IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 0, No. 0, pp. 0–0. DOI:10.46586/tosc.v0.i0.0-0

Revisiting Randomness Extraction and Key Derivation Using the CBC and Cascade Modes

Niranjan Balachandran[1], Ashwin Jha[2], Mridul Nandi[3] and Soumit Pal[3]

[1] Indian Institute of Technology, Bombay, India
niranj@math.iitb.ac.in
[2] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
ashwin.jha@cispa.de
[3] Indian Statistical Institute, Kolkata, India
{mridul.nandi,soumitpal378}@gmail.com

Abstract. In this paper, we revisit a celebrated result by Dodis et al. from CRYPTO 2004, in relation with the suitability of CBC-MAC and cascade construction for randomness extraction. We first observe that the proof of three key sub-results are missing in the paper, which makes it difficult to verify the authors' claims. Then, using a detailed and thorough analysis of the collision probability for both the CBC function and the cascade construction, we provide the missing proofs, thereby establishing the veracity of this old result. As a side-effect, we have made a significant advancement in the characterization of graph-based analysis of CBC and cascade construction, which could be of independent interest.

Keywords: CBC-MAC · cascade construction · tight bound · missing proofs

1 Introduction

At CRYPTO 2004, Dodis et al. proposed that the CBC and cascade functions are good candidates for randomness extractors [DGH+04], provided the input distribution has sufficient randomness. In the following, we briefly describe the CBC and cascade constructions.

The CBC Function was first introduced by Ehrsam et al. [EMST76] in a block cipher mode of operation for encryption. CBC-MAC — one of the most popular message authentication code (MAC) mode and a former international standard [2711] — is directly based on this function. Let n be a positive integer, and π be a permutation of $\{0,1\}^n$. The CBC function $\mathrm{CBC}_\pi : (\{0,1\}^n)^* \to \{0,1\}^n$ associated with π, is defined recursively over a sequence of n-bit blocks $M = (M[1] \cdots M[l])$ in the following manner:

$$\mathrm{CBC}_\pi(M) := \begin{cases} 0^n & M = \perp, \\ \pi(M[l] \oplus \mathrm{CBC}_\pi(M[1] \cdots M[l-1])) & \text{otherwise}, \end{cases}$$

where $\perp$ denotes the empty string. Several other modes like EMAC [BKR94, BdBB+95], ECBC, FCBC, and XCBC [BR00], TMAC [KI03], OMAC [IK03] etc. are also directly based on the CBC function and, together with CBC-MAC and several other modes [Nan09, Yas10], form the CBC-MAC family.

The Cascade Construction was first introduced in the celebrated GGM construction [GGM84] by Goldreich, Goldwasser and Micali. The Cascade function $\mathrm{Casc}_f : (\{0,1\}^n)^* \to \{0,1\}^n$, associated with a compression function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$,

is defined recursively over a sequence of n-bit blocks $M = (M[1] \cdots M[l])$ in the following manner:

$$\textsc{Casc}_f(M) := \begin{cases} 0^n & M = \perp, \\ f(\textsc{Casc}_f(M[1] \cdots M[l-1]) \| M[l]) & \text{otherwise.} \end{cases}$$

Like CBC, the cascade construction has also been used in several modes, since its first use in GGM. Perhaps the most notable examples are HMAC and NMAC by Bellare et al. [BCK96]. Apart from this the envelope MAC by Tsudik [Tsu92], and AMAC by Bellare et al. [BBT16] are also popular examples of the cascade construction. Indeed, almost any iterated construction, including the CBC function, can be viewed as a cascade construction.

The Collision Probability Problem. Coming back to [DGH+04], a key ingredient in the analysis is the study of these modes as universal hash functions, i.e., the study of collision probability problem with respect to CBC and Cascade construction. For brevity, we discuss the problem in context of CBC function. The related problem for cascade construction is defined analogously.

In the following discussion, we assume that Π is a secret permutation chosen uniformly at random from the set of all permutations of $\{0,1\}^n$. Let M and M' be two distinct inputs. Let $\mathtt{Coll}_\Pi(M, M')$ denote the event $\mathrm{CBC}_\Pi(M) = \mathrm{CBC}_\Pi(M')$. Extending the notation, we similarly define the collision event for a tuple of $q \geq 2$ distinct inputs $M^q = (M_1, \ldots, M_q)$, as

$$\mathtt{Coll}_\Pi(M^q) := \bigcup_{i<j} \mathtt{Coll}_\Pi(M_i, M_j).$$

We define the *collision probability* as $\mathsf{CP}_\Pi(M^q) := \Pr_\Pi\left(\mathtt{Coll}_\Pi(M^q)\right)$.

Let $\mathsf{CP}^{\mathsf{atk}}_\Pi(q, \ell, \sigma) = \max_{M^q} \mathsf{CP}_\Pi(M^q)$, where the maximum is taken over all tuples of q distinct inputs M^q having at most ℓ blocks each, and at most σ blocks in total, and satisfying the constraint atk, which could be one of the following:

1. eq: each input has exactly ℓ blocks;
2. pf: no input is a prefix[1] to others;
3. any: no restrictions over the choice of inputs.

The CBC collision probability is not only interesting in context of randomness extractors. Indeed, a crucial ingredient in the security proof of any mode in the CBC-MAC family is to derive a good upper bound on $\mathsf{CP}^{\mathsf{atk}}_\Pi(q, \ell, \sigma)$ for some fixed choice of atk that depends on the mode itself. In this paper, we aim to study this specific problem, that we conveniently name as the *CBC collision probability problem* or CBC CPP in short, for $q = 2$.

1.1 Related Works

In most of the previous works, the CBC CPP has been studied within the ambit of pseudorandom function advantage of some mode in CBC-MAC family. The first major result on CBC CPP appeared in [BKR94], where Bellare et al. showed that the pseudorandom function or PRF advantage of CBC-MAC is bounded by $O(q^2\ell^2 2^{-n})$, under the assumption that all the queries consist of a fixed number of blocks. This immediately gives $\mathsf{CP}^{\mathsf{eq}}_\Pi(q, \ell, \sigma) = O(q^2\ell^2 2^{-n})$. Maurer [Mau02], and later Bernstein [Ber05], also derived similar bounds using different proof techniques. Petrank and Rackoff [PR00] extended the approach in [BKR94] to obtain similar bound on $\mathsf{CP}^{\mathsf{pf}}_\Pi(q, \ell, \sigma)$. Later, Gorbunov and

[1] A string X is called a prefix of Y if $Y = X\|X'$ for some string X', where $\|$ denotes the concatenation operation.

Rackoff extended [GR16] Bernstein's proof technique [Ber05] to also obtain a similar bound.

In [DGH+04], Dodis et al. studied CBC CPP for $q = 2$ and $\mathsf{atk} = \mathsf{eq}$, in connection with randomness extractors, and claimed[2] that $\mathsf{CP}^{\mathsf{eq}}_{\Pi}(2, \ell) = O(2^{-n} + \ell(d^*(\ell))^2 2^{-2n} + \ell^6 2^{-3n})$, where $d^*(\ell)$ is the maximum, over all $\ell' \leq \ell$, of the number of positive numbers that divide ℓ'. Bellare et al. [BPR05] employed a novel graph based analysis, called *structure graphs*, to show that $\mathsf{CP}^{\mathsf{any}}_{\Pi}(2, \ell) = O(d^*(\ell)2^{-n} + \ell^4 2^{-2n})$. All subsequent works on CBC CPP, since [BPR05], have employed this approach. Most notably, Pietrzak used structure graphs to show that $\mathsf{CP}^{\mathsf{any}}_{\Pi}(q, \ell, \sigma) = O(q^2 2^{-n} + q^2 \ell^8 2^{-2n})$ in [Pie06], which simplifies to $\mathsf{CP}^{\mathsf{any}}_{\Pi}(q, \ell, \sigma) = O(q^2 2^{-n})$ while $\ell \ll 2^{n/8}$. Later, Jha and Nandi [JN16] discovered that one of the fundamental lemmata from [BPR05] on structure graphs is in fact wrong, which invalidated the results in [BPR05], and its subsequent applications [Pie06, Yas10, GPT15]. They go on to revise the bounds given in [BPR05, Pie06], specifically showing that $\mathsf{CP}^{\mathsf{any}}_{\Pi}(q, \ell, \sigma) = O(q^2 2^{-n} + q\ell^2 2^{-n} + q^2 \ell^4 2^{-2n})$, which simplifies to $\mathsf{CP}^{\mathsf{any}}_{\Pi}(q, \ell, \sigma) = O(q2^{-n/2})$ while $\ell \ll 2^{n/4}$. This is an improved and tight (in terms of the number of queries) bound for $\ell \ll 2^{n/4}$.

On the contrary, the cascade construction has been mostly studied in a reduction-based proof style [GGM84, BCK96, Bel06, GPR14, Nan21], where the collision probability problem is seldom in focus. Dodis et al. studied the cascade construction CPP [DGH+04] for $q = 2$ and $\mathsf{atk} = \mathsf{eq}$, in connection with randomness extractors, and claimed exactly the same bound as they claimed in case of CBC.

Crucially, a proof for these claims is missing till now!

The Missing Full Version of [DGH+04]: It is worth noting that, in the CRYPTO 2004 proceedings version, the authors frequently reference the availability of proofs for their claims in the full version of their paper. However, as of our knowledge up to this point, no publicly accessible proof has been available. We have made several attempts to contact the authors with the aim of obtaining the full version of the CRYPTO 2004 proceedings paper. Unfortunately, we have consistently received responses indicating that a full version might be unavailable. On one occasion, we were directed to [BPR05] as a potential source for the complete proof. It's important to clarify that [BPR05] exclusively offers a bound of $d^*(\ell)2^{-n}$ specifically for the CBC CPP (see Remark 1.1 for a comparison with our bound), with no mention of the cascade construction. Additionally, it has been identified that even this analysis, as reported in [JN16], overlooks an important case. In summary, despite multiple efforts to acquire the full version of the paper and, more significantly, a proof for the aforementioned claims, it can be concluded that a rigorously documented proof does not appear to be publicly accessible. Consequently, at this stage, the validity of the statements presented in [DGH+04, Lemma 3, Proposition 1, and Proposition 2], as well as subsequent results relying on them, remains uncertain.

Structure Graphs. Bellare et al. first introduced structure graphs [BPR05] to analyze the PRF security of CBC-MAC and ECBC, and as a result to study the CBC CPP as well. Following a long line of previous works, we also employ this useful technique in our quest for a progress on CBC CPP. Of note, Bellare et al. acknowledged some unpublished techniques of Dodis et al. [DGH+04] as the basis for a key structure graph related lemma. So it is possible that the seeds for this approach were sown in [DGH+04], although it seems unlikely as there is no explicit mention in the said paper. A fully justified description of this tool warrants a separate section, with few more notations and space. We refer the readers to section 3 for a detailed and formal description. Here, we give a brief and hopefully sufficient introduction so as to convey the motivation of this paper.

Consider the CBC evaluation illustrated in Figure 1.1, over a 4-block input M. We have $x^{\pi}_M[i] := y^{\pi}_M[i-1] \oplus M[i]$ for all $i \in \{1, 2, 3, 4\}$, where $x^{\pi}_M[i]$ and $y^{\pi}_M[i]$ are referred as the

[2] A proof of this claim is missing.

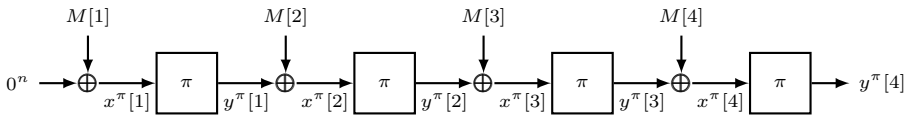

Figure 1.1: Evaluation of CBC function over a 4-block input M.

intermediate inputs and *outputs*, respectively. We will drop M from the notation whenever it is clear from the context. Additionally, suppose, $M[4] = M[2]$, and the permutation π is defined in such a way that $y^\pi[1] = y^\pi[3]$, i.e., $x^\pi[1] = M[1] = x^\pi[3] = y^\pi[2] \oplus M[3]$. Let's trace this particular CBC computation graphically. We start with an arbitrary vertex, say u_0, denoting the intermediate output $y^\pi[0] = 0^n$. Next, to represent a transition from $y^\pi[0]$ to $y^\pi[1]$, we take another vertex u_1 which corresponds to $y^\pi[1]$, and draw an edge (u_0, u_1) with label M_1. We take vertex u_2 which corresponds to $y^\pi[2]$, and draw an edge (u_1, u_2) with label $M[2]$ to represent the transition from $y^\pi[1]$ to $y^\pi[2]$. Now, we already know that $y^\pi[3] = y^\pi[1]$ due to the choice of permutation. So we simply draw an edge (u_2, u_1) with label $M[3]$ to represent the transition from $y^\pi[2]$ to $y^\pi[1]$. Further, due to $M[4] = M[2]$ and $y^\pi[3] = y^\pi[1]$, the existing edge (u_1, u_2) with label $M[2] = M[4]$ already represents the transition from $y^\pi[3]$ to $y^\pi[4]$. The resulting graph, illustrated in Figure 1.2, is the so-called structure graph corresponding to the permutation π and input M. By extending this idea for a tuple of q inputs, starting at vertex u_0 and introducing a new vertex for each new intermediate output, we can get a structure graph for q inputs.

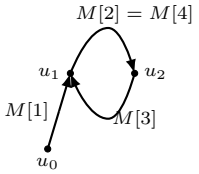

Figure 1.2: Structure graph corresponding to the 4-block input M and permutation π.

It is easy to see that the structure graph preserves the collision property, or more importantly, the *accidents*[3] among the intermediate outputs. More importantly, for two inputs M and M' having l and l' many blocks, respectively, $\mathrm{CBC}_\pi(M) = \mathrm{CBC}_\pi(M')$, if and only if the vertices corresponding to $y^\pi_M[l]$ and $y^\pi_{M'}[l']$ are the same in the corresponding structure graph. Consequently, in all structure graph based analysis, the goal is to characterize certain *good* graphs, which satisfy the collision property and are relatively easy[4] to count. The occurrence of all other graphs is upper bounded via some generic lemmata. In [BPR05], the good graphs roughly correspond to the ones in which no subgraph corresponding to any pair of messages contains more than one cycle. The characterization of good graphs in [Pie06] is slightly more involved, and ultimately futile as shown in [JN16]. Jha and Nandi employed a much simpler criteria [JN16] for good graphs. In addition to the condition used in [BPR05], they additionally require that the subgraph corresponding to each message is acyclic. This greatly reduces the counting complexity at the expense of an additional term $O(q\ell^2 2^{-n})$ in the bound, which simplifies to $q2^{-n/2}$ for $\ell \ll 2^{n/4}$. Unfortunately, this is the limit of this approach in terms of lifting the bound for larger values of ℓ, as the additional term $O(q\ell^2 2^{-n})$ is a necessary evil for such simplifications.

In this paper, our goal is to use the structure graph technique to establish the veracity of the three claims made in [DGH+04], vis-á-vis, the CBC and cascade construction collision probability problems.

[3] All the collisions which are unpredictable given the messages and temporally old collisions.
[4] Have a low number of accidents, or surprising collisions.

1.2 Our Contributions

Our chief technical contributions are twofold:

- First, we give a complete characterization of structure graphs (see section 4) for a pair of messages, having at most 2 accidents and satisfying the collision event. This encompasses a similar exercise previously performed by Jha and Nandi for at most 1 accident.
- Second, we derive tight bounds for the CBC CPP (see section 5) when $q = 2$ and $\mathsf{atk} \in \{\mathsf{any}, \mathsf{eq}\}$. In a similar manner, we derive tight bounds for the cascade construction CPP (see section 6) when $q = 2$ and $\mathsf{atk} = \mathsf{eq}$. As a side-effect, our bounds for the eq case finally validate (see section 7) [DGH+04, Lemma 3] and [DGH+04, Proposition 1 and 2] — the chief technical contributions in [DGH+04].

In a nutshell, by providing the missing proofs, we establish the complete veracity of three key results in [DGH+04].

Remark 1.1 (Comparison with the CBC CPP Bounds in [BPR05] and [JN16]). As noted before, the CBC CPP has also been studied before by Bellare et al. in [BPR05] and Jha and Nandi in [JN16]. These two previous works study the problem for $\mathsf{atk} = \mathsf{any}$, and thus, their result also imply an identical bound for $\mathsf{atk} = \mathsf{eq}$.

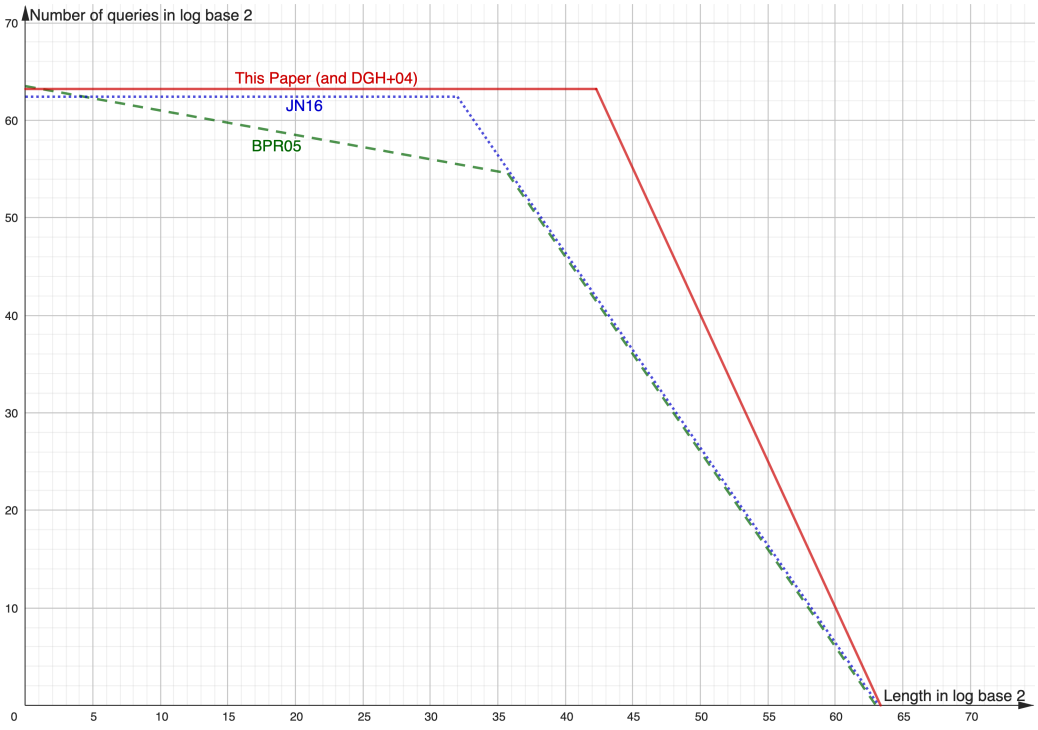

Figure 1.3: $(\log_2 \ell, \log_2 q)$-Trade-off Graph for the bounds of CBC CPP.

We remark that our — and hence the [DGH+04] — bound is tighter than both these bounds for $\mathsf{atk} = \mathsf{eq}$. A comparative graph[5] in Figure 1.3 illustrates this observation for $n = 128$ and the advantage value 1. In particular, we have

1. the [BPR05] bound,

$$B_1(\ell, q) := \frac{q^2 d^*(\ell)}{2^n} + \frac{8q^2\ell^4}{2^{2n}}$$

[5] Using GeoGebra Classic: https://www.geogebra.org/classic

Note that, we use the revised bound as stated in [JN16].

2. the [JN16] bound,
$$B_2(\ell, q) := \frac{q^2}{2^n} + \frac{q\ell^2}{2^n} + \frac{8q^2\ell^4}{2^{2n}}.$$
3. the bound in this paper (and [DGH+04]),
$$B_3(\ell, q) := \frac{q^2}{2^n} + \frac{9q^2\ell d^*(\ell)^2}{2^{2n}} + \frac{5q^2\ell^6}{2^{3n}}.$$

We further use the simplifying assumptions that $d^*(\ell) \leq 2\sqrt{\ell}$ (as stated in [DGH+04]), $q, \ell \leq 2^{n/2}$, and $q\ell^2 \leq 2^{n-1.5}$. The latter two assumptions prevent the bounds from becoming trivially greater than 1. We rewrite the three bounds using these assumptions:

$$B_1(\ell, q) \leq \begin{cases} \frac{2q^2\sqrt{\ell}}{2^n} & \ell \leq 2^{\min\{\frac{2n-6}{7}, n-1.5\}} \\ \frac{16q^2\ell^4}{2^{2n}} & \ell > 2^{\min\{\frac{2n-6}{7}, n-1.5\}} \end{cases} \quad B_2(\ell, q) \leq \begin{cases} \frac{3q}{2^{n/2}} & \ell \leq 2^{\min\{n/4, n-1.5\}} \\ \frac{3q\ell^2}{2^n} & \ell > 2^{\min\{n/4, n-1.5\}} \end{cases}$$

$$B_3(\ell, q) \leq \begin{cases} \frac{3q^2}{2^n} & \ell \leq 2^{\min\{\frac{n-6}{2}, \frac{2n-5}{6}\}} \\ \frac{15q^2\ell^6}{2^{3n}} & \ell > 2^{\min\{\frac{n-6}{2}, \frac{2n-5}{6}\}} \end{cases}$$

We can thus compare[6] the three bounds in three different intervals of message lengths:

- $[0, 2^{n/4}]$: In this range B_1 has a dominating term of $q^2\sqrt{\ell}/2^n$, B_2 has a dominating term of $q/2^{n/2}$, and B_3 has a dominating term of $q^2/2^n$. This clearly shows that B_3 (and B_2) are smaller than B_1 in this range. Furthermore, B_3 is qualitatively better than B_2.
- $(2^{n/4}, 2^{n/3}]$: Here, B_1 is dominated by $q^2\ell^4/2^{2n}$, which is qualitatively better than the dominating term, $q\ell^2/2^n$ of B_2. However, both these bounds are worse than B_3, which is still dominated by $q^2/2^n$.
- $(2^{n/3}, 2^{n/2}]$: In this range, B_1 and B_2 are still dominated by $q^2\ell^4/2^{2n}$, and $q\ell^2/2^n$, respectively, whereas B_3 is dominated by $q^2\ell^6/2^{3n}$, which is smaller than the dominating terms in B_1 and B_2 while $\ell < 2^{n/2}$.

The above discussion shows that our (and [DGH+04]) bound is strictly better than both [BPR05] and [JN16], while $\ell < 2^{n/2}$. For $\ell > 2^{n/2}$, as noted before, both the bounds become greater than 1. So, a comparison is moot beyond this point.

2 Preliminaries

For two positive integers $a \leq b$, we write $[a, b]$ to denote the set $\{a, a+1, \ldots, b\}$. We simply write $[b]$ and $(b]$ when $a = 1$ and $a = 0$ respectively. The set of all bit strings (including the empty string) is denoted $\{0,1\}^*$. The length of any bit string $X \in \{0,1\}^*$, denoted $|X|$, is the number of bits in X. For $X, Y \in \{0,1\}^*$, $Z = X\|Y$ denotes the concatenation of X and Y, where X and Y are the prefix and suffix of Z, respectively. For $n \in \mathbb{N}$, $\{0,1\}^n$ denotes the set of all bit strings of length n, and $\{0,1\}^{\leq n} := \bigcup_{i=0}^{n}\{0,1\}^i$. For $X, Y \in \{0,1\}^*$, $X\|Y$ denotes the concatenation of X and Y. For $0 \leq k \leq n$, we define the falling factorial $(n)_k := n!/(n-k)! = n(n-1)\cdots(n-k+1)$. The set of all permutations of $\{0,1\}^n$ is denoted $\mathcal{P}(n)$. For a finite set $\mathcal{X}$, $\mathsf{X} \xleftarrow{\$} \mathcal{X}$ denotes the uniform at random sampling of X from $\mathcal{X}$.

[6] We ignore the constant factors in the bounds for this comparison.

2.1 A Simple Counting Problem

Consider the following problem: Given a positive integer k, find the number of pairs (x, y) of positive integers such that k can be expressed as a positive integer linear combination of x and y. More precisely, we want to find an upper bound on the size of the following set

$$L_k := \{(x, y) \in \mathbb{N}^2 : \exists\ c, d \in \mathbb{N}, cx + dy = k\}.$$

Let $d(k)$ denote the the number of divisors of k and $d^*(k) = \max_{a \in [k]} d(a)$.

Lemma 2.1. *For any positive integer k, $|L_k| \leq k(d^*(k))^2$.*

Proof. Clearly we have at most k many choices for cx and for each choice of cx we have exactly one choice of dy. Once we fix cx and dy, the number of ways we can choose x and y is at most $d^*(k)^2$ ways as x and y are divisors of cx and dy respectively. □

2.2 The CBC Collision Probability Problem

Throughout the paper, n denotes the *block size*, and any $X \in \{0,1\}^n$ is referred as a *block*. For some positive integer ℓ, and any non-empty $M \in (\{0,1\}^n)^{\leq \ell}$, $M[1] \cdots M[m] \xleftarrow{n} M$ denotes the *block parsing* of M, where $|M_i| = n$ for all $i \in [m]$.

CBC Function. The CBC function, based on a permutation $\pi \in \mathcal{P}(n)$, takes as input a non-empty message $M \in (\{0,1\}^n)^{\leq \ell}$ and computes (see Figure 2.1) the output $\mathrm{CBC}_\pi(M) := y^\pi_M[m]$ on $(M[1] \cdots M[m]) \xleftarrow{n} M$ inductively as described below:
$y^\pi_M[0] = 0^n$ and for $1 \leq i \leq m$, we have

$$\begin{aligned} x^\pi_M[i] &:= y^\pi_M[i-1] \oplus M_i, \\ y^\pi_M[i] &:= \pi(x^\pi_M[i]). \end{aligned} \tag{1}$$

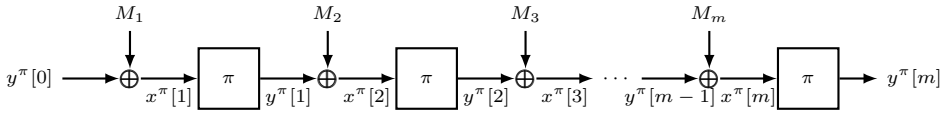

Figure 2.1: Evaluation of CBC function over an m-block message M. Note that, to lighten the notations we skipped M from the notations for intermediate input and output tuples.

We call $x^\pi_M := (x^\pi_M[i])_{i \in [m]}$ and $y^\pi_M := (y^\pi_M[i])_{i \in (m]}$, the *intermediate input* and *intermediate output* tuples, respectively, associated to π and M. Note that, the intermediate input vector x^π_M is uniquely determined by y^π_M and the message M. Going forward, we will drop π and M from the notation whenever they are clear from the context.

The CBC Collision Probability Problem. Let M and M' be two distinct inputs having m and m' many blocks, respectively, and $\pi \in \mathcal{P}(n)$. Let $\mathtt{Coll}_\pi(M, M')$ denote the event $\mathrm{CBC}_\pi(M) = y^\pi_M[m] = y^\pi_{M'}[m'] = \mathrm{CBC}_\pi(M')$. We call $\mathtt{Coll}_\pi(M, M')$ the *collision event* for a pair of inputs M and M'.

By extending the notation, we similarly define the collision event for a tuple of $q \geq 2$ distinct inputs $M^q = (M_1, \ldots, M_q)$, as

$$\mathtt{Coll}_\pi(M^q) = \bigcup_{i<j \in [q]} \mathtt{Coll}_\pi(M_i, M_j). \tag{2}$$

We define *collision probability* as $\mathsf{CP}(M^q) = \Pr\left(\mathtt{Coll}_\Pi(M^q)\right)$, where the probability is computed over the randomness of $\Pi \leftarrow\!\!\$\ \mathcal{P}(n)$. Let

$$\mathsf{CP}^{\mathsf{atk}}_{q,\ell,\sigma} = \max_{M^q} \mathsf{CP}(M^q)$$

where the maximum is taken over all q-tuples of distinct inputs M^q having at most ℓ blocks each, and the total length over all q inputs is at most σ. Further, the message tuple satisfies the input constraint atk, which could be one of the following:

1. eq: each input has exactly ℓ blocks;
2. pf: no input is a prefix to others;
3. any: no restrictions over the choice of inputs.

Note that, $\mathsf{CP}^{\mathsf{atk}}_{q,\ell,\sigma} \leq \binom{q}{2}\mathsf{CP}^{\mathsf{atk}}_{2,\ell}$ as the collision for q inputs is the union of collision events for each of the $\binom{q}{2}$ pairs of inputs. Bellare et al. [BPR05] proved that

$$\mathsf{CP}^{\mathsf{any}}_{2,\ell} \leq \frac{2d^*(\ell)}{2^n} + \frac{64\ell^4}{2^{2n}}. \tag{3}$$

where $d^*(\ell) = \max_{\ell' \leq \ell} d(\ell')$ and $d(\ell')$ is the the number of divisors of ℓ'. In [Wig07], Wigert showed that $d^*(\ell) = \ell^{1/\Theta(\ln \ln \ell)} = \ell^{o(1)}$. For equal length inputs, Dodis et al. claim (see [DGH+04, Lemma 3]) that

$$\mathsf{CP}^{\mathsf{eq}}_{2,\ell} - \frac{1}{2^n} = \mathcal{O}(\frac{\ell \times (d^*(\ell))^2}{2^{2n}}) + \mathcal{O}(\frac{\ell^6}{2^{3n}}).$$

Remark 2.1 (suffix-free message pairs). When we bound $\mathsf{CP}^{\mathsf{eq}}_{2,\ell}$ and $\mathsf{CP}^{\mathsf{any}}_{2,\ell}$, without loss of generality, we can assume that the two messages M, M' do not have any common suffix. Since otherwise, we can remove the common suffix and let M_1, M_1' be the messages after removing the common suffixes. It is easy to see that whenever collision holds for (M, M'), the collision must happen for (M_1, M_1') also.

3 Structure Graph

Fix a tuple of q distinct inputs $\widetilde{M} = (M_1, \ldots, M_q)$, where $M_i \in (\{0,1\}^n)^{m_i}$. Let $\sigma_i = \sum_{j \in [i]} m_j$, and $\sigma_q \leq \sigma$. Let $\mathcal{Q} := \{(i,a) \mid i \in [q], a \in (m_i]\}$, and $\leq$ be a natural linear ordering (known as the dictionary order) over $\mathcal{Q}$, defined as follows:

$$(i,a) \leq (i',a') \text{ if and only if } (i < i') \text{ or } (i = i' \text{ and } a \leq a').$$

In context of the linear ordering $(\mathcal{Q}, \leq) = (\alpha_1 \leq \cdots \leq \alpha_{\sigma_q})$, we can naturally define $\alpha_i + j$ as α_{i+j} for any $i \in [\sigma_q]$ and $j \in [\sigma_q - i]$. One can define subtraction analogously. Sometimes, we also use the subset $\mathcal{Q}^+ := \mathcal{Q} \setminus \{(i,0) : i \in [q]\}$.

For the input tuple $\widetilde{M}$ and a permutation $\pi \in \mathcal{P}(n)$, let y_i^π denote the output tuple corresponding to the input M_i, i.e., $y_i^\pi[0] = 0^n$, $y_i^\pi[a] = \pi(y_i^\pi[a-1] \oplus M_i[a])$, for all $(i,a) \in \mathcal{Q}$. Let $\mathrm{v}(i,a) := \min\{(j,b) \leq (i,a) : y_i^\pi[a] = y_j^\pi[b]\}$.

Structure Graphs: Given the input tuple $\widetilde{M}$ and permutation π, the *structure graph* $\mathcal{G}^\pi(\widetilde{M}) := (\mathcal{V}, \mathcal{E})$, is an edge-labeled directed graph, where the set of vertices $\mathcal{V} = \{\mathrm{v}(\alpha) : \alpha \in \mathcal{Q}\}$, the set of edges $\mathcal{E} = \{e_\alpha := (\mathrm{v}(\alpha - 1), \mathrm{v}(\alpha)) : \alpha \in \mathcal{Q}^+\}$, and edge e_α is labeled m_α for all $\alpha \in \mathcal{Q}^+$. Note that, it is possible that $e_\alpha = e_\beta$ for some $\alpha, \beta \in \mathcal{Q}^+$, i.e., they represent the same edge with obviously the same label. When we consider a single input M_r, the resulting subgraph is simply a walk, that we call an M_r-walk and denote as $\mathcal{W}_r$, starting at node $(1,0)$ and following the labels from $(M_r[1], \ldots, M_r[m_r])$. So, a structure graph can also be viewed as a union of M_i-walks for all $i \in [q]$.

Example 3.1. Let $m_1 = (1,0,2,0,7,1)$ and $m_2 = (4,1)$ be two inputs and $\pi(1) = 2$; $\pi(2) = 3$; $\pi(4) = 5$ for some $\pi \in \mathcal{P}$. As mentioned before, we can remove the common

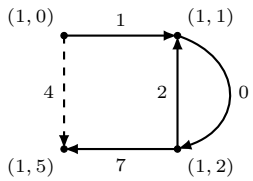

Figure 3.1: Structure graph corresponding to the inputs $M_1 = (1, 0, 2, 0, 7, 1)$ and $M_2 = (4, 1)$, and permutation π, with $\pi(1) = 2$, $\pi(2) = 3$ and $\pi(4) = 5$. The solid lines correspond to edges in $\mathcal{W}_1$, and dashed lines correspond to edges in $\mathcal{W}_2$.

suffix. So, we have $y_1 = (0, 2, 3, 2, 3, 5)$ and $y_2 = (0, 5)$. The corresponding structure graph $\mathcal{G}^\pi(M_1, M_2)$, illustrated in Figure 3.1, has vertex set $\mathcal{V} = \{(1, 0), (1, 1), (1, 2), (1, 5)\}$ and edges set

$$\mathcal{E} = \{((1, 0), (1, 1)), ((1, 1), (1, 2)), ((1, 2), (1, 1)), ((1, 2), (1, 5)), ((1, 0), (1, 5))\}.$$

Collisions and Accidents: Suppose that $\mathcal{G}^\pi(\widetilde{M})$ is revealed edge by edge in an orderly fashion following $(\mathcal{Q}^+, \leq)$. We say that an edge e_α leads to a *collision* if $\mathrm{v}(\alpha)$ is already present in the partially revealed graph. A collision formed by edges e_α and e_β is generally denoted as $(\mathrm{v}(\alpha - 1), \mathrm{v}(\beta - 1); \gamma)$, where $\gamma = \mathrm{v}(\alpha) = \mathrm{v}(\beta)$. The only exception occurs when $\gamma = (1, 0)$ and there is no prior edge to $(1, 0)$, in which case the collision is denoted as $(\mathrm{v}(\alpha - 1); \gamma)$, since prior to e_α there's no edge pointing to $(1, 0)$. This exceptional case is referred as a *zero collision*, and all other collisions are referred as *true collisions*. We refer to $\mathrm{v}(\alpha - 1)$ (and $\mathrm{v}(\beta - 1)$, if applicable) as *collision source*.

Note that it is not possible to recover the intermediate output tuple, by just looking at a given structure graph. Indeed, multiple intermediate output tuples may give the same structure graphs. However, a structure graph does preserve the collision relation between intermediate outputs. More precisely, let $\mathsf{Y}_{\mathrm{v}(\alpha)}$ denote the variable for the intermediate output corresponding to the vertex $\mathrm{v}(\alpha)$. Obviously, we must have $\mathsf{Y}_1[0] = 0^n$, otherwise the resulting intermediate output tuple is invalid. Now, any true collision $(\mathrm{v}(\alpha - 1), \mathrm{v}(\beta - 1); \gamma)$ implies a linear equation

$$\mathsf{Y}_{\mathrm{v}(\alpha-1)} \oplus \mathsf{Y}_{\mathrm{v}(\beta-1)} = m_\alpha \oplus m_\beta,$$

since both $\mathsf{Y}_{\mathrm{v}(\alpha-1)} \oplus m_\alpha$ and $\mathsf{Y}_{\mathrm{v}(\beta-1)} \oplus m_\beta$ must equal $\pi^{-1}(\mathsf{Y}_\gamma)$. Any new true collision can either give a linear equation that is linearly dependent on the linear equations due to previously discovered true collisions, or it may give an independent linear equation. True collisions of the latter type are referred as *accidents*. At a high level, accidents denote the "surprising" collisions in CBC function computation. Obviously, the number of true collisions is at least the number of accidents. The following definition due to Jha and Nandi [JN16] gives a formula for the number of accidents.

Definition 3.1 ([JN16])**.** Consider the structure graph $\mathcal{G}^\pi(\widetilde{M})$ associated with the input tuple $\widetilde{M}$ and permutation π. Let $\mathcal{S}(\mathcal{G}^\pi(\widetilde{M}))$ be the system of linear equations formed by the true collisions of $\mathcal{G}^\pi(\widetilde{M})$, and let r denote the rank of $\mathcal{S}(\mathcal{G}^\pi(\widetilde{M}))$. Let $\mathsf{Acc}(\mathcal{G}^\pi(\widetilde{M}))$ be the set of accidents of $\mathcal{G}^\pi(\widetilde{M})$. Then, the number of accidents, denoted $\mathsf{acc}(\mathcal{G}^\pi(\widetilde{M}))$ is defined as

$$\mathsf{acc}(\mathcal{G}^\pi(\widetilde{M})) := \left|\mathsf{Acc}(\mathcal{G}^\pi(\widetilde{M}))\right| = \begin{cases} r + 1 & \text{if } \mathcal{G}^\pi(\widetilde{M}) \text{ has a zero collision,} \\ r & \text{otherwise.} \end{cases}$$

Example 3.2. Consider the structure graph from Figure 3.1. Here, we have two true collisions, namely $((1, 0), (1, 2); (1, 1))$ and $((1, 0), (1, 2); (1, 5))$, and the associated system

of equations is

$$\mathsf{Y}_1[0] \oplus \mathsf{Y}_1[2] = M_1[1] \oplus M_1[3]$$
$$\mathsf{Y}_1[0] \oplus \mathsf{Y}_1[2] = M_1[5] \oplus M_2[1]$$

Clearly, the two equations are dependent. So the graph has just one accident, and that accident is $((1,0),(1,2);(1,1))$, since it occurs before $((1,0),(1,2);(1,5))$. We encourage the readers to see [BPR05, JN16] for further exposition on true collisions and accidents.

Existing Results on Structure Graphs: We now recall some known and useful combinatorial results on structure graphs. The proof of these results are already available in [BPR05, JN16].

Lemma 3.1 ([JN16]). *For any structure graph G, if there is a vertex α with in-degree d then $\mathsf{acc}(G) \geq d-1$. Moreover, if the graph has a zero collision then $\mathsf{acc}(G) \geq d$.*

Lemma 3.2 ([BPR05, JN16]). *The number of structures graphs associated to $\widetilde{M}$ with a accidents is at most $\binom{\sigma_q}{2}^a$. In particular, there exists exactly one structure graph with 0 accidents.*

Lemma 3.3 ([BPR05, JN16]). *For any structure graph G with a accidents, we have*

$$\Pr_{\Pi}\left(\mathcal{G}^{\Pi}(\widetilde{M}) = G\right) \leq \frac{1}{(2^n - \sigma_q)^a}.$$

Corollary 3.1 ([BPR05, JN16]). *For $a \in \mathbb{N}$ and $\sigma_q < 2^{n-1}$, we have*

$$\Pr_{\Pi}\left(\mathsf{acc}(\mathcal{G}^{\Pi}(\widetilde{M})) \geq a\right) \leq \frac{\sigma_q^{2a}}{2^{an}},$$

4 Characterization of Structure Graphs

We will characterize all structure graphs of rank 1 and 2 for a message pair (M_0, M_1). By counting non-isomorphic structure graphs up to rank 2, we will establish the collision probability bound of CBC MAC and Cascade. For ease of notation, we will revisit the concepts of true collisions and structure graphs before engaging in the characterization.

True Collisions and System of equations: A subset $\mathcal{C}$ is called strongly connected component of a directed graph $\mathcal{G}$ if for every $u, v \in \mathcal{V}_{\mathcal{C}}$ there is a directed path from u to v. The number of edges with $v \in \mathcal{V}_{\mathcal{G}}$ as the terminated vertex is called the *in-degree* of v in $\mathcal{G}$ and denoted as *in-deg*$_{\mathcal{G}}(v)$. The number of edges with $v \in \mathcal{V}_{\mathcal{G}}$ as the initial vertex is called the *out-degree* of v in $\mathcal{G}$ and denoted as *out-deg*$_{\mathcal{G}}(v)$. We will write *in-deg*(v) and *out-deg*(v) if the graph is understood from the context. A vertex $\gamma \in \mathcal{V}$ is called a *true collision vertex* if the in-degree of γ is at least 2. We call $\mathrm{v}(\alpha-1)$ and $\mathrm{v}(\beta-1)$ *pre-collision vertices*. A true collision $(\mathrm{v}(\alpha-1), \mathrm{v}(\beta-1); \gamma)$ yields a linear equation

$$\mathsf{Y}_{\mathrm{v}(\alpha-1)} \oplus \mathsf{Y}_{\mathrm{v}(\beta-1)} = m_\alpha \oplus m_\beta,$$

We yield a system of linear equations $\mathcal{E}_\gamma$ for every true collision with the true collision vertex being γ. Note that $\mathcal{E}_{\mathcal{G}} = \cup_v \mathcal{E}_v$, the union is taken over every true collision vertex v of $\mathcal{G}$. We denote a directed edge (u, v) by $u \to v$ and an m-labeled directed edge (u, v) by $(u \to v, m)$.

Structure Graphs: We have seen a detailed discussion on structure graphs in section 3, here we use a simpler definition for characterization purposes. A *structure graph* is an

edge labeled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$, where $\mathcal{V}$, $\mathcal{E}$, and $\mathcal{L}$ are set of vertices, set of edges, and set of labels respectively which are subsets of $\{0,1\}^n$ with the following property: If $(\mathrm{v}(\alpha-1) \to \mathrm{v}(\alpha), m_\alpha)$ and $(\mathrm{v}(\beta-1) \to \mathrm{v}(\beta), m_\beta)$ are labeled edges then

$$\mathsf{Y}_{\mathrm{v}(\alpha-1)} \oplus m_\alpha = \mathsf{Y}_{\mathrm{v}(\beta-1)} \oplus m_\beta \Leftrightarrow \mathsf{Y}_{\mathrm{v}(\alpha)} = \mathsf{Y}_{\mathrm{v}(\beta)}$$

Note that if $(\mathrm{v}(\alpha-1) \to \mathrm{v}(\alpha), m_\alpha)$ and $(\mathrm{v}(\alpha-1) \to \mathrm{v}(\beta), m_\beta)$ are two edges with labels m_α and m_β respectively, where $\mathrm{v}(\alpha) \neq \mathrm{v}(\beta)$ then we must have $m_\alpha \neq m_\beta$.

4.1 Core Component

Definition 4.1. Maximal strongly connected component of a structure graph $\mathcal{G}$ is called a *core component* of the structure graph.

We will characterize rank 1 and 2 structure graphs for a message pair $\widetilde{M} = (M_0, M_1)$. We will identify possible core components, as core components are building blocks of structure graphs. We often use messages and walks to refer to each other. For example, instead of W_0 and W_1 we say, M_0 and M_1 and vice versa.

We denote a directed edge (u, v) by $u \to v$ and an m-labeled directed edge (u, v) by $(u \to v, m)$. If $H = (V, E)$ be a graph then by $H \cup \{u \to v\}$ we mean that we are adding the edge in H i.e. $H_1 = (V_H, E_H \cup \{u \to v\})$ and by $H \backslash \{u \to v\}$ we mean that we are deleting the edge from H i.e. $H_2 = (V_H, E_H \backslash \{u \to v\})$. We denote a collision by $(u, v; w)$, where u and v are precollision vertices and w is the collision vertex, that means $u \to w$ and $v \to w$ are two edges. Let H be a subgraph of $\mathcal{G}$. Rank of H, denoted as $rank(H)$ is the rank of all equations induced by true collisions present in H. The following results will be used to characterize core components and structure graphs.

Lemma 4.1. *Let H and H' be two vertex disjoint graphs, where H' is strongly connected and $u \in V_H$ and $u' \in V'_H$. Then $rank(H \cup H' \cup \{u \to u'\}) = rank(H) + rank(H') + 1$.*

Proof. As H and H' are disjoint we clearly have $rank(H \cup H') = rank(H) + rank(H')$. Now we add one edge $e = (u, u')$ with $u \in H$ and $u' \in H'$. As H' is strongly connected, the in-degree of u' is at least one and let $v \to u'$ be an edge in H'. Thus, the edge $u \to u'$ leads to a new true collision with an equation of the form $\mathsf{Y}_{\mathrm{v}(u)} \oplus \mathsf{Y}_{\mathrm{v}(v)} = c$ for some constant c. Clearly, this newly added equation is independent of all equations presented in H and H'. This completes the proof. □

The following results are immediate corollary of lemma 4.1.

Corollary 4.1. *Let H be a connected graph, with $u \notin V_H$ and $v \in V_H$ then $rank(H \cup \{u \to v\}) = rank(H) + 1$.*

Corollary 4.2. *Only the core components of rank 0 and 1 are in rank 1 and rank 2 structure graphs. Moreover, if the set of all core components of rank 2 structure graph is $\{C_1, \ldots, C_s\}$ then we have the following possibilities:*

1. *$rank(C_1) = 1$ and $s = 1$.*
2. *$rank(C_1) = 0$ and $s = 1$.*
3. *$rank(C_1) = rank(C_2) = 0$ and $s = 2$.*

The following result says how a subgraph H can be extended (i.e., an edge is added) without increasing rank. Before going into the lemma, we describe an important concept called alternating cycle in a directed graph. Let $G = (V, E)$ be a graph, an alternating cycle of length at least 4 is a cycle with directions of edges of the cycle alternating. More formally, if C^{alt} is an alternating cycle with $V_{C^{alt}} = \{v_1, v_2, \ldots, v_{2k}\}$ then for all $i \in [k]$ we have $v_{2i-1} \to v_{2i} \in E_{C^{alt}}$ and for all $i \in [k-1]$ we obtain $v_{2i+1} \to v_{2i} \in E_{C^{alt}}$ and finally $v_1 \to v_{2k} \in E_{C^{alt}}$.

Example 4.1. Let us describe the simplest example with 4 vertices. Let the vertices be v_1, v_2, v_3, v_4. By the above description we obtain the following edges $(v_1, v_2), (v_3, v_4), (v_3, v_2), (v_1, v_4)$.

Lemma 4.2. *Let H be a graph, v is a vertex of H, in-deg$_H(v) \geq 1$ and $u \to v \notin E_H$ such that rank$(H \cup \{u \to v\}) =$ rank(H) then $u \in V_H$ and $u \to v \in E_{C^{alt}}$, where C^{alt} is an alternating cycle and $C^{alt} \backslash \{u \to v\} \subseteq H$.*

Proof. There are two conclusions of the lemma, assume the contrary, if $u \notin V_H$ then by Corollary 4.1 the hypothesis is not true i.e. $rank(H \cup u \to v) \neq rank(H)$. A contradiction. If $u \to v \notin E_{C^{alt}}$ then the hypothesis is false as the rank will get increased by 1. A contradiction. This completes the proof. □

The proof is straightforward as the only way not to increase the rank if the added edge forms an alternating cycle. Note that the given condition of the lemma ensures that there would be a newly added true collision due to adding the edge $u \to v$.

Thus, checking for rank 0 and rank 1 core components using the aforementioned lemma is sufficient. We proceed towards the characterization of rank 0 and rank 1 core components.

4.2 Characterization of Core Components

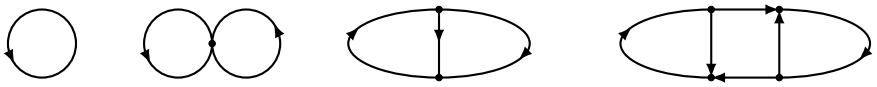

Figure 4.1: Core Components (left to right): C0, C1-8, C1-d8, C1-88.

4.2.1 Rank 0 Core Component

Lemma 4.3. *Core components of rank* 0 *are directed cycles (see Figure 4.1).*

Proof. Let $\mathcal{C} = (\mathcal{V}_\mathcal{C}, \mathcal{E}_\mathcal{C})$ be a rank 0 core component with n vertices. Since it is a strongly connected component of rank 0, in-degree of every vertex is 1. Also, out degree of every vertex is exactly 1. Since core component is a strongly connected component, *out-deg(v)* ≥ 1 for every $v \in \mathcal{V}_\mathcal{C}$. Thus we have *out-deg(v)* $= 1$ for every $v \in \mathcal{V}_\mathcal{C}$. Hence, rank 0 core component is a directed cycle. □

4.2.2 Rank 1 Core Component

The following lemma characterizes all possible rank 1 core components. There are three types of rank 1 core components, namely the 8 shaped C1-8 , the digital 8 shaped C1-d8 and the double 8 shaped C1-88.

Lemma 4.4. *Core components of rank* 1 *are C1-8, C1-d8 and C1-88 (see Figure 4.1).*

Proof. There can not be three vertices of in-degree 2 in a rank 1 core component because that will lead to rank at least 2. So, there are at most two vertices of in-degree 2. The following two cases will arise:

Case 1: Let v_1 and v_2 are 2 vertices of in-degree 2 in a rank 1 component $\mathcal{C}$. Now there are two collisions $(u_1, u_2; v_1)$ and $(u'_1, u'_2; v_2)$. Let us call $P_1 = \{u_1, u_2\}$ and $P_2 = \{u'_1, u'_2\}$. We will show that $P_1 = P_2$.

- If $P_1 \cap P_2 = \phi$ then there are 4 different variables to capture two collisions and thus we have two independent variables, hence rank will be 2, which is impossible.
- $|P_1 \cap P_2| = 1$ then there will be 2 independent variables, which yields rank 2, which is impossible. Hence $P_1 = P_2$.

Thus by Lemma 4.2 we argue that there is an alternating cycle $\mathcal{C}^{alt}$, of rank 1 in $\mathcal{C}$. Thus we obtain C1-88.

Case 2: Suppose there is exactly one vertex $v \in V_{\mathcal{C}}$ of in-degree 2 in $\mathcal{C}$, then there are two vertices u_1 and u_2 such that $(u_1, u_2; v)$ is a collision, and there are no collisions in $\mathcal{C}$. Now, there is a vertex $v' \in V_{\mathcal{C}}$ such that $\textit{out-deg}(v') = 2$, since sum of in-degrees and out-degrees are equal in a graph.

- If $v = v'$, then figure C1-8 is only possibility.
- If $v \neq v'$, then figure C1-d8 is only possibility. □

4.3 Characterization of Structure Graphs

Let us denote the corresponding walks for massages M_0 and M_1 as W_0 and W_1 respectively. We sometimes abuse the notation and denote walks as M_0 and M_1 and vice versa.
Assumptions regarding messages: The following are two assumptions we will follow throughout the discussion unless mentioned otherwise.

1. (Suffix disjoint) M_0 and M_1 are suffix free, which means W_0 and W_1 walks do not share common suffix as edge labels
2. (End vertex same) W_0 and W_1 share the same end vertex, which means both walks eventually collide at the same vertex.

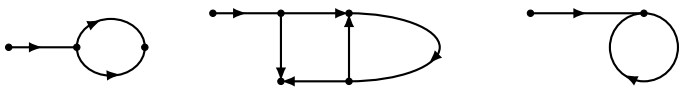

Figure 4.2: Accident 1 graphs (row major order): P-R1, C0-R1.1, C0-R1.2.

Paths and Structure graphs consisting of paths: Let $G = (V, E)$ be a graph, a walk of length s is defined as a sequence $W = (u_0, u_1, \ldots, u_s)$ such that $u_{i-1} \to u_i$ for all $i \in [s]$. When all vertices of a walk sequence are distinct, we call it a path. An M_i-walk starting from some initial vertex u_0 with no vertex having in-degree 2 must be a path. If a structure graph G is a union of M_i-walks, then it is called a structure graph comprising of paths. In the following lemmas 4.5 and 4.6, we will characterize rank 1 and 2 structure graphs, respectively, using only paths.

Lemma 4.5. *Structure graphs of rank* 1 *using only paths are of the form* P-R1 *(see Figure 4.2).*

Proof. By the End vertex same assumption there is exactly one collision $(u_1, u_2; v)$, where v is the last vertex of both walks W_0 and W_1. Any other collision in W_0 and W_1 will lead to a rank 2 graph by corollary 4.1, which is impossible. Thus, the only possibility is P-R1. □

Lemma 4.6. *Structure graphs of rank* 2 *using only paths are of the form* P-R2 *(see Figure 4.3).*

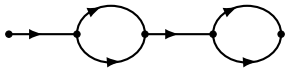

Figure 4.3: Accident 2 graphs: P-R2.

Proof. By the end vertex same assumption there is a collision $(u_1, u_2; v)$, where v is the last vertex of both walks W_0 and W_1. Since rank of the structure graph is 2, we obtain that there will be a collision $(u_1', u_2'; v')$ such that $u_1 \neq u_1'$ and $u_2 \neq u_2'$, and there is no other collision by corollary 4.1 hence we obtain P-R2. □

We will use core components to produce structure graphs of rank 1 and rank 2. We will show these are the only possibilities for structure graphs of rank 1 and 2.

4.3.1 Structure graphs of rank 1 using one rank 0 core component

Lemma 4.7. *Structure graphs of rank* 1 *using rank* 0 *component are of the forms C0-R1.1 and C0-R1.2 (see Figure 4.2).*[7]

Proof. Let $\mathcal{C} = (V, E)$ be a core component of rank 0. Without loss of generality, let us assume that M_0 walk W_0 enters $\mathcal{C}$, now there is a collision $(u_1, u_2; v)$. By Corollary 4.1 rank of the structure will be 1. Assume message walk W_0 of M_0 contains rank 0 core component $\mathcal{C}$.

Case 1: End vertex is in $\mathcal{C}$. By the Suffix Disjoint assumption of messages M_0 and M_1, we have that the end vertex is not an arbitrary vertex in $\mathcal{C}$. We can not afford to have another collision by W_1. Thus, end vertex of W_1 is v. Hence we yield C0-R1.2.

Case 2: End vertex is outside of $\mathcal{C}$. Now, M_1 walk W_1 ends at end vertex v by End Vertex Same assumption, and we need the rank of the graph to be 1. Let $u \in \mathcal{C}$ be a vertex such that $u \to v$ is an edge and $u' \in W_1$ be a vertex such that $u' \to v$ is an edge. Let $(u_{\mathcal{C}}, u_{W_0}; v_{\mathcal{C}})$ be the collision in $\mathcal{C}$. Consider the graph $\mathcal{C}' = (V', E')$ where $V' = V \cup \{u_{W_0}, v\}$ and $E' = E \cup \{u \to v, u_{W_0} \to v_{\mathcal{C}}\}$. Clearly $rank(\mathcal{C}') = 1$, also we have $rank(\mathcal{C}' \cup \{u' \to v\}) = rank(\mathcal{C}')$, thus by Lemma 4.2 $u' \in V'$ and $u' \to v \in E_{C^{alt}}$, where C^{alt} is an alternating cycle and $C^{alt} \backslash \{u' \to v\}$ is a subgraph of $\mathcal{C}'$. Hence we yield C0-R1.1.

□

4.3.2 Structure graphs of rank 2 using one rank 0 core component

Lemma 4.8. *Structure graphs of rank* 2 *using one rank* 0 *component are of the forms C0-R2.1, C0-R2.2, C0-R2.3, C0-R2.4, C0-R2.5, C0-R2.6 (see Figure 4.4).*

Proof. Let us denote the rank 0 core component as $\mathcal{C}$. We will analyze the following cases depending upon the end vertex of the walk W_0 and W_1. Let us call the end vertex v. Now, we consider different sub-cases.

Case 1: Suppose $v \in \mathcal{V}_{\mathcal{C}}$.

sub-case 1: Since W_0 enters $\mathcal{C}$ there is a collision $(u_{\mathcal{C}}, u_{W_0}; u)$. End vertex v is some arbitrary vertex other than u. By the suffix disjoint assumption, W_1 can not enter in $\mathcal{C}$ to reach the end vertex v. Thus W_1 and $\mathcal{C}$ are vertex disjoint, and

[7] In the graphs C0-R1.1 and C0-R1.2 it may happen that the directed loops will become self-loops depending on the message blocks, it is implicitly understood that such cases are there even it is not drawn separately.

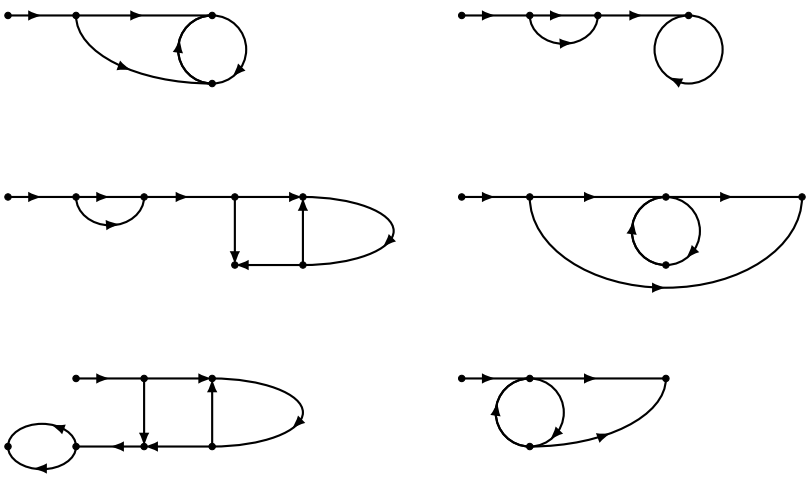

Figure 4.4: Accident 2 graphs (row major order): C0-R2.1, C0-R2.2, C0-R2.3, C0-R2.4, C0-R2.5, C0-R2.6.

W_1 reaches end vertex v in a way that and there is a collision $(u_{W_1}, u'_{\mathcal{C}}; v)$, this yields C0-R2.1.

sub-case 2: Since W_0 enters $\mathcal{C}$ there is a collision $(u_{\mathcal{C}}, u_{W_0}; u)$. Suppose end vertex is u. By the suffix disjoint assumption, W_1 can not enter $\mathcal{C}$ to reach the end vertex u. Thus there will be a collision $(v_{W_0}, v_{W_1}; v)$ where $v \notin \mathcal{V}_{\mathcal{C}}$, thus we obtain C0-R2.2.

Case 2: Suppose $v \notin \mathcal{V}_{\mathcal{C}}$.

sub-case 1: If W_0 leaves $\mathcal{C}$ from an arbitrary vertex and v be the last vertex of the walk. W_1 is either vertex disjoint with $\mathcal{C}$ or not. Thus, we have the following conclusions.

- If W_1 is vertex disjoint with $\mathcal{C}$, then it forces W_1 to collide at v, as there are two collisions $(u_{\mathcal{C}}, u_{W_0}, u)$ and $(v_{W_1}, v_{W_0}; v)$ where $u_{\mathcal{C}} \in \mathcal{C}$, $u_{W_0}, v_{W_0} \in W_0$, and $v_{W_1} \in W_1$ and by Corollary 4.1 rank will be 2. This yields C0-R2.4.
- If W_1 is not vertex disjoint with $\mathcal{C}$, then W_1 will leave the component at an arbitrary vertex other than pre-collision vertex and meets W_0 at end vertex v, thus rank will be 2 by Corollary 4.1. This yields figure C0-R2.6.

sub-case 2: If W_0 leaves $\mathcal{C}$ from a pre-collision vertex and W_1 meets W_0 such that $(u_{W_1}, u_{\mathcal{C}}; v)$ is a collision then by Lemma 4.2 we have $(u_{W_1}, u_{\mathcal{C}}; v)$ is a collision in an alternating cycle. Hence, we arrive at the following conclusions.

- If v is the end vertex, then there is a collision $(u_{W_0}, u_{W_1}; u)$ where $u_{W_0} \in W_0$ and $u_{W_1} \in W_1$, which yields C0-R2.3.
- If v is not end vertex then there is a collision $(u_{W_0}, u_{W_1}; u)$ where $u_{W_0} \in W_0$ and $u_{W_1} \in W_1$, which yields C0-R2.5.

□

4.3.3 Structure graphs of rank 2 using two rank 0 core components

Lemma 4.9. *Structure graphs of rank* 2 *using two rank* 0 *components are of the forms C00-R2.1, C00-R2.2 and C00-R2.3 (see Figure 4.5).*

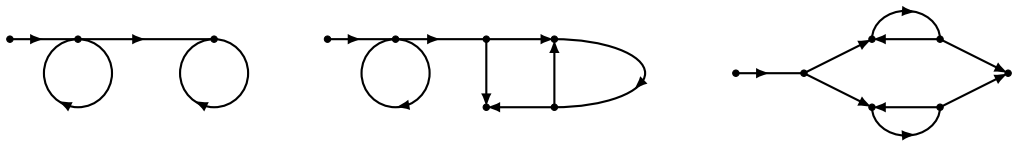

Figure 4.5: Accident 2 graphs (row major order): C00-R2.1, C00-R2.2, C00-R2.3.

Proof. Let us study the following two cases.

Case 1: Let us assume two rank 0 components are $\mathcal{C}_1$ and $\mathcal{C}_2$. Suppose there is a directed path $P = (u_{\mathcal{C}_1} \to u_1 \to \ldots \to u_n \to u_{\mathcal{C}_2})$ between $\mathcal{C}_1$ and $\mathcal{C}_2$. Now, the resulting graph is of rank 1 by Corollary 4.1. Without loss of generality, assume M_0 enters component $\mathcal{C}_1$, again by Corollary 4.1 we have the rank of the structure graph to be 2. Now, we consider different sub-cases.

sub-case 1: If end vertex v of walk W_0 and W_1 is in $\mathcal{C}_2$, then by Suffix disjoint assumption it will be only vertex with *in-degree* 2 in $\mathcal{C}_2$. This yields C00-R2.1.

sub-case 2: If end vertex v of both walk is outside $\mathcal{C}_2$, then there is a collision $(u_{W_0}, u_{W_1}; v)$, since rank remains 2, we have $u_{W_0} \in \mathcal{C}_2$ and $(u_{W_0}, u_{W_1}; v_{\mathcal{C}_2})$ is a collision by Lemma 4.2 and that u_{W_0}, u_{W_1}, v and $v_{\mathcal{C}}$ forms an alternating cycle $\mathcal{C}^{alt}$. This yields C00-R2.2.

Case 2: Suppose two rank 0 components $\mathcal{C}_1$ and $\mathcal{C}_2$ are not connected. Hence, there is no directed path between the two components. Then both message walks W_0 and W_1 can not enter a single component. Without loss of generality, assume W_0 enters $\mathcal{C}_1$ and W_1 enters $\mathcal{C}_2$ starting from the same initial vertex. Since $\mathcal{C}_1$ and $\mathcal{C}_2$ are vertex disjoint, W_0 and W_1 must collide at a vertex v such that $v \notin \mathcal{V}_{\mathcal{C}_1}$ and $v \notin \mathcal{V}_{\mathcal{C}_2}$ and there will be a collision $(u_1, u_2; v)$ where $u_1 \in W_0$ and $u_2 \in W_1$. Since there are three true collisions $(u_{W_0}, u_{\mathcal{C}_1}; v_{\mathcal{C}_1})$, $(u_{W_1}, u_{\mathcal{C}_2}; v_{\mathcal{C}_2})$ and $(u_1, u_2; v)$ and rank of the structure graph is 2, by Lemma 4.2 we have that u_{W_0}, $u_{\mathcal{C}_1}$, u_{W_1}, $u_{\mathcal{C}_2}$, u_1 and u_2 form an alternating cycle $\mathcal{C}_6$ which yields C00-R2.3.

□

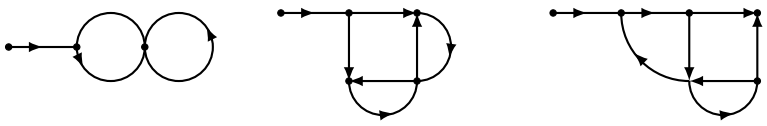

Figure 4.6: Accident 2 graphs (row major order): C1-8-R2.1, C1-8-R2.2, C1-8-R2.3.

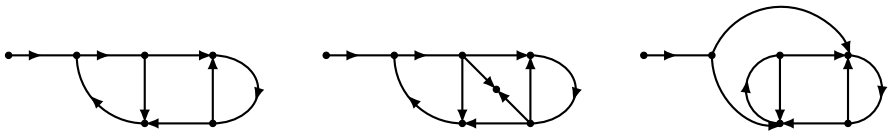

Figure 4.7: Accident 2 graphs (row major order): C1-88-R2.1, C1-88-R2.2, C1-88-R2.3.

4.3.4 Structure graphs of rank 2 using one rank 1 core component

Lemma 4.10. *Structure graphs of rank* 2 *using rank* 1 *components are of the forms C1-8-R2.1, C1-8-R2.2, C1-8-R2.3, C1-88-R2.1, C1-88-R2.2, C1-88-R2.3, C1-d8-R2.1, C1-d8-R2.2,* , *C1-d8-R2.3. (see Figure 4.6-4.8)*

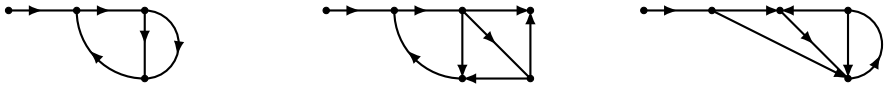

Figure 4.8: Accident 2 graphs (row major order): C1-d8-R2.1, C1-d8-R2.2, C1-d8-R2.3.

Proof. We will give a general proof for all three possible types of rank 1 core components, namely the 8 shaped C1-8 , the digital 8 shaped C1-d8, the double 8 shaped C1-88.

Let $\mathcal{C}$ be an arbitrary rank 1 component of any shape. Without loss of generality, let W_0 be the corresponding walk for M_0 enters $\mathcal{C}$, then by Corollary 4.1 rank of the resulting graph is 2.

Case 1: W_1 enters at the same vertex of $\mathcal{C}$ where W_0 will enter, following the same path. Let us consider the following sub-cases.

sub-case 1: Suppose end vertex v lies in $\mathcal{C}$, i.e. $v \in \mathcal{V}_{\mathcal{C}}$. By suffix disjoint assumption, we obtain $in\text{-}deg(v) = 2$. There is a vertex with in-degree 2 in $\mathcal{C}$ (one in C1-8, C1-d8 and two in C1-88). That vertex, being the end vertex, yields three types of graphs C1-8-R2.1, C1-d8-R2.1 and C1-88-R2.1 respectively.

sub-case 2: Suppose end vertex v does not lie in $\mathcal{C}$, i.e $v \notin \mathcal{V}_{\mathcal{C}}$. Then clearly, there is a collision $(u_1, u_2; v)$, where $u_1 \in W_0$ and $u_2 \in W_1$ by Lemma 4.2 we conclude that $u_1, u_2 \in \mathcal{V}_{\mathcal{C}}$, $u_1 \to v$ and $u_2 \to v$ are edges on an alternating cycle $\mathcal{C}^{alt}$ such that $\mathcal{C}^{alt} - \{u_1 \to v, u_2 \to v\}$ is a subgraph of $\mathcal{G}$. Hence we obtain three types of graphs C1-8-R2.3, C1-d8-R2.2 and C1-88-R2.2.

Case 2: W_1 enters at different vertex of $\mathcal{C}$ where W_0 will enter. Then clearly there are two collisions $(u_{W_0}, u_{\mathcal{C}}; v_{\mathcal{C}})$ and $(u_{W_1}, u'_{\mathcal{C}}; v'_{\mathcal{C}})$ and there is an existing collision in the component itself. By lemma 4.2, it is evident that the edges $u_{W_0} \to v_{\mathcal{C}}$, $u_{W_1} \to v'_{\mathcal{C}}$, $v \to u_{\mathcal{C}}$ and $v \to u'_{\mathcal{C}}$ will form an alternating cycle of length 4, where v is the end vertex of both message walks as well as the existing collision vertex in $\mathcal{C}$. Hence we obtain three types of graphs C1-8-R2.2, C1-d8-R2.3 and C1-88-R2.3.

□

5 CBC Collision Probability for Two Messages

We define type-1, type-2, type-2′ and type-3 structure graphs. In type-1 we have two categories of graphs: type-1-neq and type-1-any. The naming conventions are as follows: any denotes message length can be equal or unequal, and neq means the message lengths are unequal. In our notations type-1-any is P-R1 and type-1-neq are two graphs C0-R1.1 and C0-R1.2 (see figure 4.2). For type-2 we have two structure graphs of rank 2, which are C0-R2.4 and P-R2 (see figure 4.4), and in type-2′ we have several rank 2 structure graphs, namely C1-8-R2.2 and C00-R2.1 (see figure 4.5-4.8). Every structure graph other than type-1, type-2, type-2′ is called type-3 structure graph.

Let $N = 2^n$, and two messages be $M_0 = (m_1, m_2, \ldots, m_{|p|}, m_{|p|+1} \ldots, m_{\ell_0})$ and $M_1 = (m_1, m_2, \ldots, m_{|p|}, m_{\ell_0+1}, \ldots, m_{\ell_0+\ell_1-|p|})$. Here $p = (m_1, m_2, \ldots, m_{|p|})$ is the common prefix of both messages M_1 and M_2 of length $|p|$, where $0 \leq |p| \leq \ell_1 \leq \ell_0 := \ell$. Also, we assume $m_{\ell_0} \neq m_{\ell_0+\ell_1-|p|}$. In the following subsections, we consider the above two messages. We define

$$D = \{(i, j) \mid \quad |p| \leq i < \ell_0, |p| \leq j \leq \ell_0 + \ell_1 - |p|, m_i \neq m_j\}.$$

5.1 Bound on number of non-isomorphic structure graphs of type-1-neq

In type-1-neq we have two structure graphs C0-R1.1 and C0-R1.2. We observe the following:

- **Case 1 for C0-R1.1:** Let the messages be $M_0 = p \parallel m_{|p|+1} \parallel (M'_0 \parallel y)^t \parallel M'_0 \parallel m_{\ell_0}$ and $M_1 = p \parallel m_{\ell_1}$, , where y is a block of M_0, which is a label on the side of the alternating cycle other than $m_{|p|+1}$, m_{ℓ_1} and m_{ℓ_0}. Let us denote $|M'_0 \parallel y| = k$. Note that, if p, $m_{|p|+1}$, M'_0 and m_{ℓ_0} are fixed then $(M'_0 \parallel y)^t$ is determined. Clearly we have $\ell_0 = |p| + 3 + |M'_0| + t|M'_0 \parallel y|$. Thus, the upper bound for non-isomorphic structure graph of this type is $d^*(\ell)$.
- **Case 2 for C0-R1.2:** Let messages be $M_0 = (m_0, \ldots, m_{\ell_1}, \ldots m_{\ell_0})$ and $M_1 = (m_0, \ldots, m_{\ell_1})$. Let, p be common prefix, so $|p| = \ell_1$. So, $M_0 = p \parallel M'_0$ and $M_1 = p$, then M_0 enters the core component of length $(\ell_0 - \ell_1)$ and traverses several times and then stops at the end vertex. So, $M'_0 = X^t$ for some $t \in \mathbb{N}$. Clearly if M'_0 is fixed, we have $t|X| = \ell_0 - \ell_1$. Thus, by determining $|X|$, we can give an upper bound for the number of non-isomorphic structure graphs of this type, and clearly, there can be at most $d^*(\ell)$ many such choices.

We summarize the above discussion as follows:

Lemma 5.1. *The number of non-isomorphic structure graphs of type C0-R1.1 and C0-R1.2 are at most $d^*(\ell)$.*

5.2 Bound on number of non-isomorphic structure graphs of type-2

We have two type-2 graphs, namely P-R2 and C0-R2.4. Note that, up to the common prefix, the message walks M_0 and M_1 are determined. Also, the last block of M_0 and M_1 must be different due to the Suffix Disjoint assumption, and there is a collision at the end vertex, so there is a fixed collision at the last vertex. Also, there is a collision in the type-2 graph other than the end vertex collision. Note that any type-2 graph is determined by the other collision than the end vertex collision, and that collision is determined by the pre-collision vertices. For P-R2, messages will be of the form $M_0 = p \parallel X \parallel q \parallel Y$ and $M_0 = p \parallel X' \parallel q \parallel Y'$ where p is a common prefix and, q is the common part after the first collision. As pre-collision vertices of the first collision determine any such graph, the number of non-isomorphic structure graphs of type-2 is at most $|D|$. Hence, we obtain the following lemma.

Lemma 5.2. *The number of non-isomorphic structure graphs of type-2 is at most $|D|$ (as defined above).*

5.3 Bound on number of non-isomorphic type-2 structure graphs

Consider C1-8-R2.2 (see figure 4.6) and C00-R2.1 (see figure 4.5) be two rank 2 structure graphs, we will bound the number of maximum possible such structure graphs realizable by two messages M_0 and M_1.

Case 1 for C1-8-R2.2 : Let us denote M_0 and M_1 as $M_0 = p \parallel m_{p+1} \parallel M'_0$ and $M_1 = p \parallel m_{l_0+1} \parallel M'_1$ respectively where M'_0 and M'_1 are remaining message blocks which traverse core component C1-8 and collide at the end vertex. If we denote $b \parallel X' = X$ and $c \parallel Y' = Y$ where X' where b and c are message blocks on alternating cycle. Clearly we have $b \parallel M'_0 = (X^{i_1} \parallel Y^{j_1} \parallel X^{i_2} \parallel Y^{j_2} \parallel \cdots)$ and $c \parallel M'_1 = (Y^{k_1} \parallel X^{l_1} \parallel Y^{k_2} \parallel X^{l_2} \parallel \cdots)$. If M'_0 is fixed then we have $k = |M'_0| = c_1|X| + c_2|Y|$ where $c_1 = \sum_t i_t$ and $c_2 = \sum_t j_t$, also we have $c_1 \geq 2$. Note that the isomorphism of structure graphs is completely determined by $|X|$ and

$|Y|$. Thus, using Lemma 2.1, we have at most $\ell d^*(\ell)^2$ many non-isomorphic graphs of this type.

Case 2 for C00-R2.1 : M_0 can enter either of the components, the analysis is same for both. If M_0 enters both components in C-00. Let us denote the common prefix as p. Let us denote the structure graph by $p \parallel x \parallel z \parallel y$, where p is the common prefix of both messages, x is the first cycle, z is the common part of both messages till from the leaving vertex of first cycle till end vertex, and finally y be the possible labels on second cycle. We assume M_1 traverses the first cycle c_1 times, if number of blocks in M_1 is l then $\ell - (|p| + |c_1 x|) = |z|$. Fixing $c_1 x$ and choosing x can be done in $\ell d^*(\ell)$ ways, likewise for message M_0 choosing y can be done in $d^*(\ell)$ ways, thus maximum number of such graphs is $\ell(d^*(\ell))^2$.

Similar argument applies to all other rank 2 structure graphs in type-2′. Thus, we have the following result by counting the number of all possibilities in type-2′.

Lemma 5.3. *The number of non-isomorphic structure graphs of type-2′ is at most* $18ld^*(\ell)^2$.

5.4 Collision Probability for Unequal Length Messages

We recall lemma 3.3 that for any structure graph G with v vertices for two messages M_0, M_1, with a accidents, we have

$$\Pr_{\Pi}\left(\mathcal{G}^{\Pi}(\widetilde{M}) = G\right) \leq \frac{1}{(2^n - v)^a}.$$

We use this result for all structure graphs except the structure graph of type-1-any (only one such structure graph, denoted as G^* is present, which represents the collision event for the message pair (M_0, M_1)). Let $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_2', \mathcal{G}_3$ be the set of all such structure graphs with collision events and of types type-1-neq, type-2, type-2' and type-3 respectively. We have seen that $|\mathcal{G}_1| \leq d^*(\ell)$, $|\mathcal{G}_2| \leq |D|$, $|\mathcal{G}_2'| \leq 18\ell d^*(\ell)^2$ and $|\mathcal{G}_3| \leq 8\ell^6$. We write $\mathsf{CP}_{\mathcal{G}} := \Pr_{\Pi}\left(\mathcal{G}^{\Pi}(\widetilde{M}) \in \mathcal{G}\right)$ By using the upper bound for realizing a structure graph and the bound on the number of such graphs, we have the following:

$$\mathsf{CP}_{\mathcal{G}_1} \leq \frac{d^*(\ell)}{(2^n - 2\ell)}$$
$$\mathsf{CP}_{\mathcal{G}_2'} \leq \frac{18\ell d^*(\ell)^2}{(2^n - 2\ell)^2}$$
$$\mathsf{CP}_{\mathcal{G}_2} \leq \frac{|D|}{(2^n - 2\ell)^2}$$
$$\mathsf{CP}_{\mathcal{G}_3} \leq \frac{8\ell^6}{(2^n - 2\ell)^3}$$

Note that $|D|$ can be in the order ℓ^2, and hence we provide a sharper bound for CP_{G^*} to cancel the collision probability due to the class $\mathcal{G}_2$. Moreover, the term $\mathsf{CP}_{\mathcal{G}_1}$ only appears for the collision probability of two unequal messages. Now we revisit the event that $\mathcal{G}^{\Pi}(M_0, M_1) = G^*$. Note that graph G^* have set of vertices and edges be $V = V_1 \cup V_2$ and $E = E_1 \cup E_2 \cup \{((0, |p|), (1, \ell_0 + 1)), ((1, \ell_0 + \ell_1 - |p| + 1), (0, \ell_0))\}$, where

$$V_1 = \{(0, t) \mid t \in (\ell_0]\}$$
$$V_2 = \{(1, \ell_0 + t) \mid t \in [\ell_1 - |p| + 1]\}$$

$$E_1 = \{((0,t),(0,t+1)) \mid t \in (\ell_0 - 1]\}$$
$$E_2 = \{((1,\ell_0+t),(1,\ell_0+t+1)) \mid t \in [\ell_1 - |p|]\}$$

Let $v = \ell_0 + \ell_1 - |p| - 1$ denote the number of vertices and let S denote the number of pairwise distinct solutions of $(\mathsf{Y}_v)_{v\in V}$ such that

1. $\mathsf{Y}_{(0,0)} = 0^n$,
2. $\mathsf{Y}_{(0,\ell_0-1)} \oplus \mathsf{Y}_{(1,\ell_0+\ell_1-|p|-1)} = m_{(0,\ell_0)} \oplus m_{(1,\ell_0+\ell_1-|p|)}$,
3. $(\mathsf{X}_v)_{v\in V}$ are pairwise distinct where $\mathsf{X}_{(0,i)} = \mathsf{Y}_{(0,i-1)} \oplus m_{(0,i)}$, for $i \in [\ell_0]$ and $\mathsf{X}_{(1,i)} = \mathsf{Y}_{(1,i-1)} \oplus m_{(1,i)}$ where $i \in [\ell_0+2, \ell_0+\ell_1-|p|-1]$ and $\mathsf{X}_{(1,\ell_0+1)} = \mathsf{Y}_{(0,|p|)} \oplus m_{(1,\ell_0+1)}$.

Lemma 5.4. *$S \le \frac{(N)_v}{N-1} - \frac{|D|(N)_v(N-4)}{N^3} + \frac{|D|^2(N)_v}{(N-6)^3}$ where $N = 2^n$.*

Proof. Let S_0 be the set of variables such that conditions 1 and 2 hold. Let $S_{(i,j)}$ be the set of variables such that conditions 1 and 2 hold and for some i, j we have $\mathsf{X}_{(0,i)} = \mathsf{X}_{(1,j)}$, which can be written as

$$\mathsf{Y}_{(0,i-1)} \oplus \mathsf{Y}_{(1,j-1)} = m_{(0,i)} \oplus m_{(1,j)} \tag{4}$$

for $p \le i \le \ell_0 < j \le \ell_0 + \ell_1 - p$ and $m_i \ne m_j$. It is evident that

$$S = |S_0| - |\cup_{(i,j)} S_{(i,j)}|.$$

By the counting version of Bonferonni's inequality we have,

$$S \le |S_0| - \sum_{(i,j)\in D} |S_{(i,j)}| + \sum_{(i,j)\ne(r,s)\in D} |S_{(i,j)} \cap S_{(r,s)}|.$$

where $S_{(i,j)} \cap S_{(r,s)}$ be the set of variables such that 1, 2 and for some i, j, r, s we have $\mathsf{X}_{(0,i)} = \mathsf{X}_{(1,j)}$, and $\mathsf{X}_{(0,r)} = \mathsf{X}_{(1,s)}$ which can be written as

$$\mathsf{Y}_{(0,i-1)} \oplus \mathsf{Y}_{(1,j-1)} = m_{(0,i)} \oplus m_{(1,j)} \tag{5}$$
$$\mathsf{Y}_{(0,r-1)} \oplus \mathsf{Y}_{(1,s-1)} = m_{(0,r)} \oplus m_{(1,s)} \tag{6}$$

We have the following count for each of the sets:

1. For S_0 we have one constraint equation in condition 2, we can choose one of the variables in N ways, then the other variable is fixed. To hold the pairwise distinct assumption, we choose $(v-2)$ variables from the remaining $(N-2)$ possibilities. Thus we have

$$|S_0| = N(N-2)_{v-2} = \frac{(N)_v}{(N-1)}$$

2. For $S_{(i,j)}$, we have two equations to look for, namely one in condition 2 and equation 4. We can choose one variable in N ways in 4, and the other variable is fixed. Now, for the equation in condition 2, we can choose one variable in $(N-4)$ ways to hold the pairwise disjoint assumption, then the other variable is fixed. Since we have made 4 choices already, we can choose the remaining variables in $(N-4)_{(v-4)}$ ways. Thus we have

$$|S_{(i,j)}| = N(N-4)(N-4)_{(v-4)} = \frac{(N)_v(N-4)}{N(N-1)(N-2)} \ge \frac{(N)_v(N-4)}{N^3}$$

3. For $S_{(i,j)} \cap S_{(r,s)}$ we have three equations to look for, one in condition 2 and equations 5 and 6. Arguing like above, it is easy to see that the variables of the equations can be chosen in $N(N-4)(N-6)$ ways. We are left with $(N-6)$ possibilities and we have to choose $(v-6)$ variables, hence we obtain

$$\begin{aligned}|S_{(i,j)} \cap S_{(r,s)}| &= N(N-4)(N-6)(N-6)_{(v-6)} \\ &= \frac{(N)_v(N-6)}{(N-1)(N-2)(N-3)(N-5)} \\ &\leq \frac{(N)_v}{(N-6)^3}\end{aligned}$$

Hence, we obtain the desired upper bound. □

Thus,

$$\mathsf{CP}_{G^*} \leq \frac{1}{N-1} - \frac{|D|(N-4)}{N^3} + \frac{|D|^2}{(N-6)^3}$$

We will use the following result to estimate the upper bound of $\mathsf{CP}^{eq}_{2,\ell}$, for any $1 \leq m \leq \frac{N}{2}$ we have

$$\frac{1}{(N-m)} < \frac{1}{N} + \frac{2m}{N^2}.$$

Thus,

$$\begin{aligned}\mathsf{CP}^{\mathsf{eq}}_{2,\ell} &= \mathsf{CP}_{G^*} + \mathsf{CP}_{\mathcal{G}_2'} + \mathsf{CP}_{\mathcal{G}_2} + \mathsf{CP}_{\mathcal{G}_3} \\ &\leq \frac{1}{N-1} - \frac{|D|(N-4)}{N^3} + \frac{|D|^2}{(N-6)^3} + \frac{18\ell d^*(\ell)^2}{(N-2\ell)^2} + \frac{|D|}{(N-2\ell)^2} + \frac{8\ell^6}{N^3} \\ &\leq \frac{1}{N} + \frac{18\ell d^*(\ell)^2}{N^2} + \frac{10\ell^6}{N^3}\end{aligned}$$

The above expression follows through a straightforward algebra and assuming $\ell < 2^{n/3}$ (which we can assume, since otherwise the upper bound is more than one) Also, for $\mathsf{CP}^{\mathsf{any}}_{2,\ell}$ we have to consider $\mathsf{CP}_{\mathcal{G}_1}$ and add this term with $\mathsf{CP}^{\mathsf{eq}}_{2,\ell}$ and we obtain the bound stated in Theorem 5.2. Hence, we have proved our main results.

Theorem 5.1 (equal length CBC-MAC collision probability)**.**

$$\mathsf{CP}^{\mathsf{eq}}_{2,\ell} \leq \frac{1}{2^n} + \frac{18\ell d^*(\ell)^2}{2^{2n}} + \frac{10\ell^6}{2^{3n}}.$$

Theorem 5.2 (any length CBC-MAC collision probability)**.**

$$\mathsf{CP}^{\mathsf{any}}_{2,\ell} \leq \frac{d^*(\ell)}{2^n} + \frac{18\ell d^*(\ell)^2}{2^{2n}} + \frac{10\ell^6}{2^{3n}}.$$

6 Collision Probability for Cascade

6.1 Cascade Definition

Cascade Function. The Cascade function $\textsc{Casc}_f : (\{0,1\}^n)^* \to \{0,1\}^n$, associated with a compression function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$, is defined recursively over a sequence of n-bit blocks $M = (M[1] \cdots M[l])$ in the following manner:

$$\textsc{Casc}_f(M) := \begin{cases} 0^n & M = \perp, \\ f(\textsc{Casc}_f(M[1] \cdots M[l-1]) \| M[l]) & \text{otherwise}, \end{cases}$$

Going forward, we will drop f and M from the notation whenever they are understood from the context.

CASCADE Collision Probability Problem. Let M and M' be two distinct inputs having m and m' many blocks, respectively, and $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a compression function. We call $\texttt{Coll}_f(M, M')$ the *collision event* for a pair of inputs M and M'. Although we mention it for $2n$-bit to n-bit compression function, the same argument works for $(n+d)$-bit to n-bit compression function for $d > 0$. By extending the notation, we similarly define the collision event for a tuple of $q \geq 2$ distinct inputs $M^q = (M_1, \ldots, M_q)$, as

$$\texttt{Coll}_f(M^q) = \bigcup_{i<j \in [q]} \texttt{Coll}_f(M_i, M_j). \tag{7}$$

We define *collision probability* as $\textsc{Casc}\text{-}\mathsf{CP}(M^q) = \Pr\left(\texttt{Coll}_\Gamma(M^q)\right)$, where the probability is computed over the randomness of uniformly chosen random function Γ. Let

$$\textsc{Casc}\text{-}\mathsf{CP}^{\mathsf{atk}}_{q,\ell,\sigma} = \max_{M^q} \textsc{Casc}\text{-}\mathsf{CP}(M^q)$$

where the maximum is taken over all q-tuples of distinct inputs M^q having at most ℓ blocks each, and the total length over all q inputs is at most σ. Further, the message tuple satisfies the input constraint atk, which could be one of the following:

1. $\mathsf{eq\text{-}suff}$: each input has exactly ℓ blocks and there is a common suffix of the inputs;
2. $\mathsf{eq\text{-}nosuff}$: each input has exactly ℓ blocks, and there is no common suffix of the inputs.

6.2 Structure Graph for Cascade Function and Its Characterization

The structure graph for Cascade is similar to the structure graphs of CBC MAC, except for a few operational differences. Unlike CBC MAC, where we use a permutation π as a basic building block for the construction, in Cascade construction, we use a compression function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$.

Structure Graphs for Cascade: We have seen a detailed discussion on structure graphs in section 5, and here we use a simpler definition for characterization purposes. A *structure graph* is an edge labeled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$, where $\mathcal{V}$, $\mathcal{E}$, and $\mathcal{L}$ are set of vertices, set of edges and set of labels respectively which are subsets of $\{0,1\}^n$ and , with the following property: If $(\mathrm{v}(\alpha-1) \to \mathrm{v}(\alpha), m_\alpha)$ and $(\mathrm{v}(\beta-1) \to \mathrm{v}(\beta), m_\beta)$ are labeled edges then

$$\mathsf{Y}_{\mathrm{v}(\alpha-1)} \parallel m_\alpha = \mathsf{Y}_{\mathrm{v}(\beta-1)} \parallel m_\beta \Rightarrow \mathsf{Y}_{\mathrm{v}(\alpha)} = \mathsf{Y}_{\mathrm{v}(\beta)}$$

where $f(\mathsf{Y}_{\mathrm{v}(\alpha-1)} \parallel m_\alpha) = \mathsf{Y}_{\mathrm{v}(\alpha)}$ and $f(\mathsf{Y}_{\mathrm{v}(\beta-1)} \parallel m_\beta) = \mathsf{Y}_{\mathrm{v}(\beta)}$ Note that, unlike CBC construction, in Cascade equality of variables at the collision vertices will not imply

that concatenations are equal, as f applied to different concatenations can lead to same functional value.

Rank and Observations regarding Cascade structure graphs: We define rank of structure graph $\mathcal{G}$ to be *true collision* of $\mathcal{G}$ and thus we have $rank(\mathcal{G}) = \mathbf{TC}(\mathcal{G})$. Thus, there is no provision for alternating cycles in this graph, as it will increase the number of true collisions, and as a result the rank will increase. It is not possible that $(v \to u_1; m)$ and $(v \to u_2; m)$ are two labeled edges with the same message label m, then $u_1 = u_2$. There can be parallel edges $(v \to u; m_1)$ and $(v \to u; m_2)$ and this will be regarded as a collision. The characterization problem is the same as stated in section 4: message walks start at 0^n and traverse the structure graph via edges, which are message labels and stop at the same end vertex.

Assumptions regarding messages: The following are two assumptions we will follow throughout the discussion unless mentioned otherwise.

1. (Suffix disjoint) M_0 and M_1 are suffix free, which means W_0 and W_1 walks do not share common suffix as edge labels

2. (End vertex same) W_0 and W_1 share the same end vertex, which means both walks eventually collide at the same vertex

We exclude graphs with alternating cycles obtained in section 4, as those graphs will contribute to true collisions, and thus, the rank will be more than the desired. We discard all such graphs with alternating cycles and obtain a characterization of structure graphs for the Cascade function.

Different types of structure graphs for Cascade: We describe different types of structure graphs we will analyse for bounding collision probability. We will first identify possible core components for rank 1 and rank 2 structure graphs. Since taking the double 8 shaped core component C1-88 will increase true collision as there is an alternating cycle, we will consider the following core components: the directed cycle C0 , the eight(8) shaped C1-8 and the digital eight(8) shaped C1-d8 (see figure 4.1). The following graphs are possible: P-R1, P-R2, C0-R1.2, C0-R2.1, C0-R2.2, C0-R2.4, C00-R2.1, C1-d8-R2.1 and C1-8-R2.1 (see figure 4.2- 4.8) as we discard all the graphs with alternating cycles and with double 8 shaped core component C1-88. The following types are possible:

1. type-1-any: P-R1.

2. type-2: P-R2 and C0-R2.4.

3. type-2$''$: C0-R2.1, C0-R2.2, C0-R2.4, C00-R2.1, C1-d8-R2.1 and C1-8-R2.1.

4. type-3$'$: structure graphs of rank 3 and more.

We consider type-2$'' \subseteq$ type-2$'$ since all the other graphs in type-2$'$ are of higher rank. Here we have, type-3$' =$ type-2$' \setminus$ type-2$'' \cup$ type-3 We are not considering type-1-neq as we will look for equal length messages.

Note that for a structure graph G of Cascade with v vertices and e edges, we have the following relation: $\mathbf{TC}(G) = e - v + 1$. Every edge corresponds to an equation, and the probability of realizing G will be at most $\frac{(N)_{v-1}}{N^e}$. We use this observation to state the following lemma for structure graphs of Cascade.

Lemma 6.1. *For any structure graph G with a accidents, v vertices and e edges where Γ is a random function, we have*

$$\Pr_{\Gamma}\left(\mathcal{G}^{\Gamma}(\widetilde{M}) = G\right) \leq \frac{(N)_{v-1}}{N^e}.$$

since $\boldsymbol{TC}(G) = a = e - v + 1$, in particular we have

$$\Pr_{\Gamma}\left(\mathcal{G}^{\Gamma}(\widetilde{M}) = G\right) \leq \frac{1}{N^a}.$$

6.3 Collision Probability Analysis

We analyse Cascade collision probability for two equal-length messages M_0 and M_1 with the following assumptions:

1. eq-suff: each input has exactly ℓ blocks and there is a common suffix of the inputs;
2. eq-nosuff: each input has exactly ℓ blocks, and there is no common suffix of the inputs.

The analysis is essentially the same as we have already seen in section 5.4. The only difference is due to the different types of structure graphs and suffix assumptions. The number of different types of non-isomorphic structure graphs is given in the following lemma.

Lemma 6.2. *The number of non-isomorphic structure graphs of type-2, type-2$''$ and type-3$'$ are $4\ell^2$, $6\ell d^*(\ell)^2$ and $8\ell^6$ respectively.*

We recall the result of lemma 6.1

$$\Pr_{\Gamma}\left(\mathcal{G}^{\Gamma}(\widetilde{M}) = G\right) \leq \frac{1}{N^a}.$$

We use this result for all structure graphs except the structure graph of type-1-any (only one such structure graph, denoted as G^*, is present, which represents the collision event for the message pair (M_0, M_1)). Let $\mathcal{G}_2, \mathcal{G}_2'', \mathcal{G}_3'$ be the set of all such structure graphs with collision events and types type-2, type-2$''$ and type-3$'$ respectively. We have seen that $|\mathcal{G}_2| \leq 4\ell^2$, $|\mathcal{G}_2''| \leq 6\ell d^*(\ell)^2$ and $|\mathcal{G}_3'| \leq 8\ell^6$.

Let us denote $\textsc{Casc}\text{-}\mathsf{CP}_{\mathcal{G}} := \Pr_{\Gamma}\left(\mathcal{G}^{\Gamma}(\widetilde{M}) = G\right)$. By using the upper bound for realizing a structure graph and the bound on the number of such graphs, we have the following:

$$\textsc{Casc}\text{-}\mathsf{CP}_{\mathcal{G}_2} \leq \frac{4\ell^2}{N^2}$$
$$\textsc{Casc}\text{-}\mathsf{CP}_{\mathcal{G}_2''} \leq \frac{6\ell d^*(\ell)^2}{N^2}$$
$$\textsc{Casc}\text{-}\mathsf{CP}_{\mathcal{G}_3'} \leq \frac{8\ell^6}{N^3}$$

Using the first bound in lemma 6.1 we we will give a better bound for G^*, this follows through a straightforward manipulation and Bonferroni's inequality

$$\textsc{Casc}\text{-}\mathsf{CP}_{G^*} \leq \frac{(N)_{v-1}}{N^v} \leq \frac{1}{N}\prod_{i=1}^{v-2}(1 - \frac{i}{N}) \leq \frac{1}{N} - \frac{\binom{v-1}{2}}{N^2} + \frac{4\ell^4}{N^3}$$

From the above inequalities, arguing and estimating like section 5.4, we state the following theorem

Theorem 6.1 (Cascade collision probability without common suffix)**.**

$$\textsc{Casc}\text{-}\mathsf{CP}_{2,\ell}^{\mathsf{eq-nosuff}} \leq \frac{1}{2^n} + \frac{6\ell d^*(\ell)^2}{2^{2n}} + \frac{10\ell^6}{2^{3n}}$$

Note that, in the above Theorem 6.1, no common suffix essentially implies that there is a collision at the last vertex of M_0 and M_1 walk. For common suffixes, there will be a collision at some point, and after that, the messages share the same blocks. Note that the collision probability is the same in both scenarios. However, for the common suffix case, the collision can occur at any vertex, and after that messages share the same blocks. Also note that collisions occurring at different vertices are disjoint events, and there are at most ℓ such choices for suffix length. Hence, we state the following theorem.

Theorem 6.2 (Cascade collision probability with common suffix)**.**

$$\textsc{Casc}\text{-}\mathsf{CP}_{2,\ell}^{\mathsf{eq-suff}} \leq \frac{\ell}{2^n} + \frac{6\ell^2 d^*(\ell)^2}{2^{2n}} + \frac{10\ell^7}{2^{3n}}$$

7 Applications to Randomness Extraction

In this section, we revisit the results corresponding to randomness extractors as given in [DGH+04]. These results explicitly depend on good bounds for the CBC (and cascade) CPP. Based on our revised and thoroughly derived bounds for CBC and cascade CPP, we restate the updated results. But first, we briefly revisit the notion of randomness extractors, and restate some notations and definitions from [DGH+04].

7.1 A Short Note on Randomness Extractors

More Notations: For a probability distribution $\mathcal{X}$ over $\{0,1\}^\ell$, we define its *min-entropy* as the minimum integer m such that for all $x \in \{0,1\}^\ell$, $\Pr_{\mathcal{X}}(x) \leq 2^{-m}$. We denote the min-entropy of such $\mathcal{X}$ by $\mathbf{H}_\infty(\mathcal{X})$, and refer to $\mathcal{X}$ as an (ℓ, m)-distribution. The *collision probability* of $\mathcal{X}$ is $\mathtt{Col}(\mathcal{X}) = \Pr_{\mathsf{X},\mathsf{X}' \xleftarrow{\$} \mathcal{X}}(\mathsf{X} = \mathsf{X}') = \sum_x \Pr(\mathsf{X} = x)^2$. Let $\mathcal{X}_1$, $\mathcal{X}_2$ be two probability distributions over the set Ω. The statistical distance between the distributions $\mathcal{X}_1$ and $\mathcal{X}_2$ is defined as $\mathbf{SD}(\mathcal{X}_1, \mathcal{X}_2) = \frac{1}{2}\sum_{\omega\in\Omega} |\Pr_{\mathcal{X}_1}(\omega) - \Pr_{\mathcal{X}_2}(\omega)|$.

Definition 7.1 (Almost Universal Hash)**.** Let n and ℓ be integers, and $\{h_k\}_{k\in\mathcal{K}}$ be a family of hash functions with domain $\{0,1\}^\ell$, range $\{0,1\}^n$ and key space $\mathcal{K}$. We say that the family $\{h_k\}_{k\in\mathcal{K}}$ is δ-*almost universal* (δ-AU) if for every pair of different inputs x, y from $\{0,1\}^\ell$ it holds that $\Pr_{\mathsf{K} \xleftarrow{\$} \mathcal{K}}(h_{\mathsf{K}}(x) = h_{\mathsf{K}}(y)) \leq \delta$. For a given probability distribution $\mathcal{X}$ on $\{0,1\}^\ell$, we say that $\{h_k\}_{k\in\mathcal{K}}$ is δ-AU w.r.t. $\mathcal{X}$ if $\Pr(h_{\mathsf{K}}(x) = h_{\mathsf{K}}(y)) \leq \delta$, where $x, y \sim \mathcal{X}$ conditioned to $x \neq y$.

Definition 7.2 ((Strong) Extractor)**.** Let n and ℓ be integers, and $h = \{h_k\}_{k\in\mathcal{K}}$ be a family of functions with domain $\{0,1\}^\ell$, range $\{0,1\}^n$ and key space $\mathcal{K}$. We say that h is an (n,δ)-extractor if and only if for any K uniform over $\mathcal{K}$, and any (ℓ, n)-distribution, we have

$$\mathbf{SD}(h_{\mathsf{K}}(\mathcal{X}), \mathsf{U}) \leq \delta,$$

where U is uniform over $\{0,1\}^n$. Additionally, we say that h is an (n,δ)-strong extractor if and only if

$$\mathbf{SD}((\mathsf{K}, h_{\mathsf{K}}(\mathcal{X})), (\mathsf{K}, \mathsf{U})) \leq \delta.$$

The following lemma is a straightforward extension of the well-known "Leftover Hash Lemma" or LHL [HILL99].

Lemma 7.1 (Lemma 2 in [DGH+04]). *Let ℓ and n be integers, let $\mathcal{X}$ be a probability distribution over $\{0,1\}^\ell$, and let $\{h_k\}_{k\in\mathcal{K}}$ be a family of hash function with domain $\{0,1\}^\ell$ and range $\{0,1\}^n$. If $\{h_k\}_{k\in\mathcal{K}}$ is $(2^{-n}+\epsilon)$-almost universal w.r.t. $\mathcal{X}$, U is uniform over $\{0,1\}^n$ and K is uniform over $\mathcal{K}$, then*

$$\mathbf{SD}((\mathsf{K}, h_{\mathsf{K}}(\mathcal{X})), (\mathsf{K}, \mathsf{U})) \leq \frac{1}{2}\sqrt{2^n \cdot (\mathtt{Col}(\mathcal{X}) + \epsilon)} \leq \frac{1}{2} \cdot \sqrt{2^n \cdot (2^{-\mathbf{H}_\infty(\mathcal{X})} + \epsilon)}$$

7.2 CBC-MAC as Randomness Extractor

Dodis et al. showed that CBC-MAC based on a uniform random permutation is a good candidate for strong randomness extractor. Specifically, they showed the following result.

Theorem 7.1 (Theorem 1 in [DGH+04]). *Consider the CBC-MAC construction based on a uniform random permutation Π of $\{0,1\}^n$, and let $\mathcal{X}$ be an input distribution defined over ℓ-block strings. Then the statistical distance between CBC-MAC$_\Pi(\mathcal{X})$ and the uniform distribution on $\{0,1\}^n$ is at most*

$$\sqrt{2^{n-\mathbf{H}_\infty(\mathcal{X})} + O(2^n \cdot \epsilon(\ell, 2^n))},$$

where $\epsilon(\ell, 2^n) \leq \ell(d^*(\ell))^2 2^{-2n} + \ell^6 2^{-3n}$. In particular, assuming $\ell < 2^{n/4}$ and $\mathbf{H}_\infty(\mathcal{X}) \geq 2n$, the above statistical distance is at most $O(\ell/2^{n/2})$.

The proof of this theorem follows from Lemma 7.1 and the following *unproved* result in [DGH+04].

Lemma 7.2 (Lemma 3 in [DGH+04]). *Consider the CBC-MAC construction based on a uniform random permutation Π of $\{0,1\}^n$. For any $x, y \in \{0,1\}^{\ell n}$, if $x \neq y$ then*

$$\Pr_\Pi(\textit{CBC-MAC}_\Pi(x) = \textit{CBC-MAC}_\Pi(y)) \leq \frac{1}{2^n} + O(\epsilon(\ell, 2^n)),$$

where $\epsilon(\ell, 2^n)$ follows the same definition as given in Theorem 7.1.

As an application of our results on the CBC CPP, we provide the first proof for Lemma 7.2.

Proof of Lemma 7.2. This lemma is just a restatement of Theorem 5.1, and hence the proof follows directly from the proof of Theorem 5.1.

7.3 Cascade as Randomness Extractor

In a similar fashion, Dodis et al. also gave the following result for the cascade construction.

Theorem 7.2 (Theorem 2 in [DGH+04]). *Let $F = \{F_k\}$ be the cascade construction defined, as in section 6.1, over a family of random functions $\{f_k\}$. Let $\mathcal{X}$ be the input distribution to F defined over ℓ-block strings, and $\mathcal{X}_\ell$ denote the probability distribution induced by $\mathcal{X}$ on the last block $\mathsf{X}[\ell]$ for $\mathsf{X} \sim \mathcal{X}$. Then, if U is the uniform distribution over $\{0,1\}^n$, we have*

$$\mathbf{SD}(F(\mathcal{X}), \mathsf{U}) \leq \sqrt{2^{n-\mathbf{H}_\infty(\mathcal{X})} + \ell \cdot 2^{-\mathbf{H}_\infty(\mathcal{X}_\ell)} + O(2^n \cdot \epsilon(\ell, 2^n))},$$

where $\epsilon(\ell, 2^n)$ follows the same definition as given in Theorem 7.1. In particular, if $\mathbf{H}_\infty(\mathcal{X}) \geq 2n$, $\mathbf{H}_\infty(\mathcal{X}_\ell) \geq n$, and $\ell \leq 2^{n/4}$, then $\mathbf{SD}(F(\mathcal{X}), \mathsf{U}) \leq O(\ell/2^{k/2})$.

The proof of this theorem relies on Theorem 7.1 and the following two *unproved* propositions.

Proposition 7.1. *Let $F = \{F_k\}$ be the cascade construction defined over a family of random functions $\{f_k\}$. Let x, y be two inputs to F that differ (at least) in the last block, namely, $x[\ell] \neq y[\ell]$, and let k be any value of the initial key. Then $\Pr_f\left(F_k(x) = F_k(y)\right) \leq \frac{1}{2^n} + O(\epsilon(\ell, 2^n))$, where $\epsilon(\ell, 2^n)$ follows the same definition as given in Theorem 7.1.*

Proposition 7.2. *Let F be defined as above, let x, y be two different inputs to F, and let k be any value of the initial key. Then $\Pr_f\left(F_k(x) = F_k(y)\right) \leq \frac{\ell}{2^n} + O(\ell\epsilon(\ell, 2^n))$, where $\epsilon(\ell, 2^n)$ follows the same definition as given in Theorem 7.1.*

As an application of our bounds on the cascade construction CPP, we provide the first proof for Propositions 7.1 and 7.2.

Proof of Propositions 7.1 and 7.2. It is easy to see that Proposition 7.1 is just a restatement of Theorem 6.1, and similarly, Proposition 7.2 is a combined restatement of Theorem 6.1 and 6.2. Hence, their proofs directly follow from the proofs of Theorem 6.1 and 6.2.

8 Conclusion

In this paper, we provided some missing proofs from [DGH+04]. In order to derive these proofs, we employed the graph based technique from [BPR05], called the structure graphs. As a side-effect of this work, we made some significant progress in characterizing certain useful structure graphs which could be of independent interest. Specifically, this improved characterization might also be useful in obtaining better security bounds for some MAC constructions like EMAC, ECBC, FCBC, etc.

Acknowledgements: The authors would like to thank Wonseok Choi and all the anonymous reviewers who reviewed and provided valuable comments on this paper. Ashwin Jha's work was carried out under the framework of the French-German-Center for Cybersecurity, a collaboration of CISPA and LORIA.

References

[2711] ISO/IEC JTC 1/SC 27. Information technology – security techniques – message authentication codes (macs) – part 1: Mechanisms using a block cipher. ISO/IEC 9797-1, International Organization for Standardization, 2011.

[BBT16] Mihir Bellare, Daniel J. Bernstein, and Stefano Tessaro. Hash-function based prfs: AMAC and its multi-user security. In *Advances in Cryptology - EUROCRYPT 2016, Proceedings, Part I*, pages 566–595, 2016.

[BCK96] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Pseudorandom functions revisited: The cascade construction and its concrete security. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Proceedings*, pages 514–523. IEEE Computer Society, 1996.

[BdBB+95] 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. *Lecture Notes in Computer Science, Springer-Verlag, 1995*, 1007, 1995.

[Bel06] Mihir Bellare. New proofs for NMAC and HMAC: security without collision-resistance. In *Advances in Cryptology - CRYPTO 2006, Proceedings*, pages 602–619, 2006.

[Ber05] Daniel J. Bernstein. A short proof of the unpredictability of cipher block chaining. Online, 2005.

[BKR94] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of cipher block chaining. In *Advances in Cryptology - CRYPTO '94, Proceedings*, pages 341–358, 1994.

[BPR05] Mihir Bellare, Krzysztof Pietrzak, and Phillip Rogaway. Improved security analyses for CBC macs. In *Advances in Cryptology - CRYPTO 2005, Proceedings*, pages 527–545, 2005.

[BR00] John Black and Phillip Rogaway. CBC macs for arbitrary-length messages: The three-key constructions. In *Advances in Cryptology - CRYPTO 2000, Proceedings*, pages 197–215, 2000.

[DGH+04] 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 '04, Proceedings*, pages 494–510, 2004.

[EMST76] William F. Ehrsam, Carl H. W. Meyer, John L. Smith, and Walter L. Tuchman. Message verification and transmission error detection by block chaining. Patent 4074066, USPTO, 1976.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *Symposium on Foundations of Computer Science - FOCS 1984, Proceedings*, pages 464–479, 1984.

[GPR14] Peter Gazi, Krzysztof Pietrzak, and Michal Rybár. The exact prf-security of NMAC and HMAC. In *Advances in Cryptology - CRYPTO '14, Proceedings, Part I*, pages 113–130, 2014.

[GPT15] Peter Gazi, Krzysztof Pietrzak, and Stefano Tessaro. The exact PRF security of truncation: Tight bounds for keyed sponges and truncated CBC. In *Advances in Cryptology - CRYPTO 2015, Proceedings, Part I*, pages 368–387, 2015.

[GR16] Sergey Gorbunov and Charles Rackoff. On the security of cipher block chaining message authentication code. Online, 2016.

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

[IK03] Tetsu Iwata and Kaoru Kurosawa. OMAC: One-Key CBC MAC. In *Fast Software Encryption - FSE '03, Revised Papers*, pages 129–153, 2003.

[JN16] Ashwin Jha and Mridul Nandi. Revisiting structure graphs: Applications to CBC-MAC and EMAC. *J. Math. Cryptol.*, 10(3-4):157–180, 2016.

[KI03] Kaoru Kurosawa and Tetsu Iwata. TMAC: two-key CBC MAC. In *Topics in Cryptology - CT-RSA 2003, Proceedings*, pages 33–49, 2003.

[Mau02] Ueli M. Maurer. Indistinguishability of random systems. In *Advances in Cryptology - EUROCRYPT '02, Proceedings*, pages 110–132, 2002.

[Nan09] Mridul Nandi. Fast and secure cbc-type mac algorithms. In *Fast Software Encryption, Lecture Notes in Computer Science vol. 5665*, pages 375–393, 2009.

[Nan21] Mridul Nandi. A new and improved reduction proof of cascade PRF. *IACR Cryptol. ePrint Arch.*, page 97, 2021.

[Pie06] Krzysztof Pietrzak. A tight bound for EMAC. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 168–179. Springer, 2006.

[PR00] Erez Petrank and Charles Rackoff. CBC MAC for real-time data sources. *J. Cryptol.*, 13(3):315–338, 2000.

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

[Wig07] Carl Severin Wigert. Sur l'ordre de grandeur du nombre des diviseurs d'un entier. *Ark. Mat. Astron. Fys.*, 3(18), 1907.

[Yas10] Kan Yasuda. The sum of CBC macs is a secure PRF. In *Topics in Cryptology - CT-RSA 2010, Proceedings*, pages 366–381, 2010.