May 21-24, 1990, Proceedings*, volume 473 of *Lecture Notes in Computer Science*, pages 326–343. Springer, 1990.

[MPR+11] Marcel Medwed, Christophe Petit, Francesco Regazzoni, Mathieu Renauld, and François-Xavier Standaert. Fresh Re-keying II: Securing Multiple Parties against Side-Channel and Fault Attacks. In Emmanuel Prouff, editor, *Smart Card Research and Advanced Applications - 10th IFIP WG 8.8/11.2 International Conference, CARDIS 2011, Leuven, Belgium, September 14-16, 2011, Revised Selected Papers*, volume 7079 of *Lecture Notes in Computer Science*, pages 115–132. Springer, 2011.

[MRH04] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In Moni Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2004.

[MRV15] Bart Mennink, Reza Reyhanitabar, and Damian Vizár. Security of Full-State Keyed Sponge and Duplex: Applications to Authenticated Encryption. In Tetsu Iwata and Jung Hee Cheon, editors, *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*, volume 9453 of *Lecture Notes in Computer Science*, pages 465–489. Springer, 2015.

[MSGR10] Marcel Medwed, François-Xavier Standaert, Johann Großschädl, and Francesco Regazzoni. Fresh Re-keying: Security against Side-Channel and Fault Attacks for Low-Cost Devices. In Daniel J. Bernstein and Tanja Lange, editors, *Progress in Cryptology - AFRICACRYPT 2010, Third International Conference on Cryptology in Africa, Stellenbosch, South Africa, May 3-6, 2010. Proceedings*, volume 6055 of *Lecture Notes in Computer Science*, pages 279–296. Springer, 2010.

[Nat08] National Institute of Standards and Technology. FIPS 198-1: The Keyed-Hash Message Authentication Code (HMAC), July 2008.

[Nat15a] National Institute of Standards and Technology. FIPS 180-4: Secure Hash Standard (SHS), August 2015.

[Nat15b] National Institute of Standards and Technology. FIPS PUB 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, August 2015.

[PGV93] Bart Preneel, René Govaerts, and Joos Vandewalle. Hash Functions Based on Block Ciphers: A Synthetic Approach. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 368–378. Springer, 1993.

[Pie09] Krzysztof Pietrzak. A Leakage-Resilient Mode of Operation. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 462–482. Springer, 2009.

[PSV15] Olivier Pereira, François-Xavier Standaert, and Srinivas Vivek. Leakage-Resilient Authentication and Encryption from Symmetric Cryptographic

Primitives. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, pages 96–108. ACM, 2015.

[PvO95] Bart Preneel and Paul C. van Oorschot. MDx-MAC and Building Fast MACs from Hash Functions. In Don Coppersmith, editor, *Advances in Cryptology - CRYPTO '95, 15th Annual International Cryptology Conference, Santa Barbara, California, USA, August 27-31, 1995, Proceedings*, volume 963 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 1995.

[Rog06] 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.

[RS04] Phillip Rogaway and Thomas Shrimpton. Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. In Bimal K. Roy and Willi Meier, editors, *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers*, volume 3017 of *Lecture Notes in Computer Science*, pages 371–388. Springer, 2004.

[RSS11] Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton. Careful with Composition: Limitations of the Indifferentiability Framework. In Kenneth G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*, pages 487–506. Springer, 2011.

[SPY+10] François-Xavier Standaert, Olivier Pereira, Yu Yu, Jean-Jacques Quisquater, Moti Yung, and Elisabeth Oswald. Leakage Resilient Cryptography in Practice. In Ahmad-Reza Sadeghi and David Naccache, editors, *Towards Hardware-Intrinsic Security - Foundations and Practice*, Information Security and Cryptography, pages 99–134. Springer, 2010.

[Tsu92] Gene Tsudik. Message authentication with one-way hash functions. *Comput. Commun. Rev.*, 22(5):29–38, 1992.

[Yas07a] Kan Yasuda. Boosting Merkle-Damgård Hashing for Message Authentication. In Kaoru Kurosawa, editor, *Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings*, volume 4833 of *Lecture Notes in Computer Science*, pages 216–231. Springer, 2007.

[Yas07b] 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.

[YSPY10] Yu Yu, François-Xavier Standaert, Olivier Pereira, and Moti Yung. Practical leakage-resilient pseudorandom generators. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *Proceedings of the 17th ACM*

Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010, pages 141–151. ACM, 2010.