Revisiting Structure Graph and Its Applications to CBC-MAC and EMAC

Ashwin Jha and Mridul Nandi

Indian Statistical Institute, Kolkata
ashwin.jha1991@gmail.com, mridul.nandi@gmail.com

Abstract. In Crypto'05, Bellare et al. proved $O(\ell q^2/2^n)$ bound for the PRF (pseudorandom function) security of the CBC-MAC based on an n-bit random permutation Π, provided $\ell < 2^{n/3}$. Here an adversary can make at most q prefix-free queries each having at most ℓ "*blocks*" (elements of $\{0,1\}^n$). In the same paper $O(\ell^{o(1)} q^2/2^n)$ bound for EMAC (or encrypted CBC-MAC) was proved, provided $\ell < 2^{n/4}$. Both proofs are based on **structure graphs** representing all collisions among "*intermediate inputs*" to Π during the computation of CBC. The problem of bounding PRF-advantage is shown to be reduced to bounding the number of structure graphs satisfying certain collision patterns. Unfortunately, we have shown here that *the Lemma 10 in the Crypto'05 paper, stating an important result on structure graphs, is incorrect.* This is due to the fact that the authors **overlooked certain structure graphs**. This invalidates the proofs of the PRF bounds. In ICALP'06, Pietrzak improved the bound for EMAC by showing a *tight bound* $O(q^2/2^n)$ under the restriction that $\ell < 2^{n/8}$. As he used the same flawed lemma, this proof also becomes invalid. In this paper, we have revised and sometimes simplified these proofs. We revisit structure graphs in a slightly different mathematical language and provide a complete characterization of certain types of structure graphs. Using this characterization, we show that PRF security of CBC-MAC is about $\sigma q/2^n$ provided $\ell < 2^{n/3}$ where σ is the total number of blocks in all queries. We also recovered the tight bound of EMAC with a much relaxed constraint $\ell < 2^{n/4}$ than the original.

Keywords: CBC, EMAC, structure graph, accident, random permutation.

1 Introduction

Brief History on CBC and EMAC. The notion of authentication in cryptographic protocols was first introduced by Diffie and Hellman in their seminal paper [9] of 1976. In symmetric key settings, this need is fulfilled by message authentication codes, better known as MACs. CBC-MAC is a block cipher based MAC construction which is based on the CBC mode of operation invented by Ehrsam et al. [13]. The CBC-MAC was an international standard [1] which had been proved to be secure for fixed length messages [2, 5] or prefix-free message spaces [30, 16]. The fixed length constraint is not desired in practice. One way to circumvent this is to use the length of message as the first block in CBC computation. This requires prior knowledge of message length. A more reasonable and popular approach is to encrypt the CBC output with an independent keyed permutation. This later approach is called the EMAC which has been proved to be secure without any restrictions on the message [30]. We refer readers to section 2 for a brief overview of literature related to CBC-MAC.

CBC AND EMAC FUNCTIONS. Throughout the paper, we fix a positive integer n and let $\mathcal{B} := \{0,1\}^n$. Elements of these sets are called *blocks*. Let $\mathrm{Perm} := \mathrm{Perm}(n)$ be the set of all permutations over $\mathcal{B}$. The CBC (cipher block chaining) function with key $\pi \in \mathrm{Perm}$, denoted CBC_π, takes as input a message $M = (M[1], \ldots, M[m]) \in \mathcal{B}^m$ and outputs a block $\mathrm{out}^\pi(M)[m]$ which is inductively computed as $\mathrm{out}^\pi(M)[0] = 0^n$ and

$$\mathrm{out}^\pi(M)[i] = \pi(\mathrm{out}^\pi(M)[i-1] \oplus M[i]), \quad i = 1, \ldots, m.$$

For $0 < i < m$, $\mathrm{in}^\pi(M)[i] := \mathrm{out}^\pi(M)[i-1] \oplus M^i$ and $\mathrm{out}^\pi(M)[i]$ are said to be the *intermediate input and output* respectively. Fig 3 in section 3 provides an illustration of CBC computation and intermediate values.

SECURITY DEFINITIONS. In this paper we consider two types of attacks for an adversary which makes queries of at most ℓ blocks: $\mathsf{atk} = \mathsf{pf}$ and $\mathsf{atk} = \mathsf{any}$ mean no query is a prefix of another and the queries are arbitrary distinct strings, respectively. Let $\mathbf{Adv}_{\mathrm{F}}^{\mathsf{atk}}(q, \ell, \sigma)$ denote the *maximum advantage attainable by any adversary making q queries and the total number of blocks in all q queries is at most σ*, mounting an atk attack, in distinguishing whether its oracle is F,[1] or a random function that outputs n bits. To analyze the security of CBC and EMAC for the random permutation Π, the *collision probability* and *full collision probability*

$$\mathbf{CP}_n(M_1, M_2) \overset{\mathrm{def}}{=} \Pr_\Pi[\mathrm{CBC}_\Pi(M_1) = \mathrm{CBC}_\Pi(M_2)]$$
$$\mathbf{FCP}_n(M_1, M_2) \overset{\mathrm{def}}{=} \Pr_\Pi[\mathrm{out}^\Pi(M_2)[m_2] = \mathrm{out}^\Pi(M_r)[j]; \exists\ (r, j) \neq (2, m_2)]$$

had been introduced for distinct messages M_1 and M_2. Moreover, let $\mathbf{CP}_{2,\ell}^{\mathsf{atk}}$ and $\mathbf{FCP}_{2,\ell}^{\mathsf{atk}}$ denote the maximum collision and full collision probabilities respectively where the maximum is taken over all distinct messages M, M' having at most ℓ blocks and satisfies atk. In [3], the following results had been shown:

$$\mathbf{Adv}_{\mathrm{EMAC}}^{\mathsf{any}}(q, \ell) \le \binom{q}{2}(\mathbf{CP}_{2,\ell}^{\mathsf{any}} + 2^{-n}), \ \ \mathbf{Adv}_{\mathrm{CBC}}^{\mathsf{pf}}(q, \ell) \le q^2(\mathbf{FCP}_{2,\ell}^{\mathsf{pf}} + 4\ell/2^n). \tag{1}$$

As EMAC encrypts output of CBC-MAC under an independent key, as long as there is no collision in the output of CBC-MAC, the final output behaves

[1] In this paper, it is either CBC or EMAC based on the random permutation Π on n bits (i.e., Π is chosen uniformly from Perm).

randomly. This is essentially same as the Carter-Wegman construction [33]. The CBC-MAC function can be similarly viewed as a (dependent) nested construction in which the final encryption is computed under the same key as the internal computation. This is why we need an extended definition of collision which is appropriately captured by the full collision event. Thus, bounding PRF advantages are reduced to bounding (full) collision probabilities. These are again reduced to bounding the number of structure graphs as described in the following paragraph.

STRUCTURE GRAPH. **Block-vertex structure graph G** associated to a message M and a permutation π, is the directed edge labeled graph induced by the edge-set E consists of all edges

$$e_i : \mathrm{out}^{\pi}(M)[i-1] \to \mathrm{out}^{\pi}(M)[i] := (\mathrm{out}^{\pi}(M)[i-1], \mathrm{out}^{\pi}(M)[i]), \quad 1 \leq i \leq m.$$

The label for e_i is $\mathcal{L}(e_i) = M[i]$. Note that a block-vertex structure graph can be viewed simply as an M-walk. In this paper we often use this equivalent representation of block-vertex structure graphs. A **structure graph $\mathbf{G}^*$** over a vertex set $V^* \subseteq \mathcal{I}$(an index set) is an isomorphic graph of the block-vertex structure graph mapping 0^n to 0. The labelled walk of G is preserved in G^* (in isomorphism sense) due to the isomorphism between G and G^*. So we can have a similar representation of a structure graph in terms of walks. We refer readers to Definition 4 for a more formal definition of a structure graph. The (block-vertex) structure graph is also similarly defined for a tuple (or sometimes pair) of messages $\mathcal{M} = (M_1, \ldots, M_q)$.

Given a structure graph $G^* = (V^*, E^*)$, suppose we reconstruct the graph by defining edges one by one along the M-walk. Now there are three possibilities at any point of time: (1) we add a new edge heading to a new vertex (not obtained so far) (2) we get an old edge which is already defined and (3) we add a new edge heading to an already existing vertex. True collisions correspond to the last case. The number of such true collision can be equivalently defined as the following sum

$$\mathbf{TC}(G) := \mathrm{in} - \deg(0^n) + \sum_{v \in V \setminus \{0\}} (\mathrm{in} - \deg(v) - 1).$$

Let us assign a variable Y_v, meant for the intermediate output, for each node $v \in V^*$. Let $\delta := (u, v \ ; \ z)$ be a triple such that $u \to z$, $v \to z$ and $u \neq v$. We call such triple *input-collision* (also called collision). Given any such input-collision the following linear equation, denoted L_δ must hold whenever Y-variables are actually assigned as intermediate outputs:

$$Y_v \oplus Y_u = c_\delta \text{ where } c_\delta = \mathcal{L}(v, z) \oplus \mathcal{L}(u, z).$$

When 0 has no in-degree, accident of a structure graph G^*, denoted $\mathbf{Acc}(G)$, is the rank of all linear equations L_δ over all collisions of the graph. When 0 has positive in-degree we add one to the rank to define accident. In section 5, we provide a more detailed study on the structure graph.

A Flaw in [3, Lemma 10]. The Lemma 10 of [3] states that for any structure graph G^* realized by a pair of messages $\mathbf{Acc}(G^*) = 1$ implies $\mathbf{TC}(G^*) = 1$. This result has been used to bound $\mathbf{FCP}_n$ (in [3]) as well as $\mathbf{CP}_n$ (in [3, 31]). Unfortunately, the claim is incorrect as illustrated in Fig. 1 where we have two structure graphs with true collision 2 and accident one. Surprisingly, this flaw remained unobserved till now, although it has been applied for other results.

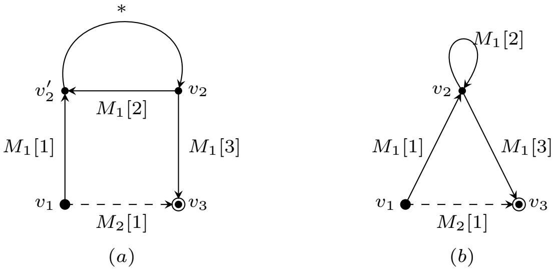


Fig. 1: Counter-example for [3, Lemma 10]. The walks corresponding to the two messages start at v_1, and end at v_3. Here $M_2[1] := M_1[1] \oplus M_1[2] \oplus M_1[3]$ and $*$ can be any number of blocks. In particular when $*$ has no block, figure (a) is same as (b). In (a) we have two input-collisions $\delta_1 := (v_1, v_2' \; ; \; v_2)$ and $\delta_2 := (v_1, v_2 \; ; \; v_3)$. The two linear equations L_{δ_1} and L_{δ_2} corresponding to the two input-collisions are same as $Y_{v_1} \oplus Y_{v_2} = M_1[1] \oplus M_1[2]$ and so the rank of all collisions (which is also accident) is one. However, true collision is two (at v_2' and v_3) which contradicts [3, Lemma 10]. Similar argument can be given for figure (b).

1.1 Our Contributions

Due to the above flaw, it is a natural question to see the impact of this flaw to those results in addition to [3], where it has been applied. This work serves this purpose. To our best knowledge, it has been applied in [31] and probably in [10, Lemma 3] (no proof of this claim is publicly available though). The bound on $\mathbf{FCP}^{\mathsf{pf}}_{2,\ell}$ [3] is also used in the PRF analysis of truncated CBC [14]. Any revision in the $\mathbf{FCP}^{\mathsf{pf}}_{2,\ell}$ bound [3] will also necessitate revision of bound in [14].

Characterization of all Accident one Structure Graphs. As the Lemma 10 is wrong and we have identified two graphs which violate lemma 10, it is important to see whether there are any more missing cases. We first settle this issue and showed that these are the only missing cases. To do so, we have characterized all structure graphs (realized by a single message) having at most one accident (see Lemma 4 in section 6). This will actually help when we study structure graphs for two messages.

Revision of the CP and FCP, and PRF bound of CBC. We revise the **FCP** bound of [3]. Fortunately, the upper bounds of **FCP** and hence PRF advantage of CBC, are only increased by a constant factor keeping the order of bound same. In section 7 we have shown this. In case of the **CP** bound due to

Bellare et al. in [3] the [3, Lemma 15], used to bound the main claim, is false. Fortunately, it can be shown that the main claim remains true after revision.

Revision of the PRF bound of EMAC. We revisit the proof of EMAC by Pietrzak [31] in section 8. Unfortunately, a straight forward revision gives a non-tight bound on EMAC. Then, we take a different approach (by consider a different bad event) to show (in Theorem 3 of the same section) the tight bound for EMAC. Our approach is much **simpler** and gives the tight bound even for a more relaxed choice of ℓ, namely $\ell < 2^{n/4}$, whereas the original constraint was $\ell < 2^{n/8}$.

2 Related Works

The security of MAC constructions has seen constant research interest. Among the block cipher based constructions CBC MAC and its variants are the most popular. Here we try to summarize the research on PRF security of CBC MAC and its variants. The aim is to list the state of the art results as well as emphasize the progress that has been made till date.

- **Analysis of CBC-MAC**. First concrete results on CBC MAC was given by Bellare et al. [2]. They showed a bound of $2\ell^2q^2/2^n$ for fixed length queries, which was further improved to $\ell^2q^2/2^n$ by Maurer [22]. Later Bernstein [5] simplified the proof for fixed-length CBC MAC. Petrank and Rackoff [30] extended the proof in [2] to prefix-free queries and similar extension on Bernstein's proof was done by Rackoff and Gorbunov [16]. Both bounds are about $\ell^2q^2/2^n$. The most recent bound on CBC MAC is by Bellare et al. [3] who improved (in terms of ℓ) the bound to $12\ell q^2/2^n + 64\ell^4q^2/2^{2n}$. Another way of improving the bound is to show the PRF bound of the form $q\sigma/2^n$ [26].
- **Analysis of EMAC**. In [2] Bellare et al. also suggested some variants of CBC MAC to handle variable length messages. In particular, they mentioned a construction where the output of CBC MAC is further encrypted by an independent key. This construction known as EMAC was first developed during the RACE [4] project. Patrank and Rackoff [30] proved that DMAC (same as EMAC) is secure up to $2.5\ell^2q^2/2^n$. Bellare et al. [3] improved the bound to $q^2 \cdot d'(\ell)/2^n$ which was further improved by Pietrzak to $q^2/2^n$ for $\ell \leq 2^{n/8}$. However, proof of the later result is invalid due to the flaw that we discussed earlier. A stated result [10] on $\mathbf{CP}^{\mathsf{eq}}_{2,\ell}$ also gives a tight bound of $O(q^2/2^n)$ for equal length messages.
- **Analysis of variants of CBC-MAC and EMAC**. Although the EMAC construction is tolerant to variable length messages it has a domain limited to $\mathcal{B}^+$. Black and Rogaway [7] introduced three refinements to EMAC, viz., ECBC, FCBC and XCBC to allow use of variable block length strings. ECBC and FCBC were shown to be secure upto $2.5\sigma^2/2^n$ [7] and the bound on XCBC was shown to be $3.75\sigma^2/2^n$ [7]. Jaulmes et al. [19] gave a randomized version of EMAC which they called RMAC and proved that the construction resists birthday attacks. However the proof seems to be incorrect

(as suggested in [3]). Other excellent variants of CBC MAC are TMAC [21], OMAC [17] and GCBC [24]. A variant of OMAC, namely OMAC1 is equivalent to CMAC which became an NIST recommendation [12] in 2005. Another design approach is the PMAC construction proposed by Black and Rogaway [6] which is inherently parallel. The improved bounds for XCBC, TMAC, PMAC and OMAC are shown in the form of $O(\ell q^2/2^n)$, $O(\sigma^2/2^n)$ and $O(\sigma q/2^n)$ as shown in [23, 18, 27, 25]. Apart from these specific constructions Jutla [20] suggested a general class of DAG-based PRF constructions.

Beyond Birthday Bound (BBB) Security. Another direction of research is BBB security, where the aim is to achieve more than $n/2$-bits security in σ. Among the block cipher based BBB secure MACs, PMAC_Plus [35] and 3kf9 [36] are two efficient candidates. Both these candidates are three-key constructions. Recently, Dutta et al. [8] proposed a one-key candidate named 1kf9, which also offers beyond birthday security of 3kf9.

Structure Graph Analysis. Structure graph is the basic tool for analyzing sequential construction based on random permutation as evident from the work on CBC based MACs [3, 31, 14] and 1kf9 [8]. Although structure graph has been mainly used in analysis of random permutation based constructions it has also found application in random function based construction as evident from the analysis of NI MAC by Gazi et al. [15] and the one key compression function based MAC by Dutta et al. [11]. From our observation these later works[15, 8, 11] are free from the flaw that we observed for [3, 31]. Gazi et al. [14] have used the **FCP** bound from [3] to bound the probability of a bad event. As the **FCP** bound [3] needs revision the bound in [14] will also get revised by a constant factor. Their proof is valid apart from the revision of some constant factor. So in the paper we focus solely on [3, 31].

3 Preliminaries

Basic Notation. Throughout the paper, we fix a positive integer n. Let Perm be the set of all permutations on $\mathcal{B} := \{0,1\}^n$. Elements of $\mathcal{B}$ are called **blocks**. For any two integers $a \leq b$, we write $[a..b]$ (or simply $[b]$, when $a = 1$) to denote the set $\{a, a+1, \ldots, b\}$. Let ϕ be a property defined for the elements of S then the subset

$$S[\phi] \stackrel{\text{def}}{=} \{x \in S : x \text{ satisfies } \phi\}.$$

The above set will appear in this paper many times for different choices of S and ϕ. Let $\mathbf{P}(m,k) := m(m-1)\cdots(m-k+1)$ denote the k-permutations of m.

3.1 Notation on Sequences

Let $\mathcal{I}$ and S be two sets. A S-sequence x over the index set $\mathcal{I}$ is denoted as $(x[\alpha])_{\alpha\in\mathcal{I}}$ where $x[\alpha] \in S$ for all $\alpha \in \mathcal{I}$. Length of the sequence is $|\mathcal{I}|$, the size of the index set. In this paper we mostly consider *block sequences*, i.e. $S = \mathcal{B}$. When the index set is $[a..b]$, we also write the sequence as a tuple or vector $x[a..b] := (x[a], \ldots, x[b])$. Sometimes, by abusing notation, x also represents the set $\{x[\alpha] : \alpha \in \mathcal{I}\}$. Similarly $x[a..b]$ represents $\{x[\alpha] : \alpha \in [a..b]\}$. We write $\#x$ to denote the number of distinct elements in the sequence x. We write S^+ and $S^{\leq \ell} := \cup_{i \leq \ell} S^i$ to represent the set of all S sequences of finite length and of length at most ℓ respectively. Now we define an equivalence relation which captures the equalities among the elements of the sequence x.

Definition 1. *Given a sequence x over an index set $\mathcal{I}$, we define an equivalence relation $\sim_x$ over the index set as follows: $\alpha \sim_x \beta$ if $x[\alpha] = x[\beta]$.*

Let $\rho : \mathcal{D} \to \mathcal{R}$ and let x and y, respectively, be $\mathcal{D}$- and $\mathcal{R}$-sequences over an index set $\mathcal{I}$. We write $x \overset{\rho}{\longmapsto} y$ to mean that $\rho(x[\alpha]) = y[\alpha]$ for all $\alpha \in \mathcal{I}$ and we simply say that ρ *multi-maps x to y*. This is a property of function ρ. When $\mathcal{D} = \mathcal{R}$, the subset $\mathrm{Perm}[x \overset{\pi}{\longmapsto} y]$ represents the set of all permutations π multi-mapping x to y. We say that (x, y) is *permutation compatible* if there exists a permutation π such that $x \overset{\pi}{\longmapsto} y$. It is easy to see that (x, y) is permutation compatible if and only if $\sim_x = \sim_y$.

3.2 Notation on Strings

We call $\mathcal{B}$ an *alphabet* and its element will be referred as *letters*. A *string* over the alphabet $\mathcal{B}$ is an element of $\mathcal{B}^*$. We can also say that a string is a finite concatenation $S := a_1 \| a_2 \| \ldots \| a_\ell$ where $a_i \in \mathcal{B}$. Note that the elements of $\mathcal{B}$ are also strings. We can also view strings as $\mathcal{B}$-sequences over an index set $\mathcal{I}$. The length of a string S, denoted by $|S|$ is defined as the total number of letters in it. Note that for an empty string the length will be 0 as it does not have any letters in it. For a string $S = X \| Y$, X (or Y) is said to be a *prefix* (or *suffix*) of S. We write $X <_1 S$ if X is a prefix of S. We write $X <_2 S$ if $X[1..x-1] <_1 S$ but $X[x] \neq S[s]$, where $x = |X|$ and $s = |S|$. For two strings S_1 and S_2 of lengths s_1 and s_2 respectively, a non-negative integer $p := \mathsf{LCP}(S_1; S_2)$ (or $s := \mathsf{LCS}(S_1; S_2)$) is called the index of the *largest common prefix* (or *largest common suffix*), if $S_1[1..p] = S_2[1..p]$ and $S_1[p+1] \neq S_2[p+1]$ (or $S_1[s..s_1] = S_2[s..s_2]$ and $S_1[s-1] \neq S_2[s-1]$).

.

3.3 Basic Definitions and Notation of Graph

Directed edge-labeled graph. A directed edge-labeled graph is a pair $G := (V, E)$ with $E \subseteq V \times V \times L$ where V is the set of vertices, L is the set of edge labels, and E is the set of edges along with their corresponding labels. In this paper we will consider only those directed edge-labeled graphs where for every vertices $u, v \in V$ there exists at most one label $a \in L$ with $((u, v)\ ;\ a) \in E$. We also write $u \xrightarrow{a} v$ to mean that $((u, v)\ ;\ a) \in E$.

Convention: *By abusing notation, E also denotes the set of unlabeled edges and the label a of the edge $e := (u, v)$ is expressed as $\mathcal{L}_G(e)$ (this notation makes sense as there is a unique choice of the label for an edge) or simply $\mathcal{L}(e)$ whenever the graph is understood.*

For an edge $e := (u, v)$, vertex u (or v) is called a *predecessor* (or *successor*) of v (or u respectively). An edge (u, v) is called a *loop* if $u = v$. We define two sets:

1. Predecessor set of a vertex v is $\mathsf{nbd}(* \to v) := \{u : (u, v) \in E\}$.
2. Similarly we define $\mathsf{nbd}(v \to *) := \{u : (v, u) \in E\}$, the successor set of v.

Size of the predecessor and successor sets of v are called **in-degree** and **out-degree** respectively. We implicitly assume that no vertex have both in-degree and out-degree zero. So the vertex set and hence the graph without the egde labels is uniquely determined by the edge set.

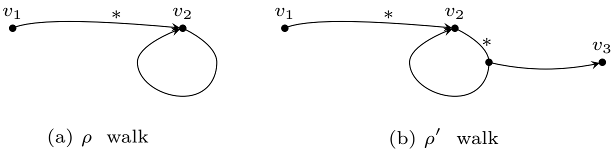
Definition 2 (walk). *A* walk of length *s is defined as a vertex sequence $w := (w[0], \ldots, w[s])$, such that $w[i-1] \to w[i]$ for all $i \in [s]$. We define label of the walk as $\mathcal{L}(w) := (a_1, \ldots, a_s)$ where $a_i = \mathcal{L}(w[i-1], w[i])$, $i \in [s]$.*

Since a walk is a V-sequence over the index set $\{0, 1, \ldots, s\}$, we define a subwalk $w[a..b] := (w[a], \ldots, w[b])$ where $0 \leq a \leq b \leq s$.

When all vertices of a walk sequence are distinct, we call it a **path**. When all vertices $w[0], \ldots, w[s-1]$ are distinct and $w[s] = w[0]$ then we call it a **cycle**. Other special examples of walks, which will be studied later in the paper, are ρ walks and ρ' walks.

A ρ walk is a walk $w := (w[0], \ldots, w[s])$ such that for some $0 \leq i < j$, $w[0..j-1]$ is a path, $w[j] = w[i]$ and for all $j < k \leq s$, $w[k] = w[i + r]$ where $0 \leq r < (j - i)$ and $(k - r)$ is multiple of $(j - i)$. It is illustrated in Fig 2(a). In words, a ρ walk comes back to one of previous vertex (which makes a cycle) and afterwards it remains in the cycle.

A ρ' walk is an extension of a ρ walk that leaves the cycle and does not come back. It is illustrated in Fig 2(b). Note that length of the subwalks labelled with $*$ can be zero.



(a) ρ walk

(b) ρ' walk

Fig. 2: The graph corresponding to ρ and ρ' walks. Note that the lengths of the parts mentioned by $*$ can be zero.

A directed edge-label graph $G = (V, E)$ is called a *function graph* if for all $v \in V$, there does not exists two distinct successors v_1 and v_2 of v with $\mathcal{L}_G(v, v_1) = \mathcal{L}_G(v, v_2)$. In other words, for every vertex v and a label a we can find at most one successor w for which the label of the edge (u, v) is a. This observation can be extended for a walk in a function graph G as follows:

$$w_1[0] = w_2[0], \mathcal{L}(w_1) = \mathcal{L}(w_2) \;\Rightarrow\; \quad w_1 = w_2.$$

So if there is a walk with label M then it must be unique and we call such a walk M-walk.

3.4 PRF Advantage of a Keyed function

If S is a finite set, then $x \xleftarrow{\$} S$ denotes the uniform random sampling of x from S. Let $\mathcal{D} \subseteq \mathcal{B}^+$ be a finite set. A **random function** from $\mathcal{D}$ to $\mathcal{B}$ is $\mathsf{RF}(\mathcal{D}) \overset{\$}{\leftarrow} \mathrm{Func}(\mathcal{D}, \mathcal{B})$, the set of all functions from $\mathcal{D}$ to $\mathcal{B}$. When the domain $\mathcal{D}$ is understood, we simply write the random function as RF.

Definition 3. *Let F be a keyed function from $\mathcal{D}$ to $\mathcal{B}$ with a finite key space $\mathcal{K}$. We define the* **prf-advantage** *(or pseudorandom function advantage) of an adversary A against F as*

$$\mathbf{Adv}_F^{\mathrm{atk}}(A) \overset{\mathrm{def}}{=} |\mathsf{Pr}[A^{F_K} = 1 : K \overset{\$}{\leftarrow} \mathcal{K}] - \mathsf{Pr}[A^{\mathsf{RF}} = 1]|.$$

The *maximum prf-advantage of F* is defined as

$$\mathbf{Adv}_F^{\mathrm{atk}}(q, \ell, \sigma) = \max_A \mathbf{Adv}_F^{\mathrm{atk}}(A)$$

where the maximum is taken over all adversaries A making at most q queries from the domain $\mathcal{D}$, say $M_1, \ldots, M_q$ with $M_i \in \mathcal{B}^{m_i}$, such that $\sum_i m_i \leq \sigma$ and $\max_i m_i \leq \ell$. Note that **atk** = **pf** means none of the query is a prefix of another; **atk** = **eq** means the queries are of equal length; and **atk** = **any** means all queries are arbitrary distinct strings. This is an information theoretic definition and we allow an unbounded time adversary. There is no loss to assume that A always makes exactly q distinct queries, represented by a sequence say $\mathcal{M} = (M_1, \ldots, M_q)$. In this case for any $T = (T_1, \ldots, T_q) \in \mathcal{B}^q$, we have

$$\mathsf{Pr}_{\mathsf{RF}}[\mathcal{M} \overset{\mathsf{RF}}{\longmapsto} T] = 2^{-nq}.$$

3.5 Coefficient-H Technique

Let A be an adversary which makes q distinct queries (possibly adaptive) to F. Let the queries be $x_1, \ldots, x_q$ and the corresponding F outputs be $y_1, \ldots, y_q$. We write the view, $\mathrm{view}(A^F)$ by the q-tuple of pairs $((x_1, y_1), \ldots, (x_q, y_q))$ where x_i denotes the i^{th} query and y_i is the corresponding response.

For any q-tuple of pairs $\tau = ((x_1, y_1), \ldots, (x_q, y_q))$, the following probability

$$\mathrm{IP}^F(\tau) \overset{\mathrm{def}}{=} \mathsf{Pr}_F[(x_1, \ldots, x_q) \overset{F}{\longmapsto} (y_1, \ldots, y_q)]$$

is called the *interpolation probability*, where the probability is taken under the randomness of F's key. Here we assume that F is stateless and so the above probability is independent of the order of the pairs.

Theorem 1. *[Coefficient-H Technique] Let $\mathcal{T}_{good}$ be some set of q-tuples of pairs. Suppose the interpolation probability for a (stateless) oracle $\mathcal{O}$ follows the inequality*

$$\mathrm{IP}^{\mathcal{O}}(\tau) \geq (1-\epsilon) \cdot \mathrm{IP}^{\mathsf{RF}}(\tau) = (1-\epsilon)2^{-nq} \quad \forall \tau \in \mathcal{T}_{good}.$$

Then, for any adversary A we have,

$$\mathbf{Adv}_F^{\mathrm{atk}}(A) \leq \epsilon + \mathsf{Pr}[\mathrm{view}(A^{\mathsf{RF}}) \notin \mathcal{T}_{good}].$$

This technique was first introduced by Patarin in his PhD thesis [28] (as mentioned in [32]). The proof of this theorem can be found in [29]. So we skip the proof. We use this theorem while we bound PRF advantage of CBC and EMAC function defined next.

3.6 (E)CBC Function based on a Permutation

CBC Function. The CBC (cipher block chaining) function with an oracle $\pi \in \mathrm{Perm}$, viewed as a key of the construction, takes as input a message $M = (M[1], \ldots, M[m]) \in \mathcal{B}^m$ with m blocks and outputs $\mathrm{CBC}_\pi(M) := \mathsf{out}^\pi(M)[m]$. This is inductively computed as follows: $\mathsf{out}^\pi(M)[0] = 0^n$ and

$$\mathsf{out}^\pi(M)[i] = \pi(\mathsf{in}^\pi(M)[i]), \quad \mathsf{in}^\pi(M)[i] = \mathsf{out}^\pi(M)[i-1] \oplus M[i], \quad i \in [m]. \quad (2)$$

We call $\mathsf{in}^\pi(M)$ and $\mathsf{out}^\pi(M)$ **intermediate input** and **output vectors** respectively, associated to π. Note that the intermediate input vector in^π is uniquely determined by out^π (and not depends on the permutation π). We can write down this association generically as a function $\mathsf{out2in}_M : \mathcal{B}^m \to \mathcal{B}^m$ mapping any block vector y to a block vector x where $x[1] = M[1]$ and $x[i] = y[i-1] \oplus M[i]$ if $1 < i \leq m$. So for all permutation $\pi \in \mathsf{Perm}$, $\mathsf{out2in}(\mathsf{out}^\pi) = \mathsf{in}^\pi$.

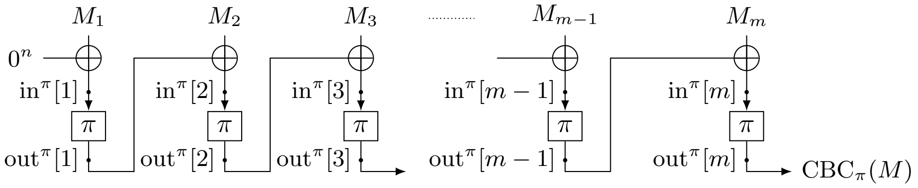


Fig. 3: CBC function and its intermediate values.

EMAC Function. The EMAC function (E for encrypted) is derived from the CBC function by additionally encrypting the output with another permutation $\pi' \in \mathrm{Perm}$. Formally, $\mathrm{EMAC}_{\pi,\pi'}(M) \stackrel{\mathrm{def}}{=} \pi'(\mathrm{CBC}_\pi(M))$.

4 PRF Analysis of CBC and EMAC

In this section we quickly recall the PRF analysis of CBC and EMAC as done in [3, 31]. Here CBC is based on uniform random permutation Π chosen uniformly from Perm and EMAC is based on two independent random permutations Π and Π'. In this section we reduce the bounding PRF advantages of CBC and EMAC to the full bounding collision and collision probability respectively. We use coefficient-H technique rather than the game playing technique used in [3].

4.1 PRF Advantage of EMAC

Let M_1 and M_2 be two distinct tuple of blocks. Let $\mathsf{coll}_\pi(M_1; M_2)$ denote the event that $\mathrm{CBC}_\pi(M_1) = \mathrm{CBC}_\pi(M_2)$ and we call it collision event for a pair of messages M_1 and M_2. We similarly define collision event for a tuple of $q \geq 2$ distinct messages $\mathcal{M} = (M_1, \ldots, M_q)$ as

$$\mathsf{coll}_\pi(\mathcal{M}) = \bigcup_{i \neq j} \mathsf{coll}_\pi(M_i; M_j).$$

We define **collision probability** as $\mathbf{CP}_n(\mathcal{M}) = \mathsf{Pr}[\mathsf{coll}_\Pi(\mathcal{M})]$. Let $\mathbf{CP}^{\mathsf{atk}}_{q,\ell} = \max_{\mathcal{M}} \mathbf{CP}_n(\mathcal{M})$ where the maximum is taken over all q-tuple of distinct messages $\mathcal{M}$ having at most ℓ blocks each and satisfy atk (i.e., when $\mathsf{atk} = \mathsf{eq}$, messages must have equal length, similarly when $\mathsf{atk} = \mathsf{pf}$ no message is prefix to other and finally $\mathsf{atk} = \mathsf{any}$ means no restriction other than length restriction). Following [3], we view EMAC as an instance of the Carter-Wegman paradigm [33]. This enables us to reduce the problem of bounding the prf-advantage of EMAC to bounding the collision probability as

$$\mathbf{Adv}^{\mathsf{any}}_{\mathrm{EMAC}}(q, \ell) \leq \mathbf{CP}^{\mathsf{any}}_{q,\ell} + \frac{q(q-1)}{2^{n+1}}. \tag{3}$$

Note that $\mathbf{CP}^{\mathsf{any}}_{q,\ell} \leq \binom{q}{2} \mathbf{CP}^{\mathsf{any}}_{2,\ell}$ as the collision for q messages is $\binom{q}{2}$ union of collision events for two messages. Bellare et al. [3] proved that

$$\mathbf{CP}^{\mathsf{any}}_{2,\ell} \leq \frac{2d'(\ell)}{2^n} + \frac{64\ell^4}{2^{2n}}. \tag{4}$$

where $d'(\ell) = \max_{\ell' \leq \ell} d(\ell')$ and $d(\ell')$ is the the number of divisors of ℓ'. In [34], it was shown that $d'(\ell) = \ell^{1/\Theta(\ln \ln \ell)} = \ell^{o(1)}$. Using this bound of collision probability for a pair of messages the prf-advantage of EMAC is about $\mathrm{O}(d'(\ell) q^2 / 2^n)$ for $\ell < 2^{n/4}$. Later [31] provided an improved analysis of EMAC and proved that the PRF advantage of EMAC is about $\mathrm{O}(q^2/2^n)$ for $\ell < \min\{q^{1/2}, 2^{n/8}\}$. We revisit this improved analysis later in section 8. A related claim on $\mathbf{CP}$ is $\mathbf{CP}^{\mathsf{eq}}_{2,\ell} = 2^{-n} + (d(\ell))^2 \cdot \ell \cdot 2^{-2n} + \ell^6 \cdot 2^{-3n}$ [10] which gives tight bound for equal length messages.

4.2 PRF Advantage of CBC

Now we revisit the security analysis of CBC-MAC construction. Let $\mathsf{Fcoll}_{\pi}(M_1; M_2)$, called **full collision**, denote the event that

$$\mathsf{in}^{\pi}(M_2)[m_2] = \mathsf{in}^{\pi}(M_r)[j] \text{ for some } (r, j) \neq (2, m_2).$$

In other words, if the full collision event does not hold then the last intermediate input of π is "fresh" (not appeared before) while computing $\mathrm{CBC}_{\pi}(M_2)$. So when π is replaced by a random permutation and this event does not hold then the CBC-output should behave "almost" randomly. We use this intuition while we provide a bound of prf-advantage of CBC.

Remark 1. We would like to remark that in the original paper [3], full collision event is defined through the intermediate outputs instead of inputs. Since we consider CBC based on permutation only, equalities among inputs and equalities among outputs are same.

For a q-tuple of messages $\mathcal{M}$, the union of full collision event similarly denoted by $\mathsf{Fcoll}_{\pi}(\mathcal{M})$. The probability of this event, called **full collision probability**, is denoted by $\mathbf{FCP}_n(\mathcal{M})$. The maximum full collision probability is denoted by $\mathbf{FCP}^{\mathsf{atk}}_{q,\ell}$. Similar to the previous lemma, the following result has been proved in [3].

$$\mathbf{Adv}^{\mathsf{pf}}_{\mathrm{CBC}}(q, \ell) \leq q^2(\mathbf{FCP}^{\mathsf{pf}}_{2,\ell} + 4\ell/2^n). \tag{5}$$

Note that we must restrict adversary to make prefix-free queries, since otherwise it would be easy to distinguish CBC from a random function (using the classical length extension attack). Similarly, if M_2 is prefix of M_1, it is easy to see that $\mathbf{FCP}_n(M_1, M_2) = 1$ so the above result becomes meaningless. As before, we also state an equivalent form of PRF advantage of CBC in terms of full collision probability among q messages. The above equation 5 would be again straight forward application of the following result.

Proposition 1. $\mathbf{Adv}^{\mathit{pf}}_{\mathrm{CBC}}(q, \ell, \sigma) \leq \mathbf{FCP}^{\mathit{pf}}_{q,\ell} + \dfrac{2\sigma q}{2^n} + \dfrac{q^2}{2^{n+1}}$.

Proof. Let $\mathcal{T}_{good} := ((M_1, T_1), (M_2, T_2), \ldots, (M_q, T_q))$ be the set of all pairs of $\mathcal{M} = (M_1, \ldots, M_q) \in (\mathcal{B}^+)^q$ and $T = (T_1, \ldots, T_q) \in \mathcal{B}^q$ such that M_i's are distinct and T_i's are also distinct. Trivially, random function RF returns a collision pair on any q distinct queries with probability at most $\binom{q}{2}2^{-n}$ for any adversary A. Thus,

$$\mathsf{Pr}[\mathrm{view}(A^{\mathsf{RF}}) \notin \mathcal{T}_{good}] \leq \frac{q^2}{2^{n+1}}.$$

Using coefficient H-technique, now we only need to bound the relationship between the interpolation probabilities. We fix $\mathcal{M} = (M_1, \ldots, M_q) \in (\mathcal{B}^+)^q$ and $T = (T_1, \ldots, T_q) \in \mathcal{B}^q$ such that $M_i \in \mathcal{B}^{m_i}$'s are distinct and T_i's are also distinct. Let $m_i \leq \ell$ for all i and we write $\sum_i m_i = m \leq \sigma$. Now, A permutation π is called **bad** if

1. $\mathsf{Fcoll}_\pi(\mathcal{M})$ holds, or
2. for some $r, r' \in [q], i \in [m_r]$, $\mathsf{out}^\pi(M_r)[i] = T_{r'}$.

All other permutations are called good. We define an equivalence relation $\sim$ on Perm as $\pi \sim \pi'$ if $\mathsf{in}^\pi(M_r) = \mathsf{in}^{\pi'}(M_r)$ for all r. It is clearly an equivalence relation and a good permutation can only be related with another good permutation. Let $\mathcal{C}$ be an equivalence class consisting of some good permutations. Let s be the number of distinct intermediate inputs for the computation of all $\mathrm{CBC}_\pi(M_r)$ where $\pi \in \mathcal{C}$. Note that s is same for all $\pi \in \mathcal{C}$. Then, $|\mathcal{C}| = (2^n - s - q)!$ as the output of exactly $(2^n - s - q)$ inputs of π are determined. Since T_i's are not intermediate outputs, $|\mathcal{C}[\mathcal{M} \overset{\mathsf{CBC}_\Pi}{\longmapsto} T]| = (2^n - s)!$ (since q additional restrictions on input-output are being added). So for any class of good permutations $\mathcal{C}$, $\mathsf{Pr}[\mathcal{M} \overset{\mathrm{CBC}_\Pi}{\longmapsto} T \mid \Pi \in \mathcal{C}] = (2^n - s)!/(2^n - s - q)! \geq 2^{-nq}$. Thus,

$$\begin{aligned} \mathsf{Pr}[\mathcal{M} \overset{\mathrm{CBC}}{\longmapsto} T] &\geq \sum_{\mathcal{C} \textit{ is good}} \mathsf{Pr}[\mathcal{M} \overset{\mathrm{CBC}}{\longmapsto} T \mid \Pi \in \mathcal{C}] \times \mathsf{Pr}[\Pi \in \mathcal{C}] \\ &\geq \mathsf{Pr}[\Pi \text{ is good}] \times 2^{-nq}. \end{aligned}$$

So it is sufficient to bound a random permutation being bad. Then we will be done by using coefficient H-technique as stated in theorem 1 in section 3.5. By definition of full collision probability, the first condition of bad can happen with probability at most $\mathbf{FCP}^{\mathsf{pf}}_{q,\ell}$. The second condition says that we sample at most m many outputs of a random permutation and one of them belongs to the set $\{T_1, \ldots, T_q\}$. This can happen with probability at most $\frac{mq}{2^n - m}$ which is further less than $mq/2^{n-1}$ provided $m < 2^{n-1}$. Note that $m \leq \sigma$. If $m \geq 2^{n-1}$ then the above bound holds trivially. So probability of bad permutation is bounded by $\mathbf{FCP}^{\mathsf{pf}}_{q,\ell} + mq/2^{n-1}$. After applying coefficient-H technique, we have proved the result. □

Remark 2. Note that $\mathbf{FCP}^{\mathsf{pf}}_{q,\ell} \leq q(q-1)\mathbf{FCP}^{\mathsf{any}}_{2,\ell}$ by considering all ordered pair (M_i, M_j). This also proves the original claim by [3] as stated in Eq. 5. In fact, it is potentially a better bound than the original as it uses the total number of blocks σ instead of ℓq. In [3] Bellare et al. proved that

$$\mathbf{FCP}^{\mathsf{pf}}_{2,\ell} \leq \frac{8\ell}{2^n} + \frac{64\ell^4}{2^{2n}}. \tag{6}$$

In section 7 we revisit the above bound. In particular we revise the proof in light of the flaw in [3, Lemma 10] and get an increment in the multiplication factor. Moreover, our revised bound of $\mathbf{FCP}^{\mathsf{pf}}_{q,\ell}$ would be in the order $\frac{\sigma q}{2^n}$ instead of $\frac{\ell q^2}{2^n}$ (whenever $\ell \leq 2^{n/3}$). So our analysis rectifies the previous proof and also provides better bound in some cases (e.g., average message length is much smaller than the length of longest messages which may occur when message lengths are very skewed).

5 Revisiting Structure Graph

In the previous section we have seen that how the PRF advantage of CBC or EMAC is essentially reduced to bound some collision events of internal inputs or outputs of the underlying permutation. Thus, it would be useful to have an object which deals with the intermediate inputs and outputs. Structure graph does so and it has been used to bound the (full) collision probabilities in [BPR05] paper. In this section we revisit the structure graph and show that one of the main claims in [3]([3, Lemma 10]) about a structure graph is false.

Notation and Conventions for this section: Let us fix a tuple of messages $\mathcal{M} = (M_1, \ldots, M_q)$ throughout this section where $M_i \in \mathcal{B}^{m_i}$ and let $m := \sum_{i=1}^{q} m_i$ and $\max_i m_i = \ell$.

5.1 Intermediate Inputs and Outputs

Index Set. We first collect all intermediate inputs and outputs which are obtained through the computation of $\mathrm{CBC}_\pi(M_r)$ for all r. These intermediate values will be defined as a sequence over a two-dimensional index set. Each index is a pair where the first element of the pair corresponds to the message number and the second element is the block number of that message. More formally, we define the *index set*

$$\mathcal{I} = \{(r,i) : r \in [q], i \in [m_r]\}$$

and the dictionary order $\prec$ on it as follows: $(r,i) \prec (r',i')$ if $r < r'$ or $r = r'$ and $i < i'$. Let x be a sequence over this index set. For any $r \in [q]$, we denote the subsequence $(x[r,1], \ldots, x[r,m_r])$ by $x[r,*]$. Sometimes we also consider the index set $\mathcal{I}_0 = \mathcal{I} \cup \{(r,0) : r \in [q]\}$, and the natural extension of the order $\prec$ on $\mathcal{I}_0$.

Sequences for Intermediate Inputs and Outputs. We denote the *sequence of intermediate outputs* and *inputs* over the index set $\mathcal{I}$ as $\mathsf{out}^\pi(\mathcal{M})$ and $\mathsf{in}^\pi(\mathcal{M})$ respectively where

$$\mathsf{out}^\pi(\mathcal{M})[r,*] = \mathsf{out}^\pi(M_r), \quad \mathsf{in}^\pi(\mathcal{M})[r,*] = \mathsf{in}^\pi(M_r), \quad \forall r \in [q].$$

For a single message, we have seen before that the intermediate input sequence is uniquely determined by intermediate output sequence and we denote the association by a function $\mathsf{out2in}$. The same is true for q messages and we extend this definition as follows: Given any block sequence y over the index set $\mathcal{I}$, we define $\mathsf{out2in}(y)$ as a block sequence x over the same index space where $x[r,*] = \mathsf{out2in}_{M_r}(y[r,*])$, $r \in [q]$. Thus, for any π, we have $\mathsf{out2in}(\mathsf{out}^\pi) = \mathsf{in}^\pi$.

5.2 Structure Graphs and Block-Vertex Structure Graphs

A block-vertex structure graph is a graph theoretical representation of intermediate output out^π. The **block-vertex structure graph** $\mathsf{Bstruct}^\pi$ for a permutation π is defined by the set of labeled edges

$$E \overset{\text{def}}{=} \cup_{r=1}^{q} \{(\mathsf{out}^\pi[r,i-1], \mathsf{out}^\pi[r,i] \ ; \ M_r[i]) : i \in [m_r]\}.$$

Let $\mathsf{Bstruct}(\mathcal{M})$ denote the set of all block structure graphs. Clearly, $0^n \in V$ has positive out-degree and it is a union of M_i-walks, for $i \in [q]$. Note that as explained below,

$$v \xrightarrow{A} w \;\Rightarrow\; \pi(v \oplus A) = w. \tag{7}$$

So *for every $v \in V$, all outward edges (similarly for inward edges) have distinct edge labels.* Using this property, it is easy to see that **the walks are unique** and we denote them by w_{M_i} or simply w_i whenever the message tuple is understood. See Fig. 4 for a single message (i.e., $q = 1$) in which the input and output vectors are stored in a directed graph.

While storing the intermediate sequences as a set of labeled edges, we may loose the order as well as the repetition of the elements. Interestingly, we see that we can uniquely reconstruct the intermediate sequences from such an edge-labeled graph by using uniqueness of M_i-walks. More precisely, $\mathsf{out}^{\pi}[r, i] = w_r[i]$.

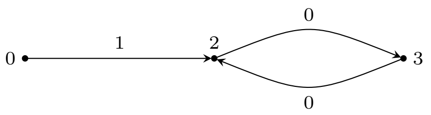


Fig. 4: Let $M_1 = (1, 0, 0, 0, 0)$ and $\pi(1) = 2$, $\pi(2) = 3$, $\pi(3) = 2$. For any such π, we have $\mathsf{out}^{\pi} = (2, 3, 2, 3, 2)$ and $\mathsf{in}^{\pi} = (1, 2, 3, 2, 3)$. However, the graph consists of three vertices $\{0, 2, 3\}$ and edge set $E = \{(0, 2), (2, 3), (3, 2)\}$ with labels 1, 0 and 0 respectively. We see that the intermediate input and output sequences actually can be reconstructed from this labeled structure graph. The walk corresponding to the message M_1 will uniquely identify the output vector as $\mathsf{out}^{\pi} = (2, 3, 2, 3, 2)$ and the input vector $\mathsf{in}^{\pi} = (1, 2, 3, 2, 3)$ can be constructed using the relation between input, output and message.

Let $G = (V, E)$ be a labeled directed graph and $f : V \to V^*$ is a bijective function. Then one can define a labeled directed graph $G^* = (V^*, E^*)$ isomorphic to G for which f is an isomorphism. More precisely, $((u, v)\ ,\ a) \in E$ if and only if $((f(u), f(v))\ ;\ a) \in E^*$. When f is an injective function we can view the function where the range set is the image set of the function and this makes the function bijective. We call the graph obtained as described above a transformed G with respect to f.

Definition 4 (structure graph). *For every vertex v of a block-vertex structure graphs $G = (V, E)$, we define a mapping $\alpha : V \to \mathcal{I}$ as $\alpha_v = \alpha(v) = (r, i)$ where (r, i) is the minimum index such that $w_r[i] = v$. Clearly, it is an injective mapping with an image set say V^*. Structure graph $G^* = (V^*, E^*)$ associated to π is the α-transformed block-vertex structure graph.*

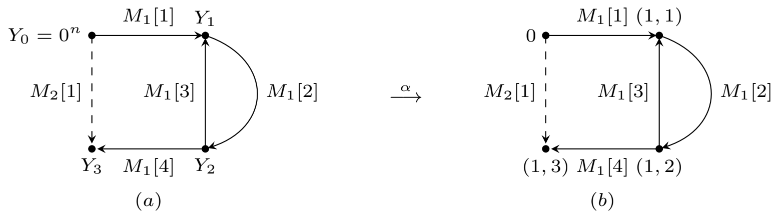


Fig. 5: Structure Graph corresponding to the labeled structure graph.

Example 1. Let $M_1 = (M_1^1, M_1^2, M_1^3, M_1^4)$ and $M_2 = (M_2^1)$ be two messages and for $\pi \in$ Perm let $\mathsf{in}^\pi[1,*] = (Y_0 = 0^n, Y_1, Y_2, Y_3)$; $\mathsf{out}^\pi[1,*] = (Y_1, Y_2, Y_3)$ and $in^\pi[2,*] = (Y_0)$;$\mathsf{out}^\pi[2,*] = (Y_3)$. The corresponding block labeled structure graph $\mathsf{Bstruct}^\pi$ is as shown in fig. 5(a). Following the above steps we arrive at a valid structure graph struct^π in fig. 5(b).

Let w_r* denote the M_r-walk in G^*. It is easy to see that a structure graph is again an **union of M_r-walks** w_r^* starting from 0.[2] A structure graph is called a **zero-output graph** if 0 has positive in-degree, otherwise we call it **nonzero-output graph**. To express it mathematically, we define a binary function Iszero such that for all zero-output graph G^*, Iszero$(G^*) = 1$, otherwise it maps to 0.

To reconstruct a block-vertex structure graph realizing G^* we have to find labels from $\mathcal{B}$ for all the vertices in a "consistent manner" and we call those labeling **valid**. Basically, we need to find an injective mapping $\alpha^{-1} : V^* \to \mathcal{B}$ such that image set of α^{-1} is V and $\alpha := (\alpha^{-1})^{-1}$ is an isomorphism.

Definition 5 (valid block label). An injective function $Y : V^* \to \mathcal{B}$ *is called* **valid block label** *for a structure graph* $\mathcal{S} = (V^*, E^*)$ *if the graph* $G = (V, E)$ *is a block-vertex structure graph where*

1. $V = \{0^n\} \cup \{Y_i := Y(i) : i \in V^*\}$ *and*
2. E *is the edge set after relabeling* i *by* Y_i *(we assume* $Y_0 := 0^n$*).*

Necessary Condition of Valid Labeling Function Y. Now we try to find necessary conditions of a valid labeling. First of all, by definition, Y_i should be all distinct as the valid block label is injective (distinct vertex should get distinct block label). In addition to this, whenever $e_1 := (u, z), e_2 := (v, z) \in E$ we must have $Y_u \oplus \mathcal{L}(e_1) = Y_v \oplus \mathcal{L}(e_2)$ as these are input for the vertex z. An **input-collision or simply a collision** of a graph G is defined by such a triple $\delta = (u, v; z)$. The set $\{u, v\}$ is called the *source of the collision* whereas z is called the *head of the collision.* We also say the edges e_1 and e_2 colliding edges. Thus, an input-collision $\delta = (u, v; z)$ induces a linear restriction $L_\delta : Y_u \oplus Y_v = c_\delta$ where $c_\delta = \mathcal{L}(u, z) \oplus \mathcal{L}(v, z) \in \mathcal{B}$. Thus, a valid block label must satisfy the above condition for all collisions δ. Let $\varDelta_{G^*}$ denote the set of all collisions of G^*. Let rank(G^*) denote the rank of all linear equations $\{L_\delta : \delta \in \varDelta_{G^*}\}$. Accident

[2] Note that, as per the convention used here and in the preceding discussion $w_r^*[i] = \alpha(w_r[i])$.

of a structure graph is defined depending on whether the graph is zero-output or not.

Definition 6 (Accident of a structure graph). *We define accident of a structure graph G^* as* $\mathbf{Acc}(G^*) \overset{\text{def}}{=} \text{rank}(G^*) + \text{Iszero}(G^*)$*. Thus, accident of a non-zero structure graph G^* is defined to be* $\text{rank}(G^*)$ *rank of the graph, whereas accident of a zero-output graph is* $\text{rank}(G^*) + 1$.

Lemma 1. *If there is a vertex v with in-degree d then $\mathsf{rank}(G^*) \geq d - 1$. Moreover, if the graph is a zero-output graph then $\mathbf{Acc}(G^*) \geq d$.*

Proof. Let $v_1, \ldots, v_d$ be all predecessor of v. Let us define an input-collision $\delta_{i,j} := (v_i, v_j; v)$. It is now easy to see that $L_{\delta_{i,j}} = L_{1,i} \oplus L_{1,j}$. Moreover, $L_{1,i}$'s are linearly independent. Thus, the first part is proved, The second part is also trivial from the first part and the definition of accident. □

Remark 3. Another simple but useful observation is as follows: if a structure graph G^* has at least two collisions with different source, then $\mathsf{rank}(G^*) \geq 2$.

Remark 4. Our definition of accident is based on linear algebra whereas the original definition of accident in [3] is based on graph theory. One can check that both definitions are eventually equivalent. We find this definition is useful proving lemma 2 as stated below. Moreover, as the definition is more non-visual than the original, it would be less prone to have an error while applying the definition.

Let $S = (V^*, E^*)$ be a structure graph with rank r and $|V^*| = s + 1$. Then from linear algebra we know that some $s - r$ choices of Y_i values will uniquely determine the rest and so the number of valid block labeling is at most $\mathbf{P}(2^n, s - r)$. Any valid choice of Y induces a block-vertex structure graph $G = (V, E)$ such that $G^* = S$. Note that $s+\text{Iszero}(G)$ is the number of vertices $v \in V$ with positive in-degree. So exactly $(2^n - s - \text{Iszero}(G))!$ number of permutations can result in block-vertex structure graph G. Therefore,

$$\mathsf{Pr}[\mathsf{Bstruct}^{\Pi} = G] = \frac{2^n - (s + \text{Iszero}(G))!}{2^n!} \quad = \frac{1}{\mathbf{P}(2^n, s + \text{Iszero}(G))}. \tag{8}$$

So $\mathsf{Pr}[\mathsf{struct}^{\Pi} = S] = \sum_{G:G^*=S} \mathsf{Pr}[\mathsf{Bstruct}^{\Pi} = G]$. Here the sum is taken over all block-vertex structure graphs G such that the induced structure graph $G^* = S$. As there are at most $\mathbf{P}(2^n, s - r)$ many vertex-label structure graphs (by bounding the number of valid block label functions as described above and using $s + 1 \leq m$), we proved the following important result.

Lemma 2. *For any structure graph S with accident a,*

$$\mathit{Pr}[\mathit{struct}^{\Pi} = S] \leq \frac{1}{(2^n - m)^a}.$$

Now we state another important result which bounds the number of structure graphs with a accident. The proof of this result can be found in [3, 31]. So we skip the proof here.

Lemma 3. *The number of structures graphs associated to $\mathcal{M} = (M_1, \ldots, M_t)$ with accident a is at most $\binom{m}{2}^a$. In particular, there exists exactly one structure graph with zero accident.*

Corollary 1. *Let $a \geq 1$ be an integer. Then,*

$$\Pr[\mathbf{Acc}(\mathit{struct}^{\Pi}) \geq a : \Pi \xleftarrow{\$} \mathrm{Perm}] \leq (\frac{m^2}{2^n})^a.$$

This can be shown by making a straightforward algebraic simplification after applying the lemma 2 and lemma 3. So we skip the proof. □

5.3 True Collision and an Observation on [3, Lemma 10]

The definition of accident is not obvious by looking at the structure graph. It would be good to have some transparent definition for a structure graph. True collision is such a metric. Let G^* be a structure graph and w_i^* are the M_i-walks. Suppose we reconstruct the graph G^* again by making all the walks w_i^* for $i = 1$ to q. While we walk along w_i^* for all i we count how many times we reach an existing vertex which increases its current in-degree. The total count is defined to be the number of true collisions of the graph. Mathematically, one can define it as follows: For a vertex $v \in V^* \setminus \{0\}$, we define number of true collision at v by $\mathbf{TC}(v) := |\mathsf{nbd}(* \to v)| - 1$ and $\mathbf{TC}(0) = |\mathsf{nbd}(* \to 0)|$. So the above count is actually the sum $\mathbf{TC}(G^*) \overset{\text{def}}{=} \sum_{v \in V^*} \mathbf{TC}(v)$. By lemma 1 we know that $\mathbf{Acc}(G^*) \geq \mathbf{TC}(v)$ for all $v \in V^*$. From the definition of accidents it is also obvious that $\mathbf{Acc}(G^*) \leq \mathbf{TC}(G^*)$.

Lemma 10 of [3]. To identify all structure graphs with accident one it would be good if we have some relationship between true collision and accident. Lemma 10 of [3] was meant for this. It says that when $q = 2$, $\mathbf{Acc}(G^*) = 1 \Rightarrow \mathbf{TC}(G^*) = 1$. This lemma is wrong due to the following counter example. This lemma has been used to bound the PRF advantage of CBC [3] and EMAC [31, 3]. As this becomes wrong, it would be very important to look back the proof and rectify the results as much as possible.

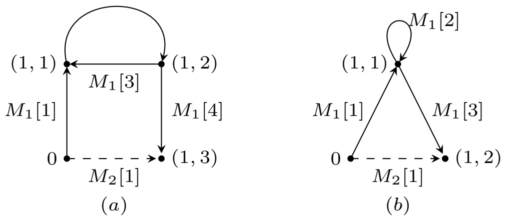


Fig. 6: The counter examples. This is because one equation is generated by the other. We call it alternating closed walk.

6 Characterization of Accident one Structure Graphs

In this section we characterize all structure graphs with zero and one accidents. We have already seen that the authors have missed some structure graphs for two messages. Thus it is important to see whether there are other such graphs or not. To do so we characterize single message structure graph which is much easier to convince. Later in section we characterize all structure graphs for a pair of messages satisfying some event. Note that from here onwards we won't be dealing with the block-vertex structure graph. So for simplicity from here onwards we will use G (instead of G^*) to represent a structure graph and w_r (instead of w_r*) to represent the M_r-walk in the structure graph.

Let $\mathsf{struct}_a(\mathcal{M}) = \{G \in \mathsf{struct}(\mathcal{M}) : \mathbf{Acc}(G) = a\}$, the set of all structure graphs associated to $\mathcal{M}$ with accident a. In particular we are interested in $\mathsf{struct}_0(\mathcal{M})$ and $\mathsf{struct}_1(\mathcal{M})$ the set of all structure graphs with accident 0 and 1 respectively. Lemma 3 says that the number of graphs with accident one is at most $\binom{m}{2}$ where $m = \sum_i m_i$ and $M_i \in \mathcal{B}^{m_i}$. Thus, the number of structure graphs with accident zero is at most one. In the following we actually identify a structure graph and hence it is unique, We call it the **free graph** associated to $\mathcal{M}$.

Free Graphs. As there is no accident every non-zero vertex has in-degree one and 0 has in-degree zero (i.e., non-zero output graph). Being a structure graph, it is union of M_i-walks w_{M_i}. A M_i-walk starting from 0 with no vertex having in-degree two must be a path. So G is an union of M_i-paths w_{M_i}. Now for any $i \neq j$, let $p = \mathsf{LCP}(M_i; M_j)$. Then, $w_i[1..p] = w_j[1..p]$ and $w_i[p+1] \neq w_j[p+1]$ (if these are defined). It is also easy to see that $w_i[1..p]$, $w_i[p+1..m_i]$, and $w_j[p+1..m_j]$ are disjoint paths. Thus, *any two paths w_i and w_j are same up to the length of the largest common prefix of M_i and M_j and afterwards* they remain disjoint. We call this unique graph **free graph**. A free graph for three messages is illustrated in Fig 7.

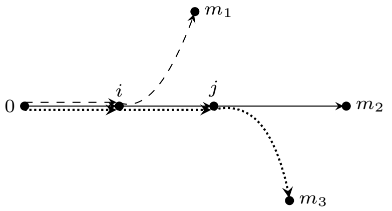


Fig. 7: Free structure graph for three messages.

6.1 Accident one for a single message

Now we consider a single message $M \in \mathcal{B}^+$ structure graph. Note the any such structure graph must be a walk w of length m. We say a node $w[i]$ **is fresh** in the walk if $w[i] \neq w[j]$ for all $j \neq i$.

Case A: 0 has positive in-degree As 0 has positive in-degree there can not be any more collision pairs otherwise the accident would be at least two. Let c be the minimum positive integer such that $w[c] = 0$, so we have a cycle $(w[0], w[1], \ldots, w[c])$ and let X be its label. Suppose $M = X^i \| Y$ where i is the maximum positive integer for which we can write M in this form. So X is not a prefix of Y. Let $s = \mathsf{LCP}(X; Y)$. Thus, $w[ic + j] = w[j] \;\; \forall j \in [0..s]$.

1. If Y is a prefix of X then the structure graph is a cycle of size c where it ends at $w[s]$. It is illustrated in fig. 8(a) where the $*$ is empty.
2. If Y is not a prefix of X then $w[ic + s] = w[s]$ and $w[ic + s + 1] \neq w[s + 1]$. Further, $w[ic+s+1] \neq w[j]$ for all $j \in [c]$ since otherwise we get a collision. In fact it can be shown that all subsequent nodes are fresh. Suppose not, then let $j > ic + s + 1$ be the first such integer for which $w[j] = w[k]$ for some $k < j$ and hence we obtain a collision. So the structure graph is an edge disjoint union of a cycle of size c and a path starting from s, as illustrated in fig. 8(a). Length of the cycle is c where as the length of the path is $m - ic - s$. We also called this graph ρ' graph. The tail (path from 0 to the cycle) of the ρ' walk is empty.

Case B: 0 has zero in-degree As 0 has zero in-degree, there is a collision $\delta = (u_0, v_0; z)$. In fact, all other collisions must have same source as that of δ.

Consider the M-walk $(w[0], w[1], \ldots)$ which is clearly not a path. Let (i_0, j_0) be the smallest positive distinct integers such that $w[i_0] = w[j_0]$.[3] As 0 has zero in-degree so $1 \leq i_0 < j_0$. So we can assume that $w[i_0 - 1] = u_0$ and $w[j_0 - 1] = v_0$. Now we let, as done in the case A, $A = \mathcal{L}(w[0..i_0])$, $X = \mathcal{L}(w[i_0..j_0])$, $j_0 - i_0 = c$. Then, $A \| X$ is the prefix of M. Let t be the largest positive integers such that $M = A \| X^t \| Y$. So X is not a prefix of Y. If Y is a prefix of X then we have a structure graph as illustrated in fig. 8(d) and 8(f) (the end point lies inside the cycle). Suppose Y is not a prefix and let $s = \mathsf{LCP}(X; Y)$.

Claim. The walk after $A \| X^t \| Y[1..s]$ is a path and disjoint from the rest (illustrated in fig. 8(c)).

Proof of the claim. Suppose $\exists v \neq w[s] \in w[tc + s..m] \cap w[1..tc + s]$. Then,

Case B.1: $w[tc + s + 1] = w[i] \quad i \in [tc + s]$. If $s \neq j_0 - 1$ then we have a new collision $\delta' = (w[i - 1], w[tc + s]; w[i])$ independent of δ which increases the number of accidents to 2. If $s = j_0 - 1$ then $i \neq i_0$ as $X[s + 1] \neq Y[1]$. Now the only way to make δ' dependent on δ is to have $i - 1 = i_0 - 1$. This implies a collision at $w[j]$ where $j \in [1..i_0 - 1]$, as the walk must come back to $i_0 - 1$ at the $(i - 1)$-th step. This again gives a new accident.

Case B.2: $w[tc+s+1] \notin w[1..tc+s]$ **and** $w[j] = w[i] \;\; j \in [tc+s+2..m]$, $i \in [tc+s]$. So, there is a new collision $\delta' = (w[j - 1], w[i - 1]; w[i])$ which is independent of δ. This gives a new accident. So $w[tc + s + 1..m] \cap w[1..tc + s] = \emptyset$.

[3] i_0 and j_0 can be fixed one by one. First fix i_0 to be the smallest positive integer such that $w_[i_0] = w[j] \;\; j \in [i_0 + 1..m]$. Now, fix the smallest positive integer j_0 such that $w[j_0] = w[i_0]$.

Case B.3: $w[tc+s..m]$ **is not a path.** Therefore $\exists i, j \in [tc+s..m]$ such that $(w[i], w[j]; w[i+1])$ is a collision. Clearly this will be independent from δ and hence gives a new accident. So none of the case 1, 2 or 3 is possible. $\square$

Observe that $s = j_0 - 1$ is a special case. In addition to this condition, suppose we have an edge $e := (w[i_0-1], w[tc+s+1])$ which creates a collision $\delta' = (w[i_0-1], w[j_0-1]; w[tc+s+1])$ dependent on δ. e cannot occur in a single message graph, as that will imply $\mathsf{nbd}(* \to w[j]) \geq 2$ for some $j \in [0..i_0-1]$ which gives a new accident. But for a two message graph this is realizable (counter-examples) as illustrated in fig. 8(b) and 8(e). We summarise our discussions in the following lemma.

Lemma 4. *For $m \geq 1$, $M \in \mathcal{B}^m$ and $\pi \in \mathsf{Perm}$, following graphs exhaust all possible forms for $G_\pi(M)$.*

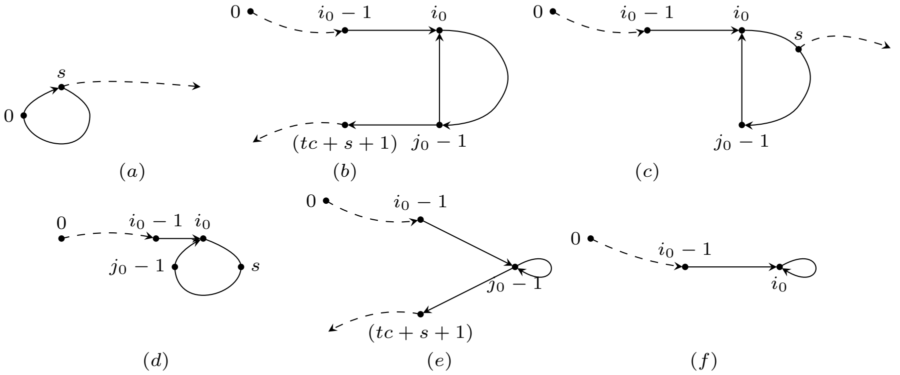


Fig. 8: *Characterizing all accident 1 structure graphs realizable by a single message. The dashed lines in these illustrations represent optional subwalks.* **Here the vertex $w[i]$ is represented by i, for notational simplicity**.

7 Revisiting $\mathsf{CP}_n(M_1, M_2)$ and $\mathsf{FCP}_n(M_1, M_2)$ Bounds [3]

In this section our main aim is to revise the proofs of (CP) and (FCP) bounds and consequently the PRF advantages in [3]. As mentioned earlier the motivation for this revision is our observation that one of the main tools [3, Lemma 10] in bounding $|\mathsf{struct}_1[\mathsf{coll}]|$ and $|\mathsf{struct}_1[\mathsf{Fcoll}]|$ is false.

We start off with a discussion that establishes the role of structure graphs in the PRF security analysis of CBC-MAC and EMAC. Note that we have already seen that bounding PRF advantages of CBC-MAC and EMAC are reduced to bounding full collision probability $\mathbf{FCP}^{\mathsf{pf}}_{2,\ell}$ and collision probability $\mathbf{CP}^{\mathsf{any}}_{2,\ell}$ respectively. So it would be sufficient to bound these probabilities. For this we first prove a general claim as stated in proposition 2.

Structure Graph Events. Let $\mathcal{M} = (M_1, \ldots, M_q)$ be a tuple of q messages. Let E be an event defined on the intermediate output sequence $\mathsf{out}^\pi(\mathcal{M})$ for a permutation π. We say that the event E **is defined by structure graph** if there is an event E' defined on the structure graph struct^{π} such that E holds if and only if E' holds. We call such an event a **structure graph event**. Moreover, we say that E is *non-free* if it is false for the free structure graph (the structure graph with accident zero). Note the the collision event for any distinct messages as well as the full collision event for prefix-free messages are examples of non-free structure graph events. In consistency with our notation, we denote the set of all structure graphs with a accidents and satisfying a non-free event E by $\mathsf{struct}_a(E)$.

Proposition 2. *Let E be a non-free structure graph event for the message tuple $\mathcal{M}$. Then,*

$$\mathsf{Pr}_{\Pi}[E] \leq \frac{|\mathsf{struct}_1[E]|}{2^n - m} + \frac{m^4}{2^{2n}}.$$

Proof. Note that for any structure graph event E,

$$\mathsf{Pr}_{\Pi}[E] = \sum_{a \geq 0} \mathsf{Pr}[\mathsf{struct}^{\Pi} \in \mathsf{struct}_a[E]].$$

As the event is non-free, the sum can be done for $a \geq 1$. Moreover, we know that $\mathsf{Pr}[\mathbf{Acc}(\mathsf{struct}^{\Pi}) \geq 2] \leq \frac{m^4}{2^{2n}}$. So the result follows from the Lemma 2 which bounds probability of realizing a structure graph with a accidents. □

7.1 Revisiting The $\mathbf{CP}_{2,\ell}$ Bound

Suppose $M_1 \in \mathcal{B}^{m_1}$ and $M_2 \in \mathcal{B}^{m_2}$ such that $M_1[m_1] \neq M_2[m_2]$, $0 \leq m_1 \leq m_2$, since otherwise we can remove the largest common suffix which does not change the collision probability. Note that the first message M_1 now can be empty (then M_2 is not as they are distinct) and in this case collision event means that $\mathsf{out}^{\Pi}(M_2)[m_2] = 0^n$. This is a structure graph event because 0 is a vertex of the structure graph. Due to proposition 2, we only need to bound the number of structure graphs with accident one satisfying coll event for the pair of messages. More precisely, we have to bound the size of the set

$$\mathsf{struct}_1(M_1, M_2)[\mathsf{coll}].$$

Case 1: M_1 is an empty message. In this case we have

$$\mathsf{struct}_1(M_1, M_2)[\mathsf{coll}] = \mathsf{struct}_1(M_2)[w_{M_2}[m_2] = 0].$$

Now, we make the following claim which is essentially [3, Lemma 14]:

Claim. $|\mathsf{struct}_1(M_2)[w_2[m_2] = 0]| \leq d(m_2)$

Proof of the claim. Let x be the smallest positive integer such that $w_{M_2}[x] = 0$. Let X be the label of the walk $w_{M_2}[0..x]$. If $M_2 = X^d$ some positive integer d, then $\mathsf{struct}_1(M_2)[w_{M_2}[x] = 0]$ contains exactly one structure graph. Note that x must divide m_2 and hence possible choices of such x is at most $d(m_2)$, the number of divisors of m_2. If $M_2 = X^d \| Y$ for some non-empty Y where d is the largest such integer of this form. If Y is a prefix then $W_2[m_2]$ is the point in the cycle and it must be 0. This can be zero only if $Y = X$ which contradicts the maximality of d. So now assume that $Y = Y_1 \| Y_2$ such that Y_1 is the largest common prefix of X and Y, and Y_2 is some non-empty string. If s is length of Y_1 then $Y_2[1] \neq Y[s+1]$. Thus, $w_2[dx+s+1] \neq w_2[s+1]$. As it is a zero-output structure graph, we can not have any collision. So there is no way to obtain $w_2[m_2] = 0$. This proves the claim.

Case 2: M_1 is not an empty message. In this case we have a collision

$$(u := w_1[m_1 - 1], v := w_2[m_2 - 1], z := w_2[m_2])$$

as the labels of the last edges for walks w_1 and w_2 are different. Any other collision, if any, must have the same source set $\{u, v\}$. Moreover, 0 can not have positive in-degree. Now we consider different sub-cases:

Case 2.1: Both w_1 and w_2 are paths: In this case, the union of $w_1[1..m_1-1]$ and $w_2[1..m_2-1]$ is a free graph (as $w_1[m_1-1]$ and $w_2[m_2-1]$ can not appear before in the graph and so no collision among the path can occur). This gives only one choice of the graph as shown in the figure 9(a). So the number of choices is bounded by at most 1. This is proved as part of the incorrect lemma [3, Lemma 15].

Case 2.2: w_2 is not a path: Then we have already characterized all possibilities of w_2. So there exists some integers t, c such that $w_2[1..t]$ is a path with $w_2[t-1] = u$ and $w_2[t] = p$, $w_2[t..t+c]$ is a cycle of length c such that $w_2[t+c-1] = v$. (Note that $w_2[t-1] \neq w_2[m_2-1]$.) Now, $w_1[m_1-1] = u$.

Claim 2.2.1: $w_1[1..m_1-1] = w_2[1..t-1]$ **and so** $m_1 = t$**.**

Proof. Let s be the length of largest common prefix of $w_1[1..m_1-1]$ and $w_2[1..t-1]$. If $s < t-1$ then in the walk w_1 there is no way to reach u without coming back to the walk $w_2[1..t-1]$. Coming back is not possible as it leads a collision with a different generator set. Similarly we can disprove that $s = t-1$ and $m_1 > t$. Thus, we have $m_1 = t$ and $w_1[1..m_1-1] = w_2[1..t-1]$. □

Now, we make two cases for the choices of $p = \mathsf{LCP}(M_1; M_2)$.

1. **Case 2.2.1.(a):** If $w_1[p] = z$ then we have the structure graph as illustrated in Fig 9(b). In this case M_1 is a prefix of M_2. The number of such structure graphs is again at most $d(m_2 - m_1)$ (similar to the previous case where M_1 is the empty message). This is also [3, Lemma 13].

2. **Case 2.2.1.(b):** If $w_1[p] \neq z$. Then we get a case which was not considered in [3]. In this case $w_1[p]$ should be a fresh node otherwise we get a collision with different source set. Thus, we get a structure graph which is shown in the Fig 9(c). Let $M_1 = A\|a$ where $A = M_1[1..t-1]$ and $a = M_1[t]$. Note that $t-1$ is the length of the largest common prefix of M_1 and M_2. Then,

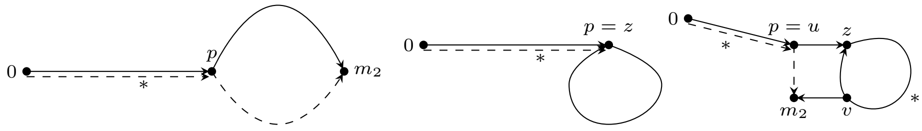
$$M_2 = A\|b\|(X\|x)^{d-1}\|X\|c, \text{ where } c = M_2[m_2], b = M_2[t], x = a \oplus b \oplus c.$$

The choice of X is variable. But it must satisfy the above for some $d > 1$. In fact X is determined by its length which is c. Again, c must divide $m_2 - m_1$ and hence the number of choices of c is at most $d(m_2 - m_1) - 1$.

This completes the characterization of all structure graphs satisfying coll with accident one and bounds the number of such graphs for all cases. Note that the cases 2.2.1.(a) and 2.2.1.(b) cannot hold simultaneously. But, case 2.2.1.(b) and 2.1 can hold simultaneously which makes the total count of these two cases at most $d(m_2 - m_1)$. Since the order of messages do not matter in coll we are done.

Lemma 5. *For $M_1 \in \mathcal{B}^{m_1}$, $M_2 \in \mathcal{B}^{m_2}$,*

1. *If $M_1 <_1 M_2$ then $\mathsf{struct}_1(M_1, M_2)[\mathsf{coll}]$ is of the form illustrated in Fig 9(b) and the number of such graphs is at most $d'(m_2)$.*
2. *If $M_1 <_2 M_2$ then $\mathsf{struct}_1(M_1, M_2)[\mathsf{coll}]$ is of the form illustrated in Fig 9(c) and the number of such graphs is at most $d'(m_2)$.*
3. *In all other cases, $\mathsf{struct}_1(M_1, M_2)[\mathsf{coll}]$ is of the form illustrated in Fig 9(a) and the number of such graphs is at most one.*



(a) (b) (c)

Fig. 9: *Characterizing all accident 1 structure graphs realizable by two messages which satisfy coll event. Dashed lines represents w_1 and solid lines represents w_2.*

Corollary 2. *$|\mathsf{struct}_1(M_1, M_2)[\mathsf{coll}]| \le d'(m_2)$ for any distinct messages M_1, M_2 with $m_1 \le m_2$. Thus,*

$$\mathbf{CP}^{\mathsf{any}}_{2,\ell} \le \frac{d'(m_2)}{2^n - m_1 - m_2} + \frac{(m_1+m_2)^4}{2^{2n}}$$

7.2 Revision of $\mathbf{FCP}^{\mathsf{pf}}_{2,\ell}$ Bound

Since Fcoll is a non-free structure graph event. So, by using proposition 2 we have,

$$\mathbf{FCP}^{(}_{n}\mathsf{pf})(M_1, M_2) \le \frac{|\mathsf{struct}_1(\mathsf{Fcoll})|}{2^n - m_1 - m_2} + \frac{(m_1+m_2)^4}{2^{2n}}.$$

Thus, it would be again sufficient to bound the number of structure graphs for two messages with accident one and satisfy full collision property. Bellare et al. [3] proved that $|\mathsf{struct}_1(\mathsf{Fcoll})| \leq 4\max\{m_1, m_2\}$. While bounding the $|\mathsf{struct}_1(\mathsf{Fcoll})|$, Bellare et al. proved a strong result that will be useful in our analysis also. We reproduce the lemma [3, Lemma 19] here in our notations.

Lemma 6. *For $b \in \{1, 2\}$ and any $i \in [0..m_b]$,*

$$|\mathsf{struct}_1(M_1, M_2)[w_b[i] \in w_b[0..i-1, i+1..m_b]]| \leq m_b.$$

Since the proof for lemma 6 can be found in [3], we skip it here. Equipped with the observation that the [3, Lemma 10] is incorrect, we revise the bound to $|\mathsf{struct}_1(\mathsf{Fcoll})| \leq 3(m_1 + m_2)$ and the new bound is

Lemma 7. $\mathbf{FCP}_n^{pf}(M_1, M_2) \leq \frac{3(m_1+m_2)}{2^n - m_1 - m_2} + \frac{(m_1+m_2)^4}{2^{2n}}$.

Proof. We need to bound the number of structure graphs for a pair of prefix-free messages $M_1 \in \mathcal{B}^{m_1}$ and $M_2 \in \mathcal{B}^{m_2}$ which has at most one accident and satisfies the Fcoll event. Note that the event implies that the structure graph must have at least one accident as the messages are prefix-free. The event Fcoll can be written as $w_2[m_2] \in w_2[0..m_2-1] \cup w_2[m_2] \in w_1[1..m_1]$.

Case 1: $\boldsymbol{w_2[m_2] \in w_2[0..m_2-1]}$. This case can be bounded by a straightforward application of lemma 6 to at most m_2.

Case 2: $\boldsymbol{w_2[m_2] \in w_1[1..m_1]}$. Suppose $\mathsf{Fcoll}(M_1; M_2)$ happens due to $w_2[m_2] = w_1[r]$ for an arbitrary $r \in [1..m_1 - 1]$. Then $\mathsf{Fcoll}(M_1; M_2)$ is equivalent to $\mathsf{coll}(M_1[1..r], M_2)$. For simplicity let $M_1' := M_1[1..r]$. Let $s := \mathsf{LCS}(M_1'; M_2)$. Then $M_1'[s-1] \neq M_2[m_2 - r + s - 1]$. Let $M_1^* = M_1'[1..s-1]$ and $M_2^* = M_2[1..m_2 - r + s - 1]$. From lemma 5 we know that $G^* \in \mathsf{struct}_1(M_1^*; M_2^*)[\mathsf{coll}]$ must be one of (a), (b) or (c) in fig. 9. Note that G^* is a subgraph of some $G \in \mathsf{struct}_1(M_1; M_2)[\mathsf{Fcoll}]$.

Case 2.1: $\boldsymbol{G^*}$ is as in fig. 9(a). Therefore w_1^* and w_2^* are paths. For a fixed r the only possible collision is at $(w_1^*[s-2], w_2^*[m_2 - r + s - 2]; w_1^*[s-1])$ and hence the number of such graphs is at most 1. There are at most m_1 possible values for r. So, the number of choices for $G \in \mathsf{struct}_1(M_1; M_2)[\mathsf{Fcoll}]$ is at most m_1.

Case 2.2: $\boldsymbol{G^*}$ is either as in fig. 9(b) or (c). Therefore, $b \in \{1, 2\}$ w_b^* is not a path. Without loss of generality assume $b = 1$. Let $p^* = \mathsf{LCP}(M_1^*; M_2^*)$. We know that $M_1^* <_1 M_1$ and $M_2^* <_1 M_2$. Therefore $M_1[1..p^*] = M_2[1..p^*]$. Now we must have a collision $(u, v\ ;\ z)$ in w_1^*. From lemma 5 we know that the graph can be either fig. 9(b) or (c) depending on whether $z = w_1^*[p^*]$ or $z = w_1^*[p^* + 1]$. Next we make two claims which will enable us to bound the two cases. The proofs for these two claims are given later in the section.

Claim 2.2.1: If $\boldsymbol{G^*}$ is fig. 9(b) then $\mathbf{w_1[\mathsf{LCP}(M_1; M_2)]}$ is not fresh in w_1.

Claim 2.2.2: If G^* is fig. 9(c) then $\mathbf{w_1[\mathsf{LCP}(M_1;M_2)+1]}$ is not fresh in w_1.

Recall that in a walk w a vertex $w[i]$ is not fresh if $\exists\ j \neq i$ such that $w[j] = w[i]$. By claim 2.2.1 we know that $w_1[\mathsf{LCP}(M_1; M_2)]$ is not fresh when G^* is as in fig. 9(b). Similarly, by claim 2.2.2 we know that $w_1[\mathsf{LCP}(M_1; M_2) + 1]$ is not fresh when G^* is as in fig. 9(c). So using lemma 6 we bound the number of such graphs G to at most $m_1 + m_1 = 2m_1$ when w_1^* is not a path. Similarly we have at most $2m_2$ choices when w_2^* is not a path. Therefore the total number of choices in case 2.2 is at most $2(m_1 + m_2)$. Combining case 1, 2.1 and 2.2 we have at most $3(m_1 + m_2)$ number of choices. The result follows. □

Proof for claim 2.2.1: Therefore $z = w_1^*[p^*]$. Let q be the minimum index such that $w_1^*[q] = w_1^*[p^*]$. Let $P = \mathcal{L}(w_1^*[0..p^*])$ and $X = \mathcal{L}(w_1^*[p^*..q])$, $c = q - p^*$. Then $M_1^* = P\|X$ and $M_2^* = P$. As M_1^* and M_2^* are formed by removing the largest common suffix from of M_1' and M_2 respectively, therefore $M_1' = (M_1^*\|X^{i_1}\|Y) = (P\|X^{i_1+1}\|Y)$ and $M_2 = (M_2^*\|X^{i_2}\|Y) = (P\|X^{i_2}\|Y)$ where $i_1, i_2 \geq 0$ are the largest such indices. Since M_1' and M_2 are prefix-free, we have $i_1 + 1 > i_2$. Now $M_1 = (M_1'\|Z) = (P\|X^{i_1+1}\|Y\|Z)$, where $|Z| \geq 0$. From now onwards we will work on the walk w_1 (instead of w_1^* which is a subwalk of w_1) corresponding to M_1. If Y is a prefix of X then $M_2 <_1 M_1$ which contradicts the prefix-free condition. So Y is not a preifx of X. If X is a prefix of Y then it contradicts the maximality of i_1, i_2. So X is not a preifx of Y. Assume $Y = Y_1\|Y_2$ such that Y_1 is the largest common prefix of X and Y, and Y_2 is some non-empty string. If p is the length of Y_1, then $Y_2[1] = Y[p + 1] \neq X[p + 1]$. Thus $M_1[1..i_2c + p] = M_2[1..i_2c + p]$ and $M_1[i_2c + p + 1] \neq M_2[i_2c + p + 1]$. So, $p = \mathsf{LCP}(M_1; M_2)$. Further since $i_2 < i_1 + 1$, $w_1[p]$ is traversed twice $\Rightarrow$ $\mathsf{LCP}(M_1; M_2)$ will not be fresh. Note that we started off with an arbitrary r. So $\mathsf{LCP}(M_1; M_2)$ will not be fresh irrespective of the value of r.

Proof for claim 2.2.2: Therefore $z = w_1^*[p^* + 1]$. As noted earlier in the revision of CP bound, this case was missing in [3] proof. Using a similar line of argument as in the previous case we can conclude that irrespective of the value of r, the cycle goes through $w_1[\mathsf{LCP}(M_1; M_2) + 1]$ twice $\Rightarrow$ $\mathsf{LCP}(M_1; M_2) + 1$ is not fresh.

Note that our approach in Case 2.2 above is a bit subtle. We used lemma 5 to identify a fundamental property (cycle goes through p or $p + 1$ twice) and then exploited this property to bound the counting. A straightforward approach of summing the counts for graphs in fig. 9(b) and (c) over all values of r will give a worse bound of $m_b d'(m_b)\ \ b \in \{1, 2\}$. To get a tighter bound of m_b we needed this subtlety. Now we extend the bound for $\mathbf{FCP}_n^{\mathsf{pf}}(M_1; M_2)$ to $\mathbf{FCP}_{q,\ell}^{\mathsf{pf}}$, in order to get the revised prf bound for CBC MAC.

$$\begin{aligned}
\mathbf{FCP}^{\mathsf{pf}}_{q,\ell} &\leq \sum_{i\neq j\in[q]} \mathbf{FCP}^{\mathsf{pf}}_{n}(M_i;M_j) \\
&\leq \sum_{i\neq j\in[q]} (\frac{3(m_i+m_j)}{2^n-m_1-m_2}+\frac{(m_i+m_j)^4}{2^{2n}}) \\
&\leq \sum_{i\neq j\in[q]} \frac{6(m_i+m_j)}{2^n}+\frac{(m_i+m_j)^4}{2^{2n}} \\
&\leq \frac{12mq}{2^n}+\frac{16mq\ell^3}{2^{2n}} \leq \frac{12\sigma q}{2^n}+\frac{16\sigma q\ell^3}{2^{2n}} \qquad (9)
\end{aligned}$$

Here we have computed the bound in terms of q, ℓ and σ. Another approach (as used in [3]) is to bound the value using q and ℓ only, in which case the bound will be

$$\mathbf{FCP}^{\mathsf{pf}}_{q,\ell} \leq \frac{12\ell q^2}{2^n}+\frac{16\ell^4 q^2}{2^{2n}}$$

Using proposition 1 and eq. 9 we get,

Theorem 2. $\mathbf{Adv}^{\mathit{pf}}_{\mathrm{CBC}}(q,\ell,\sigma) \leq \frac{14\sigma q}{2^n}+\frac{16\sigma q\ell^3}{2^{2n}}+\frac{q^2}{2^{n+1}}$

This gives a bound of $\mathrm{O}(\frac{\sigma q}{2^n})$ for $\ell < 2^{n/3}$. As noted earlier, this is a better bound whenever the average message length is much smaller than the length of the longest message.

8 Revised Security Analysis of EMAC

In this section we revisit the PRF analysis of EMAC due to Pietrzack [31]. We first identify the actual flaw in the proof and then provide a different proof to obtain, in fact, a better bound of EMAC (in terms of ℓ). For notational simplicity we will keep our bounds in order notation and avoid the constant factors.

8.1 Flaw and Revisision of PRF advantage of EMAC [31]

The proposed bound for EMAC as stated in [31]

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathrm{EMAC}}(q,\ell,\sigma) = O(\frac{q^2}{2^n}(1+\frac{\ell^8}{2^n}))$$

provided $\ell^2 \leq q$. Thus, it becomes tight bound $q^2/2^n$ when $\ell \leq \mathsf{min}(q^{1/2}, 2^{n/8})$. To show that we need to bound the collision probability $\mathbf{CP}_{q,\ell}$. We can group the q message into $O(q/\ell^2)$ groups, each group consists of about ℓ^2 messages. So collision event among q messages implies that collision occur in two of the groups. Since coll is a non-free event, we have seen in proposition 2 that

$$\mathbf{CP}_{q,\ell} = O(\frac{|\mathsf{struct}_1(\mathcal{M})[\mathsf{coll}]|}{2^n}) + O(\frac{q^4\ell^4}{2^{2n}}).$$

Applying this with $q = 2\ell^2$ (i.e. for two groups) we have

$$\mathbf{CP}_{q,\ell} = O(\frac{q^2}{\ell^4}) \times \mathbf{CP}_{\ell^2,\ell} = O(\frac{q^2 N}{\ell^4 2^n}) + O(\frac{q^2 \ell^8}{2^{2n}})$$

where N denote the number of accident one structure graphs satisfying coll for ℓ^2 messages with maximum length ℓ. The $O(q^2/\ell^4)$ term is due to the number of ways in which we can choose two groups. Author claimed that $N = O(\ell^4)$ ([31, Lemma 4]) and so plugging this bound for ℓ we have the desired bound. Now for proving this bound for N author considered two cases for a pair of messages M and M' (note that accident one and collision must occur for a pair of messages). More precisely, it can be shown that

$$N = \ell^4 \max_{M \not<_1 M'} |\mathsf{struct}_1(M.M')[\mathsf{coll}]| + \ell^4. \tag{10}$$

Recall that $M \not<_1 M'$ means that they become prefix-free after removing largest common suffix of M and M'.

Claim 1 of [31]. If $M \not<_1 M'$ then $|\mathsf{struct}_1(M.M')[\mathsf{coll}]| = 1$.

If this claim was happened to be true then $N = O(\ell^4)$. However, we have seen before there exists $M <_2 M'$ (such that $M \not<_1 M'$) with $|\mathsf{struct}_1(M, M')[\mathsf{coll}]| = d(\ell - 1)$. Thus, $|\mathsf{struct}_1(M, M')[\mathsf{coll} \wedge M \not<_1 M']| = O(d'(\ell))$. If we plug in this, we find the modified bound as $N = O(\ell^4(d'(\ell))^2)$ and so the revised bound for the collision probability becomes $O(q^2 d'(\ell)/2^n)$ which is not tight.

8.2 Simple Proof of EMAC

We have seen in the last subsection that the influence of the flaw from [3, Lemma 10] is more serious having tight bound of EMAC. So it is very crucial to revisit the security analysis of EMAC. One possible approach to fix the proof of [31], by bounding N in a different way. For example, we can consider two cases $M <_1 M'$ and $M <_2 M'$ (i.e., $M[1..m-1] <_1 M_2$ but $M \not<_1 M'$). For any pair of messages which are not related by any one of these two relations then the number of structure graphs can be shown to be one. However, we need to show that the remaining graphs is still about ℓ^4 (see second term of Eq. 10).

In this section we actually took a slightly different and, in fact simpler, approach. Instead of making groups of q messages we directly bound the number of structure graphs for a slightly different choices of permutations. We will ignore all those permutations (i.e. bad permutations) which induces one of the following:

1. For some pair of messages M_i and M_j the number of accident is two or more.
2. For some message M_i, the accident is one.

Let ϕ be the property to represent the complement of the event. Let S be a structure graph associated to a q-tuple of messages. We recall that S is an union of q walks w_i. We denote the sub-graph S_i and $S_{i,j}$ to represent the walk w_i

and $w_i \cup w_j$. Note that these are again structure graphs associated to M_i and (M_i, M_j) respectively. In this notation, ϕ is a property on all structure graphs S on $\mathcal{M}$ such that $\mathbf{Acc}(S_{i,j}) \leq 1$ for all $i \neq j$ and $\mathbf{Acc}(S_i) = 0$ for all i. We call a permutation good if its induced structure graph satisfies ϕ, otherwise we call bad. Now we claim our new bound.

Lemma 8. $\mathbf{CP}_{q,\ell}(\mathcal{M}) \leq \frac{q^2}{2^n} + \frac{\ell^2 q}{2^n} + \frac{\ell^4 q^2}{2^{2n}}$.

Proof. We first bound the probability of bad random permutation. For a bad permutation (1) there exists i and j such that the accident for the pair of message M_i and M_j is at least 2 or (2) there exists i, such that the accident for M_i is at least one. The first event can happen with probability $O(\ell^4 q^2/2^{2n})$ by using corollary 1. Similarly the second event can happen with $O(\ell^2 q/2^n)$. Now we bound the probability $p := \mathsf{Pr}[\mathsf{coll} \wedge \phi]$. Note that collision event implies that there exists i and j such that collision event holds for the message M_i and M_j. Now ϕ implies that accident of $S_{i,j}$ is one whereas accident of S_i and S_j are zero. In section 6 we have characterized all structure graphs for a pair of messages with accident one satisfying collision. Among all possibilities only one structure graph satisfies ϕ. Hence there is exactly one structure graph. This implies that $\mathsf{Pr}[\mathsf{coll}(M_i, M_j) \wedge \phi] = O(2^{-n})$. Hence, by summing over all possible i, j we have

$$\mathsf{Pr}[\mathsf{coll}(\mathcal{M}) \wedge \phi] = O(q^2/2^n).$$

Now we summarize above discussion as

$$\begin{aligned}
\mathbf{CP}_{q,\ell}(\mathcal{M}) &= \mathsf{Pr}_{\Pi}[\mathsf{coll}_{\Pi}(\mathcal{M}) \wedge (\mathsf{struct}^{\Pi}(\mathcal{M}) \in \mathsf{struct}(\mathcal{M})[\phi])] \\
&\quad + \mathsf{Pr}[\mathsf{struct}^{\Pi}(\mathcal{M}) \notin \mathsf{struct}^{\Pi}(\mathcal{M})[\phi]]] \\
&= \sum_{i \neq j} O(\frac{|\mathsf{struct}(M_i, M_j)[\mathsf{coll} \wedge \phi]|}{2^n}) + O(\ell^2 q/2^n) + O(\ell^4 q^2/2^{2n}) \\
&= O(q^2/2^n) + O(\ell^2 q/2^n) + O(\ell^4 q^2/2^{2n}) \qquad (11)
\end{aligned}$$

This completes the proof. □

Theorem 3. $\mathbf{Adv}^{\mathit{any}}_{\mathit{EMAC}}(q, \ell, \sigma) = O(\frac{2q^2}{2^n} + \frac{q\ell^2}{2^{2n}} + \frac{q^2\ell^4}{2^{2n}})$. *So if* $\ell \leq \min\{q^{1/2}, 2^{n/4}\}$ *then* $\mathbf{Adv}^{\mathit{any}}_{\mathit{EMAC}}(q, \ell, \sigma) = O(\frac{q^2}{2^n})$.

Note that our theorem gives tight bound for a better constraint than what we had before in [31]. The condition $q > \ell^2$ can be dropped if we assume $\ell \leq 2^{n/4-k}$ for some small k such that 2^{-k} is negligible. More precisely, if $\ell \leq 2^{n/4-k}$ then the PRF advantage of EMAC is about $\frac{q^2}{2^n} + \frac{1}{2^k}$.

9 Conclusion and Future Work

In this paper we have revisited the PRF security analysis of CBC-MAC and EMAC. We made the revision as we have found one of the main claims in the original papers providing improved bounds is not correct. This claim, in fact, influences some of the other claims. More importantly, the tight bound claim of EMAC becomes invalid even after a simple fix of the claim. So we feel that revision is essential and this paper serves this. Fortunately we have recovered same bounds, at least in terms of the order, for both constructions. While revising we attain potentially better bound of $O(\sigma q/2^n)$ for CBC-MAC. Moreover, we have found a better way to analyze EMAC which provides tight bound with a much relaxed constraint on message length ℓ. Namely our constraint is $\ell < 2^{n/4}$ whereas the original constraint was $\ell < 2^{n/8}$.

Acknowledgement

We have communicated with the authors of the papers [3, 31] and they have acknowledged our findings. We would like to thank them for giving their valuable time to go through our findings.

References

1. Information Technology – Security Techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms Using A Block Cipher. ISO/IEC 9797-1, International Organization for Standardization, Geneva, CH, 1999.
2. Mihir Bellare, Joe Kilian, and Phillip Rogaway. The Security of The Cipher Block Chaining Message Authentication Code. *J. Comput. Syst. Sci.*, 61(3):362–399, 2000.
3. Mihir Bellare, Krzysztof Pietrzak, and Phillip Rogaway. Improved Security Analyses for CBC MACs. In *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, pages 527–545, 2005.
4. A. Berendschot, B. den Boer, J. Boly, A. Bosselaers, J. Brandt, D. Chaum, I. Damgård, M. Dichtl, W. Fumy, M. van der Ham, C. Jansen, P. Landrock, B. Preneel, G. Roelofsen, P. de Rooij, and J Vandewalle. Final Report of Race Integrity Primitives, 1995.
5. D. J. Bernstein. A Short Proof of the Unpredictability of Cipher Block Chaining, 2005.
6. John Black and Phillip Rogaway. A Block-Cipher Mode of Operation for Parallelizable Message Authentication. In *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, pages 384–397, 2002.
7. John Black and Phillip Rogaway. CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions. *J. Cryptology*, 18(2):111–131, 2005.
8. Nilanjan Datta, Avijit Dutta, Mridul Nandi, Goutam Paul, and Liting Zhang. One-key Double-Sum MAC with Beyond-Birthday Security. *IACR Cryptology ePrint Archive*, 2015:958, 2015.
9. Whitfield Diffie and Martin E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

10. Yevgeniy Dodis, Rosario Gennaro, Johan Håstad, Hugo Krawczyk, and Tal Rabin. Randomness Extraction and Key Derivation Using the CBC, Cascade and HMAC Modes. In *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, pages 494–510, 2004.
11. Avijit Dutta, Mridul Nandi, and Goutam Paul. One-Key Compression Function Based MAC with BBB Security. *IACR Cryptology ePrint Archive*, 2015:1016, 2015.
12. M Dworkin. Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. NIST Special Publication 800-38b, National Institute of Standards and Technology, U. S. Department of Commerce, 2005.
13. William F. Ehrsam, Carl H. W. Meyer, John L. Smith, and Walter L. Tuchman. Message Verification and Transmission Error Detection by Block Chaining. US Patent 4074066, 1976.
14. Peter Gaži, Krzysztof Pietrzak, and Stefano Tessaro. Tight Bounds for Keyed Sponges and Truncated CBC, 2015.
15. Peter Gazi, Krzysztof Pietrzak, and Michal Rybár. The Exact PRF-Security of NMAC and HMAC. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 113–130, 2014.
16. Sergey Gorbunov and Charles Rackoff. On the Security of Cipher Block Chaining Message Authentication Code.
17. Tetsu Iwata and Kaoru 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.
18. Tetsu Iwata and Kaoru Kurosawa. Stronger Security Bounds for OMAC, TMAC, and XCBC. In *Progress in Cryptology - INDOCRYPT 2003, 4th International Conference on Cryptology in India, New Delhi, India, December 8-10, 2003, Proceedings*, pages 402–415, 2003.
19. Éliane Jaulmes, Antoine Joux, and Frédéric Valette. On the Security of Randomized CBC-MAC Beyond the Birthday Paradox Limit: A New Construction. In *Fast Software Encryption, 9th International Workshop, FSE 2002, Leuven, Belgium, February 4-6, 2002, Revised Papers*, pages 237–251, 2002.
20. Charanjit S. Jutla. PRF Domain Extension Using DAGs. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, pages 561–580, 2006.
21. Kaoru Kurosawa and Tetsu Iwata. TMAC: Two-Key CBC MAC. *IEICE Transactions*, 87-A(1):46–52, 2004.
22. Ueli M. Maurer. Indistinguishability of Random Systems. In *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, pages 110–132, 2002.
23. Kazuhiko Minematsu and Toshiyasu Matsushima. New Bounds for PMAC, TMAC, and XCBC. In *Fast Software Encryption, 14th International Workshop, FSE 2007, Luxembourg, Luxembourg, March 26-28, 2007, Revised Selected Papers*, pages 434–451, 2007.
24. Mridul Nandi. Fast and Secure CBC-type MAC Algorithms. In *Fast Software Encryption, 16th International Workshop, FSE 2009, Leuven, Belgium, February 22-25, 2009, Revised Selected Papers*, pages 375–393, 2009.
25. Mridul Nandi. Improved Security Analysis for OMAC as a Pseudorandom Function. *J. Mathematical Cryptology*, 3(2):133–148, 2009.

26. Mridul Nandi. A Unified Method for Improving PRF Bounds for a Class of Blockcipher Based MACs. In *Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, Revised Selected Papers*, pages 212–229, 2010.
27. Mridul Nandi and Avradip Mandal. Improved Security Analysis of PMAC. *J. Mathematical Cryptology*, 2(2):149–162, 2008.
28. Jacques Patarin. *Etude des Générateurs de Permutations Pseudo-aléatoires Basés sur le Schéma du DES*. PhD thesis, Université de Paris, 1991.
29. Jacques Patarin. The "Coefficients H" Technique. In *Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers*, pages 328–345, 2008.
30. Erez Petrank and Charles Rackoff. CBC MAC for Real-Time Data Sources. *J. Cryptology*, 13(3):315–338, 2000.
31. Krzysztof Pietrzak. A Tight Bound for EMAC. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, pages 168–179, 2006.
32. Serge Vaudenay. Decorrelation: A Theory for Block Cipher Security. *J. Cryptology*, 16(4):249–286, 2003.
33. Mark N. Wegman and Larry Carter. New Classes and Applications of Hash Functions. In *20th Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 29-31 October 1979*, pages 175–182, 1979.
34. S. Wigert. Sur l'ordre de grandeur du nombre des diviseurs d'un entier. *Ark. Mat. Astron. Fys.*, 3(18):9, 1907.
35. Kan Yasuda. A New Variant of PMAC: Beyond the Birthday Bound. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, pages 596–609, 2011.
36. Liting Zhang, Wenling Wu, Han Sui, and Peng Wang. 3kf9: Enhancing 3GPP-MAC Beyond the Birthday Bound. In *Advances in Cryptology – ASIACRYPT 2012: 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012, Proceedings*, pages 296–312, 2012.