Security Analysis of NIST Key Derivation Using Pseudorandom Functions

Yaobin Shen[1,2], Lei Wang[3(✉)], and Dawu Gu[3(✉)]

[1] School of Informatics, Xiamen University, Xiamen, China
[2] State Key Laboratory of Cryptology, P. O. Box 5159, Beijing, 100878, China
yaobin.shen@xmu.edu.cn
[3] Shanghai Jiao Tong University, Shanghai, China
wanglei_hb@sjtu.edu.cn, dwgu@sjtu.edu.cn

Abstract. Key derivation functions can be used to derive variable-length random strings that serve as cryptographic keys. They are integral to many widely-used communication protocols such as TLS, IPsec and Signal. NIST SP 800-108 specifies several key derivation functions based on pseudorandom functions such as CMAC and HMAC, that can be used to derive additional keys from an existing cryptographic key. This standard either explicitly or implicitly requests their KDFs to be variable output length pseudorandom function, collision resistant, and preimage resistant, which are also demanded by practical applications. Yet, since the publication of this standard dating back to the year of 2008, until now, there is no formal analysis to justify these security properties of KDFs.

In this work, we give the formal security analysis of key derivation functions in NIST SP 800-108. We show both positive and negative results regarding these key derivation functions. For KCTR-CMAC, KFB-CMAC, and KDPL-CMAC that are key derivation functions based on CMAC in counter mode, feedback mode, and double-pipeline mode respectively, we prove that all of them are secure variable output length pseudorandom functions and preimage resistant. We show that KFB-CMAC and KDPL-CMAC are collision resistant. While for KCTR-CMAC, we can mount constant-time collision attack against it. For KCTR-HMAC, KFB-HMAC, and KDPL-HMAC that are key derivation functions based on HMAC in modes, we show that all of them behave like variable output length pseudorandom functions. When the key of these key derivation functions is of variable length, they suffer from collision attacks. For the case when the key of these key derivation function is of fixed length and less than $d - 1$ bits where d is the input block size of the underlying compression function, we can prove that they are collision resistant and preimage resistant. Finally, we extend our analysis to the plain CMAC-based KDFs for which mitigation techniques against key-control attacks do not apply, as well as to the KMAC-based KDF.

Keywords: Key derivation functions · CMAC · HMAC · KMAC · Pseudorandom function

1 Introduction

Key derivation functions (KDFs) play an important role in many cryptographic protocols. They can be used to derive arbitrary amount of cryptographic keys from either a raw source containing some good amount of randomness but not necessarily uniform or a fixed-length cryptographic key. KDFs are integral to many widely-used communication protocols such as TLS, IPsec, or Signal to derive keys. There are two common methods to build a key derivation function. The first one follows a two-step paradigm called extract-then-expand as formalized by Krawczyk [27] and specified in NIST SP 800-56C [4]. The first step extracts a fixed-length pseudorandom key from a source that is entropic but not necessarily uniform. The second step takes this fixed-length key and expands it to generate a key of variable length. The second method simply expands a fixed-length key to a variable-length key. This method can be used to derive additional keys from an existing cryptographic key. Following the second method, NIST SP 800-108 [21] specifies several KDFs based on pseudorandom functions including HMAC [6], CMAC [25,24], and KMAC [26]. KDFs specified in NIST SP 800-108 can not only be used as an expansion component in a two-step key-derivation method, but can also be solely used to derive keys as general KDFs. Yet, until now, these KDFs lack a formal security analysis as explained below.

History of NIST SP 800-108. KDFs in NIST SP 800-108 mainly uses the following procedure to derive variable-length keying material. Given a fixed-length K and a message M that contains the information of the derived keying material, the key derivation function $\mathsf{KDF}(K, M)$ outputs a random string C of the desired length. The first version of NIST SP 100-108 was published in 2008 [19]. In this version, it specified three modes to build a KDF based on a pseudorandom function, including KDF in counter mode, KDF in feedback mode, and KDF in double-pipeline mode. The reason for specifying these modes is that the output of a pseudorandom function is typically of fixed length and cannot be solely used to generate variable-length strings. It used HMAC and CMAC as the underlying pseudorandom function in these modes. In the second version that is named as NIST SP 800-108r1 and published in 2022 [20], a KDF using KMAC [26] was included. Compared to HMAC and CMAC, KMAC is not equipped with any modes as it is based on a sponge function and its output can be of variable length. However, as acknowledged in [21, Appendix B], Arciszewski et al. revealed a serious key control security issue regarding KDFs based on CMAC in these three modes. Namely, for any n-bit value C, an entity that knows the key K can easily finds a message M such that $\mathsf{KDF}(K, M) = C$. Hence, this entity can force the output keying material to a pre-selected value as she wants. Obviously, the key control security can also be captured as the preimage resistance of a hash function. As a response to this key control attack, an updated version NIST SP 800-108r1-upd1 was published in 2024 [21]. In this updated version, specifications of KDFs based on CMAC in these three modes are revised to implement mitigation techniques against this key-control attack, while other KDFs based on different pseudorandom functions remain the same

as they do not suffer from this key-control attack. The key control security is explicitly written as a desirable security feature in this standard[4]. Yet, whether the revised version of KDFs based on CMAC or the other KDFs in this standard can provably achieve key control security remains open as currently we only know there are no existing attacks.

Recently, Bhaumik et al. [13] found that the context binding security requested by NIST SP 800-108r1-upd1[5] can be formalized as the context-committing security for MAC that is essentially the same as the collision resistance of a hash function, and they show that CMAC is not context-committing secure. However, the context binding security of these KDFs still remain open, as neither the context-committing attack against CMAC works for KDFs based on CMAC in these three modes nor it says anything about the context binding security of other KDFs that are based on different pseudorandom functions.

Hence, so far we can see, besides the basic security property that a KDF should behave like a variable output length pseudorandom function (volPRF), it is also desirable that a KDF should have key control security and context binding security, where the former can be captured as preimage resistance and the later can be captured as collision resistance (we use these two notions in the rest of this paper). The preimage resistance and collision resistance of KDFs are demanded in several important applications in practice, including TLS 1.3, IPSec, and key management service. In NIST standard reviewing project, the cryptography expert Mattsson from Ericsson Research [28] explicitly suggested that collision resistance and preimage resistance should be requirements for a key derivation function [28].

However, until now, there is no formal security analysis of NIST SP 800-108r1-upd1, even for the basic security as a variable output length pseudorandom function, despite of the fact that KDFs are widely-used. We have no idea whether KDFs in this standard can achieve the volPRF security, or being collision resistant, or being preimage resistant.

Our contributions. In this paper, we give formal security analysis of NIST SP 800-108r1-upd1. We show both positive and negative results for the security of KDFs in this standard. A summary of our results is presented in Fig. 1. We mainly focus on KDFs based on fixed-output length pseudorandom functions equipped with modes to generate variable-output length strings, including KCTR-CMAC, KFB-CMAC, and KDPL-CMAC which are KDFs based on CMAC in counter mode, feedback mode, and double-pipeline mode respectively, and KCTR-HMAC, KFB-HMAC, and KDPL-HMAC which are KDFs based on HMAC in modes. As mentioned above, to resist against key-control attack, there are

[4] The key control security provides assurance that "(even with knowledge of the input key K) no single party can manipulate the process in such a way as to force output keying material to a preselected value" [21, Section 6.7].

[5] The context binding provides "assurance that all parties who (correctly) derive the keying material share the same understanding of who will access it and in which session it will be used. If those parties have different understandings, then they will derive different keying material" [21, Section 6.6].

Scheme	volPRF	Collision	Preimage	Analysis
KCTR-CMAC	$O(\frac{q^2b^2}{2^n} + \frac{qb\ell^2}{2^n})$	no	$O(\frac{p}{2^n} + \frac{p^2\ell}{2^n})$	Sec. 3.1
KCTR-HMAC	$O(\frac{q^2b^2}{2^n})$	$O(\frac{p^2}{2^n})$	$O(\frac{p^2}{2^n})$	Sec. 3.2
pKCTR-CMAC	$O(\frac{q^2b^2}{2^n} + \frac{qb\ell^2}{2^n})$	no	no	Sec. 3.3 & [21]
KFB-CMAC	$O(\frac{q^2b^2}{2^n} + \frac{qb\ell^2}{2^n})$	$O(\frac{p^2}{2^n} + \frac{p^2\ell}{2^n})$	$O(\frac{p}{2^n} + \frac{p^2\ell}{2^n})$	Sec. 4.1
KFB-HMAC	$O(\frac{q^2b^2}{2^n} + \frac{qb^2}{2^n})$	$O(\frac{p^2}{2^n})$	$O(\frac{p^2}{2^n})$	Sec. 4.2
pKFB-CMAC	$O(\frac{q^2b^2}{2^n} + \frac{qb\ell^2}{2^n})$	no	no	Sec. 4.3 & [21]
KDPL-CMAC	$O(\frac{q^2b^2}{2^n} + \frac{qb\ell^2}{2^n})$	$O(\frac{p^2}{2^n} + \frac{p^2\ell}{2^n})$	$O(\frac{p}{2^n} + \frac{p^2\ell}{2^n})$	Sec. 5.1
KDPL-HMAC	$O(\frac{q^2b^2}{2^n} + \frac{qb^2}{2^n})$	$O(\frac{p^2}{2^n})$	$O(\frac{p^2}{2^n})$	Sec. 5.2
pKDPL-CMAC	$O(\frac{q^2b^2}{2^n} + \frac{qb\ell^2}{2^n})$	$O(\frac{p^2}{2^n} + \frac{p^2\ell}{2^n})$	$O(\frac{p}{2^n} + \frac{p^2\ell}{2^n})$	Sec. 5.3
KDF-KMAC	$O(\frac{pq+q^2}{2^c} + \frac{\sigma^2}{2^n} + \frac{\sigma q}{2^{c+k}} + \frac{pq}{2^k})$	$O(\frac{p^2}{2^c})$	$O(\frac{p^2}{2^c})$	Sec. 6

Fig. 1: Summary of NIST SP 800-108r1-upd1 KDFs security. Here q denotes the number of queries to a key derivation function, with each query being of block length at most ℓ and the total number of blocks among these queries being at most σ, b the output block length of a key derivation function, p the number of queries to the underlying primitive, n the block size of the underlying primitive, c the capacity, k the key length. The collision resistance and preimage resistance bounds of KDFs based on HMAC only hold when the key is of fixed length and less than $d-1$ bits.

subtle differences in the specification between KDFs based on CMAC and KDFs based on HMAC, even in the case where they are equipped with the same mode. Hence, it requires to perform independent security analysis for these KDFs based on CMAC and HMAC.

For KDFs based on CMAC, there are a total of three different KDFs depending on the modes, including KCTR-CMAC, KFB-CMAC, and KDPL-CMAC. We prove that all of these three KDFs are a secure volPRF with the bound $O(\frac{q^2b^2}{2^n} + \frac{qb\ell^2}{2^n})$ by reducing the volPRF security of these three KDFs to the PRF security of the underlying CMAC, where n is the block size of the underlying block cipher, q the number of queries to the key derivation function with each query being of block length at most ℓ, b the output block length of the key derivation function. While CMAC itself is not collision resistant and preimage resistant, we prove that after being properly instantiated in modes, the resulting KDFs can be collision resistant and preimage resistant. Specially, we prove that all of these three KDFs based on CMAC are preimage resistant with the bound $O(\frac{p}{2^n} + \frac{p^2\ell}{2^n})$, where p denotes the number of queries to the underlying block cipher. We prove that KFB-CMAC and KDPL-CMAC are collision resistant with the bound $O(\frac{p^2}{2^n} + \frac{p^2\ell}{2^n})$. However, for KCTR-CMAC that is the KDF based on CMAC in counter mode, we mount constant-time collision attack against it. Thus, it shows distinction

that KCTR-CMAC does not meet the collision resistance as required by this NIST standard and may not be used in scenarios where collision resistance is demanded.

For KDFs based on HMAC, we prove that all of KCTR-HMAC, KFB-HMAC, and KDPL-HMAC are a secure volPRF with the bound around $O(\frac{q^2b^2}{2^n} + \frac{qb^2}{2^n})$, where n is the output block size of the underlying compression function, q the number of queries to the key derivation function, b the output block length of the key derivation function. Proofs are done by reducing the volPRF security of KDFs to the PRF security of their underlying HMAC. Regarding collision resistance, we exhibit both negative and positive results. When the key of these KDFs is of variable length, we show collision attack against these KDFs that only requires two queries but can succeed with probability 1. This attack is due to the weak key pairs of HMAC when its key allows to have variable length that have been highlighted in previous attacks [23,35,13]. This sets an alert for these KDFs as the key of variable length is allowed in this standard (see [21, Section 6.2]). While for the case when the key is of fixed length and less than $d-1$ bits where d is the size of input block of underlying compression function, we show that these KDFs are collision resistant and preimage resistant with the bound $O(\frac{p^2}{2^n})$, where p denotes the number of queries to the underlying compression function.

We also extend our analysis to the plain CMAC-based KDFs for which the mitigations against key-control attacks proposed in the updated version of [21] do not apply, namely pKCTR-CMAC, pKFB-CMAC, and pKDPL-CMAC, as well as to KDF-KMAC, a KMAC-based KDF where KMAC is a variable-output-length keyed hash function. The results of this analysis are also presented in Fig. 1. We show that these plain CMAC-based KDFs achieve the same volPRF security as their corresponding enhanced variants. Interestingly, although pKCTR-CMAC and pKFB-CMAC are neither preimage resistant nor collision resistant via the key-control attack of [21, Appendix B], pKDPL-CMAC retains the same collision resistance and preimage resistance as its enhanced version. This demonstrates that plain pKDPL-CMAC can achieve preimage resistance and collision resistance even without the proposed mitigation. In addition, we prove that KDF-KMAC is a secure volPRF and is both collision resistant and preimage resistant. Our results for KDF-KMAC follow by leveraging prior PRF-security analyses of keyed sponge constructions [22,29] together with the indifferentiability of the sponge construction [10].

Concurrent and related work. Bhaumik et al. [14,15] concurrently and independently studied key-control security for NIST SP 800-108. Their work primarily focuses on key-control security and introduces a generalized definition in which the adversary may specify a set of target images rather than a single one, potentially of exponential size. In this sense, their notion is more general than preimage resistance, and accordingly their key-control analysis is more extensive than ours. We refer to [14,15] for a detailed discussion of related notions and for more real-world applications of NIST SP 800-108.

That said, Bhaumik et al. only present birthday-complexity attacks against CMAC-based KDFs, whereas we provide preimage-resistance proofs for CMAC-based KDFs. The two sets of results are therefore complementary and may be relevant in broader use cases. We also note that our bounds are stated for an n-bit target image; when the derived keying material has length exceeding n bits, the same bounds implicitly cover a set of images whose first n bits are fixed to the target value, which can likewise be exponentially large. Finally, beyond key-control security, we give formal analyses for volPRF security and collision resistance, which are also required by the NIST standard.

Organization. We introduce relevant notations and security notions in Section 2. We give security analysis of KDFs based on CMAC and HMAC in counter mode in Section 3. We give security analysis of KDFs based on CMAC and HMAC in feedback mode in Section 4. We present security analysis of KDFs based on CMAC and HMAC in double-pipeline mode in Section 5. We present security analysis of KDF based on KMAC in Section 6. Finally, we conclude the paper in Section 7.

2 Preliminaries

Notation. Let ε denote the empty string. Let $\{0,1\}^n$ be the set of all n-bit strings and $\{0,1\}^*$ be the set of all finite bit strings including the empty string ε. For a finite set $\mathcal{X}$, let $X \leftarrow\!\!\$\, \mathcal{X}$ denote the uniform sampling from $\mathcal{X}$ and assigning the value to X. Let $|X|$ denote the length of string X. Let $X[i]$ denote the i-th bit of string X, and let $X[i:j]$ denote the substring of X from the i-th bit to the j-th bit both inclusively for $i \leq j$. Let $[i]_2$ denote the binary encoding of an integer i. Concatenation of strings X and Y is written as $X \,\|\, Y$ or simply XY. $X10^*$ denotes the padding that first appended with a single 1 and follow by as few 0 bits so that the length of string to be a multiple of n. Let $Y \leftarrow \mathcal{A}(X_1, \ldots; r)$ denote running algorithm $\mathcal{A}$ with randomness r on inputs $X_1, \ldots$ and assigning the output to Y. We let $Y \leftarrow\!\!\$\, \mathcal{A}(X_1, \ldots)$ be the result of picking r at random and letting $Y \leftarrow \mathcal{A}(X_1, \ldots; r)$. Let $\mathrm{Perm}(n)$ denote the set of all permutations over $\{0,1\}^n$. Let $\mathrm{Func}(\mathcal{X}, \mathcal{Y})$ denote the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$.

PRP security. An adversary $\mathcal{A}$ is a probabilistic algorithm with access to one or more oracles. Let $\mathcal{A}^{O_1, O_2, \cdots} = 1$ denote the event that an adversary $\mathcal{A}$ outputs 1 after interacting with oracles $O_1, O_2, \ldots$. A block cipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ is a family of permutations over $\{0,1\}^n$ indexed by a key $K \in \{0,1\}^k$. We write $E_K(\cdot)$ or $E(K, \cdot)$ as a permutation over $\{0,1\}^n$ for a key $K \in \{0,1\}^k$. We denote by $E_K^{-1}(\cdot)$ the inverse of $E_K(\cdot)$ for a fixed key K. Let $\pi \leftarrow\!\!\$\, \mathrm{Perm}(n)$ be a random permutation. The advantage of $\mathcal{A}$ against the pseudorandom permutation (PRP) security of E is defined as

$$\mathsf{Adv}_E^{\mathrm{prp}} = \left| \Pr\left[\mathcal{A}^{E_K(\cdot)} = 1 \right] - \Pr\left[\mathcal{A}^{\pi(\cdot)} = 1 \right] \right| \ ,$$

where K is selected uniformly at random from $\{0,1\}^k$.

Variable output length PRF security. Let $F : \mathcal{K} \times \{0,1\}^* \times \mathbb{N} \to \{0,1\}^*$ be a keyed function whose output length is specified by the third input parameter. For example, for the input tuple (K, M, L) where $K \in \mathcal{K}, M \in \{0,1\}^*$, and $L \in \mathbb{N}$, the function $F_K(M, L)$ outputs a string $Y \in \{0,1\}^L$. Let $\mathrm{Func}(\{0,1\}^* \times \mathbb{N}, \{0,1\}^*)$ denote the set of all functions from $\{0,1\}^* \times \mathbb{N}$ to $\{0,1\}^*$ whose output length is specified by the second parameter in the set $\mathbb{N}$. Let $\mathcal{R} \leftarrow\!\!\$\, \mathrm{Func}(\{0,1\}^* \times \mathbb{N}, \{0,1\}^*)$ be a random function of variable output length whose output length is specified by the second parameter. The advantage of the adversary $\mathcal{A}$ against the variable output length PRF (volPRF) security of F is defined as

$$\mathsf{Adv}_F^{\mathrm{volprf}}(\mathcal{A}) = \left| \Pr\left[\mathcal{A}^{F_K(\cdot,\cdot)} = 1 \right] - \Pr\left[\mathcal{A}^{\mathcal{R}(\cdot,\cdot)} = 1 \right] \right| \ ,$$

where K is chosen uniformly at random from $\mathcal{K}$.

Collision resistance and preimage resistance. Let $H : \mathcal{M} \to \mathcal{Y}$ be a hash function. We consider collision resistance and everywhere preimage resistance [34] as follows.

Definition 1. *For any adversary $\mathcal{A}$, we define*

$$\mathsf{Adv}_H^{\mathsf{coll}}(\mathcal{A}) = \Pr\left[(M, M') \leftarrow\!\!\$\, \mathcal{A} : M \neq M' \wedge H(M) = H(M') \right] \ .$$

Definition 2. *For any adversary $\mathcal{A}$, we define*

$$\mathsf{Adv}_H^{\mathsf{epre}}(\mathcal{A}) = \max_{Y \in \mathcal{Y}} \Pr\left[M \leftarrow\!\!\$\, \mathcal{A} : H(M) = Y \right] \ .$$

Note that in the rest of this paper, the hash function is either based on a block cipher or a compression function that will be modeled as an ideal cipher or a random oracle when considering collision resistance and preimage resistance. Hence, the randomness of the probability in definitions of collision resistance and preimage resistance comes from the ideal cipher or the random oracle.

3 Security Analysis of KDF in Counter Mode

3.1 Security Analysis of KCTR-CMAC

In this subsection, we present security analysis for the key derivation function KCTR-CMAC based on CMAC in counter mode. We first prove that KCTR-CMAC is a secure volPRF with the bound $O(\frac{q^2b^2}{2^n} + \frac{qb\ell^2}{2^n})$. We then show constant-time collision attack against KCTR-CMAC which illustrates that KCTR-CMAC is not collision resistant. Finally, we prove that KCTR-CMAC is preimage resistant with the bound $O(\frac{p}{2^n} + \frac{p^2\ell}{2^n})$.

Scheme description. The key derivation function KCTR-CMAC based on CMAC in counter mode is illustrated in Fig. 2. It takes as input a key K, two strings

procedure KCTR-CMAC $(K, \text{Label}, \text{Context}, L)$
$b \leftarrow \lceil L/n \rceil$; $C \leftarrow \varepsilon$
if $b > 2^r - 1$ **then return** $\perp$
$S_0 \leftarrow \mathsf{CMAC}(K, \text{Label} \parallel 0x00 \parallel \text{Context} \parallel [L]_2)$
for $i \leftarrow 1$ **to** b **do**
  $S_i \leftarrow \mathsf{CMAC}(K, [i]_2 \parallel \text{Label} \parallel 0x00 \parallel \text{Context} \parallel [L]_2 \parallel S_0)$
  $C = C \parallel S_i$
return $C[1:L]$

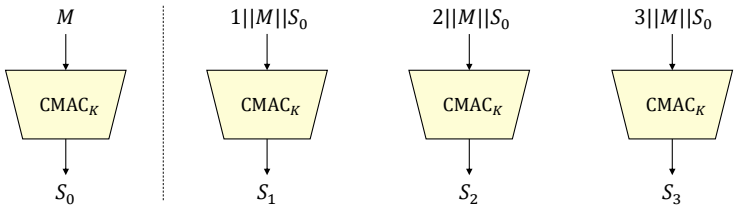


Fig. 2: The pseudocode definition of KCTR-CMAC in counter mode based on CMAC and its pictorial illustration for producing $3n$-bit keying material $C = S_1 \parallel S_2 \parallel S_3$ where $M = \text{Label} \parallel 0x00 \parallel \text{Context} \parallel [L]_2$.

procedure CMAC (K, M)
$K_0 \leftarrow E_K(0^n)$
$M_1 \ldots M_m \leftarrow_n M$
$Y_0 \leftarrow 0^n$
for $i \leftarrow 1$ **to** $m-1$
  $X_i \leftarrow M_i \oplus Y_{i-1}$
  $Y_i \leftarrow E_K(X_i)$
if $|M_m| = n$ **then**
  $X_m \leftarrow Y_{m-1} \oplus M_m \oplus u \cdot K_0$
else
  $X_m \leftarrow Y_{m-1} \oplus M_m \parallel 10^* \oplus u^2 \cdot K_0$
$T \leftarrow E_K(X_m)$
return T

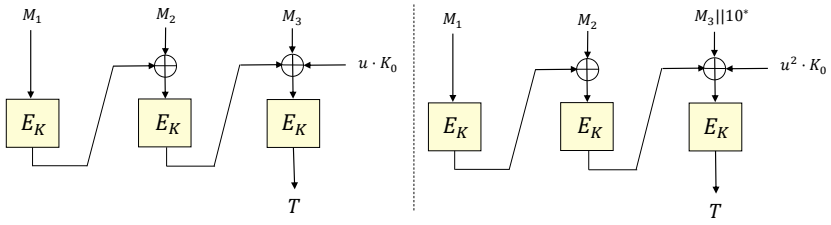


Fig. 3: The pseudocode definition of CMAC and its pictorial illustration for a three-block message M. Here u is specified as the polynomial "x" in $\mathbf{GF}(2^n)$.

called Label and Context[6], an integer L that specifies the requested length

[6] Here 'Label' is a bit string that identifies the purpose for the derived keying material and 'Context' is a bit string that contains the information related to the derived keying material as explained in [21].

in bits of the derived keying material, and outputs L bits of keying material by iterating the CMAC algorithm $\lceil L/n \rceil$ times and truncating the leftmost L bits. In each iteration of a CMAC execution, the fixed input data is the string $\text{Label} \,\|\, 0x00 \,\|\, \text{Context} \,\|\, [L]_2 \,\|\, S_0$ where $0x00$ is an all-zero octet and S_0 is defined as

$$S_0 = \mathsf{CMAC}(K, \text{Label} \,\|\, 0x00 \,\|\, \text{Context} \,\|\, [L]_2) \ .$$

The counter $[i]_2$ is prepended to the fixed input data and is encoded as a string of r bits with $1 \leq r \leq 32$. As required in NIST SP 800-108r1-upd1 [21], the length for each data field and their order shall be defined unambiguously. Hence, in the rest of this paper, besides the counter, we assume that the Label, the separation indicator 0x00, the Context, and $[L]_2$ are represented with a specific length and in a specific order. The specification of CMAC is presented in Fig. 3 for the reference.

volPRF security. Assuming that the underlying block cipher E_K is a good PRP, the following theorem shows that KCTR-CMAC is a good volPRF. Recall that the volPRF security is defined in Section 2.

Theorem 1. *For any adversary $\mathcal{A}$ against the volPRF security of* KCTR-CMAC *that runs in time at most t, makes at most q queries, with each query being of block length at most ℓ and being of output block length at most b where $b \leq 2^r - 1$, we have*

$$\mathsf{Adv}^{\text{volprf}}_{\mathsf{KCTR\text{-}CMAC}}(\mathcal{A}) \leq \mathsf{Adv}^{\text{prp}}_{E}(\mathcal{B}) + \frac{20q^2(b+1)^2}{2^n} + \frac{23q(b+1)(\ell+2)^2}{2^n} \ ,$$

by assuming $q(b+1) \leq 2^{n/2-1}$ and $\ell + 2 \leq 2^{n/4-0.5}$, where $\mathcal{B}$ is an adversary against the PRP security of the block cipher E that runs in time at most $t' = t + q(b+1)(\ell+3)t_E$ and makes at most $q(b+1)(\ell+3)$ block cipher queries where t_E denotes the running time for one computation of E.

Proof. Without loss of generality, we assume that the adversary $\mathcal{A}$ does not repeat a previous query as otherwise she will receive the same answer. We first replace the underlying block cipher E_K with a random permutation $\pi \leftarrow\!\!\$\, \text{Perm}(n)$ and denote the resulting key derivation function by $\mathsf{KCTR\text{-}CMAC}[\pi]$. From the standard argument, we have

$$\mathsf{Adv}^{\text{volprf}}_{\mathsf{KCTR\text{-}CMAC}[E]}(\mathcal{A}) \leq \mathsf{Adv}^{\text{prp}}_{E}(\mathcal{B}) + \mathsf{Adv}^{\text{volprf}}_{\mathsf{KCTR\text{-}CMAC}[\pi]}(\mathcal{A}) \ ,$$

where $\mathcal{B}$ is an adversary against the PRP security of block cipher E that makes at most $q(b+1)(\ell+3)$ queries and runs in times at most $t' = t + q(b+1)(\ell+3)t_E$.

Next, we replace the CMAC algorithm in KCTR-CMAC with a random function $f \leftarrow\!\!\$\, \text{Func}(\{0,1\}^*, \{0,1\}^n)$ and denote the resulting key derivation function by $\mathsf{KCTR\text{-}CMAC}[f]$. By using the results of [18,25,32] that show CMAC is a good PRF, we have

$$\mathsf{Adv}^{\text{volprf}}_{\mathsf{KCTR\text{-}CMAC}[\pi]}(\mathcal{A}) \leq \frac{20q^2(b+1)^2}{2^n} + \frac{23q(b+1)(\ell+2)^2}{2^n} + \mathsf{Adv}^{\text{volprf}}_{\mathsf{KCTR\text{-}CMAC}[f]}(\mathcal{A}) \ .$$

Here we use the state-of-the-art PRF security bound of CMAC in [18, Theorem 3.1] that is proved by assuming $q(b+1) \leq 2^{n/2-1}$ and $\ell + 2 \leq 2^{n/4-0.5}$. We remark that any previous PRF bounds of CMAC could be used, e.g., the ones in [25,32], and may be simpler than the bound here.

Finally, we show that the key derivation function KCTR-CMAC[f] based on a random function f behaves exactly the same as a variable output length random function. For any query M to KCTR-CMAC[f] as illustrated in the bottom of Fig. 2, the inputs $1 \,\|\, M \,\|\, S_0, 2 \,\|\, M \,\|\, S_0, \ldots, b \,\|\, M \,\|\, S_0$ of the random function f to generate the keying material C are distinct due to the use of counter i. Hence, the corresponding outputs of f are independently random strings, therewith the derived keying material C being a L-bit random string as it is a L-bit truncation from $S_1 \,\|\, S_2 \,\|\, \cdots \,\|\, S_b$. For any two distinct queries M and M' to KCTR-CMAC[f], the inputs of the random function f to generate keying material C and C' are also distinct as $M \neq M'$. Therefore, the corresponding two derived keying material C and C' are independently random strings. Following this argument, we can obtain that for any q distinct queries to KCTR-CMAC[f], the outputs of KCTR-CMAC[f] are q independently random strings as the corresponding inputs of the random function f to produce these strings are distinct due to the use of counter and M being non-repeating in each query. Hence, we have

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KCTR\text{-}CMAC}[f]}(\mathcal{A}) = 0 \ .$$

Summing over above equations, we obtain

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KCTR\text{-}CMAC}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \frac{20q^2(b+1)^2}{2^n} + \frac{23q(b+1)(\ell+2)^2}{2^n} \ ,$$

and complete the proof.

Collision attack. We mount collision attack against KCTR-CMAC as follows. The goal of the adversary is to find a pair of $(K, \mathrm{Label}, \mathrm{Context}, L)$ and $(K', \mathrm{Label}', \mathrm{Context}', L')$ such that

$$\mathsf{KCTR\text{-}CMAC}(K, \mathrm{Label}, \mathrm{Context}, L) = \mathsf{KCTR\text{-}CMAC}(K', \mathrm{Label}', \mathrm{Context}', L')$$

with $(K, \mathrm{Label}, \mathrm{Context}, L) \neq (K', \mathrm{Label}', \mathrm{Context}', L')$. Obviously, L should equal to L' as otherwise two strings of different length will not collide. Fix K and K' such that $K = K'$. Assume $M = \mathrm{Label} \,\|\, 0x00 \,\|\, \mathrm{Context} \,\|\, [L]_2$ and $M' = \mathrm{Label}' \,\|\, 0x00 \,\|\, \mathrm{Context}' \,\|\, [L]_2$ that both have the length of $2n - r$ bits, and thus the length of $[i]_2 \,\|\, M \,\|\, S_0$ and $[i]_2 \,\|\, M \,\|\, S'_0$ are both of $3n$ bits. Assume that $r = 1$ and $L = n$, namely that the counter is encoded as 1-bit string and KCTR-CMAC outputs only n-bit keying material by iterating the CMAC once with the counter $[1]_2$. Informally speaking, the intuition behind this attack is to construct a pair of message M and M' such that $S_0 = Y_2$ and $S'_0 = Y'_2$ where Y_2 and Y'_2 are outputs of the second last block cipher in CMAC respectively and result in a collision in the inputs to the last block cipher call. We now show in

detail how the adversary can construct a pair of M and M' to violate the collision resistance of KCTR-CMAC. To find a collision on the output of KCTR-CMAC that is the same as $C[1:n] = C'[1:n]$, it requires the following equation holds:

$$\begin{aligned} & E_K(S_0 \oplus K_1 \oplus E_K(M[n:2n-1] \oplus E_K([1]_2 \parallel M[1:n-1]))) \\ = & E_K(S_0' \oplus K_1 \oplus E_K(M'[n:2n-1] \oplus E_K([1]_2 \parallel M'[1:n-1]))) \ , \end{aligned}$$

where the subkey key K_1 is derived as $K_1 = u \cdot E_K(0^n)$ following the specification of CMAC [24,25]. After removing the outer block cipher call, the above equation becomes

$$\begin{aligned} & S_0 \oplus E_K(M[n:2n-1] \oplus E_K([1]_2 \parallel M[1:n-1])) \\ = & S_0' \oplus E_K(M'[n:2n-1] \oplus E_K([1]_2 \parallel M'[1:n-1])) \ . \end{aligned} \tag{1}$$

If the adversary can find a pair of M and M' such that

$$S_0 = E_K(M[n:2n-1] \oplus E_K([1]_2 \parallel M[1:n-1])) \tag{2}$$

and

$$S_0' = E_K(M'[n:2n-1] \oplus E_K([1]_2 \parallel M'[1:n-1])) \ , \tag{3}$$

then the equation (1) holds as both left part and right part of it become 0^n. The adversary can first construct M to satisfy the above condition as follows. The condition

$$S_0 = E_K(M[n:2n-1] \oplus E_K([1]_2 \parallel M[1:n-1]))$$

is the same as

$$\begin{aligned} & E_K((M[n+1:2n-1] \parallel 1) \oplus K_2 \oplus E_K(M[1:n])) \\ = & E_K(M[n:2n-1] \oplus E_K([1]_2 \parallel M[1:n-1])) \end{aligned}$$

by writing down the computation of S_0 where $K_2 = u^2 \cdot E_K(0^n)$ is another subkey following the specification of CMAC. By removing the outer block cipher call, the above condition becomes

$$(M[n+1:2n-1] \| 1) \oplus K_2 \oplus E_K(M[1:n]) = M[n:2n-1] \oplus E_K([1]_2 \| M[1:n-1]) \ .$$

The adversary can choose a value for $M[1:n]$ to satisfy the format of M so that an all-zero octet $0x00$ following after the string 'Label' (we assume the length of 'Label' is less than or equal to $n-8$ bits here). After that, the adversary can compute the values of $E_K(M[1:n])$ and $E_K([1]_2 \parallel M[1:n-1])$. Let

$$\mathsf{cst} = K_2 \oplus E_K(M[1:n]) \oplus E_K([1]_2 \parallel M[1:n-1]) \ .$$

The adversary then needs to compute the value for $M[n+1:2n-1]$ that should satisfy the following condition:

$$(M[n+1:2n-1] \parallel 1) \oplus M[n:2n-1] = \mathsf{cst} \ .$$

She can compute the value of $M[n+1:2n-1]$ one bit by one bit from the following equations:

$$\begin{aligned} M[n+1] \oplus M[n] &= \mathsf{cst}[1] \\ M[n+2] \oplus M[n+1] &= \mathsf{cst}[2] \\ &\vdots \\ M[2n-1] \oplus M[2n-2] &= \mathsf{cst}[n-1] \\ 1 \oplus M[2n-1] &= \mathsf{cst}[n] \ . \end{aligned}$$

Namely, from the last equation, she can obtain $M[2n-1]$ as $M[2n-1] = 1 \oplus \mathsf{cst}[n]$. Then she can obtain $M[2n-2]$ as $M[2n-2] = M[2n-1] \oplus \mathsf{cst}[n-1]$. Repeating this procedure, she can finally obtain $M[n+1] = M[n+2] \oplus \mathsf{cst}[2]$ and $M[n] = M[n+1] \oplus \mathsf{cst}[1]$. Note that the first bit $M[n]$ has been determined when the adversary chooses $M[1:n]$, and the last $\lceil \log_2(n) \rceil$ bits $M[(2n - \lceil \log_2(n) \rceil) : 2n-1]$ should be the encoding of L. Hence, there is probability that these bits should be consistent with the pre-determined values which is about $2^{-\lceil \log_2(n) \rceil - 1}$ by relying on the randomness of cst which is the xor sum of some outputs of block cipher E_K. At this stage, the adversary has constructed a message M such that the equation (2) holds. Following similar procedures, she can construct another message M' such that $M' \neq M$ by choosing $M'[1:n] \neq M[1:n]$ and makes the equation (3) hold with probability about $2^{-\lceil \log_2(n) \rceil - 1}$. Hence, the complexity of finding this pair of M and M' is about $2 \cdot 2^{\lceil \log_2(n) \rceil + 1} = 2^{\lceil \log_2(n) \rceil + 2}$. If the block size of the underlying block cipher instantiated in CMAC is 128 bits like AES, then the attack complexity to find a collision against KCTR-CMAC becomes 2^9.

Note that as specified in NIST SP 800-108r1-upd1 [21], the order of each data field can be defined as part of a KDF specification. The requirement is that the length for each data field and their order shall be defined unambiguously. Hence, if the fixed input data is defined in the order of Label $\| \, 0x00 \| \, [L]_2 \| \,$ Context $\| \, S_0$, then the adversary may choose $M[1:n]$ and $M'[1:n]$ to satisfy the format imposed by Label $\| \, 0x00 \| \, [L]_2$ and Label$' \| \, 0x00 \| \, [L]_2$ (assume that the total length of each of these two strings is not greater than n bits). By doing so, the complexity of finding a collision against KCTR-CMAC becomes 4 as there may not be format restriction in the strings Context and Context'. Hence, as a summary, this collision attack is constant time.

Preimage resistance. For any fixed value $T \in \{0,1\}^n$, we say an adversary finds a preimage against KCTR-CMAC if she outputs a pair of (K, M) such that KCTR-CMAC$(K, M)[1:n] = T$. We suspect that although the derived keying material can allow to have more than n bits, the preimage resistance security may not be more than the case of n-bit output. Hence, we focus on the case when the image is of n bits. The following result illustrates that KCTR-CMAC is preimage resistant.

Theorem 2. *For any adversary $\mathcal{A}$ that makes at most p ideal-cipher queries to E and E^{-1}, we have*

$$\mathsf{Adv}^{\mathsf{epre}}_{\mathsf{KCTR\text{-}CMAC}}(\mathcal{A}) \leq \frac{4p}{2^n} + \frac{2p^2\ell}{2^n} \ ,$$

by assuming $p \leq 2^{n-1}$ where ℓ is the maximum block length of a query to the key derivation function.

Proof overview. Prior to the formal proof, we present a conceptual overview of the underlying proof strategy. We bound the preimage advantage for a fixed target $T \in \{0,1\}^n$ by letting P_i denote the event that the adversary succeeds on its i-th construction query (K_i, M_i) and then estimating $\Pr[P_i]$ under an ideal-cipher model with at most p total E/E^{-1} queries. Writing the computation of KCTR-CMAC as two chained CMAC evaluations under the same key K_i (a first chain producing S_0^i, followed by a second chain producing $S_1^i = \mathsf{KCTR\text{-}CMAC}(K_i, M_i)[1:n]$), the proof conditions on whether each internal block-cipher call is forward or backward and applies the standard ideal-cipher argument: unless a newly formed input repeats a previous one, its output is uniform over a set of size at least $2^n - p$, so hitting the specific value needed to obtain T costs about 2^{-n}. If there is no input collision, each construction query requires at least one new ideal-cipher query, which implies $q \leq p$. The main case analysis shows that any success must be accompanied by either (i) directly guessing a required internal value (giving an $O(2^{-n})$ term) or (ii) creating an internal input collision in the second CMAC chain; by tracking how a "random" message block from the first chain propagates into at most ℓ subsequent inputs of the second chain and union-bounding the corresponding collision events, we obtain $\Pr[P_i] \leq 4/2^n + 2p\ell/2^n$ (for $p \leq 2^{n-1}$). Summing over at most $q \leq p$ construction queries yields $\sum_i \Pr[P_i] \leq 4p/2^n + 2p^2\ell/2^n$, which gives the claimed advantage bound.

Proof. Let $\mathcal{A}$ be an adversary that aims at finding a preimage against the key derivation function KCTR-CMAC for any given image $T \in \{0,1\}^n$. Namely, given an image $T \in \{0,1\}^n$, the goal of the adversary is to find a preimage (K, M) such that $\mathsf{KCTR\text{-}CMAC}(K, M)[1:n] = T$. The query to the key derivation function KCTR-CMAC is named as construction query, while the query to the block cipher E or E^{-1} is named as ideal cipher query. Assume that the adversary can make at most p queries to E and E^{-1}.

Fix an image $T \in \{0,1\}^n$. Denote by P_i the event that the adversary find the preimage for T at the i-th construction query (K_i, M_i), i.e., the event that $\mathsf{KCTR\text{-}CMAC}(K_i, M_i)[1:n] = T$.[7] Let m be the bit length of the message M_i, and $\ell = \lceil m/n \rceil$. We consider the case of $m + r < n\ell$ where r is the bit length

[7] Note that although we consider event P_i at the i-th construction query, the following analysis explicitly considers ideal cipher queries inside this construction query. Eventually, if the adversary finds the preimage for a given image T, it will anyway form a construction query (K_i, M_i) from ideal cipher queries such that $\mathsf{KCTR\text{-}CMAC}(K_i, M_i)[1:n] = T$.

of the binary encoding of the counter i and n is the block size of the underlying block cipher. The other case of $m + r \geq n\ell$ can be analyzed similarly. The output of $\mathsf{KCTR\text{-}CMAC}(K_i, M_i)[1:n]$ is defined as $S_1^i = Y_{\ell+1}^i$, where internal block cipher calls among the computation of S_1^i are:

$$\begin{aligned}
B_0^i &= 0^n \\
A_j^i &= B_{j-1}^i \oplus M_i[n(j-1)+1:nj] \text{ for } 1 \leq j \leq \ell-1 \\
A_\ell^i &= B_{\ell-1}^i \oplus M_i[n(\ell-1)+1:m] \parallel 10^* \oplus u^2 \cdot E_{K_i}(0^n) \\
B_j^i &= E_{K_i}(A_j^i) \text{ for } 1 \leq j \leq \ell \\
S_0^i &= B_\ell^i \\
Y_0^i &= 0^n \\
X_1^i &= Y_0^i \oplus [1]_r \parallel M_i[1:n-r] \\
X_j^i &= Y_{j-1}^i \oplus M_i[n(j-1)-r+1:nj-r] \text{ for } 2 \leq j \leq \ell-1 \\
X_\ell^i &= Y_{\ell-1}^i \oplus M_i[n(\ell-1)-r+1:m] \parallel S_0^i[1:n\ell-m-r] \\
X_{\ell+1}^i &= Y_\ell^i \oplus S_0^i[n\ell-m-r+1:n] \parallel 10^* \oplus u^2 \cdot E_{K_i}(0^n) \\
Y_j^i &= E_{K_i}(X_j^i) \text{ for } 1 \leq j \leq \ell+1 \ .
\end{aligned}$$

We call the chain to produce the value S_0^i as the first CMAC chain, and the chain to produce the value S_1^i as the second CMAC chain. We proceed to analyze the preimage security by considering the type of ideal cipher queries to build these two CMAC chains.

If both of $(K_i, A_\ell^i, B_\ell^i)$ and $(K_i, X_\ell^i, Y_\ell^i)$ are forward block cipher queries, then the probability that $Y_\ell^i \oplus B_\ell^i[n\ell-m-r+1:n] \parallel 10^* \oplus u^2 \cdot E_{K_i}(0^n) = E_{K_i}^{-1}(T)$ is at most $1/(2^n-p)$ as Y_ℓ^i is distributed uniformly at random in a set of size at least $2^n - p$. One exception is when $Y_\ell^i = B_\ell^i[n\ell-m-r+1:n] \parallel 10^*$ as shown in the collision attack against $\mathsf{KCTR\text{-}CMAC}$. Yet, in this case, it requires that $u^2 \cdot E_{K_i}(0^n) = E_{K_i}^{-1}(T)$ which happens with probability at most $1/(2^n-p)$ for a given image T. Hence, summing over at most $\binom{p}{2}$ different pairs of $(K_i, A_\ell^i, B_\ell^i)$ and $(K_i, X_\ell^i, Y_\ell^i)$, we have

$$\sum_{i=1}^{q} \Pr[\mathsf{P}_i] \leq \frac{p(p-1)}{2(2^n-p)} \leq \frac{p^2}{2^n}$$

by assuming $p \leq 2^{n-1}$.

If at least one of $(K_i, A_\ell^i, B_\ell^i)$ and $(K_i, X_\ell^i, Y_\ell^i)$ is a backward block cipher query, then without loss of generality, we assume that $(K_i, A_\ell^i, B_\ell^i)$ is a backward block cipher query. The other case that $(K_i, X_\ell^i, Y_\ell^i)$ is a backward query can be analyzed similarly. We consider different cases depending on whether the preceding block cipher call is a forward query or not. If the preceding block cipher call $(K_i, A_{\ell-1}^i, B_{\ell-1}^i)$ is a forward block cipher query, then $M_i[n(\ell-1)+1:m] \parallel 10^* = A_\ell^i \oplus B_{\ell-1}^i \oplus u^2 \cdot E_{K_i}(0^n)$ is distributed uniformly at random in a set of size at least $2^n - \max\{p_{i,\ell-1}, p_{i,\ell}\}$ where $p_{i,j}$ denotes the total number of ideal

cipher queries at the j-th block cipher call in this CMAC chain. If the preceding block cipher call $(K_i, A^i_{\ell-1}, B^i_{\ell-1})$ is a backward block cipher query, then the adversary can set $B^i_{\ell-1} = A^i_\ell \oplus M_i[n(\ell-1)+1:m] \,\|\, 10^* \oplus u^2 \cdot E_{K_i}(0^n)$ and does the backward query to obtain $A^i_{\ell-1}$. We can repeat this argument until some preceding block cipher call (K_i, A^i_j, B^i_j) is a forward query for $1 \le j \le \ell-1$. In this case, the message block $M_i[nj+1:n(j+1)] = A^i_{j+1} \oplus B^i_j$ is uniformly distributed in a set of size at least $2^n - \max\{p_{i,j}, p_{i,j+1}\}$ as A^i_{j+1} is uniformly distributed in a set of size at least $2^n - p_{i,j+1}$ and B^i_j is uniformly distributed in a set of size at least $2^n - p_{i,j}$. In the case that all of preceding block cipher calls are backward queries, then the first message block $M_i[1:n]$ will be distributed uniformly at random in a set of size at least $2^n - p_{i,1}$ due to the randomness of the output of $E^{-1}_{K_i}(B^i_1)$. Let $M_i[nc+1:n(c+1)]$ be this random message block where $0 \le c < \ell-1$. Note that the case when the random message block is $M_i[n(\ell-1)+1:m] \,\|\, 10^*$ can be analyzed similarly, and in the following we focus on the more general case of $M_i[nc+1:n(c+1)]$ with $0 \le c < \ell-1$. Without loss of generality, we assume that all of message blocks including this random message block has been determined during the computation of the first CMAC chain. We then analyze the influence of this random message block on the second CMAC chain. This message block may spread into two block cipher calls in the second CMAC chain. If $c = 0$, then it will spread into two block cipher calls with input

$$X^i_{c+1} = Y^i_c \oplus [1]_2 \,\|\, M_i[1:n-r]$$

and

$$X^i_{c+2} = Y^i_{c+1} \oplus M_i[n-r+1:2n-r] \ .$$

If $c > 0$, then it will spread into two block cipher calls with inputs

$$X^i_{c+1} = Y^i_c \oplus M_i[nc-r+1:n(c+1)-r]$$

and

$$X^i_{c+2} = Y^i_{c+1} \oplus M_i[n(c+1)-r+1:n(c+2)-r] \ .$$

Let E_1 be the event that X^i_{c+1} hits the previous inputs of the block cipher E_{K_i}, E_2 the event that X^i_{c+2} hits the previous inputs of the block cipher E_{K_i}. Then we have

$$\begin{aligned}
\Pr[\mathsf{E}_2] &= \Pr[\mathsf{E}_2 \wedge \mathsf{E}_1] + \Pr[\mathsf{E}_2 \wedge \overline{\mathsf{E}_1}] \\
&= \Pr[\mathsf{E}_2 \mid \mathsf{E}_1] \cdot \Pr[\mathsf{E}_1] + \Pr[\mathsf{E}_2 \mid \overline{\mathsf{E}_1}] \cdot \Pr[\overline{\mathsf{E}_1}] \\
&\le \Pr[\mathsf{E}_2 \mid \mathsf{E}_1] \cdot \Pr[\mathsf{E}_1] + \Pr[\mathsf{E}_2 \mid \overline{\mathsf{E}_1}] \qquad (4)
\end{aligned}$$

as $\Pr[\overline{\mathsf{E}_1}] \le 1$. We first analyze the probability of the first term on the right side of this equation. As all of message blocks have been determined in the first CMAC chain, we can use message blocks $[1]_2 \,\|\, M_i[1:n-r]$, $M_i[n-r+1:2n-r]$, …, $M_i[n(c-1)-r+1:nc-r]$ to compute the chain that reaches the output Y^i_c. Let $\mathrm{Dom}(E^j_{K_i})$ denote the set of inputs of queried block cipher calls E_{K_i}

and outputs of queried block cipher calls $E_{K_i}^{-1}$ at the j-th call to the block cipher E_{K_i} in the second CMAC chain. Denote by $\widetilde{p}_{i,j}$ the size of $\mathrm{Dom}(E_{K_i}^j)$. Let

$$\begin{aligned} D_1 =&\{X[r+1:n] \mid X \in \mathrm{Dom}(E_{K_i}^c) \\ &\wedge X[1:r] = Y_c^i[1:r] \oplus M_i[nc-r+1:nc]\} \ . \end{aligned}$$

Then event E_1 requires that

$$M_i[nc+1:n(c+1)-r] \oplus Y_c^i[r+1:n] \in D_1 \ .$$

Let

$$\begin{aligned} D_2 =&\{X[1:r] \mid X \in \mathrm{Dom}(E_{K_i}^{c+1}) \\ &\wedge X[r+1:n] = Y_{c+1}^i[r+1:n] \oplus M_i[n(c+1)+1:n(c+2)-r]\} \ . \end{aligned}$$

Conditioned on that event E_1 happens, event E_2 requires that

$$M_i[n(c+1)-r+1:n(c+1)] \oplus Y_{c+1}^i[1:r] \in D_2 \ .$$

As $\mathrm{Dom}(E_{K_i}^c) \subset \mathrm{Dom}(E_{K_i}^{c+1})$ (without loss of generality, we assume that the c-th block cipher call is made before the $(c+1)$-th block cipher call in the second CMAC chain), we have

$$|D_1| + |D_2| \leq |\mathrm{Dom}(E_{K_i}^{c+1})| + 1 = \widetilde{p}_{i,c+1} + 1 \ ,$$

as there is only one possible element in $\mathrm{Dom}(E_{K_i}^{c+1})$ that can both lie in the set of D_1 and D_2, namely the only potential value is X where $X[1:r] = Y_c^i[1:r] \oplus M_i[nc-r+1:nc]\}$ and $X[r+1:n] = Y_{c+1}^i[r+1:n] \oplus M_i[n(c+1)+1:n(c+2)-r]\}$. Hence, we have

$$\Pr[\mathsf{E}_2 \wedge \mathsf{E}_1] \leq \frac{\widetilde{p}_{i,c+1}+1}{2^n - p_{i,c+1}} \ ,$$

as the message block $M_i[nc+1:n(c+1)]$ is distributed uniformly at random in a set of size at least $2^n - p_{i,c+1}$. We proceed to analyze the second term on the right side of the equation (4). Conditioned on that event E_1 does not happen, the value Y_{c+1}^i is distributed uniformly at random in a set of size at least $2^n - \widetilde{p}_{i,c+1}$, as it is the output of $E_{K_i}(X_{c+1}^i)$. Hence, we have

$$\Pr[\mathsf{E}_2 \mid \overline{\mathsf{E}_1}] \leq \frac{\widetilde{p}_{i,c+1}}{2^n - \widetilde{p}_{i,c+1}} \ .$$

Therefore,

$$\Pr[\mathsf{E}_2] \leq \frac{\widetilde{p}_{i,c+1}+1}{2^n - p_{i,c+1}} + \frac{\widetilde{p}_{i,c+1}}{2^n - \widetilde{p}_{i,c+1}} \ .$$

We then continue to analyze subsequent block cipher calls. Denote by $\mathsf{bcoll}_{i,c+j}$ the event that $X_{c+j}^i = Y_{c+j-1}^i \oplus M_i[n(c+j-1)-r+1 : n(c+j)-r]$ hits

previous inputs of the block cipher E_{K_i} for $3 \le j \le \ell + 1 - c$. Then we have $\Pr[\mathsf{bcoll}_{i,c+j}] \le \widetilde{p}_{i,c+j-1}/(2^n - \widetilde{p}_{i,c+j-1})$. Let $\mathsf{Bcoll}_i = \mathsf{E}_2 \vee (\vee_{j=3}^{\ell+1-c} \mathsf{bcoll}_{i,c+j})$. Conditioned on that the event Bcoll_i does not happen, S_1^i is distributed uniformly at random in a set of size at least $2^n - \widetilde{p}_{i,\ell+1}$. Thus, the probability of the preimage event P_i can be bounded by

$$\begin{aligned}
\Pr[\mathsf{P}_i] &\le \Pr[\mathsf{P}_i \mid \overline{\mathsf{Bcoll}}_i] + \Pr[\mathsf{Bcoll}_i] \\
&\le \frac{1}{2^n - \widetilde{p}_{i,\ell+1}} + \Pr[\mathsf{E}_2 \vee (\vee_{j=3}^{\ell+1-c} \mathsf{bcoll}_{i,\ell+c+j})] \\
&\le \frac{1}{2^n - \widetilde{p}_{i,\ell+1}} + \Pr[\mathsf{E}_2] + \Pr[\mathsf{bcoll}_{i,c+3}] \\
&\quad + \sum_{j=4}^{\ell+1-c} \Pr[\mathsf{bcoll}_{i,c+j} \mid \overline{\mathsf{bcoll}}_{i,c+3} \wedge \cdots \wedge \overline{\mathsf{bcoll}}_{i,c+j-1} \wedge \overline{\mathsf{E}}_2] \\
&\le \frac{4}{2^n} + \frac{2p\ell}{2^n} \quad ,
\end{aligned}$$

by assuming $p_{i,j}, \widetilde{p}_{i,j} \le p \le 2^{n-1}$. Summing over at most $q \le p$ construction queries, we have

$$\sum_{i=1}^{q} \Pr[\mathsf{P}_i] \le \frac{4p}{2^n} + \frac{2p^2\ell}{2^n} \quad .$$

As $p^2/2^n \le 4p/2^n + 2p^2\ell/2^n$, we have

$$\mathsf{Adv}^{\mathsf{epre}}_{\mathsf{KCTR\text{-}CMAC}}(\mathcal{A}) \le \frac{4p}{2^n} + \frac{2p^2\ell}{2^n}$$

that completes the proof.

3.2 Security Analysis of KCTR-HMAC

In this subsection, we give security analysis of key derivation function KCTR-HMAC based on HMAC in counter mode. We prove that KCTR-HMAC is a secure volPRF with the bound $O(\frac{q^2b^2}{2^n})$. We show that KCTR-HMAC is collision resistant and preimage resistant both with the bound $O(\frac{p^2}{2^n})$ when the key is of fixed length and less than $d-1$ bits. We also show collision attack against KCTR-HMAC when the key is of variable length.

Scheme description. The key derivation function KCTR-HMAC is the same as KCTR-CMAC, except that the underlying PRF is replaced with HMAC, and the fixed input data is simplified from Label $\|$ $0x00$ $\|$ Context $\|$ $[L]_2$ $\|$ S_0 to Label $\|$ $0x00$ $\|$ Context $\|$ $[L]_2$, namely the string S_0 is removed. See Fig. 4 for pseudocode specification and pictorial illustration of KCTR-HMAC.

volPRF security. Let $h : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^n$ be a compression function. Let pad be the padding function such that $M^* = M \| \mathrm{pad}(|M|) \in (\{0,1\}^n)^+$ for any string M. The cryptographic hash function H associated to h is defined

procedure KCTR-HMAC $(K, \text{Label}, \text{Context}, L)$
$b \leftarrow \lceil L/n \rceil$; $C \leftarrow \varepsilon$
if $b > 2^r - 1$ **then return** $\perp$
for $i \leftarrow 1$ **to** b **do**
  $S_i \leftarrow \mathsf{HMAC}(K, [i]_2 \parallel \text{Label} \parallel 0x00 \parallel \text{Context} \parallel [L]_2)$
  $C = C \parallel S_i$
return $C[1:L]$

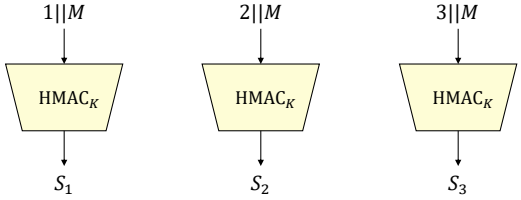


Fig. 4: The pseudocode definition of KCTR-HMAC in counter mode based on HMAC and its pictorial illustration for producing $3n$-bit keying material $C = S_1 \parallel S_2 \parallel S_3$ where $M = \text{Label} \parallel 0x00 \parallel \text{Context} \parallel [L]_2$.

procedure $h^*(iv, M^*)$
$M_1 \dots M_m \xleftarrow{d} M^*$
$Y_0 \leftarrow iv$
for $i \leftarrow 1$ **to** m
  $Y_i \leftarrow h(Y_{i-1}, M_i)$
return Y_m

Fig. 5: The definition of h^* which is the classical Merkle-Damgård construction.

as $H(M) = h^*(iv, M^*)$, where iv is a n-bit initial value that is fixed as part of the description of hash function H and M is a string of arbitrary length such that $M^* = M \parallel \text{pad}(|M|)$. The pseudocode description of h^* is given in Fig. 5. Then $\mathsf{HMAC} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$ is defined as

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

where opad and ipad are two distinct d-bit constant strings, $K' = H(K) \parallel 0^{d-n}$ if K is larger than the block size of h and $K' = K \parallel 0^{d-|K|}$ is K is less than or equal to the block size of h.

Let $\overline{h} : \{0,1\}^d \times \{0,1\}^n \to \{0,1\}^n$ be the dual of h, namely $\overline{h}(x, y) = h(y, x)$. The assumption that h is a PRF when keyed by its data input is equivalent to the assumption that $\overline{h}$ is a PRF. We say that h is a secure dual-PRF if both h and $\overline{h}$ are a secure PRF. We will show that KCTR-HMAC is a secure volPRF as long as h is a dual-PRF and $\overline{h}$ is a PRF against an appropriate class of related key attacks. Before that, we recall the formal definition of related key security from [7].

A related-key attack on $\overline{h} : \{0,1\}^d \times \{0,1\}^n \to \{0,1\}^n$ is parameterized by a set of mappings $\Phi : \{0,1\}^d \to \{0,1\}^d$. Let the related key function RK :

$\Phi \times \{0,1\}^d \to \{0,1\}^d$ be $\mathrm{RK}(\phi, K) = \phi(K)$ where $\phi \in \Phi$ and $K \in \{0,1\}^d$. A related-key attack adversary against $\overline{h}$ may make an oracle query of the form (ϕ, x) where $\phi \in \Phi$ and $x \in \{0,1\}^n$. Let $\mathrm{Func}(\{0,1\}^d \times \{0,1\}^n, \{0,1\}^n)$ denote the set of all functions from $\{0,1\}^d \times \{0,1\}^n$ to $\{0,1\}^n$. The rka-PRF advantage is defined by

$$\mathsf{Adv}^{\mathrm{rkaprf}}_{\overline{h},\Phi}(\mathcal{A}) = \left|\Pr[\mathcal{A}^{\overline{h}(\mathrm{RK}(\cdot,K),\cdot)} = 1] - \Pr[\mathcal{A}^{G(\mathrm{RK}(\cdot,K),\cdot)} = 1]\right|, \tag{5}$$

where K is chosen uniformly at random from $\{0,1\}^d$ and G is chosen uniformly at random from $\mathrm{Func}(\{0,1\}^d \times \{0,1\}^n, \{0,1\}^n)$. The related key function set of interest for HMAC is

$$\Phi_{\mathsf{zio},e} : \{0,\ldots,d\} \times \{\mathsf{opad},\mathsf{ipad}\} \times \{0,1\}^d \to \{0,1\}^d \cup \{\perp\}$$

defined as follows: on inputs $i \in \{0,\ldots,d\}$, $\mathsf{const} \in \{\mathsf{opad},\mathsf{ipad}\}$, and $K \in \{0,1\}^d$, it returns $(K[1:i] \,\|\, 0^{d-i}) \oplus \mathsf{const}$ if $i \geq e$ and $\perp$ otherwise. Here $e = \min\{n,k\}$, where n is the output size of $\overline{h}$ and k is the key length of HMAC. Namely, the related key function $\Phi_{\mathsf{zio},e}$ overwrites the right $(d-i)$ bits of the key of $\overline{h}$ with zeros, and then XORs the result with either opad or ipad. The following theorem establishes the volPRF security for KCTR-HMAC, assuming dual-PRF security of h and rka-PRF security of $\overline{h}$ under $\Phi_{\mathsf{zio},e}$-restricted related-key attacks. We refer to [5,3] for more discussion about these two security notions. We remark that $\Phi_{\mathsf{zio},e}$ is a small set of simple functions, and the RKA assumption on $\overline{h}$ is mild by the fact that the adversary makes only two oracle queries, in both of which the message is the same, namely the *iv* of the hash function.

Theorem 3. *Let k be the key length of* KCTR-HMAC*, and $\alpha = \lceil k/d \rceil$. For any adversary $\mathcal{A}$ against the volPRF security of* KCTR-HMAC *that runs in time at most t, makes at most q queries, with each query being of block length at most ℓ and being of output block length at most b where $b \leq 2^r - 1$, we have*

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KCTR\text{-}HMAC}}(\mathcal{A}) \leq (\ell{+}3{+}\alpha)\,\mathsf{Adv}^{\mathrm{prf}}_{h}(\mathcal{A}_1){+}\mathsf{Adv}^{\mathrm{prf}}_{\overline{h}}(\mathcal{A}_2){+}\mathsf{Adv}^{\mathrm{rkaprf}}_{\Phi_{\mathsf{zio},e},\overline{h}}(\mathcal{B}){+}\frac{qb(qb-1)}{2^{n+1}} \ .$$

Adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ are against the PRF security of h and $\overline{h}$ respectively, where $\mathcal{A}_1$ makes at most qb queries and $\mathcal{A}_2$ makes one query. Adversary $\mathcal{B}$ makes two queries. The running times of adversaries $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{B}$ are about the same as that of $\mathcal{A}$.

Proof. Let $\mathcal{A}$ be an adversary against the volPRF security of KCTR-HMAC. We first replace HMAC with a random function $f \leftarrow\!\!\$\ \mathrm{Func}(\{0,1\}^* \times \{0,1\}^*, \{0,1\}^n)$ in KCTR-HMAC and denote the key derivation function by KCTR-HMAC$[f]$. Using the result of [3, Theorem 10][8] that shows HMAC is a good PRF under

[8] Note that this result covers the multi-user PRF security of HMAC in the standard model. Here we adapt it for the single-user PRF security of HMAC by setting the number of users to be one.

the assumption of dual-PRF security of h and rka-PRF security of $\overline{h}$, we have

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KCTR\text{-}HMAC}}(\mathcal{A}) \leq (\ell + 3 + \alpha)\,\mathsf{Adv}^{\mathrm{prf}}_{h}(\mathcal{A}_1) + \mathsf{Adv}^{\mathrm{prf}}_{\overline{h}}(\mathcal{A}_2) + \mathsf{Adv}^{\mathrm{rkaprf}}_{\Phi_{\mathsf{zio},e},\overline{h}}(\mathcal{B}) + \frac{qb(qb-1)}{2^{n+1}} + \mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KCTR\text{-}HMAC}[f]}(\mathcal{A}) \ ,$$

where $\mathcal{A}_1$ and $\mathcal{A}_2$ are adversaries against the PRF security of h and $\overline{h}$ respectively, who make at most qb queries and one query respectively, $\mathcal{B}$ is an adversary against the rka-PRF security of $\overline{h}$ that makes two queries, the running times of who are all about the same as that of $\mathcal{A}$. We remark that the previous PRF bound by Bellare [5] could also be used here. Yet, as noted by Backendal et al. [3], Bellare's bound assumes a full-block key, while the bound by Backendal et al. [3] allows the key of HMAC of varying length that matches the suggestion in the NIST standard [21, Section 6.2].

We then analyze the volPRF security of $\mathsf{KCTR\text{-}HMAC}[f]$. For any q distinct queries to $\mathsf{KCTR\text{-}HMAC}[f]$, as the inputs to the underlying random function f are all distinct due to use of counter and the constraint that $M \neq M'$ for any two different queries M and M', the corresponding outputs are merely q random strings. Hence, we have

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KCTR\text{-}HMAC}[f]}(\mathcal{A}) = 0$$

that concludes the proof.

Collision resistance. As indicated in NIST SP 800-108r1-upd1 [21], it requires that derived keying material should be bound to the input of the key derivation function that is called context binding. The purpose of context binding is to provide "assurance that all parties who (correctly) derive the keying material share the same understanding of who will access it and in which session it will be used. If those parties have different understandings, then they will derive different keying material". It is therefore expected that it is hard for an adversary to obtain a pair of $(K_1, M_1) \neq (K_2, M_2)$ such that $\mathsf{KCTR\text{-}HMAC}(K_1, M_1) = \mathsf{KCTR\text{-}HMAC}(K_2, M_2)$, namely finding a collision against the key derivation function KCTR-HMAC. Here we say a collision happens if the two derived keying materials have the same first n bits. We suspect that although the derived keying material can allow to have arbitrary length, the collision resistance of KCTR-HMAC cannot be better than the underlying HMAC. The following theorem shows that KCTR-HMAC is collision resistant by assuming the compression function $h : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^n$ is a random oracle.

Theorem 4. *Suppose that the key length of* KCTR-HMAC *is fixed and less than* $d-1$ *bits. For any adversary* $\mathcal{A}$ *that makes at most* p *queries to the compression function* h*, we have*

$$\mathsf{Adv}^{\mathsf{coll}}_{\mathsf{KCTR\text{-}HMAC}}(\mathcal{A}) \leq \frac{13p^2}{2^n} \ .$$

Proof. Let $\mathcal{A}$ be an adversary against the collision resistance security of KCTR-HMAC with oracle access to the compression function h. Suppose that the adversary

$\mathcal{A}$ finds a pair of (K_1, M_1) and (K_2, M_2) such that $\mathsf{KCTR\text{-}HMAC}(K_1, M_1)[1 : n] = \mathsf{KCTR\text{-}HMAC}(K_2, M_2)[1 : n]$, then we can transform this pair into another colliding pair of $(K_1, 1 \parallel M_1)$ and $(K_2, 1 \parallel M_2)$ against HMAC such that $\mathsf{HMAC}(K_1, 1 \| M_1) = \mathsf{HMAC}(K_2, 1 \| M_2)$. As shown by Dodis et al. [23], HMAC is indifferentiable from a random oracle when the key is of a fixed length less than $d - 1$ bits. Hence, we can use this colliding pair to attack the indifferentiability of HMAC. By [23, Theorem 4.4], we have

$$\mathsf{Adv}^{\mathsf{coll}}_{\mathsf{KCTR\text{-}HMAC}}(\mathcal{A}) \leq \frac{13p^2}{2^n} \ .$$

Collision attack. Note that when the key is of variable length, there is a collision attack against HMAC as shown in [23,13]. The attack works as follows. The adversary first computes $T = \mathsf{HMAC}(K_1, 1 \parallel M_1)$ for a pair of (K_1, M_1). If $|K_1| < d$, then she sets $K_2 = K_1 \parallel 0$. If $|K_1| > d$, then she sets $K_2 = H(K_1)$. One can check that $\mathsf{HMAC}(K_1, 1 \parallel M_1) = \mathsf{HMAC}(K_2, 1 \parallel M_1)$ from the definition of HMAC. Obviously, the pair of (K_1, M_1) and (K_2, M_1) can also result in a collision on the first n-bit output of KCTR-HMAC. Hence, KCTR-HMAC is not collision resistant when the key is of variable length. This collision attack also works for KFB-HMAC and KDPL-HMAC when the key is of variable length.

Preimage resistance. For a given image $T \in \{0,1\}^n$, we say an adversary $\mathcal{A}$ finds a preimage (K, M) against KCTR-HMAC if $\mathsf{KCTR\text{-}HMAC}(K, M)[1 : n] = T$. We also reduce the preimage resistance of KCTR-HMAC to the preimage resistance of HMAC that again is reduced to the indifferentiablity of HMAC by Dodis et al. [23]. The proof is omitted as the reduction is similar to the above proof.

Theorem 5. *Suppose that the key length of* KCTR-HMAC *is fixed and less than* $d - 1$ *bits. For any adversary* $\mathcal{A}$ *that makes at most* p *queries to the compression function* h*, we have*

$$\mathsf{Adv}^{\mathsf{epre}}_{\mathsf{KCTR\text{-}HMAC}}(\mathcal{A}) \leq \frac{13p^2}{2^n} \ .$$

3.3 Security Analysis of pKCTR-CMAC

In this subsection, we prove the volPRF security of the plain key derivation function pKCTR-CMAC based on CMAC in counter mode. Our result shows that pKCTR-CMAC enjoys the same volPRF security as that of KCTR-CMAC. In contrast, pKCTR-CMAC is neither preimage resistant nor collision resistant by applying the previous key-control attack [21], which we discuss at the end of this section.

The plain key derivation function pKCTR-CMAC is the same as KCTR-HMAC, except that the underlying PRF is replaced with CMAC. It is also the same as the enhanced version KCTR-CMAC by removing the initial string S_0.

volPRF security. The following theorem shows that pKCTR-CMAC is a good volPRF by assuming the underlying block cipher E_K is a good PRP.

Theorem 6. *For any adversary $\mathcal{A}$ against the volPRF security of* pKCTR-CMAC *that runs in time at most t, makes at most q queries, with each query being of block length at most ℓ and being of output block length at most b where $b \leq 2^r - 1$, we have*

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{pKCTR\text{-}CMAC}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \frac{20q^2b^2}{2^n} + \frac{23qb(\ell+1)^2}{2^n} \ ,$$

by assuming $qb \leq 2^{n/2-1}$ and $\ell + 1 \leq 2^{n/4-0.5}$, where $\mathcal{B}$ is an adversary against the PRP security of the block cipher E that runs in time at most $t' = t + qb(\ell + 2)t_E$ and makes at most $qb(\ell + 2)$ block cipher queries where t_E denotes the running time for one computation of E.

Proof. The proof of this theorem is the same as that of Theorem 3, except that the terms corresponding to the PRF security of HMAC are replaced by those of CMAC which follow from [18, Theorem 3.1].

Discussion. As shown in [21, Appendix B], pKCTR-CMAC is vulnerable to a key control attack (thus, it is not preimage resistant) by Arciszewski et al. of Amazon. At a high level, this attack works as follows. Given a target value $T \in \{0,1\}^n$ and a known key K, we can find a two-block message

$$M = M_1 \parallel M_2 = [1]_2 \parallel \mathrm{Label} \parallel 0x00 \parallel \mathrm{Context} \parallel [L]_2$$

where

$$M_1 = [1]_2 \parallel \mathrm{Label} \parallel 0x00 \parallel \mathrm{Context}_1 \ ,$$

$$M_2 = \mathrm{Context}_2 \parallel [L]_2$$

and

$$\mathrm{Context} = \mathrm{Context}_1 \parallel \mathrm{Context}_2$$

that satisfies $\mathsf{pKCTR\text{-}CMAC}(K, M) = T$ with a high probability, by first selecting $\mathrm{Context}_1$ and then determining $\mathrm{Context}_2$ via the inverse of the block cipher. Using the same method and selecting a different value $\mathrm{Context}'_1$, we can find another two-block message

$$M' = M'_1 \parallel M'_2 = [1]_2 \parallel \mathrm{Label} \parallel 0x00 \parallel \mathrm{Context}' \parallel [L]_2$$

such that $\mathsf{pKCTR\text{-}CMAC}(K, M') = T$ with a high probability. Hence, the pair of (K, M) and (K, M') will result in the same output and pKCTR-CMAC is not collision resistant. We note that Bhaumik et al. [15, Appendix A] demonstrate a similar key-control attack as that by Arciszewski et al. against pKCTR-CMAC for three-block messages. Following the similar idea as above, this key control attack can also be easily turned into a collision attack against pKCTR-CMAC.

procedure KFB-CMAC $(K, \text{Label}, \text{Context}, L)$
$b \leftarrow \lceil L/n \rceil$; $C \leftarrow \varepsilon$
if $b > 2^r - 1$ **then return** $\perp$
$S_0 \leftarrow \mathsf{CMAC}(K, \text{Label} \parallel 0x00 \parallel \text{Context} \parallel [L]_2)$
for $i \leftarrow 1$ **to** b **do**
  $S_i \leftarrow \mathsf{CMAC}(K, S_{i-1} \parallel [i]_2 \parallel \text{Label} \parallel 0x00 \parallel \text{Context} \parallel [L]_2)$
  $C = C \parallel S_i$
return $C[1:L]$

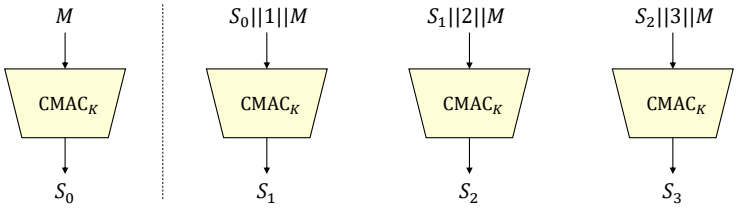


Fig. 6: The pseudocode definition of KFB-CMAC in feedback mode based on CMAC and its pictorial illustration for producing $3n$-bit keying material $C = S_1 \parallel S_2 \parallel S_3$. Here $M = \text{Label} \parallel 0x00 \parallel \text{Context} \parallel [L]_2$.

4 Security Analysis of KDF in Feedback Mode

4.1 Security Analysis of KFB-CMAC

In this subsection, we present security analysis of key derivation function KFB-CMAC based on CMAC in feedback mode. We prove that KFB-CMAC is a secure vol-PRF with the bound $O(\frac{q^2b^2}{2^n} + \frac{qb\ell^2}{2^n})$. We show that KFB-CMAC is collision resistant with the bound $O(\frac{p^2}{2^n} + \frac{p^2\ell}{2^n})$, and is preimage resistant with the bound $O(\frac{p}{2^n} + \frac{p^2\ell}{2^n})$.

Scheme description. The key derivation function KFB-CMAC based on CMAC in feedback mode takes as input a key K, two bit strings called Label and Context, an integer L that specifies the requested length in bits of the derived keying material, and outputs a L-bit keying material C by iterating $\lceil L/n \rceil$ times CMAC where n is the block size of the underlying block cipher. In each iteration of a CMAC execution, the fixed input data is the string $\text{Label} \parallel 0x00 \parallel \text{Context} \parallel [L]_2$. Let

$$S_0 = \mathsf{CMAC}(K, \text{Label} \parallel 0x00 \parallel \text{Context} \parallel [L]_2) \ .$$

The iteration-dependent input data is $S_i \parallel [i]_2$ where

$$S_i = \mathsf{CMAC}(K, S_{i-1} \parallel [i]_2 \parallel \text{Label} \parallel 0x00 \parallel \text{Context} \parallel [L]_2) \ ,$$

for $i \geq 1$ and $[i]_2$ is a counter that is encoded as a string of r bits with $1 \leq r \leq 32$. The pseudocode specification and pictorial illustration of KFB-CMAC are given in Fig. 6.

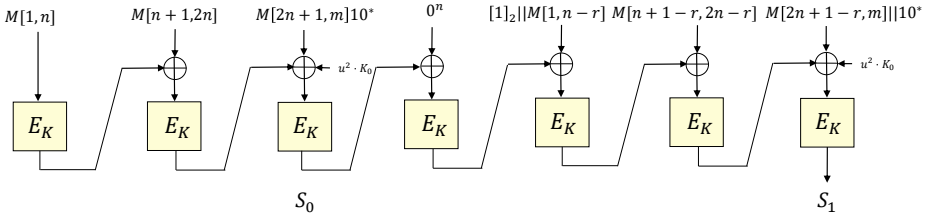


Fig. 7: The pictorial illustration for producing n-bit keying material $C = S_1$ with a message M of m bits where $2n \leq m < 3n$ and $r + m < 3n$.

volPRF security. Given that the block cipher E_K is a good PRP, the key derivation function KFB-CMAC is a good volPRF as shown in the following theorem.

Theorem 7. *For any adversary $\mathcal{A}$ against the volPRF security of* KFB-CMAC *that runs in time at most t, makes at most q queries, with each query being of block length at most ℓ and being of output block length at most b where $b \leq 2^r - 1$, we have*

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KFB\text{-}CMAC}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \frac{20q^2(b+1)^2}{2^n} + \frac{23q(b+1)(\ell+2)^2}{2^n} \ ,$$

by assuming $q(b+1) \leq 2^{n/2-1}$ and $\ell + 2 \leq 2^{n/4-0.5}$, where $\mathcal{B}$ is an adversary against the PRP security of the block cipher E that runs in time at most $t' = t + q(b+1)(\ell+3)t_E$ and makes at most $q(b+1)(\ell+3)$ queries where t_E denotes the running time of one computation of E.

Proof. The procedures to replace the block cipher E_K with a random permutation $\pi \leftarrow\!\!\$\ \mathrm{Perm}(n)$ and to replace the CMAC algorithm with a random function $f \leftarrow\!\!\$\ \mathrm{Func}(*, n)$ are the same as the ones in the proof of Theorem 1. Following similar arguments, we have

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KFB\text{-}CMAC}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \frac{20q^2(b+1)^2}{2^n} + \frac{23q(b+1)(\ell+2)^2}{2^n} + \mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KFB\text{-}CMAC}[f]}(\mathcal{A}) \ ,$$

where $\mathcal{B}$ is an adversary against the PRP security of block cipher E that makes at most $q(b+1)(\ell+3)$ queries and runs in times at most $t' = t + q(b+1)(\ell+3)t_E$.

We then show that the key derivation function KFB-CMAC$[f]$ based on a random function f is the same as a variable output length random function. For any query M to KFB-CMAC$[f]$, the inputs of the random function f to generate the keying material C are $S_0 \,\|\, 1 \,\|\, M, S_1 \,\|\, 2 \,\|\, M, \ldots, S_{b-1} \,\|\, b \,\|\, M$ that are distinct from each other as each input contains a distinct counter i for $1 \leq i \leq b$. Therefore, the corresponding outputs $S_1, S_2, \ldots, S_b$ are independently random strings. Hence, the final L-bit keying material C is a random string as it is a L-bit truncation of the random string $S_1 \parallel S_2 \parallel \ldots \parallel S_b$. For any two distinct queries M and M' to KFB-CMAC$[f]$, the inputs to produce keying materials C

and C' are distinct from each other as $M \neq M'$. Hence, the two keying materials C and C' are independently random strings. From the similar argument, for any q distinct queries to $\mathsf{KFB\text{-}CMAC}[f]$, the outputs of $\mathsf{KFB\text{-}CMAC}[f]$ are q random and independent strings, as the inputs of the underlying random function f to generate these strings are distinct with the help of counter i and non-repeating M in each query. Therefore,

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KFB\text{-}CMAC}[f]}(\mathcal{A}) = 0 \ ,$$

which completes the proof.

Collision resistance. Here we say a collision happens if two derived keying materials have the same first n bits, namely the adversary finds a pair of (K_1, M_1) and (K_2, M_2) such that $\mathsf{KFB\text{-}CMAC}(K_1, M_1)[1:n] = \mathsf{KFB\text{-}CMAC}(K_2, M_2)[1:n]$. The following theorem shows that $\mathsf{KFB\text{-}CMAC}$ is collision resistant.

Theorem 8. *For any adversary $\mathcal{A}$ that makes at most p ideal-cipher queries to E and E^{-1}, we have*

$$\mathsf{Adv}^{\mathsf{coll}}_{\mathsf{KFB\text{-}CMAC}}(\mathcal{A}) \leq \frac{2p^2\ell + p(p+3)}{2^n} \ ,$$

where ℓ is the maximum block length of a query to the key derivation function.

Proof overview. Prior to the formal proof, we provide a high-level overview of the main proof strategy. We upper-bound the probability that the i-th construction query (K_i, M_i) yields an output prefix $S_1^i = \mathsf{KFB\text{-}CMAC}(K_i, M_i)[1:n]$ colliding with some earlier S_1^j for $j < i$. Since one evaluation of $\mathsf{KFB\text{-}CMAC}$ is realized by two iterated CMAC chains of ideal-cipher calls, a collision can only occur if some internal block-cipher input repeats a previous input (an internal collision), or else the final determining internal value is fresh and thus (almost) uniform, so S_1^i hits one of the $i-1$ prior outputs with probability about $(i-1)/2^n$. The proof treats the two structural cases (the last determining step being a forward vs. backward query) by exhibiting a fresh internal variable that is uniform over a set of size at least $2^n - p$ unless an internal collision happens, and then union-bounding all possible internal collisions over at most $O(\ell)$ relevant internal calls, each contributing at most $p/(2^n - p)$. If there is no input collision, each construction query requires at least one new ideal-cipher call, hence the number of construction queries satisfies $q \leq p$. This gives $\Pr[\mathsf{Coll}_i] \leq 2(i+1)/2^n + 2p\ell/2^n$ (using $p \leq 2^{n-1}$), and summing over all $q \leq p$ construction queries yields $\sum_i \Pr[\mathsf{Coll}_i] \leq (2p^2\ell + p(p+3))/2^n$.

Proof. Let $\mathcal{A}$ be an adversary that aims at finding a collision pair for the key derivation function $\mathsf{KFB\text{-}CMAC}$, namely a pair of $(K_1, M_1) \neq (K_2, M_2)$ such that $\mathsf{KFB\text{-}CMAC}(K_1, M_1)[1:n] = \mathsf{KFB\text{-}CMAC}(K_2, M_2)[1:n]$. Assume that $\mathcal{A}$ asks at most p queries to E and E^{-1} in total. Note that to determine the output of key derivation function $\mathsf{KFB\text{-}CMAC}$, it requires one of iterated CMAC chains is fully determined so that the rest of CMAC chains can be determined subsequently.

In the following, we call the query to the key derivation function KFB-CMAC as construction query, and the query to E or E^{-1} as ideal cipher query.

We now consider the probability that a collision happens at the i-th construction query (K_i, M_i) to KFB-CMAC. It implies that there exists some $1 \leq j < i$ such that $\mathsf{KFB\text{-}CMAC}(K_i, M_i)[1:n] = \mathsf{KFB\text{-}CMAC}(K_j, M_j)[1:n]$. The query (K_i, M_i) to KFB-CMAC is built from ideal cipher queries to E and E^{-1}. Denote by m the bit length of message M_i. Let $\ell = \lceil m/n \rceil$. We consider the case of $m + r < n\ell$ where r is the bit length of the binary encoding of counter i. The other case of $m + r \geq n\ell$ can be analyzed similarly. Then the output of $\mathsf{KFB\text{-}CMAC}(K_i, M_i)[1:n]$ is defined as $S_1^i = Y_{2\ell+1}^i$ (see Fig. 7 for the pictorial illustration for the case of a three-block message M), where the internal block cipher calls during the computation of this query are:

$$Y_j^i = E_{K_i}(X_j^i) \text{ for } 1 \leq j \leq 2\ell + 1 \ ,$$

and

$$\begin{aligned}
Y_0^i &= 0^n \\
X_j^i &= Y_{j-1}^i \oplus M_i[n(j-1)+1 : nj] \text{ for } 1 \leq j \leq \ell - 1 \\
X_\ell^i &= Y_{\ell-1}^i \oplus M_i[n(\ell-1)+1 : m] \parallel 10^* \oplus u^2 \cdot E_{K_i}(0^n) \\
X_{\ell+1}^i &= Y_\ell^i \oplus 0^n \\
X_{\ell+2}^i &= Y_{\ell+1}^i \oplus [1]_2 \parallel M_i[1 : n-r] \\
X_{\ell+j+1}^i &= Y_{\ell+j-1}^i \oplus M_i[n(j-1)-r+1 : nj-r] \text{ for } 2 \leq j \leq \ell \\
X_{2\ell+1}^i &= Y_{2\ell}^i \oplus M_i[n(\ell-1)+1-r : m] \parallel 10^* \oplus u^2 \cdot E_{K_i}(0^n) \ .
\end{aligned}$$

As mentioned above, in order to determine the output of this construction query, it requires one of two iterated CMAC chains among the computation of S_1^i is fully determined. Without loss of generality, we assume that the first CMAC chain is fully determined firstly. We then proceed to analyze the collision probability by considering the type of ideal cipher queries to build this CMAC chain.

If the last block cipher call $(K_i, X_{\ell+1}^i, Y_{\ell+1}^i)$ in the first CMAC chain is a forward query, then we can rely on the randomness of $Y_{\ell+1}^i$. However, as the adversary may control the first CMAC chain and choose ideal cipher queries such that $S_0^i = S_0^j$ for some $j < i$ with $K_i = K_j$. In this case, if $[1]_2 \parallel M_i[1, n-r] = [1]_2 \parallel M_j[1, n-r]$, then subsequently $X_{\ell+2}^i = X_{\ell+2}^j$ holds. Namely, there may exist a common prefix between this message and some previous message in the second CMAC chain. To handle this issue, let

$$c = \max\{d \mid \exists j < i\colon\ K_i = K_j \wedge S_0^i = S_0^j \wedge M_i[1 : nd-r] = M_j[1 : nd-r]\} \ .$$

Note that the value d should be in $[0, \ell)$ and cannot be greater than or equal to ℓ, otherwise the input (K_j, M_j) will be the same as the input (K_i, M_i). We proceed to consider the randomness of $Y_{\ell+1+c}^i$. The input to the next block cipher call is $X_{\ell+c+2}^i = Y_{\ell+c+1}^i \oplus M_i[nc-r+1 : n(c+1)-r]$ if $c > 0$ and

$X^i_{\ell+c+2} = Y^i_{\ell+c+1} \oplus [1]_2 \,\|\, M_i[1:n-r]$ if $c = 0$. We say a collision event $\mathsf{fcoll}_{i,\ell+c+2}$ happens if $X^i_{\ell+c+2}$ hits previous inputs of the block cipher E_{K_i}. Then we have

$$\Pr[\mathsf{fcoll}_{i,c}] \le \frac{p_{i,\ell+c+1}}{2^n - p_{i,\ell+c+1}} \ ,$$

as $Y^i_{\ell+c+1}$ is selected uniformly at random from a set of size at least $2^n - p_{i,\ell+c+1}$ where $p_{i,\ell+c+1}$ denotes the total number of ideal cipher queries at the $(\ell+c+1)$-th block cipher call in this construction query. More generally, let $\mathsf{fcoll}_{i,\ell+2+j}$ be the event that $X^i_{\ell+2+j}$ hits previous inputs of the block cipher E_{K_i} for $c \le j \le \ell-1$. Let $\mathsf{Fcoll}_i = \vee_{j=c}^{\ell-1} \mathsf{fcoll}_{i,\ell+2+j}$. Obviously, if the event Fcoll_i does not happen, then S^i_1 is distributed uniformly at random in a set of size at least $2^n - p_{i,2\ell+1}$. Let $\widetilde{\mathsf{Fcoll}}_i$ be the event that S^i_1 does not collide with previous S^j_1 for $1 \le j \le i-1$. Hence, the probability of the event $\widetilde{\mathsf{Fcoll}}_i$ can be bounded by

$$\begin{aligned}
\Pr[\widetilde{\mathsf{Fcoll}}_i] &\le \Pr[\widetilde{\mathsf{Fcoll}}_i \mid \overline{\mathsf{Fcoll}}_i] + \Pr[\mathsf{Fcoll}_i] \\
&\le \frac{i-1}{2^n - p_{i,2\ell+1}} + \Pr[\vee_{j=c}^{\ell-1} \mathsf{fcoll}_{i,\ell+2+j}] \\
&\le \frac{i-1}{2^n - p_{i,2\ell+1}} + \Pr[\mathsf{fcoll}_{i,\ell+2+c}] \\
&\quad + \sum_{j=c+1}^{\ell-1} \Pr[\mathsf{fcoll}_{i,\ell+2+j} \mid \overline{\mathsf{fcoll}}_{i,\ell+2+c} \wedge \cdots \wedge \overline{\mathsf{fcoll}}_{i,\ell+1+j}] \\
&\le \frac{i-1}{2^n - p_{i,2\ell+1}} + \sum_{j=c}^{\ell-1} \frac{p_{i,\ell+1+j}}{2^n - p_{i,\ell+1+j}} \\
&\le \frac{2(i-1)}{2^n} + \frac{2p\ell}{2^n} \ ,
\end{aligned}$$

by assuming $p_{i,j} \le p \le 2^{n-1}$.

If the last block cipher call $(K_i, X^i_{\ell+1}, Y^i_{\ell+1})$ in the first CMAC chain is a backward query, then we can rely on the randomness of its inverse output $X^i_{\ell+1}$. We consider different cases depending on whether the preceding block cipher call is a forward query or a backward query. If the preceding block cipher call $(K_i, X^i_\ell, Y^i_\ell)$ is a forward query, then it requires that $X^i_{\ell+1} = Y^i_\ell \oplus 0^n$, and we denote this event by E_0. Then event E_0 happens with probability $1/(2^n - p_{i,\ell})$ as Y^i_ℓ is distributed uniformly at random in a set of size at least $2^n - p_{i,\ell}$. If the preceding block cipher call $(K_i, X^i_\ell, Y^i_\ell)$ is a backward query, then the adversary can simply set $Y^i_\ell = X^i_{\ell+1}$ and does the backward query to obtain X^i_ℓ. Then we continue to consider previous preceding queries until there is a forward query (K_i, X^i_j, Y^i_j) with $1 < j < \ell$. In this case, as (K_i, X^i_j, Y^i_j) is a forward query, Y^i_j is selected uniformly at random from a set of size at least $2^n - p_{i,j}$. In addition, its succeeding block cipher call $(K_i, X^i_{j+1}, Y^i_{j+1})$ is a backward query, which means that X^i_{j+1} is distributed uniformly at random from a set of size at least $2^n - p_{i,j+1}$. Hence, the message block $M_i[nj+1 : n(j+1)] = X^i_{j+1} \oplus Y^i_j$ is

uniformly distributed in a set of size at least $2^n - \max\{p_{i,j}, p_{i,j+1}\}$. In the case that all of preceding queries are backward queries, then we will reach the first block $M_i[1:n]$ that is the output of $E_{K_i}^{-1}(X_1^i)$ which is distributed uniformly at random in a set of size at least $2^n - p_{i,1}$. Without loss of generality, assume that all of message blocks besides this random message block has been determined during the computation of the first CMAC chain. Denote by $M_i[nc+1:n(c+1)]$ this random message block where $0 \leq c < \ell - 1$. We then analyze the influence of this message block on the second CMAC chain. As this random message block may spread into two block cipher calls with inputs

$$X_{\ell+c+2}^i = Y_{\ell+c+1}^i \oplus M_i[nc - r + 1 : n(c+1) - r]$$

and

$$X_{\ell+c+3}^i = Y_{\ell+c+2}^i \oplus M_i[n(c+1) - r + 1 : n(c+2) - r]$$

if $c > 0$, and with inputs

$$X_{\ell+c+2}^i = Y_{\ell+c+1}^i \oplus [1]_2 \parallel M_i[1 : n - r]$$

and

$$X_{\ell+c+3}^i = Y_{\ell+c+2}^i \oplus M_i[n - r + 1 : 2n - r]$$

if $c = 0$, we consider a joint event for these two block cipher calls. We say the event E_1 happens if $X_{\ell+c+2}^i$ appears in previous inputs of the block cipher E_{K_i}, and the event E_2 happens if $X_{\ell+c+3}^i$ appears in previous inputs of the block cipher E_{K_i}. Then we have

$$\begin{aligned}
\Pr[\mathsf{E}_2] &= \Pr[\mathsf{E}_2 \wedge \mathsf{E}_1] + \Pr[\mathsf{E}_2 \wedge \overline{\mathsf{E}_1}] \\
&= \Pr[\mathsf{E}_2 \mid \mathsf{E}_1] \cdot \Pr[\mathsf{E}_1] + \Pr[\mathsf{E}_2 \mid \overline{\mathsf{E}_1}] \cdot \Pr[\overline{\mathsf{E}_1}] \\
&\leq \Pr[\mathsf{E}_2 \mid \mathsf{E}_1] \cdot \Pr[\mathsf{E}_1] + \Pr[\mathsf{E}_2 \mid \overline{\mathsf{E}_1}] \qquad (6)
\end{aligned}$$

as $\Pr[\overline{\mathsf{E}_1}] \leq 1$. Following similar arguments as in the proof of Theorem 2, we can obtain

$$\Pr[\mathsf{E}_2] \leq \frac{p_{i,\ell+c+2} + 1}{2^n - p_{i,c+1}} + \frac{p_{i,\ell+c+2}}{2^n - p_{i,\ell+c+2}} \ .$$

We then continue to analyze subsequent block cipher calls. Denote by $\mathsf{bcoll}_{i,\ell+c+j}$ the collision event that $X_{\ell+c+j}^i = Y_{\ell+c+j-1}^i \oplus M_i[n(c+j-1) - r + 1 : n(c+j) - r]$ hits previous inputs of the block cipher E_{K_i} for $3 \leq j \leq \ell + 1 - c$. Let $\mathsf{Bcoll}_i = \mathsf{E}_0 \vee \mathsf{E}_2 \vee (\vee_{j=3}^{\ell+1-c} \mathsf{bcoll}_{i,\ell+c+j})$. If Bcoll_i does not happen, then obviously S_1^i is distributed uniformly at random in a set of size at least $2^n - p_{i,2\ell+1}$. Let $\widetilde{\mathsf{Bcoll}}_i$ be the event that S_1^i collides with previous S_1^j for $1 \leq j \leq i - 1$. Thus, the collision

probability at the i-th construction query can be bounded by

$$\begin{aligned}
\Pr[\widetilde{\mathsf{Bcoll}}_i] &\le \Pr[\widetilde{\mathsf{Bcoll}}_i \mid \overline{\mathsf{Bcoll}}_i] + \Pr[\mathsf{Bcoll}_i] \\
&\le \frac{i-1}{2^n - p_{i,2\ell+1}} + \Pr[\mathsf{E}_0 \vee \mathsf{E}_2 \vee (\vee_{j=3}^{\ell+1-c} \mathsf{bcoll}_{i,\ell+c+j})] \\
&\le \frac{i-1}{2^n - p_{i,2\ell+1}} + \Pr[\mathsf{E}_0] + \Pr[\mathsf{E}_2] + \Pr[\mathsf{bcoll}_{i,\ell+c+3}] \\
&\quad + \sum_{j=4}^{\ell+1-c} \Pr[\mathsf{bcoll}_{i,\ell+c+j} \mid \overline{\mathsf{bcoll}_{i,\ell+c+3}} \wedge \cdots \wedge \overline{\mathsf{bcoll}_{i,\ell+c+j-1}} \wedge \overline{\mathsf{E}_2} \wedge \overline{\mathsf{E}_0}] \\
&\le \frac{i-1}{2^n - p_{i,2\ell+1}} + \frac{1}{2^n - p_{i,\ell}} + \frac{p_{i,\ell+c+2}+1}{2^n - p_{i,c+1}} + \sum_{j=3}^{\ell+1-c} \frac{p_{i,\ell+c+j-1}}{2^n - p_{i,\ell+c+j-1}} \\
&\le \frac{2(i+1)}{2^n} + \frac{2p\ell}{2^n} \quad ,
\end{aligned}$$

by assuming $p_{i,j} \le p \le 2^{n-1}$.

Let $\mathsf{Coll}_i = \max\{\widetilde{\mathsf{Fcoll}}_i, \widetilde{\mathsf{Bcoll}}_i\}$ for $1 \le i \le q$. Let $\mathsf{Coll} = \vee_{i=1}^{q} \mathsf{Coll}_i$. Then we finally have

$$\Pr[\mathsf{Coll}] \le \sum_{i=1}^{q} \Pr[\mathsf{Coll}_i] \le \sum_{i=1}^{q} \left(\frac{2(i+1)}{2^n} + \frac{2p\ell}{2^n}\right) \le \frac{2p^2\ell + p(p+3)}{2^n}$$

as $q \le p$ and finish the proof.

Preimage resistance. Given a n-bit value T, we say that the adversary $\mathcal{A}$ finds a preimage against KFB-CMAC if she outputs a pair of (K, M) such that $\mathsf{KFB\text{-}CMAC}(K, M)[1:n] = T$. The following theorem shows that KFB-CMAC is preimage resistant.

Theorem 9. *For any adversary $\mathcal{A}$ that makes at most p ideal-cipher queries to E and E^{-1}, we have*

$$\mathsf{Adv}^{\mathsf{epre}}_{\mathsf{KFB\text{-}CMAC}}(\mathcal{A}) \le \frac{4p}{2^n} + \frac{2p^2\ell}{2^n} \quad ,$$

where ℓ is the maximum block length of a query to the key derivation function.

Proof overview. Before turning to the formal proof, we first outline the main ideas behind the proof. Fix a target value $T \in \{0,1\}^n$ and bound the probability that an adversary outputs a pair (K_i, M_i) such that $S_1^i = \mathsf{KFB\text{-}CMAC}(K_i, M_i)[1:n] = T$ on its i-th construction query. As in the collision analysis, one evaluation of KFB-CMAC is realized by two iterated CMAC chains built from ideal-cipher calls, and the output becomes determined once one of the chains is fully fixed; the proof therefore analyzes the first time a fresh internal block-cipher output (or an induced random message block) is generated during this determination

process. Conditioned on the event that the relevant internal block-cipher inputs do not repeat any prior inputs under K_i, this fresh value is uniform over a set of size at least $2^n - p$, so the chance that it equals the fixed target T is at most $1/(2^n - p)$, and a second such independent opportunity may arise in the backward-last-call case, giving a constant-factor term bounded by $4/2^n$ (using $p \leq 2^{n-1}$). The remaining contribution comes from internal input collisions among the at most $O(\ell)$ block-cipher calls within the evaluation; by a union bound, their total probability is at most about $2p\ell/2^n$. Thus each construction query succeeds with probability at most $4/2^n + 2p\ell/2^n$, and since (if there is no input collision) each construction query requires at least one new ideal-cipher query, we have $q \leq p$; summing over at most $q \leq p$ construction queries yields $\mathsf{Adv}^{\mathsf{pre}}_{\mathsf{KFB\text{-}CMAC}}(\mathcal{A}) \leq 4p/2^n + 2p^2\ell/2^n$.

Proof. Let $\mathcal{A}$ be an adversary against the preimage security of the key derivation function KFB-CMAC. For any fixed image $C \in \{0,1\}^n$, the adversary $\mathcal{A}$ aims at finding a pair of (K, M) satisfying $\mathsf{KFB\text{-}CMAC}(K, M)[1:n] = T$. Assume that the adversary makes at most p queries to E and E^{-1}. We call the query to key derivation function KFB-CMAC as construction query, and the query to E or E^{-1} as ideal cipher query.

Fix an image $T \in \{0,1\}^n$. We then analyze the event P_i that the adversary finds the preimage for T at the i-th construction query (K_i, M_i), namely the event that $\mathsf{KFB\text{-}CMAC}(K_i, M_i)[1:n] = T$. The construction query (K_i, M_i) to KFB-CMAC consists of ideal cipher queries to E and E^{-1}. Following the same notations as in the proof of Theorem 8, let m be the bit length of message M_i. We consider the case of $m + r < n\ell$ where r is the bit length of the encoding of the counter i. The other case of $m + r \geq n\ell$ can be analyzed similarly. Then the output of $\mathsf{KFB\text{-}CMAC}(K_i, M_i)$ is defined as $S_1^i = Y_{2\ell+1}^i$, where internal block cipher calls among the computation of this query are:

$$Y_j^i = E_{K_i}(X_j^i) \text{ for } 1 \leq j \leq 2\ell + 1 \ ,$$

and

$$\begin{aligned}
Y_0^i &= 0^n \\
X_j^i &= Y_{j-1}^i \oplus M_i[n(j-1)+1 : nj] \text{ for } 1 \leq j \leq \ell - 1 \\
X_\ell^i &= Y_{\ell-1}^i \oplus M_i[n(\ell-1)+1 : m] \parallel 10^* \oplus u^2 \cdot E_{K_i}(0^n) \\
X_{\ell+1}^i &= Y_\ell^i \oplus 0^n \\
X_{\ell+2}^i &= Y_{\ell+1}^i \oplus [1]_2 \parallel M_i[1 : n-r] \\
X_{\ell+j+1}^i &= Y_{\ell+j}^i \oplus M_i[n(j-1)-r+1 : nj-r] \text{ for } 2 \leq j \leq \ell \\
X_{2\ell+1}^i &= Y_{2\ell}^i \oplus M_i[n(\ell-1)+1-r : m] \parallel 10^* \oplus u^2 \cdot E_{K_i}(0^n) \ .
\end{aligned}$$

To determine the output of this construction query, it requires that one of two iterated CMAC chains among the computation of S_1^i is fully determined. Without loss of generality, we assume that the first CMAC chain is fully determined at first. We then proceed to analyze the probability that at the i-th construction

query, the output equals to the given image T by considering the type of ideal cipher queries. The following preimage analysis resorts to some results in the proof of Theorem 8.

We first consider the case when the last block cipher call $(K_i, X^i_{\ell+1}, Y^i_{\ell+1})$ in the first CMAC chain is a forward query. Let

$$c = \max\{d \mid \exists j < i\colon\ K_i = K_j \wedge S^i_0 = S^j_0 \wedge M_i[1 : nd - r] = M_j[1 : nd - r]\}\ .$$

Namely, c is the length of the longest common prefix between this message and previous messages in the second CMAC chain. We consider the randomness of $Y^i_{\ell+1+c}$. The input to the next block cipher call is $X^i_{\ell+c+2} = Y^i_{\ell+c+1} \oplus M_i[nc - r + 1 : n(c+1) - r]$ if $c > 0$ and $X^i_{\ell+c+2} = Y^i_{\ell+c+1} \oplus [1]_2 \,\|\, M_i[1 : n - r]$ if $c = 0$. Similar to the proof of Theorem 8, let $\mathsf{fcoll}_{i,\ell+c+2}$ be the event that $X^i_{\ell+c+2}$ hits previous inputs of the block cipher E_{K_i}. Then we have

$$\Pr[\mathsf{fcoll}_{i,\ell+c+2}] \le \frac{p_{i,\ell+c+1}}{2^n - p_{i,\ell+c+1}}\ ,$$

since $Y^i_{\ell+c+1}$ is selected uniformly at random from a set of size at least $2^n - p_{i,\ell+c+1}$ where $p_{i,\ell+c+1}$ denotes the total number of ideal cipher queries at the $(\ell + c + 1)$-th block cipher call in this construction query. More generally, let $\mathsf{fcoll}_{i,\ell+2+j}$ be the event that $X^i_{\ell+2+j}$ hits previous inputs of the block cipher E_{K_i} for $c \le j \le \ell - 1$, and $\mathsf{Fcoll}_i = \vee_{j=c}^{\ell-1}\mathsf{fcoll}_{i,\ell+2+j}$. If the event Fcoll_i does not happen, then S^i_1 is distributed uniformly at random in a set of size at least $2^n - p_{i,2\ell+1}$. Hence, the probability of the event P_i can be bounded by

$$\begin{aligned}\Pr[\mathsf{P}_i] &\le \Pr[\mathsf{P}_i \mid \overline{\mathsf{Fcoll}}_i] + \Pr[\mathsf{Fcoll}_i] \\ &\le \frac{1}{2^n - p_{i,2\ell+1}} + \sum_{j=c}^{\ell-1} \frac{p_{i,\ell+1+j}}{2^n - p_{i,\ell+1+j}} \\ &\le \frac{2}{2^n} + \frac{2p\ell}{2^n}\ ,\end{aligned}$$

by assuming $p_{i,j} \le p \le 2^{n-1}$.

Next, we consider the case when the last block cipher call $(K_i, X^i_{\ell+1}, Y^i_{\ell+1})$ in the first CMAC chain is a backward query. We proceed the analysis depending on whether the preceding block cipher call is a forward query or a backward query. If the preceding block cipher call $(K_i, X^i_\ell, Y^i_\ell)$ is a forward query, then it requires that $X^i_{\ell+1} = Y^i_\ell \oplus 0^n$ that is denoted as the event E_0. Event E_0 happens with probability at most $1/(2^n - p_{i,\ell})$ as Y^i_ℓ is distributed uniformly at random in a set of size at least $2^n - p_{i,\ell}$. If the preceding block cipher call $(K_i, X^i_\ell, Y^i_\ell)$ is a backward query, then the adversary can set $Y^i_\ell = X^i_{\ell+1}$ and asks the backward query to obtain X^i_ℓ. Then we continue to consider previous preceding queries until (K_i, X^i_j, Y^i_j) is a forward query for $1 \le j \le \ell$. In this case, Y^i_j is distributed uniformly at random in a set of size at least $2^n - p_{i,j}$ as (K_i, X^i_j, Y^i_j) is a forward query, while X^i_{j+1} is distributed uniformly at random in a set of size at least $2^n - p_{i,j+1}$ as $(K_i, X^i_{j+1}, Y^i_{j+1})$ is a backward query. Hence,

the message block $M_i[nj+1:n(j+1)] = X^i_{j+1} \oplus Y^i_j$ is uniformly distributed at random in a set of size at least $2^n - \max\{p_{i,j+1}, p_{i,j}\}$. In the case that all of preceding queries are backward queries, then the first message block $M_i[1:n]$ is distributed uniformly at random in a set of size at least $2^n - p_{i,1}$ as it is the output of $E^{-1}_{K_i}(X^i_1)$. Denote by $M_i[nc+1:n(c+1)]$ this random message block where $0 \le c < \ell - 1$. Without loss of generality, assume that all of message blocks including this random message block has been determined during the computation of the first CMAC chain. We then analyze the influence of this random message block on the computation of second CMAC chain. If $c = 0$, then this random message block will spread into two block cipher calls with inputs

$$X^i_{\ell+c+2} = Y^i_{\ell+c+1} \oplus [1]_2 \,\|\, M_i[1:n-r]$$

and

$$X^i_{\ell+c+3} = Y^i_{\ell+c+2} \oplus M_i[n-r+1:2n-r] \ .$$

If $c > 0$, then it will spread into two block cipher calls with inputs

$$X^i_{\ell+c+2} = Y^i_{\ell+c+1} \oplus M_i[nc-r+1:n(c+1)-r]$$

and

$$X^i_{\ell+c+3} = Y^i_{\ell+c+2} \oplus M_i[n(c+1)-r+1:n(c+2)-r] \ .$$

Let E_2 be the event that $X^i_{\ell+c+3}$ hits the previous inputs of the block cipher E_{K_i}. Following the analysis in the proof of Theorem 8, we have

$$\Pr[\mathsf{E}_2] \le \frac{p_{i,\ell+c+2}+1}{2^n - p_{i,c+1}} + \frac{p_{i,\ell+c+2}}{2^n - p_{i,\ell+c+2}} \ .$$

We then analyze subsequent block cipher calls in the second CMAC chain. Let $\mathsf{bcoll}_{i,\ell+c+j}$ be the event that $X^i_{\ell+c+j} = Y^i_{\ell+c+j-1} \oplus M_i[n(j+c-1)-r+1 : n(j+c)-r]$ hits previous inputs of the block cipher E_{K_i} for $3 \le j \le \ell+1-c$. Let $\mathsf{Bcoll}_i = \mathsf{E}_0 \vee \mathsf{E}_2 \vee (\vee_{j=3}^{\ell+1-c} \mathsf{bcoll}_{i,\ell+c+j})$. If Bcoll_i does not happen, then S^i_1 is distributed uniformly at random in a set of size at least $2^n - p_{i,2\ell+1}$. Hence, the probability of the event P_i can be bounded by

$$\begin{aligned}
\Pr[\mathsf{P}_i] &\le \Pr[\mathsf{P}_i \mid \overline{\mathsf{Bcoll}}_i] + \Pr[\mathsf{Bcoll}_i] \\
&\le \frac{1}{2^n - p_{i,2\ell+1}} + \Pr[\mathsf{E}_0 \vee \mathsf{E}_2 \vee (\vee_{j=3}^{\ell+1-c} \mathsf{bcoll}_{i,\ell+c+j})] \\
&\le \frac{1}{2^n - p_{i,2\ell+1}} + \Pr[\mathsf{E}_0] + \Pr[\mathsf{E}_2] + \Pr[\mathsf{bcoll}_{i,\ell+c+3}] \\
&\quad + \sum_{j=4}^{\ell+1-c} \Pr[\mathsf{bcoll}_{i,\ell+c+j} \mid \overline{\mathsf{bcoll}_{i,\ell+c+3}} \wedge \cdots \wedge \overline{\mathsf{bcoll}_{i,\ell+c+j-1}} \wedge \overline{\mathsf{E}_2} \wedge \overline{\mathsf{E}_0}] \\
&\le \frac{1}{2^n - p_{i,2\ell+1}} + \frac{1}{2^n - p_{i,\ell}} + \frac{p_{i,\ell+c+2}+1}{2^n - p_{i,c+1}} + \sum_{j=3}^{\ell+1-c} \frac{p_{i,\ell+c+j-1}}{2^n - p_{i,\ell+c+j-1}} \\
&\le \frac{4}{2^n} + \frac{2p\ell}{2^n} \quad ,
\end{aligned}$$

procedure KFB-HMAC $(K, \text{Label}, \text{Context}, IV, L)$
$b \leftarrow \lceil L/n \rceil$; $C \leftarrow \varepsilon$
if $b > 2^r - 1$ **then return** $\perp$
$S_0 \leftarrow IV$
for $i \leftarrow 1$ **to** b **do**
  $S_i \leftarrow \mathsf{HMAC}(K, S_{i-1} \parallel [i]_2 \parallel \text{Label} \parallel 0x00 \parallel \text{Context} \parallel [L]_2)$
  $C = C \parallel S_i$
return $C[1:L]$

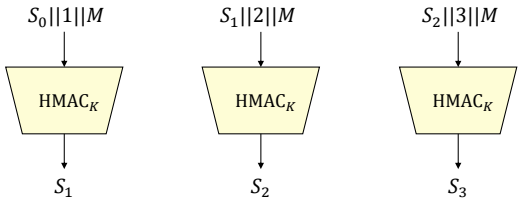


Fig. 8: The pseudocode definition of KFB-HMAC in feedback mode based on HMAC and its pictorial illustration for producing $3n$-bit keying material $C = S_1 \parallel S_2 \parallel S_3$. Here $M = \text{Label} \parallel 0x00 \parallel \text{Context} \parallel [L]_2$. The counter i is optional in KFB-HMAC.

by assuming $p_{i,j} \leq p \leq 2^{n-1}$.

Let $\Pr[\mathsf{P}_i] = \max\{\frac{2}{2^n} + \frac{2p\ell}{2^n}, \frac{4}{2^n} + \frac{2p\ell}{2^n}\} = \frac{4}{2^n} + \frac{2p\ell}{2^n}$. Summing over at most $q \leq p$ construction queries, we have

$$\begin{aligned}\mathsf{Adv}^{\mathsf{epre}}_{\mathsf{KFB\text{-}CMAC}}(\mathcal{A}) &\leq \sum_{i=1}^{q} \Pr[\mathsf{P}_i] \\ &\leq \sum_{i=1}^{q} \frac{4}{2^n} + \frac{2p\ell}{2^n} \\ &\leq \frac{4p}{2^n} + \frac{2p^2\ell}{2^n} \ ,\end{aligned}$$

by assuming $q \leq p \leq 2^{n-1}$ which completes the proof.

4.2 Security Analysis of KFB-HMAC

In this subsection, we present security proofs for key derivation function KFB-HMAC based on HMAC in feedback mode. We prove that KFB-HMAC is a secure volPRF with the bound $O(\frac{q^2b^2}{2^n} + \frac{qb^2}{2^n})$. We show that KFB-HMAC is collision resistant and preimage resistant with the bound $O(\frac{p^2}{2^n})$ when the key is of fixed length and less than $d-1$ bits.

Scheme description. By setting S_0 in the specification of KFB-CMAC to a fixed initial value IV and replacing CMAC with HMAC, we can obtain KFB-HMAC. In addition, the counter i used in each iteration of HMAC becomes optional. The pseudocode description and pictorial illustration of KFB-HMAC are given in Fig. 8.

volPRF security. Assuming that h is a dual-PRF and $\overline{h}$ is a rka-PRF, the following theorem shows that KFB-HMAC is a volPRF. Here the proof is done when the optional counter i is not included in the input of each iteration of HMAC. The proof applies to the case when the counter i is included as the counter can only enhance the security of the scheme and make the analysis easier.

Theorem 10. *Let k be the key length of* KFB-HMAC*, and $\alpha = \lceil k/d \rceil$. For any adversary $\mathcal{A}$ against the volPRF security of* KFB-HMAC *that runs in time at most t, makes at most q queries, with each query being of block length at most ℓ and being of output block length at most b where $b \leq 2^r - 1$, we have*

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KFB\text{-}HMAC}}(\mathcal{A}) \leq (\ell + \alpha + 3)\,\mathsf{Adv}^{\mathrm{prf}}_{h}(\mathcal{A}_1) + \mathsf{Adv}^{\mathrm{prf}}_{\overline{h}}(\mathcal{A}_2) + \mathsf{Adv}^{\mathrm{rkaprf}}_{\Phi_{\mathrm{zio},e},\overline{h}}(\mathcal{B}) + \frac{qb(qb-1)}{2^{n+1}} + \frac{qb^2}{2^{n+1}} \ .$$

Adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ are against the PRF security of h and $\overline{h}$ respectively, where $\mathcal{A}_1$ makes at most qb queries and $\mathcal{A}_2$ makes one query. Adversary $\mathcal{B}$ makes two queries. The running times of adversaries $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{B}$ are about the same as that of $\mathcal{A}$.

Proof. Similar to the proof of Theorem 3, we first replace HMAC of KFB-HMAC with a random function $f \leftarrow\!\!\$\ \mathrm{Func}(\{0,1\}^* \times \{0,1\}^*, \{0,1\}^n)$ and let the resulting key derivation function be KFB-HMAC$[f]$. By [3, Theorem 8] that shows HMAC is a good PRF, we have

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KFB\text{-}HMAC}}(\mathcal{A}) \leq (\ell + \alpha + 3)\,\mathsf{Adv}^{\mathrm{prf}}_{h}(\mathcal{A}_1) + \mathsf{Adv}^{\mathrm{prf}}_{\overline{h}}(\mathcal{A}_2) + \mathsf{Adv}^{\mathrm{rkaprf}}_{\Phi_{\mathrm{zio},e},\overline{h}}(\mathcal{B}) + \frac{qb(qb-1)}{2^{n+1}} + \mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KFB\text{-}HMAC}[f]}(\mathcal{A}) \ ,$$

where $\mathcal{A}_1$ is an adversary against the PRF security of h that makes at most qb queries, $\mathcal{A}_2$ is an adversary against the PRF security of $\overline{h}$ that makes one query, and $\mathcal{B}$ is an adversary against the rka-PRF security of $\overline{h}$ that makes two queries, the running times of who are all about the same as that of $\mathcal{A}$.

We proceed to analyze the volPRF security of KFB-HMAC$[f]$. Recall that we analyze the case where the counter i is not included in the input. At the i-th query M_i to KFB-HMAC$[f]$, the inputs of random function f to generate strings $S^i_1, \ldots, S^i_b$ are $S^i_0 \parallel M_i, \ldots, S^i_{b-1} \parallel M_i$. If S^i_j does not collide with previous $S^i_0, \ldots, S^i_{j-1}$, then the input of random function f is fresh, therewith the corresponding output S^i_{j+1} at this block is a random string. Let $\mathsf{bad}_{i,j}$ be the event that at the i-th query, S^i_j collides with previous $S^i_0, \ldots, S^i_{j-1}$. Note that the inputs of random function f at the i-th query will not collide with the inputs of random function f at the j-th query for $i \neq j$ as they contain two different strings that are M_i and M_j. Let $\mathsf{bad} = \vee^q_{i=1}(\vee^b_{j=1}\mathsf{bad}_{i,j})$. If the event bad does not happen, the outputs of KFB-HMAC$[f]$ are merely random strings. Hence, KFB-HMAC behaves exactly the same as a variable length output random

function if bad does not happen. From the fundamental lemma of game-playing technique [8], we have

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KFB\text{-}HMAC}[f]}(\mathcal{A}) \leq \Pr[\mathsf{bad}] \ ,$$

where

$$\begin{aligned}\Pr[\mathsf{bad}] &\leq \sum_{i=1}^{q}\sum_{j=1}^{b} \mathsf{bad}_{i,j} \\ &\leq \sum_{i=1}^{q}\sum_{j=1}^{b} \frac{j-1}{2^n} \\ &\leq \frac{qb^2}{2^{n+1}} \ .\end{aligned}$$

Combining above equations, we prove the result of this theorem.

Collision resistance. We say a collision happens against KFB-HMAC if the adversary finds a pair of (K_1, M_1) and (K_2, M_2) such that the first n bits of $\mathsf{KFB\text{-}HMAC}(K_1, M_1)$ and $\mathsf{KFB\text{-}HMAC}(K_2, M_2)$ collides. The following theorem shows that KFB-HMAC is collision resistant when the key is of fixed length and less than $d-1$ bits. The proof is omitted as it is similar to the one of Theorem 4. Note that similar to KCTR-HMAC, we can mount collision attack against KFB-HMAC when the key is of arbitrary length.

Theorem 11. *Suppose that the key length of* KFB-HMAC *is fixed and less than* $d-1$ *bits. For any adversary* $\mathcal{A}$ *that makes at most* p *queries to the compression function* h*, we have*

$$\mathsf{Adv}^{\mathsf{coll}}_{\mathsf{KFB\text{-}HMAC}}(\mathcal{A}) \leq \frac{13p^2}{2^n} \ .$$

Preimage resistance. For any $T \in \{0,1\}^n$, we say the adversary finds a preimage against KFB-HMAC if she outputs a pair of (K, M) such that the first n-bit of KFB-HMAC equals to T. The following theorem shows that KFB-HMAC is preimage resistant. The proof is similar to that of Theorem 5, and thus is omitted.

Theorem 12. *Suppose that the key length of* KFB-HMAC *is fixed and less than* $d-1$ *bits. For any adversary* $\mathcal{A}$ *that makes at most* p *queries to the compression function* h*, we have*

$$\mathsf{Adv}^{\mathsf{epre}}_{\mathsf{KFB\text{-}HMAC}}(\mathcal{A}) \leq \frac{13p^2}{2^n} \ .$$

4.3 Security Analysis of pKFB-CMAC

In this subsection, we prove that the plain key derivation function pKFB-CMAC is a secure volPRF with the bound $O(\frac{q^2b^2}{2^n} + \frac{qb\ell^2}{2^n})$. Note that pKFB-CMAC is the

same as pKCTR-CMAC when the output is one block and the initial value IV is an empty string. Hence, similar preimage attack and collision attack as those against pKCTR-CMAC that are discussed in Section 3.3 can be mounted against pKFB-CMAC.

The plain key derivation function pKFB-CMAC is the same as KFB-HMAC when the underlying PRF is replaced with CMAC. It can also be obtained from the enhanced version KFB-CMAC by setting the string S_0 to be a fixed initial value IV and setting the counter i being an optional string.

volPRF security. Assuming that the block cipher E_K is a good PRP, the following theorem demonstrates that pKFB-CMAC is a good volPRF.

Theorem 13. *For any adversary $\mathcal{A}$ against the volPRF security of* pKFB-CMAC *that runs in time at most t, makes at most q queries, with each query being of block length at most ℓ and being of output block length at most b where $b \leq 2^r - 1$, we have*

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{pKFB\text{-}CMAC}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \frac{20q^2b^2}{2^n} + \frac{23qb(\ell+2)^2}{2^n} + \frac{qb^2}{2^{n+1}} \ ,$$

by assuming $qb \leq 2^{n/2-1}$ and $\ell + 2 \leq 2^{n/4-0.5}$, where $\mathcal{B}$ is an adversary against the PRP security of the block cipher E that runs in time at most $t' = t + qb(\ell + 3)t_E$ and makes at most $qb(\ell + 3)$ queries where t_E denotes the running time of one computation of E.

Proof. Note that the only difference between KFB-HMAC and pKFB-CMAC comes from the underlying PRF. Hence, by replacing the terms corresponding to the PRF security of HMAC in Theorem 10 by those of CMAC which again follows from [18, Theorem 3.1], we can obtain the above result.

Remark. Recently, Bhaumik et al. [16] presented an efficient PRF-distinguishing attack on a particular instantiation of a feedback-mode KDF that is independent of the choice of the underlying PRF. Their attack crucially assumes that the IV can be varied and that no counter is used. Consequently, it does not apply to the settings considered in this section, which either fix the IV or incorporate a counter, as also noted by Bhaumik et al. [16].

5 Security Analysis of KDF in Double-Pipeline Mode

5.1 Security Analysis of KDPL-CMAC

In this subsection, we show that the key derivation function KDPL-CMAC based on CMAC in double-pipeline mode is a secure volPRF with the bound $O(\frac{q^2b^2}{2^n} + \frac{qb\ell^2}{2^n})$. We also show that KDPL-CMAC is collision resistant and preimage resistant with bounds $O(\frac{p^2}{2^n} + \frac{p^2\ell}{2^n})$ and $O(\frac{p}{2^n} + \frac{p^2\ell}{2^n})$ respectively.

Scheme description. The key derivation function KDPL-CMAC using CMAC as the PRF in double-pipeline mode takes as input a key K, two strings called

procedure KDPL-CMAC $(K, \text{Label}, \text{Context}, L)$
$b \leftarrow \lceil L/n \rceil$; $C \leftarrow \varepsilon$
if $b > 2^r - 1$ **then return** $\perp$
$A_0 \leftarrow \text{Label} \,\|\, 0x00 \,\|\, \text{Context} \,\|\, [L]_2$
for $i \leftarrow 1$ **to** b **do**
  $A_i \leftarrow \mathsf{CMAC}(K, A_{i-1})$
  $S_i \leftarrow \mathsf{CMAC}(K, A_i \,\|\, [i]_2 \,\|\, \text{Label} \,\|\, 0x00 \,\|\, \text{Context} \,\|\, [L]_2)$
  $C = C \,\|\, S_i$
return $C[1:L]$

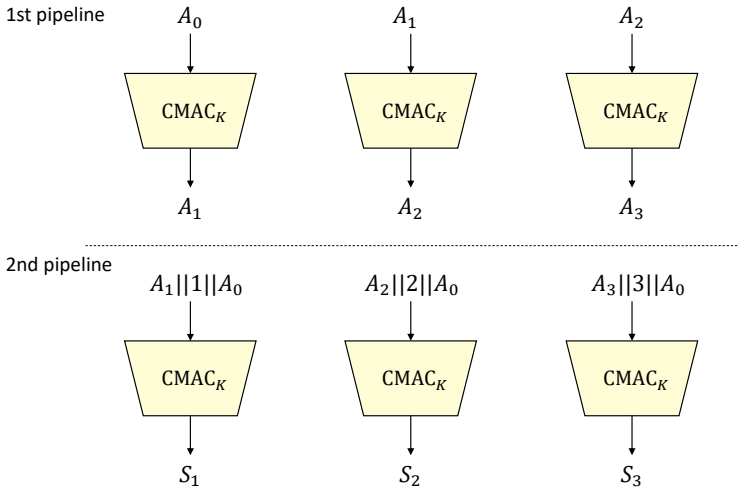


Fig. 9: The pseudocode definition of KDPL-CMAC in double-pipeline mode based on CMAC and its pictorial illustration for producing $3n$-bit keying material $C = S_1 \| S_2 \| S_3$.

Label and Context, an integer L specifying the output length of the derived keying material, and outputs a L-bit string as the derived keying material. In the first pipeline, the CMAC iterations work in a feedback mode with a key K and an initial value $A_0 = \text{Label} \,\|\, 0x00 \,\|\, \text{Context} \,\|\, [L]_2$ to generate strings A_i. In the second pipeline, each CMAC iteration generates a n-bit string S_i with the key K from the fixed input data $\text{Label} \,\|\, 0x00 \,\|\, \text{Context} \,\|\, [L]_2$, the iteration-dependent input data A_i, and a counter $[i]_2$ that is represented as a r-bit string with $1 \leq r \leq 32$. The pseudocode description and pictorial illustration of KDPL-CMAC can be found in Fig. 9.

volPRF security. The following theorem shows that KDPL-CMAC is a good volPRF under the assumption that E_K is a good PRP.

Theorem 14. *For any adversary $\mathcal{A}$ against the volPRF security of* KDPL-CMAC *that runs in time at most t, makes at most q queries, with each query being of block length at most ℓ and being of output block length at most b where $b \leq 2^r - 1$,*

we have

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KDPL\text{-}CMAC}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \frac{80q^2b^2}{2^n} + \frac{46qb(\ell+2)^2}{2^n} \ ,$$

by assuming $2qb \leq 2^{n/2-1}$ *and* $\ell+2 \leq 2^{n/4-0.5}$*, where* $\mathcal{B}$ *is an adversary against the PRP security of the block cipher* E *that runs in time at most* $t' = t + 2qb(\ell+3)t_E$ *and makes at most* $2qb(\ell+3)$ *block cipher queries where* t_E *denotes the running time of* E*.*

Proof. Following similar arguments as in the proof of Theorem 1, we can first replace the block cipher E_K with a random permutation $\pi \leftarrow\!\!\$\ \mathrm{Perm}(n)$ and then replace the CMAC algorithm with a random function $f \leftarrow\!\!\$\ \mathrm{Func}(\{0,1\}^*, \{0,1\}^n)$. We obtain

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KDPL\text{-}CMAC}[E]}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \frac{80q^2b^2}{2^n} + \frac{46qb(\ell+2)^2}{2^n} + \mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KDPL\text{-}CMAC}[f]}(\mathcal{A}) \ ,$$

where $\mathcal{B}$ is an adversary against the PRP security of block cipher E, making at most $2qb(\ell+3)$ queries and runs in times at most $t' = t + 2qb(\ell+3)t_E$.

Next, we consider the advantage of the adversary $\mathcal{A}$ in distinguishing the key derivation function $\mathsf{KDPL\text{-}CMAC}[f]$ based on a random function f from a variable output length random function. As shown in Fig. 9, for any query $M = A_0$ to $\mathsf{KDPL\text{-}CMAC}[f]$, the inputs of the random function f to produce the keying material C are $A_1 \,\|\, 1 \,\|\, A_0, A_2 \,\|\, 2 \,\|\, A_0, \ldots, A_b \,\|\, b \,\|\, A_0$ that are all distinct values. Hence, the outputs $S_1, S_2, \ldots, S_b$ are independently random strings. The final output string C is a L-bit random string as it is the L-bit truncation from the concatenation of these strings. Similarly, for any two distinct queries M and M', the inputs of the random function f to produce keying materials C and C' are all distinct values due to the use of counter i and the constraint $M \neq M'$. Hence, the two derived keying materials are independently random strings. Following similar argument, for any q queries, the corresponding derived keying materials are random strings and independent from each other. Therefore,

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KDPL\text{-}CMAC}[f]}(\mathcal{A}) = 0 \ .$$

Collision resistance. As we can see, for the case of n-bit derived keying material, the key derivation function $\mathsf{KDPL\text{-}CMAC}$ behaves exactly the same as $\mathsf{KFB\text{-}CMAC}$. Hence, we can use the result of Theorem 8 to justify the collision resistance security of $\mathsf{KDPL\text{-}CMAC}$ that is stated as follows.

Theorem 15. *For any adversary* $\mathcal{A}$ *that makes at most* p *ideal-cipher queries to* E *and* E^{-1}*, we have*

$$\mathsf{Adv}^{\mathsf{coll}}_{\mathsf{KDPL\text{-}CMAC}}(\mathcal{A}) \leq \frac{2p^2\ell + p(p+3)}{2^n} \ ,$$

where ℓ *is the maximum block length of a message to the key derivation function.*

PREIMAGE RESISTANCE. Fix an image $T \in \{0,1\}^n$. We say that the adversary finds a preimage against KDPL-CMAC if she finds a pair of (K, M) satisfying $\mathsf{KDPL\text{-}CMAC}(K, M)[1:n] = T$. The following theorem shows that KDPL-CMAC is preimage resistant. As for the n-bit derived keying material, KDPL-CMAC is the same as KFB-CMAC, the proof of this theorem is the same as that of Theorem 9.

Theorem 16. *For any adversary $\mathcal{A}$ that makes at most p ideal-cipher queries to E and E^{-1}, we have*

$$\mathsf{Adv}^{\mathsf{epre}}_{\mathsf{KDPL\text{-}CMAC}}(\mathcal{A}) \leq \frac{4p}{2^n} + \frac{2p^2\ell}{2^n} \ ,$$

where ℓ is the maximum block length of a query to the key derivation function.

5.2 Security Analysis of KDPL-HMAC

In this subsection, we give analysis of the key derivation function KDPL-HMAC based on HMAC in double-pipeline mode. We prove that KDPL-HMAC is a secure volPRF with the bound $O(\frac{q^2b^2}{2^n} + \frac{qb^2}{2^n})$. We show that KDPL-HMAC is collision resistant and preimage resistant with the bound $O(\frac{p^2}{2^n})$.

SCHEME DESCRIPTION. The key derivation function KDPL-HMAC is the same as KDPL-CMAC, except that the CMAC algorithm is replaced with the HMAC algorithm and the counter i becomes optional. The specification and pictorial illustration of KDPL-HMAC can be found in Fig. 10.

VOLPRF SECURITY. The following theorem shows that KDPL-HMAC is a secure volPRF by assuming h is a secure dual-PRF and $\overline{h}$ is a secure rka-PRF. The proof is done for the case when the counter i is not included as the input to HMAC. It can trivially works for the case when the counter i is included.

Theorem 17. *Let k be the key length of* KDPL-HMAC*, and $\alpha = \lceil k/d \rceil$. For any adversary $\mathcal{A}$ against the volPRF security of* KDPL-HMAC *that runs in time at most t, makes at most q queries, with each query being of block length at most ℓ and being of output block length at most b where $b \leq 2^r - 1$, we have*

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KDPL\text{-}HMAC}}(\mathcal{A}) \leq (\ell + \alpha + 3)\,\mathsf{Adv}^{\mathrm{prf}}_{h}(\mathcal{A}_1) + \mathsf{Adv}^{\mathrm{prf}}_{\overline{h}}(\mathcal{A}_2) + \mathsf{Adv}^{\mathrm{rkaprf}}_{\Phi_{\mathsf{zio},e},\overline{h}}(\mathcal{B}) + \frac{2qb(2qb-1)}{2^{n+1}} + \frac{qb^2}{2^n} \ .$$

Adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ are against the PRF security of h and $\overline{h}$ respectively, where $\mathcal{A}_1$ makes at most $2qb$ queries and $\mathcal{A}_2$ makes one query. Adversary $\mathcal{B}$ makes two queries. The running times of adversaries $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{B}$ are about the same as that of $\mathcal{A}$.

Proof. Firstly, we replace the HMAC algorithm in KDPL-HMAC with a random function $f \leftarrow\!\!\$\, \mathrm{Func}(\{0,1\}^* \times \{0,1\}^*, \{0,1\}^n)$ and denote the resulting key

procedure KDPL-HMAC $(K, \text{Label}, \text{Context}, L)$
$b \leftarrow \lceil L/n \rceil$; $C \leftarrow \varepsilon$
if $b > 2^r - 1$ **then return** $\perp$
$A_0 \leftarrow \text{Label} \,\|\, 0x00 \,\|\, \text{Context} \,\|\, [L]_2$
for $i \leftarrow 1$ **to** b **do**
  $A_i \leftarrow \mathsf{HMAC}(K, A_{i-1})$
  $S_i \leftarrow \mathsf{HMAC}(K, A_i \,\|\, [i]_2 \,\|\, \text{Label} \,\|\, 0x00 \,\|\, \text{Context} \,\|\, [L]_2)$
  $C = C \,\|\, S_i$
return $C[1:L]$

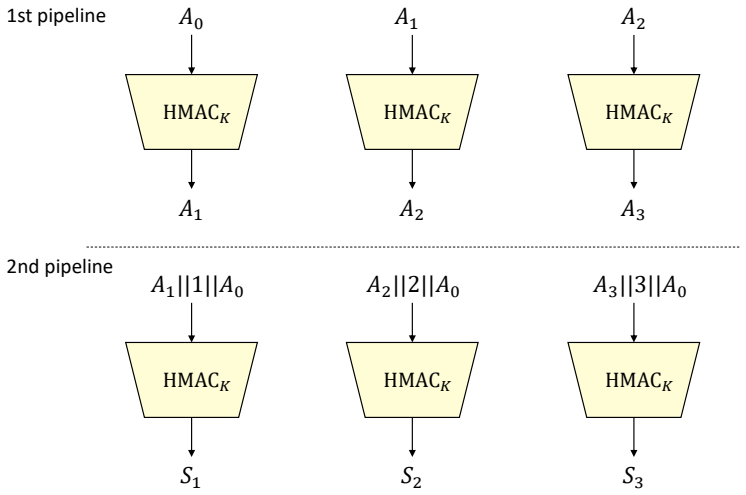


Fig. 10: The pseudocode definition of KDPL-HMAC in double-pipeline mode based on HMAC and its pictorial illustration for producing $3n$-bit keying material $C = S_1 \| S_2 \| S_3$. The counter i is optional in KDPL-HMAC.

derivation function by KDPL-HMAC$[f]$. Following the result of [3, Theorem 8], we obtain

$$\mathsf{Adv}^{\text{volprf}}_{\mathsf{KDPL\text{-}HMAC}}(\mathcal{A}) \le (\ell + \alpha + 3)\,\mathsf{Adv}^{\text{prf}}_{h}(\mathcal{A}_1) + \mathsf{Adv}^{\text{prf}}_{\overline{h}}(\mathcal{A}_2) + \mathsf{Adv}^{\text{rkaprf}}_{\Phi_{\text{zio},e},\overline{h}}(\mathcal{B}) + \frac{2qb(2qb-1)}{2^{n+1}} + \mathsf{Adv}^{\text{volprf}}_{\mathsf{KDPL\text{-}HMAC}[f]}(\mathcal{A}) \ ,$$

where $\mathcal{A}_1$ is an adversary against the PRF security of h that makes at most $2qb$ queries, $\mathcal{A}_2$ is an adversary against the PRF security of $\overline{h}$ that makes one query, and $\mathcal{B}$ is an adversary against the rka-PRF security of $\overline{h}$ that makes two queries, the time complexity of who are all about the same as that of $\mathcal{A}$.

Next, we analyze the volPRF security of KDPL-HMAC$[f]$. For the i-th query M_i to KDPL-HMAC$[f]$, the inputs of random function f to produce strings $S^i_1, \ldots, S^i_b$ are $A^i_1 \,\|\, A^i_0, \ldots, A^i_b \,\|\, A^i_0$ where $A^i_0 = M_i$. If A^i_j is different from $A^i_1, \ldots, A^i_{j-1}$, then the input $A^i_j \,\|\, A^i_0$ of random function f is fresh. Hence, in this case the output S^i_j will be a random string. Denote by $\mathsf{bad}_{i,j}$ the event that at

the i-th query, A^i_j collides with previous $A^i_1, \ldots, A^i_{j-1}$. For two different queries M_i and M_j to $\mathsf{KDPL\text{-}HMAC}[f]$, the inputs of random function f regarding M_i are different from those inputs of random function f regarding M_j as their inputs contain two different strings M_i and M_j. Denote by $\mathsf{bad} = \vee^q_{i=1}(\vee^b_{i=j})\mathsf{bad}_{i,j}$ the union of these events $\mathsf{bad}_{i,j}$ for $1 \le i \le q$ and $1 \le j \le b$. If the event bad does not happen, then the outputs of $\mathsf{KDPL\text{-}HMAC}[f]$ are random strings that are the same as the outputs of a variable output length random function. Hence, by the fundamental lemma of game-playing technique [8], we have

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KDPL\text{-}HMAC}[f]}(\mathcal{A}) \le \Pr[\mathsf{bad}] \ .$$

We now bound the probability of event bad. The occurrence of event $\mathsf{bad}_{i,j}$ requires either as an input A^i_{j-1} hits previous inputs of random function f that are $A^i_0, \ldots, A^i_{j-2}$, or as an output A^i_j hits previous outputs of random function f that are $A^i_1, \ldots, A^i_{j-1}$. Hence, conditioned on previous bad events $\mathsf{bad}_{i,1}, \ldots, \mathsf{bad}_{i,j-1}$ do not happen, we have

$$\Pr[\mathsf{bad}_{i,j} \mid \overline{\mathsf{bad}}_{i,1} \wedge \cdots \wedge \overline{\mathsf{bad}}_{i,j-1}] \le \frac{2(j-1)}{2^n} \ .$$

Therefore,

$$\begin{aligned}
\Pr[\mathsf{bad}] &= \Pr[\vee^q_{i=1}(\vee^b_{i=1})\mathsf{bad}_{i,j}] \\
&\le \sum_{i=1}^{q}\sum_{j=2}^{b} \Pr[\mathsf{bad}_{i,j} \mid \overline{\mathsf{bad}}_{i,1} \wedge \cdots \wedge \overline{\mathsf{bad}}_{i,j-1}] \\
&\le \sum_{i=1}^{q}\sum_{j=1}^{b} \frac{2(j-1)}{2^n} \\
&\le \frac{qb^2}{2^n}
\end{aligned}$$

that concludes the proof.

Collision resistance. Assuming that the compression function $h : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^n$ is a random oracle, the following theorem shows that $\mathsf{KDPL\text{-}HMAC}$ is collision resistant. The proof is similar to the one of Theorem 4, and thus is omitted. Similar to $\mathsf{KCTR\text{-}HMAC}$, we can mount collision attack against $\mathsf{KDPL\text{-}HMAC}$ when the key is of arbitrary length.

Theorem 18. *Suppose that the key length of* $\mathsf{KDPL\text{-}HMAC}$ *is fixed and less than* $d-1$ *bits. For any adversary* $\mathcal{A}$ *that makes at most* p *queries to the compression function* h*, we have*

$$\mathsf{Adv}^{\mathsf{coll}}_{\mathsf{KDPL\text{-}HMAC}}(\mathcal{A}) \le \frac{13p^2}{2^n} \ .$$

Preimage resistance. Assuming that the compression function $h : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^n$ is a random oracle, the following theorem shows $\mathsf{KDPL\text{-}HMAC}$ is preimage resistant. The proof relies on the indifferentialibity of HMAC which is similar to the proof of Theorem 12, and thus is omitted.

Theorem 19. *Suppose that the key length of* KDPL-HMAC *is fixed and less than* $d-1$ *bits. For any adversary* $\mathcal{A}$ *that makes at most* p *queries to the compression function* h*, we have*

$$\mathsf{Adv}^{\mathsf{epre}}_{\mathsf{KDPL\text{-}HMAC}}(\mathcal{A}) \leq \frac{13p^2}{2^n} \ .$$

5.3 Security Analysis of pKDPL-CMAC

In this subsection, we prove that the plain key derivation function pKDPL-CMAC based on CMAC in double-pipeline mode is a secure volPRF with the bound $O(\frac{q^2b^2}{2^n} + \frac{qb\ell^2}{2^n})$. We show that pKDPL-CMAC is collision resistant and preimage resistant with bounds $O(\frac{p^2}{2^n} + \frac{p^2\ell}{2^n})$ and $O(\frac{p}{2^n} + \frac{p^2\ell}{2^n})$, respectively. Our provable security results demonstrate that the plain key derivation function pKDPL-CMAC enjoys almost the same security as that of the enhanced version KDPL-CMAC. Hence, the effect of requiring the inclusion of a counter in double-pipeline mode is limited. Interestingly, similar implication can also be obtained from the birthday-bound attacks against pKDPL-CMAC and KDPL-CMAC as shown by Bhaumik et al. [14].

The plain key derivation function pKDPL-CMAC is the same as KDPL-HMAC, except that the underlying PRF is instantiated with CMAC. It is also the same as the enhanced version KDPL-CMAC when the counter i is set to be an optional string.

volPRF security. Given that the block cipher E_K is a good PRP, the following theorem shows that pKDPL-CMAC is a good volPRF.

Theorem 20. *For any adversary* $\mathcal{A}$ *against the volPRF security of* pKDPL-CMAC *that runs in time at most* t*, makes at most* q *queries, with each query being of block length at most* ℓ *and being of output block length at most* b *where* $b \leq 2^r - 1$*, we have*

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{pKDPL\text{-}CMAC}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{prp}}_{E}(\mathcal{B}) + \frac{80q^2b^2}{2^n} + \frac{46qb(\ell+2)^2}{2^n} + \frac{qb^2}{2^n} \ ,$$

by assuming $2qb \leq 2^{n/2-1}$ *and* $\ell + 2 \leq 2^{n/4-0.5}$*, where* $\mathcal{B}$ *is an adversary against the PRP security of the block cipher* E *that runs in time at most* $t' = t + 2qb(\ell + 3)t_E$ *and makes at most* $2qb(\ell + 3)$ *block cipher queries where* t_E *denotes the running time of* E*.*

Proof. Similar to the proof of Theorem 13, the above result follows by replacing the terms corresponding to the PRF security of HMAC in Theorem 17 with those of CMAC.

Collision resistance. We say that a collision occurs if the adversary finds a pair of (K_1, M_1) and (K_2, M_2) such that $\mathsf{pKDPL\text{-}CMAC}(K_1, M_1)[1 : n] = \mathsf{pKDPL\text{-}CMAC}(K_2, M_2)[1 : n]$. As for the case of n-bit derived keying material, pKDPL-CMAC is the same as KFB-CMAC except that the counter i becomes an optional string. As demonstrated in the collision-resistance analysis of

KFB-CMAC in Section 4.1, the counter does not affect the proof. Thus, the proof of Theorem 8 also works for pKDPL-CMAC.

Theorem 21. *For any adversary $\mathcal{A}$ that makes at most p ideal-cipher queries to E and E^{-1}, we have*

$$\mathsf{Adv}^{\mathsf{coll}}_{\mathsf{pKDPL\text{-}CMAC}}(\mathcal{A}) \leq \frac{2p^2\ell + p(p+3)}{2^n} \ ,$$

where ℓ is the maximum block length of a message to the key derivation function.

Preimage resistance. Given an image $T \in \{0,1\}^n$, we say that an adversary finds a preimage against pKDPL-CMAC if she finds a pair of (K, M) such that $\mathsf{pKDPL\text{-}CMAC}(K, M)[1:n] = T$. The following theorem demonstrates that pKDPL-CMAC is preimage resistant. Note that for the case of n-bit derived keying material, pKDPL-CMAC is the same as KFB-CMAC when the counter i is set to be an optional string. As we can see in the preimage-resistance analysis of KFB-CMAC in Section 4.1, the counter has no impact on the proof. Hence, the proof of Theorem 9 also works for pKDPL-CMAC.

Theorem 22. *For any adversary $\mathcal{A}$ that makes at most p ideal-cipher queries to E and E^{-1}, we have*

$$\mathsf{Adv}^{\mathsf{epre}}_{\mathsf{pKDPL\text{-}CMAC}}(\mathcal{A}) \leq \frac{4p}{2^n} + \frac{2p^2\ell}{2^n} \ ,$$

where ℓ is the maximum block length of a query to the key derivation function.

6 Security Analysis of KMAC-based KDF

In this section, we present security analysis of the key derivation function KDF-KMAC based on KMAC. We prove that KDF-KMAC is a secure volPRF with the bound of $O(\frac{pq+q^2}{2^c} + \frac{\sigma^2}{2^n} + \frac{\sigma q}{2^{c+k}} + \frac{pq}{2^k})$. We show that KDF-KMAC is collision resistant and preimage resistant with the bound $O(\frac{p^2}{2^c})$.

Scheme description. The key derivation function KDF-KMAC based on KMAC takes as input a key K, two strings called Label and Context, an integer L specifying the output length of the derived keying material, and outputs a L-bit string as the derived keying material. The pseudocode description and pictorial illustration of KDF-KMAC are presented in Fig. 11.

We also present the specification of KMAC [26] in Fig. 12 for the reference. KMAC (Keccak Message Authentication Code) is a keyed hash function whose output is of variable length. It follows the sponge construction [9] and is built from a random permutation P with size n. Here $n = c+r$, where c is the capacity and r is the rate. KMAC takes the following parameters: a key K of length k, the main input string X of variable length, an integer L representing the requested output length, an optional customization string S of variable length. It also

procedure KDF-KMAC $(K, \text{Label}, \text{Context}, L)$
if $L > 2^{1040} - 1$ **then return** $\perp$
$C \leftarrow \mathsf{KMAC}(K, \text{Context}, L, \text{Label})$
return C

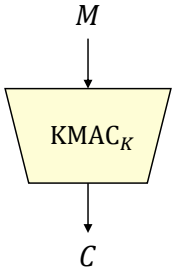


Fig. 11: The pseudocode definition of key derivation function KDF-KMAC based on KMAC and its pictorial illustration where $M = (\text{Context}, L, \text{Label})$.

procedure KMAC (K, X, L, S)
$b \leftarrow \lceil L/r \rceil$; $C = \varepsilon$; $N =$ "KMAC"
$M \leftarrow \text{kpad}(N \parallel S) \parallel \text{kpad}(K) \parallel \text{kpad}(X \parallel L \parallel 00)$
$M_1 \ldots M_m \leftarrow_r M$; $Z \leftarrow 0^n$
for $i \leftarrow 1$ **to** m **do**
  $Z \leftarrow P(Z \oplus (M_i \parallel 0^c))$
for $i \leftarrow 1$ **to** b **do**
  $C = C \parallel Z[1:r]$
  $Z = P(Z)$
return $C[1:L]$

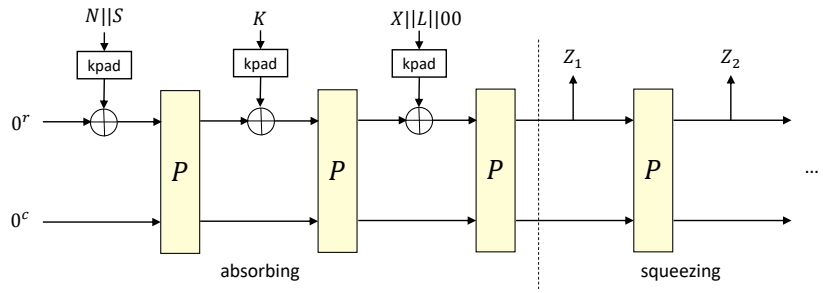


Fig. 12: The pseudocode definition of KMAC and its pictorial illustration where $X \| L \| 00$ becomes a r-bit block after being padded.

includes a fixed string N that is the name "KMAC". Given these inputs, KMAC first processes a block formatted from the strings N and S. Then it handles a block of the padded key K, and absorbs message blocks. After absorbing, it squeezes an L-bit string by iteratively invoking the permutation P. Each input string is padded first before being absorbed via an injective padding function

kpad. The details of this padding function are irrelevant to the security analysis here, and thus is omitted.

volPRF security. KMAC can be regarded as a keyed sponge construction. The security of the key sponge construction has been studied in a series of works [11,17,1,31,12,30]. Daemen et al. [22] and Mennink [29] observed that the keyed sponge construction can also be viewed as an extension of the generalized keyed duplex construction. They proved the security for this general keyed duplex construction. Based on their results [22, Theorem 1] and [29, Theorem 6], we can derive the volPRF security of KDF-KMAC that is formalized by the following theorem.

Theorem 23. *Let n, c, r, k be positive integers where n is the size of permutation P with $n = c+r$ and k is the key length. For any adversary $\mathcal{A}$ against the volPRF security of* KDF-KMAC *that makes at most q queries to* KMAC *of total length at most σ permutation calls, p queries to the ideal permutation P, we have*

$$\mathsf{Adv}^{\mathrm{volprf}}_{\mathsf{KDF\text{-}KMAC}}(\mathcal{A}) \leq \frac{2\nu^{2\sigma}_{r,c}(p+1)}{2^c} + \frac{(q-1)p + q(q-1)/2}{2^c} + \frac{(\sigma-q)q}{2^n - q} + \frac{\sigma(\sigma-1)}{2^n} + \frac{q(\sigma-q)}{2^{\min\{c+k,n\}}} + \frac{pq}{2^k} \ ,$$

by assuming $\sigma + p \leq 2^c/10$.

Proof. The above result is a direct application of [22, Theorem 1] to the volPRF security of KMAC. The proof is almost the same as that of [22, Theorem 1]. The only difference is that KMAC uses a slightly different initialization to initiate the state with the key K. Taking a closer look at the analysis of Daement et al. [22], the only influenced term is the last term in their bound. The rooted event behind this term comes from the collision between two initial states, namely there exists two distinct pairs of (N, S) and (N', S') such that $P(\mathrm{kpad}(N \parallel S) \parallel 0^c) = P(\mathrm{kpad}(N' \parallel S') \parallel 0^c)$. This event occurs with probability 0 for KMAC as two initial states never collide for a distinct pair of (N, S) and (N', S'). Hence, this term becomes 0. The rest parts of proof are the same. We also refer to [29, Theorem 6] for a detailed explanation of how to understand the bound of [22, Theorem 1] for the case of keyed sponge construction.

Discussion. The dominating terms in the above theorem are $O(\frac{pq+q^2}{2^c} + \frac{\sigma^2}{2^n} + \frac{\sigma q}{2^{c+k}} + \frac{pq}{2^k})$. Here $\nu^{2\sigma}_{r,c}$ is the multicollision limit function that is determined by parameters σ, r, and c. It is a constant value for most practical parameters. We refer to [22, Section 6.5] for the detailed discussion about the behavior of this function.

Collision resistance. Assuming that P is an ideal random permutation and the output length of KDF-KMAC satisfies $L \geq c$, the following theorem shows that KDF-KMAC is collision resistant. The result of this theorem can be obtained from the indifferentiability security result of the sponge construction by Bertoni et al. [10, Theorem 2].

Theorem 24. *For any adversary $\mathcal{A}$ that makes at most p queries to the permutation P, we have*

$$\mathsf{Adv}^{\mathsf{coll}}_{\mathsf{KDF\text{-}KMAC}}(\mathcal{A}) \leq \frac{p^2}{2^{c+1}} + \frac{p}{2^c} \ .$$

Preimage resistance. Similarly, assuming that P is an ideal random permutation and the output length of KDF-KMAC satisfies $L \geq c$, the preimage resistance of KDF-KMAC can be obtained from the indifferentiability of the sponge construction, which is formalized by the following theorem.

Theorem 25. *For any adversary $\mathcal{A}$ that makes at most p queries to the permutation P, we have*

$$\mathsf{Adv}^{\mathsf{pre}}_{\mathsf{KDF\text{-}KMAC}}(\mathcal{A}) \leq \frac{p^2}{2^{c+1}} + \frac{p}{2^c} \ .$$

7 Concluding Remarks

In this article, we give a comprehensive security analysis of key derivation functions using pseudorandom functions in NIST SP 800-108 [21]. Based on our analyses, KCTR-CMAC may not be a good choice in general as a KDF as it is vulnerable to constant-time collision attack. While for other KDFs, including KFB-CMAC, KDPL-CMAC (as well as its plain version pKDPL-CMAC), KCTR-HMAC, KFB-HMAC, KDPL-HMAC (KDFs based on HMAC should use a key of fixed length that is less than $d-1$ bits), and KDF-KMAC, they are all basically good as a KDF, as all of them are volPRF, collision resistant, and preimage resistant. Overall, we believe these results contribute to the community for deeper understanding of the current version of standard NIST SP 800-108. We personally suggest that the standard should be revised for a stronger security, especially regarding the algorithm KCTR-CMAC as it is not collision resistant (or not context binding) as required by this standard.

We remark that most security bounds proved in this article are essentially tight in terms of their query complexity, although a few are not known to be tight. Concretely, the birthday bounds for the volPRF security of CMAC-based and HMAC-based KDFs are tight, in light of the generic PRF distinguishing attack against iterated MACs due to Preneel and Oorschot [33]. The birthday bounds for collision resistance of CMAC-based and HMAC-based KDFs are tight as well, since the above PRF distinguisher can be converted into a collision-finding attack as it succeeds by detecting internal-state collisions. The birthday bound for collision resistance of the KMAC-based KDF is tight as demonstrated by Andreeva et al. [2]. Finally, the birthday bounds for preimage resistance of CMAC-based KDFs are also tight due to the recent birthday-complexity key-control attacks of Bhaumik et al. [14,15]. In contrast, for the volPRF security of the KMAC-based KDF, as well as the preimage-resistance bounds for the HMAC-based and KMAC-based KDFs, no explicit matching attacks are currently known. We leave closing these gaps to future work.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2024YFB4504800), the National Natural Science Foundation of China (62402404, 62372294, U2336210), the Fundamental Research Funds for the Central Universities (20720250030). This work was also partly supported by Shanghai Pudong Research Institute of Cryptology (SPRIC) under its R&D Program in 2026.

References

1. E. Andreeva, J. Daemen, B. Mennink, and G. V. Assche. Security of keyed sponge constructions using a modular proof approach. In *Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers*, pages 364–384, 2015.
2. E. Andreeva, B. Mennink, and B. Preneel. Security reductions of the second round SHA-3 candidates. In *Information Security - 13th International Conference, ISC 2010, Boca Raton, FL, USA, October 25-28, 2010, Revised Selected Papers*, pages 39–53, 2010.
3. M. Backendal, M. Bellare, F. Günther, and M. Scarlata. When messages are keys: Is HMAC a dual-prf? In *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III*, pages 661–693, 2023.
4. E. Barker, L. Chen, R. Davis, et al. Recommendation for key-derivation methods in key-establishment schemes. *NIST Special Publication*, 800:56C, 2018.
5. M. Bellare. New proofs for NMAC and HMAC: security without collision-resistance. In *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, pages 602–619, 2006.
6. M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, pages 1–15, 1996.
7. M. Bellare and T. Kohno. A theoretical treatment of related-key attacks: Rka-prps, rka-prfs, and applications. In *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, pages 491–506, 2003.
8. M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In *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*, pages 409–426, 2006.
9. G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche. Sponge functions. *Ecrypt Hash Workshop 2007*, 2007.
10. G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche. On the indifferentiability of the sponge construction. In *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, pages 181–197, 2008.

11. G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche. On the security of the keyed sponge construction. *Symmetric Key Encryption Workshop*, 2011.
12. G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche. Permutation-based encryption, authentication and authenticated encryption. *Directions in Authenticated Ciphers*, 2012.
13. R. Bhaumik, B. Chakraborty, W. Choi, A. Dutta, J. Govinden, and Y. Shen. The committing security of macs with applications to generic composition. In *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part IV*, pages 425–462, 2024.
14. R. Bhaumik, A. Dutta, A. Inoue, T. Iwata, A. Jha, K. Minematsu, M. Nandi, Y. Sasaki, M. S. Turan, and S. Tessaro. Cryptographic treatment of key control security - in light of NIST SP 800-108. In *Advances in Cryptology - CRYPTO 2025 - 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2025, Proceedings, Part V*, pages 371–403, 2025.
15. R. Bhaumik, A. Dutta, A. Inoue, T. Iwata, A. Jha, K. Minematsu, M. Nandi, Y. Sasaki, M. S. Turan, and S. Tessaro. Cryptographic treatment of key control security - in light of NIST SP 800-108. *IACR Cryptol. ePrint Arch.*, page 1123, 2025.
16. R. Bhaumik, A. Dutta, T. Iwata, A. Jha, K. Minematsu, M. Nandi, Y. Sasaki, M. S. Turan, and S. Tessaro. A note on feedback-prf mode of KDF from NIST SP 800-108. *IACR Cryptol. ePrint Arch.*, page 1586, 2025.
17. D. Chang, M. Dworkin, S. Hong, J. Kelsey, and M. Nandi. A keyed sponge construction with pseudorandomness in the standard model. *NIST SHA–3 Workshop*, 2012.
18. S. Chattopadhyay, A. Jha, and M. Nandi. Towards tight security bounds for omac, XCBC and TMAC. In *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part I*, pages 348–378, 2022.
19. L. Chen. Recommendation for key derivation using pseudorandom functions - nist sp 800-108. 2008.
20. L. Chen. Recommendation for key derivation using pseudorandom functions - nist sp 800-108r1. 2022.
21. L. Chen. Recommendation for key derivation using pseudorandom functions - nist sp 800-108r1-upd1. 2024.
22. J. Daemen, B. Mennink, and G. V. Assche. Full-state keyed duplex with built-in multi-user support. In *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*, pages 606–637, 2017.
23. Y. Dodis, T. Ristenpart, J. P. Steinberger, and S. Tessaro. To hash or not to hash again? (in)differentiability results for H 2 and HMAC. In *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, pages 348–366, 2012.
24. M. J. Dworkin. Recommendation for block cipher modes of operation: The cmac mode for authentication. 2016.
25. T. Iwata and K. Kurosawa. OMAC: one-key CBC MAC. In *Fast Software Encryption, 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003, Revised Papers*, pages 129–153, 2003.

26. J. Kelsey, S.-j. Chang, and R. Perlner. Nist special publication 800–185: sha-3 derived functions: cshake, kmac, tuplehash and parallelhash. *National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep*, 2016.
27. H. Krawczyk. Cryptographic extraction and key derivation: The HKDF scheme. In *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, pages 631–648, 2010.
28. J. Mattsson. Public comments on sp 800-38c. 2024.
29. B. Mennink. Understanding the duplex and its security. *IACR Trans. Symmetric Cryptol.*, 2023(2):1–46, 2023.
30. B. Mennink, R. Reyhanitabar, and D. Vizár. Security of full-state keyed sponge and duplex: Applications to authenticated encryption. In *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, pages 465–489, 2015.
31. Y. Naito and K. Yasuda. New bounds for keyed sponges with extendable output: Independence between capacity and message length. In *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, pages 3–22, 2016.
32. M. Nandi. Improved security analysis for OMAC as a pseudorandom function. *J. Math. Cryptol.*, 3(2):133–148, 2009.
33. B. Preneel and P. C. van Oorschot. Mdx-mac and building fast macs from hash functions. In *Advances in Cryptology - CRYPTO '95, 15th Annual International Cryptology Conference, Santa Barbara, California, USA, August 27-31, 1995, Proceedings*, pages 1–14, 1995.
34. P. Rogaway and T. Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers*, pages 371–388, 2004.
35. M. Stäuble. Actually good encryption? confusing users by changing nonces. *Semester project, Department of Computer Science, ETH Zürich*, page 8, 2022.