Revisiting Sum of CBC-MACs and Extending NI2-MAC to Achieve Beyond-Birthday Security

Avijit Dutta and Goutam Paul

Cryptology and Security Research Unit (CSRU),
R. C. Bose Centre for Cryptology & Security,
Indian Statistical Institute, Kolkata 700 108, India.
avirocks.dutta13@gmail.com, goutam.paul@isical.ac.in

Abstract. In CT-RSA 2010, Kan Yasuda has shown that the sum of two independent Encrypted CBC (ECBC) MACs is a secure PRF with security beyond birthday bound. It was mentioned in the abstract of the paper that "no proof of security above the birthday bound ($2^{n/2}$) has been known for the sum of CBC MACs" (where n is the tag size in bits). Kan Yasuda's paper did not consider the sum of actual CBC outputs and hence the PRF-security of the same has been left open. In this paper, we solve this problem by proving the beyond birthday security of sum of two CBC MACs. As a tool to prove this result, we develope a generalization of the result of S. Lucks from EUROCRYPT 2000 that the sum of two secure PRPs is a secure PRF. We extend this to the case when the domain and the range of the permutations may have some restrictions. Finally, we also lift the birthday bound of NI2 MAC construction (the bound was proven in CRYPTO 2014 by Gazi et al.) to beyond birthday by a small change in the existing construction.

Keywords: Beyond Birthday, CBC, ECBC, MAC, NI, NI2, Sum of PRP

1 Introduction

In symmetric key paradigm, MAC (Message Authentication Code) is used for preserving message integrity and message origin authentication. The design of a MAC should not only consider achieving security, but also target attaining efficiency. In the literature, three different approaches of designing a MAC exists: (a) universal hash function based MAC, a popular example of which is UMAC [8], (b) a compression function based MAC, like NMAC [2], HMAC [2], NI [1] etc. (c) Block cipher based MAC, such as CBC MAC [5], PMAC [9], OMAC [11]. etc.

Most of the popular MACs are block cipher based MACs, but each one of them suffers from the same problem - security is guaranteed up to the *birthday bound*. When the block length of the underlying block cipher is 128-bit, then birthday bound does not seem to be a problem, as we are guaranteed to have 64 bits of security which is well acceptable for many practical applications. But when we deal with 64-bit block

cipher as used in many light weight crypto devices, then birthday bound problem becomes the main bottleneck.

In recent researches, many MAC constructions have been proposed with security beyond the birthday barrier without degrading the performance. The first attempt was made in ISO 9797-1 [3] without security proof. But Algorithm 4 of ISO 9797-1 was attacked by Joux et al. [13] that falsified the security bound. Algorithm 6 of ISI 9797-1 was proven to be secure against $O(2^{2n/3})$ queries with restrictions on the message length [19].

1.1 Motivations and Contributions

Our present work has three distinct contributions. We discuss the motivation behind and the summary of each contribution one by one.

First contribution. We generalize the result of S. Lucks [15] in EUROCRYPT 2000 related to the sum of permutation. In [15], it was shown that the sum of two independent random permutations is a secure PRF with security beyond birthday bound, however there was no restrictions on the domain or range of the permutations. When the inputs to the permutations come from a nested MAC construction (like CBC), that uses the same permutation(s) as its primitives, then one needs to consider restricted domain and range for the security analysis of the sum of permutation. We show that even with such restrictions, the sum of two permutations is a secure PRF (Section 4). This result is used in our subsequent analysis.

Second contribution. CBC-MAC [5] allows only prefix-free queries. To circumvent this restriction, the ECBC construction [17] was proposed, though it achieves only birthday security just like CBC. In CT-RSA 2010, K. Yasuda [19] came up with an extension of ECBC to lift the security beyond birthday. We note that though the title of the paper [19] says that it is the sum of two CBC constructions, actually the proposal of [19] involved the sum of two ECBC constructions, requiring 4 keys. In the conclusion, the author mentioned that their approach "do not seem to be directly applicable to the sum of one-key CBC MACs" and "the problem of reducing the number of keys remains open". In this paper, we show that the sum of two CBC MACs is a secure MAC with PRF-security beyond birthday bound, requiring only two keys instead of 4-keys (used in Sum of Two ECBC construction [19]). Thus we solve (Section 5) an open problem since the work of [19].

Third contribution. In CRYPTO 1999, J. An and M. Bellare [1] proposed a Merkle-Damgård iteration based MAC construction called NI-MAC. The construction of NI-MAC is similar to that of NMAC [2], the only difference is that in NI-MAC the compression function f takes an additional input key k at each invocation. The motivation of designing NI was to avoid constant rekeying on multiblock messages in NMAC and to allow for a security proof starting by the standard switch from a PRF to a random function, followed by information-theoretic analysis.

In CRYPTO 2014, Gazi et al. [10] revisited the proof of NI-MAC in the view of structure graph introduced by Bellare et al. in CRYPTO 2005 [6] and gave a tight bound of order $\frac{lq^2}{2^n}$, which is an improvement over trivial bound of order $\frac{l^2q^2}{2^n}$, for q queries, each of length at most ℓ blocks. But this is again restricted to the birthday security. In order to prove the security of NI-MAC, Gazi et al. [10] introduced a variant of NI-MAC, called NI2-MAC, and then derived the security of NI-MAC from the security analysis of NI2-MAC. As our third contribution in this paper, we propose an extension of NI2-MAC with a single invocation of an additional pseudo-random function and prove (Section 6) that it achieves beyond-birthday security using similar proof-technique of NI-MAC.

2 Preliminaries

In this section, we briefly discuss the notations and definitions used in this paper. We also state some existing basic results.

2.1 PRP, PRF and Secure MAC

We denote $|S|$ as the cardinality of set S and S^c as the complement set of S. Let $x \xleftarrow{\$} S$ denote that x is chosen uniformly at random from S. Let $Perm(n)$ denote the set of all permutations over $\{0,1\}^n$, $Func(A,B)$ denote the set of all functions from A to B and $Bij(A,B)$ denote the set of all bijective functions from A to B (of course, here we need $|A| = |B|$). A permutation π is said to be a random permutation over $\{0,1\}^n$, if $\pi \xleftarrow{\$} Perm(n)$. A function $\rho : A \to B$ is said to be a random function, if ρ is chosen uniformly at random from the $Func(A,B)$.

We will specify a random permutation π which is defined over $\{0,1\}^n$ by performing *lazy sampling*. In lazy sampling initially the permutation π is undefined at every point of its domain. We maintain two sets that grows dynamically. One is domain, $\mathrm{Dom}(\pi)$ and another is Range, $\mathrm{Ran}(\pi)$ both of them are initialized to be empty. $\mathrm{Dom}(\pi)$, $\mathrm{Ran}(\pi)$ keeps the record of already defined domain points and range points of permutation π respectively. Therefore, if $x \notin \mathrm{Dom}(\pi)$ then we will choose

$y \xleftarrow{\$} \{0,1\}^n \setminus \mathrm{Ran}(\pi)$ and add y in $\mathrm{Ran}(\pi)$ and x in $\mathrm{Dom}(\pi)$. In this regard, x is said to be *fresh*. Similarly, we can lazy-sample a random function ρ maintaining consistency.

We consider that an adversary $\mathcal{A}$ is an oracle machine with access to its oracle $\mathcal{O}(\cdot)$ and outputs either 1 or 0. Accordingly, we write $\mathcal{A}^{\mathcal{O}(\cdot)} = 1$ or 0. The resource of $\mathcal{A}$ is measured in terms of the time complexity $T(n)$ that it takes to interacts with its oracle $\mathcal{O}(\cdot)$ and the query complexity $q(n)$ which says the number of queries and replies exchanged between the adversary and its oracle. For practical purpose, we restrict to probabilistic polynomial time (PPT) adversaries only.

Let E_k be a keyed permutation over $\{0,1\}^n$, i.e., a bijective function from $\{0,1\}^n$ to $\{0,1\}^n$. We define the PRP-advantage of E_k with respect to an adversary $\mathcal{A}$ as

$$\mathbf{Adv}^{\mathbf{PRP}}_{\mathbf{E_k}}(\mathcal{A}) = \Pr\left[\mathcal{A}^{E_k(\cdot)} = 1 : k \xleftarrow{\$} \mathcal{K}\right] - \Pr\left[\mathcal{A}^{\pi(\cdot)} = 1 : \pi \xleftarrow{\$} Perm(n)\right].$$

If this advantage is negligible in n for all PPT adversaries, we say that E_k is a secure PRP. Note that the first probability in the definition of advantage is calculated over the internal coin tosses of $\mathcal{A}$ and the randomness of $k \xleftarrow{\$} \mathcal{K}$ and the second probability is calculated over the randomness of $\pi \xleftarrow{\$} Perm(n)$.

Similarly, the PRF-advantage of a function $F_k : A \to B$ is defined as

$$\mathbf{Adv}^{\mathbf{PRF}}_{\mathbf{F_k}}(\mathcal{A}) = \Pr\left[\mathcal{A}^{F_k(\cdot)} = 1 : k \xleftarrow{\$} \mathcal{K}\right] - \Pr\left[\mathcal{A}^{f(\cdot)} = 1 : f \xleftarrow{\$} Func(A,B)\right].$$

If this advantage is negligible in the length of the input for all PPT adversaries, F is said to be a secure PRF. Note that the first probability is calculated over the internal coin tosses of the algorithm $\mathcal{A}$ and randomness of $k \xleftarrow{\$} \mathcal{K}$ and second probability is calculated over the randomness of $f \xleftarrow{\$} Func(A,B)$.

The length of M in bits is denoted by $len(M)$. When it is not a multiple of n, we append $10^{n-1-len(M) \mod n}$ to M to make $len(M)$ a multiple of n. We denote the maximum number of block in a query by l. We denote the partition of a message M as $M = M_1||M_2||\ldots||M_l$ where each M_i is an n-bit block and the number of blocks of M is denoted by l.

A MAC construction uses smaller primitives such as PRP or PRF that works on n-bit message blocks to build a larger primitive that works on variable-length message to produce a fixed-length tag. An adversary attacking a MAC with q queries obtains q tags for q distinct messages and produces a valid tag of a fresh message that he has not queried earlier. It is known that any secure PRF is a secure MAC. Thus, to show that a MAC construction is secure, one needs to show that the PRF-advantage (which is a function of q, l and n) of an adversary for the construction is negligible.

2.2 Structure Graphs

In this section, we briefly revisit the structure graph analysis of CBC-MAC [6] by Bellare et al. and that of NI-MAC [10] by Gazi et al.

Consider an iterated/cascaded construction with a function f, where f could be a random permutation or a random function, that works on a message $M = M_1||M_2||\ldots||M_l$ of length l blocks as follows:

$$Y_0 = \mathbf{0}, \text{ and } Y_i = f(Y_{i-1}, M_i) \text{ for } i = 1, \ldots, l.$$

Note that for CBC-MAC analysis, $f(\alpha, \beta)$ is taken as $\pi(\alpha \oplus \beta)$ and for the NI-MAC analysis, $f(\alpha, \beta)$ is taken as $\rho(\alpha||\beta)$, where π is a random permutation from $Perm(n)$ and ρ is a random function from $b+n$ bits to n bits, where b is the message block-length and n is the length of the chaining variable as well as the tag.

For a set of any two fixed distinct messages $\mathcal{M} = \{M^{(1)}, M^{(2)}\}$ and a function f, we construct the structure graph $\mathcal{G}^f(\mathcal{M})$ with $\{0,1\}^n$ as the set of nodes as follows. We follow the computations for $M^{(1)}$ followed by those of $M^{(2)}$ by creating nodes labelled by the values y_i of the intermediate chaining variables Y_i with the edge (Y_i, Y_{i+1}) labelled by the block M_{i+1}. In this process, if we arrive at a vertex already labelled, while not following an existing edge, we call this event an f-collision. An accident is an f-collision that does not close a cycle with alternating edge-directions such that the XOR of the labels of the cycle becomes 0.

More formally, let for two distinct messages $M^{(1)}$ and $M^{(2)}$ of l_1 and l_2 blocks respectively, where

$$M^{(1)} = M_1^{(1)}||M_2^{(1)}||\ldots||M_{l_1}^{(1)} \text{ and } M^{(2)} = M_1^{(2)}||M_2^{(2)}||\ldots||M_{l_2}^{(2)},$$

the corresponding Y-values be given by

$$Y_0^{(1)}, Y_1^{(1)}, Y_2^{(1)}, \ldots, Y_{l_1}^{(1)} \text{ and } Y_0^{(2)}, Y_1^{(2)}, Y_2^{(2)}, \ldots, Y_{l_2}^{(2)}$$

respectively. Let $\sigma = l_1 + l_2$. We use the notation M_i to refer to the block $M_i^{(1)}$, when $i < l_1$, otherwise to refer to the block $M_{i-l_1}^{(2)}$. Similarly, let Y_i to refer to $\mathbf{0}$ when $i = 0$; $Y_i^{(1)}$, when $1 \leq i \leq l_1$; and $Y_{i-l_1}^{(2)}$, when $l_1 + 1 \leq i \leq \sigma$. Now, consider the mappings

$$[[\cdot]] \text{ and } [[\cdot]]' \text{ on } \{0, \ldots, \sigma\}$$

so that $[[i]] = \min\{j : Y_i = Y_j\}$ and $[[i']] = [[i]]$ for $i \neq l_1$ except that $[[l_1]]' = 0$.

For any fixed f and any two distinct messages $\mathcal{M} = \{M^{(1)}, M^{(2)}\}$, we define the structure graph $\mathcal{G}^f(\mathcal{M})$ to be the triple $\mathcal{G}^f(\mathcal{M}) = (V, E, L)$, where

$$V = \{[[i]] : 0 \leq i \leq \sigma\}, \qquad E = \{([[i-1]]', [[i]]) : 1 \leq i \leq \sigma\}$$

and $L = E \to \{0,1\}^n$ is an edge-labeling function defined as

$$L((u,v)) = \{M_i : [[i-1]]' = u \text{ and } [[i]] = v\}.$$

Let (V_i, E_i, L_i) be the graph obtained after processing only the first i out of σ blocks of $\mathcal{M}$. We say that $(i, [[i]])$ is an f-collision if $[[i]] < i$ and $M_i \notin L_{i-1}([[i-1]]', [[i]])$. Note that the last condition on M_i implies that collision occurred due to parallel edges with the same message label is not considered.

In [6], a general collision is called a *true collision* (except the collision that occurs due to parallel edges with same label on the edges). Further, a true collision is called an *accident* if it is not followed from a cycle C with alternating edges with the sum of the labels of the edges involved in C to $\mathbf{0}$, otherwise it is called an *induced collision*. However, for NI2-MAC, all f-collisions are accidents. In our work, irrespective of whether $f = \pi$ or ρ, we need to consider the accidents in $\mathcal{G}^f(\mathcal{M})$. Let $\mathcal{G}(\mathcal{M})$ denote the set of all structure graphs corresponding to the set of messages $\mathcal{M}$ (by varying f over a function family). For a fixed graph G, let $Acc(G)$ denote the set of all accidents in G. We state the following known results.

Proposition 1. *[10, Lemma 2] For a fixed graph G, $\Pr_f[\mathcal{G}^f(\mathcal{M}) = G] \leq 2^{-n|Acc(G)|}$.*

Proposition 2. *[6, Lemma 7]* $\Pr[G \xleftarrow{\$} \mathcal{G}(\mathcal{M}) : |Acc(G)| \geq 2] \leq \frac{8l^4}{2^{2n}}$.

3 Proposed Constructions for Beyond-Birthday Secure MAC

We introduce here the construction of two separate MACs. One is Sum of Two CBC MACs and another is NI2$^+$ MAC.

3.1 Sum of Two CBC MACs

In this section we present the algorithm for Sum of Two CBC MACs followed by its schematic diagram in Fig.3.1.

For any message $M \in \{0,1\}^*$, the sum of CBC algorithm calls the two subroutines Internal(K_1, M) and Internal(K_2, M), described in Algorithm 1 and 2 respectively, and obtains Σ and Θ respectively. Final tag T is computed by XOR-ing the outputs of $E_{K_1}(\Sigma)$ and $E_{K_2}(\Theta)$.

Internal subroutine (after suitably padding with 10^* if the message length is not a multiple of the block length n) partitions the message M into l many blocks each of which is n bits long. Then it iteratively processes the message up to $l-1$ many blocks. The output of $(l-1)^{th}$ block is XOR-ed with the last block of the message M_l and the output is returned.

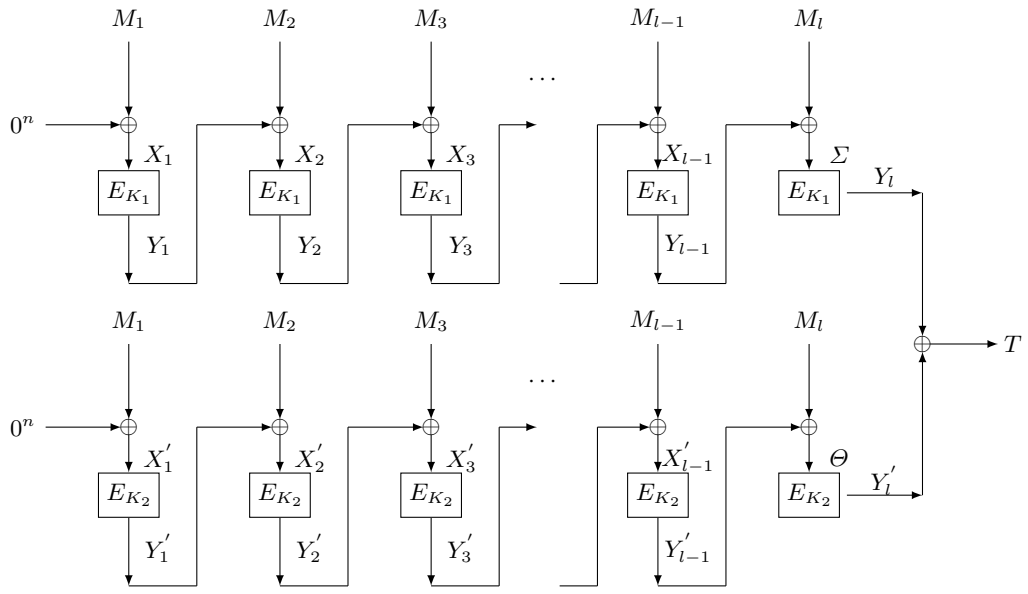

Fig. 3.1: Construction of Sum of Two CBC MACs

Input: $K_1, K_2 \xleftarrow{\$} \mathcal{K}$, $M \leftarrow \{0,1\}^*$
Output: $T \in \{0,1\}^n$
1 $\Sigma \leftarrow Internal(K_1, M)$;
2 $\Theta \leftarrow Internal(K_2, M)$;
3 $T \leftarrow E_{K_1}(\Sigma) \oplus E_{K_2}(\Theta)$;
4 Return T;

Algorithm 1: Algorithm for Sum of Two CBC MACs

Input: $K \xleftarrow{\$} \mathcal{K}$, $M \leftarrow \{0,1\}^*$
Output: $Z \in \{0,1\}^n$
1 $M_1||M_2||\dots M_l \leftarrow M||10^*$;
2 $Y \leftarrow 0^n$;
for $i = 1$ *to* $l-1$ **do**
3 $\quad X \leftarrow Y \oplus M_i$; $Y \leftarrow E_K(X)$;
end
4 $Z \leftarrow Y \oplus M_l$;
5 Return Z;

Algorithm 2: Algorithm for Subroutine Internal

3.2 NI2$^+$ MAC

For this construction, we present the schematic diagram in Fig. 3.2 followed by the description in Algorithm 3.

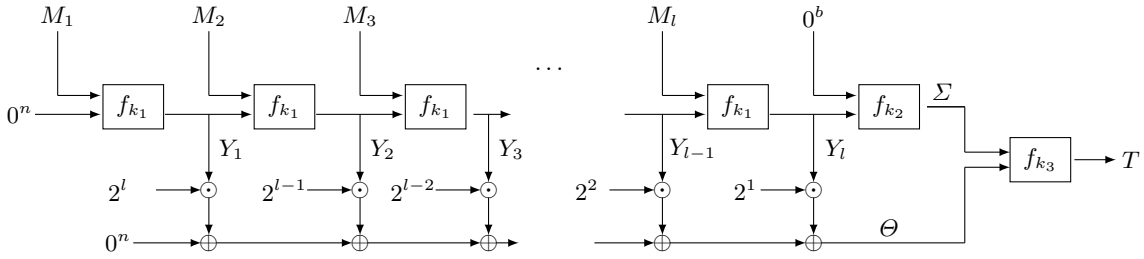

Fig. 3.2: Construction of NI2$^+$ MAC

Input: $f_{K_1}, f_{K_2}, f_{K_3} : K_1, K_2, K_3 \xleftarrow{\$} \mathcal{K}$, $M \leftarrow \{0,1\}^*$
Output: $T \in \{0,1\}^n$

$M_1||M_2||\dots M_l \leftarrow M||10^*$; //$l$ is the number of message blocks in M
$Z \leftarrow 0^n$;
$Y \leftarrow 0^n$;
for $i = 1$ *to* l **do**
  $Y \leftarrow f_{K_1}(M_i, Y)$;
  $Z \leftarrow 2 \cdot (Z \oplus Y)$;
end
$\Theta \leftarrow Z$;
$\Sigma \leftarrow f_{K_2}(0^b, Y)$;
$T \leftarrow f_{K_3}(\Sigma, \Theta)$;
Return T;

Algorithm 3: Algorithm for NI2$^+$ MAC

f_{K_1}, f_{K_2} and f_{K_3} are three independently chosen keyed functions such that $f_{K_1}, f_{K_2} : \{0,1\}^{n+b} \to \{0,1\}^n$ and $f_{K_3} : \{0,1\}^{2n} \to \{0,1\}^n$. We denote

$$Casc^{f_{K_1}}(M) := f_{K_1}(\dots(f_{K_1}(f_{K_1}(f_{K_1}(0, M_1), M_2), M_3), \dots,)M_l)$$

to be the output of the last message block in the upper lane of the construction depicted in Fig.3.2.

For any message $M \in \{0,1\}^*$, NI2$^+$ MAC (after suitably padding with 10^* if the message length is not a multiple of the block length b) partitions M into l many blocks

each of which is b bits long. Then the blocks are iteratively processed as depicted in Fig.3.2. Final output Y_l of $Casc^{f_{K_1}}(M)$ as depicted in Fig.3.2 and 0^b becomes the input of $f_{K_2}(\cdot,\cdot)$ and the output of $f_{K_2}(\cdot,\cdot)$ is denoted as Σ. This is the so-called NI2 construction which we extend as follows. A linear combination of the intermediate chaining value of $Casc^{f_{K_1}}(M)$ is denoted as Θ. The symbol '2' in the construction is the root of an irreducible polynomial of degree n. Σ and Θ are then fed into $f_{K_3}(\cdot,\cdot)$ and the output is returned as tag T.

4 Analysis of Sum of Two Independent Random Permutations over Restricted Domain and Range

Stefan Lucks, in [15], used the concept of a *fair* set, to show that the statistical distance between uniform distribution and that of the sum of $(d \geq 2)$ independent random permutations sampled from $Perm(n)$ is $\frac{1}{2^{dn-1}} \cdot \sum_{i=0}^{q} i^d$. For $d = 2$, when the inputs are distinct, $\Pi_1(x) \oplus \Pi_2(x)$ is a secure PRF with security bound $\frac{q^3}{3*2^{2n-1}}$. In [4, Section 4.3], using ratio based comparison theorem, Bellare et al. have shown that sum construction $S(x) = \Pi(x||0) \oplus \Pi(x||1)$ where $x \in \{0,1\}^{n-1}$ is a secure PRF with security bound $\frac{q}{2^n} + O(n) \cdot \frac{q^{3/2}}{2^{3n/2}}$ where Π is a random permutation sampled from $Perm(n)$. In this section, for the first time, we analyze the sum of two independent random permutations in a more general setting, where the domains and the ranges of the permutations may have certain *restrictions*.

4.1 Modeling the Restrictions

Let $D = R = \{0,1\}^n$. Let $D_1, D_2 \subset D$ and $R_1, R_2 \subset R$ (all arbitrary but large subsets; bound on the cardinality of the subset is given in Theorem 1), such that $|D_1| = |R_1| = \sigma_1$ and $|D_2| = |R_2| = \sigma_2$. Consider two *fixed* bijective functions $g_1 : D_1 \rightarrow R_1$ and $g_2 : D_2 \rightarrow R_2$. For $s = 1, 2$, we set $D_s^{'} = D \backslash D_s$, $R_s^{'} = R \backslash R_s$, and sample Π_s uniformly at random from $Bij(D_s^{'}, R_s^{'})$, and then extend its definition over the full domain D and range R such that $\forall x \in D_s$, $\Pi_s(x) = g_s(x)$. Thus, the fixed mapping g_s determines the common restriction on each randomly selected permutation Π_s, $s = 1, 2$. We show that the sum construction $C_S(x, y) = \Pi_1(x) \oplus \Pi_2(y)$ where $x \in D_1^{'}$ and $y \in D_2^{'}$ is a secure PRF with security bound $O(\frac{q(\sigma+q)^2}{2^{2n}})$. In this regard, like [15] our analysis is carried out using the fair-set technique.

4.2 Analysis of the PRF advantage of the construction C_S

We consider an adversary $\mathcal{A}$ that is presented with an on oracle $\mathcal{O}$. $\mathcal{A}$ is allowed to make pair of query (x_i, y_i) to oracle $\mathcal{O}$ such that $x_i \neq x_j$ and $y_i \neq y_j$, $\forall 1 \leq i, j \leq q$. We define the advantage of $\mathcal{A}$ with respect to C_S as follows:

$$\begin{aligned}\mathbf{Adv}^{\mathbf{PRF}}_{\mathbf{C_S}}(\mathcal{A}) := \Pr[\mathcal{A}^{C_S} = 1 : \Pi_1 \xleftarrow{\$} Bij(D_1^{'}, R_1^{'}), \Pi_2 \xleftarrow{\$} Bij(D_2^{'}, R_2^{'})] \\ - \Pr[\mathcal{A}^{\rho} = 1 : \rho \xleftarrow{\$} Func(\{0,1\}^n \times \{0,1\}^n, \{0,1\}^n)].\end{aligned}$$

For any adaptive adversary $\mathcal{A}$, Theorem 1 establishes an upper bound of the advantage for construction C_S.

Theorem 1. *Let $\Pi_1 \xleftarrow{\$} Bij(D_1^{'}, R_1^{'})$ and $\Pi_2 \xleftarrow{\$} Bij(D_2^{'}, R_2^{'})$, where $|D_s^{'}| = (2^n - \sigma_s)$ and the definition of Π_s is extended to the full domain and range following the restrictions g_s as defined above, for $s = 1, 2$. Suppose $z_1 = \Pi_1(x)$ and $z_2 = \Pi_2(y)$, where $x \in D_1^{'}$ and $y \in D_2^{'}$. Then $C_S(x, y) := \Pi_1(x) \oplus \Pi_2(y) = z_1 \oplus z_2$ is a secure PRF with advantage at most $\frac{q(\sigma+q)^2}{2^{2n}}$.*

The proof of Theorem 1 requires a fair-set construction. For the analysis purpose, we use the same definition of fair set which was originally proposed by Lucks in [15]:

Definition 1. *A set $F \subseteq \{0,1\}^n \times \{0,1\}^n$ is said to be a fair set, if for any $v \in \{0,1\}^n$, we have $|\{(z_1, z_2) \in F : (z_1 \oplus z_2) = v\}| = \frac{|F|}{2^n}$.*

Note that, if $(z_1, z_2) \in F$ then $(z_1 \oplus z_2)$ is uniformly distributed over $\{0,1\}^n$.

Construction of the Fair Set. Let $R_1^{''} = \{y_1, y_2, \ldots, y_{\sigma_1+i}\}$ denotes the consumed range set of Π_1 up to i^{th} query. $R_2^{''} = \{y_1^{'}, y_2^{'}, \ldots, y_{\sigma_2+i}^{'}\}$ denotes the consumed range set of Π_2 up to i^{th} query. More formally,

$$R_1^{''} = R_1 \cup \left\{ y_j : \Pi_1(x_j) = y_j, x_j \in D_1^{'}, \sigma_1 + 1 \leq j \leq \sigma_1 + i \right\}.$$

Similarly,

$$R_2^{''} = R_2 \cup \left\{ y_j^{'} : \Pi_2(x_j^{'}) = y_j^{'}, x_j^{'} \in D_2^{'}, \sigma_2 + 1 \leq j \leq \sigma_2 + i \right\}.$$

At the time of $(i+1)^{th}$ query where $i \geq 0$, we define the *Consumed set* S_{i+1} to be the set of values that (z_1, z_2) can't take due to domain reduction. It is easy to see that,

$$S_{i+1} = (y_1, *) \cup (y_2, *) \cup \ldots \cup (y_{\sigma_1+i}, *) \cup (*, y_1^{'}) \cup (*, y_2^{'}) \cup \ldots \cup (*, y_{\sigma_2+i}^{'}),$$

where $(a, *)$ denotes the set $\{(a, i) : i \in \{0,1\}^n\}$ and $(*, a)$ denotes the set $\{(i, a) : i \in \{0,1\}^n\}$.

It is to be noted that if all of these $(\sigma_1 + \sigma_2 + 2i)$ sets were distinct then we would have $|S_{i+1}| = 2^n(\sigma_1 + \sigma_2 + 2i)$. But for each two sets $(y_s, *)$ and $(*, y_t^{'})$, where $s \in \{1, \ldots, \sigma_1 + i\}$ and $t \in \{1, \ldots, \sigma_2 + i\}$, there is a common element namely $(y_s, y_t^{'})$. Let $I = \{(y_\alpha, y_\beta^{'}) : y_\alpha \in R_1^{''}, y_\beta^{'} \in R_2^{''}\}$ be the set of $(\sigma_1 + i)(\sigma_2 + i)$ common elements in S_{i+1}. Therefore,

$$|S_{i+1}| = 2^n(\sigma_1 + \sigma_2 + 2i) - (\sigma_1 + i)(\sigma_2 + i).$$

Suppose $U = \{(x, y) : x, y \in \{0,1\}^n\}$ be the set of values that (z_1, z_2) can take when we consider two independent random permutation sampled from $Perm(n)$. We call $S_{i+1}^c = U \setminus S_{i+1}$ as the *Unconsumed set*. It is easy to see that

$$|S_{i+1}^c| = 2^{2n} - 2^n(\sigma_1 + \sigma_2 + 2i) + (\sigma_1 + i)(\sigma_2 + i).$$

It is easy to see that S_{i+1}^c is not a fair set as $|S_{i+1}^c|$ is not multiple of 2^n. Thus at the time of $(i+1)^{th}$ query, we will construct a set $F_{i+1} \subset S_{i+1}^c$ so that F_{i+1} becomes a fair set. Therefore, at the time of $(i+1)^{th}$ query, we will construct a set $F_{i+1} \subset S_{i+1}^c$ according to Algorithm 4.

For each of the element $(y_\alpha, y_\beta^{'}) \in I$, we find an element $(y_g, y_h) \in S_{i+1}^c$, such that $y_\alpha \oplus y_\beta^{'} = y_g \oplus y_h$. We say (y_g, y_h) to be a representative of $(y_\alpha, y_\beta^{'})$. Therefore, for each element of I we find a unique representative in S_{i+1}^c. Let

$$L_{i+1} := \left\{(y_g, y_h) \in S_{i+1}^c : \exists (y_\alpha, y_\beta^{'}) \in I, y_\alpha \oplus y_\beta^{'} = y_g \oplus y_h\right\}.$$

Now, we define a set $\mathcal{S}_{i+1}$ which is initially to be empty. For each $(y_\alpha, y_\beta^{'}) \in I$, we choose a pair $(y_g, y_h) \in L_{i+1}$ such that $y_\alpha \oplus y_\beta^{'} = y_g \oplus y_h$. Then, $\mathcal{S}_{i+1} \leftarrow \mathcal{S}_{i+1} \cup \{(y_g, y_h)\}$. Note that $|\mathcal{S}_{i+1}| = (\sigma_1 + i)(\sigma_2 + i)$. The fair-set thus we construct $F_{i+1} := S_{i+1}^c \setminus \mathcal{S}_{i+1}$. We show the fair-set construction in Algorithm 4.

Note that the algorithm doesn't abort, i.e., Y will not be an empty set set, i.e., some (y_g, y_h) pair will exist in S_{i+1}^c in step 3 of Algorithm 4, can be easily verified from the following Lemma.

Lemma 1. *At the time of making $(i+1)^{th}$ query and for any field element v, there is exactly $2^n - (\sigma_1 + \sigma_2 + 2i) + t_v$ many pairs (y_g, y_h) in S_{i+1}^c, for which $y_g \oplus y_h = v$, where $I_v = \{(y_\alpha, y_\beta^{'}) \in I : y_\alpha \oplus y_\beta^{'} = v\}$ with $|I_v| = t_v \leq (\sigma_1 + i)(\sigma_2 + i)$.*

Proof. The first thing to observe that any finite field element can be written as a sum of two distinct field elements in 2^n ways. Since $|I_v| = t_v$, therefore, to express

Input: S^c_{i+1}, I
Output: F_{i+1}

Set $F_{i+1} \leftarrow S^c_{i+1}$;
For each element $(y_\alpha, y'_\beta) \in I$;
    Find the set $Y = \{(y_g, y_h) \in S^c_{i+1} : (y_g \oplus y_h) = (y_\alpha \oplus y'_\beta)\}$;
    If $Y \leftarrow \phi$, Abort ;
    Randomly choose an element $(z_1, z_2) \in Y$;
    Update $F_{i+1} \leftarrow F_{i+1} \setminus \{(z_1, z_2)\}$;

Algorithm 4: Algorithm for Construction of Fair Set $F_{i+1} \subset S^c_{i+1}$.

v as a sum of two distinct field elements we have already exhausted t_v many field elements.
Now if we look at $S_{i+1} \setminus I_v$, then we will have $(\sigma_1+\sigma_2+2i)-t_v$ many $(y_\gamma, y_\delta) \in S_{i+1} \setminus I_v$ such that $(y_\gamma \oplus y_\delta) = v$. Therefore, total $(\sigma_1 + \sigma_2 + 2i) - t_v$ field elements are consumed to write v as a sum of two distinct field elements. Therefore, the number of remaining field element is $2^n - (\sigma_1 + \sigma_2 + 2i) + t_v$. Therefore the number of ways we can write v as a sum of two distinct field elements (y_g, y_h) such that $(y_g, y_h) \in S^c_{i+1}$ is $2^n - (\sigma_1 + \sigma_2 + 2i) + t_v$. Therefore, to write v as a sum of $(y_g, y_h) \in S^c_{i+1}$ is $2^n - (\sigma_1 + \sigma_2 + 2i) + t_v$. □

Thus, we have

$$|F_{i+1}| = 2^{2n} - 2^n(\sigma_1 + \sigma_2 + 2i).$$

The next lemma establishes the fairness of F_{i+1}.

Lemma 2. *The set F_{i+1} constructed in Algorithm 4 is fair.*

Proof. Fix any field element v. Now two cases are possible: (a) when there exists t_v many pairs $(y_\alpha, y'_\beta) \in I$ such that $y_\alpha \oplus y'_\beta = v$. (b) when there exists no pairs $(y_\alpha, y'_\beta) \in I$ such that $y_\alpha \oplus y'_\beta = v$. Consider set S_{i+1}. For case (a), the number of pairs $(y_\gamma, y_\delta) \in S_{i+1} \setminus I_v$ such that $y_\gamma \oplus y_\delta = v$ is $(\sigma_1 + \sigma_2 + 2i) - t_v$. For case (b), the number of pairs $(y_\gamma, y_\delta) \in S_{i+1}$ such that $y_\gamma \oplus y_\delta = v$ is $(\sigma_1 + \sigma_2 + 2i)$. That means, for case (a), number of pairs $(y_g, y_h) \in S^c_{i+1}$ is $2^n - (\sigma_1 + \sigma_2 + 2i) + t_v$ and for case (b), number of pairs $(y_g, y_h) \in S^c_{i+1}$ is $2^n - (\sigma_1 + \sigma_2 + 2i)$. Therefore to keep the count same for each element we remove all the t_v many pairs (y_γ, y_δ) from I. Therefore, we remove a unique representative for each of the elements of I and thus producing a set F_{i+1}, where for each non zero element z, number of pairs (y_γ, y_δ) such that $y_\gamma \oplus y_\delta = z$ is same. Thus F_{i+1} is a fair set. □

Proof of Theorem 1. Now we are in a position to complete the proof of Theorem 1. At the time of sampling the output for $(i+1)^{th}$ query, **bad** event will occur when $(z_1^{i+1}, z_2^{i+1}) \in S_{i+1}^c \setminus F_{i+1}$. Therefore, we bound the probability of the **bad** event as follows:

$$\Pr[\mathbf{bad} = 1] = \sum_{i=0}^{q-1} \Pr[(z_1^{i+1}, z_2^{i+1}) \notin F_{i+1}] = \sum_{i=0}^{q-1} \frac{|S_{i+1}^c| - |F_{i+1}|}{|S_{i+1}^c|}$$
$$= \sum_{i=0}^{q-1} \frac{(\sigma_1 + i)(\sigma_2 + i)}{2^{2n} - 2^n(\sigma_1 + \sigma_2 + 2i) + (\sigma_1 + i)(\sigma_2 + i)} \leq \frac{q(\sigma_1 + q)^2}{2^{2n}} \leq \frac{q(\sigma + q)^2}{2^{2n}},$$

assuming $(\sigma_1 + q) < 2^{n-1}$, and (without loss of generality) $\sigma_2 \leq \sigma_1 \leq \sigma$.

5 Security Analysis of Sum of Two CBC MACs

In this section, we show that the Sum of Two CBC-MACs construction is a secure PRF with security beyond birthday bound. We start our analysis in the same line as [19, Section 5] that showed the security beyond birthday of the Sum of Two ECBC-MACs construction. However, we observe that the proof for Sum of Two independent ECBC-MACs is relatively simpler as the output of two independent CBC-MACs, namely, $\Sigma = CBC_{\pi_1}(M)$ and $\Theta = CBC_{\pi_2}(M)$, are fed into two independent random permutations π_3 and π_4 respectively. However author mentioned in [19, Section 6] that their result for sum of two ECBC-MACs do not seem to be directly applicable to the sum of one key CBC-MACs such as CMAC or GCBC-MAC. In this section, we proceed to analyze the case of sum of two independent CBC-MACs which was remain open in [19, Section 6].

We use the standard trick of replacing the block ciphers E_{K_1} and E_{K_2} with independent random permutations π_1 and π_2 from n bits to n bits respectively. We call the resulting scheme SUM-CBC$[\pi_1, \pi_2]$. The following result is immediate.

$$\mathbf{Adv}^{PRF}_{\text{SUM-CBC}[E_{K_1}, E_{K_2}]}(t, q, l) \leq \mathbf{Adv}^{PRF}_{\text{SUM-CBC}[\pi_1, \pi_2]}(q, l) + 2\epsilon, \tag{1}$$

where $\epsilon = \mathbf{Adv}^{PRP}_{E}(t', lq)$. The time complexity t' is of the order of the original running time t plus the time to compute block cipher E for lq times.

5.1 Our Main Result

Now we state our main theorem related to the PRF-advantage of SUM-CBC$[\pi_1, \pi_2]$.

Theorem 2. *We have*

$$\mathbf{Adv}^{PRF}_{\text{SUM-CBC}[\pi_1,\pi_2]}(q,l) \leq \frac{196l^3q^3}{2^{2n}}.$$

Proof. Let $\mathcal{A}$ be an adaptive adversary that makes at most q queries, each query being at most l blocks. The goal of $\mathcal{A}$ is to distinguish between the SUM-CBC$[\pi_1, \pi_2](\cdot)$ oracle and a truly random function $\rho : \{0,1\}^* \to \{0,1\}^n$. Upon making a query M, we consider the algorithm in Algorithm 5.

Input: Message M
Output: Tag T
$\pi_1, \pi_2 \xleftarrow{\$} Perm(n)$; $\Sigma \leftarrow$ **Internal**(π_1, M);
$\Theta \leftarrow$ **Internal**(π_2, M);
if $\Sigma \notin Dom(\pi_1)$ *and* $\Theta \notin Dom(\pi_2)$ **then**
  go to **Case A**; (computes T and may set **bad** flag true)
end
if $\Sigma \in Dom(\pi_1)$ *and* $\Theta \notin Dom(\pi_2)$ **then**
  go to **Case B**; (computes T and may set **bad** flag true)
end
if $\Sigma \notin Dom(\pi_1)$ *and* $\Theta \in Dom(\pi_2)$ **then**
  go to **Case C**; (computes T and may set **bad** flag true)
end
if $\Sigma \in Dom(\pi_1)$ *and* $\Theta \in Dom(\pi_2)$ **then**
  go to **Case D**; (computes T and set **bad** flag true)
end
Return T;

Algorithm 5: Main Algorithm

Algorithm 5 shows the algorithm using which the random permutations π_1 and π_2 are lazily sampled. The sub-routines for **Case A**, **Case B** and **Case D** are given in subsequent subsections.

For all of the queried messages $M^{(i)}$ where $i \in [1, q]$, if the **bad** flag is not set to be true, then the distribution of the output of SUM-CBC$[\pi_1, \pi_2]$ will be identical to the uniformly random distribution. Therefore, by the fundamental lemma of game playing [7],

$$\begin{aligned}\mathbf{Adv}^{PRF}_{\text{SUM-CBC}[\pi_1,\pi_2]}(\mathcal{A}) &= \Pr[\mathcal{A}^{\text{SUM-CBC}[\pi_1,\pi_2](\cdot)} = 1] - \Pr[\mathcal{A}^{\rho(\cdot)} = 1] \\ &\leq \Pr[\textbf{bad} \text{ event occurs}].\end{aligned}$$

Since our setting is information-theoretic, we can assume that $\mathcal{A}$ is deterministic. Also, the adversary $\mathcal{A}$ learns nothing from the values returned by the oracles, as the

values are mere random strings and do not help $\mathcal{A}$ set **bad** flags (until one of the **bad** flags gets set). Hence, we can assume that $\mathcal{A}$ is non-adaptive and so we consider only a fixed sequence of queries $M^{(1)}, \ldots, M^{(q)}$ output by $\mathcal{A}$. Therefore, it amounts to compute the probability that the sequence of queries $M^{(1)}, \ldots, M^{(q)}$ output by $\mathcal{A}$ sets the **bad** flag true. Let l_i be the number of blocks in $M^{(i)}$, and as already assumed, each $l_i \leq l$.

In subsequent sections we will analyze the probability of **bad** flag set to true corresponding to the four different cases and will show the upper-bounds on these probabilities as $\frac{4l^2q^3}{2^{2n}}$, $\frac{32l^3q^2}{2^{2n}}$, $\frac{32l^3q^2}{2^{2n}}$ and $\frac{128l^3q^3}{2^{2n}}$ respectively. Adding these, we get the result. □

5.2 Analysis of Case A: $\Sigma \notin \mathbf{Dom}(\pi_1)$ and $\Theta \notin \mathbf{Dom}(\pi_2)$

The subroutine for this case is given in Algorithm 6. Since Σ and Θ are fresh, output T is calculated by sampling two range points $\pi_1(\Sigma)$ and $\pi_2(\Theta)$ as $z_1 \xleftarrow{\$} \{0,1\}^n \setminus \mathrm{Ran}(\pi_1)$ and $z_2 \xleftarrow{\$} \{0,1\}^n \setminus \mathrm{Ran}(\pi_2)$ respectively. Note that T will not be random if $(z_1, z_2) \notin F$ where F is a fair-set.

```
1 Z_1 ← {0,1}^n \ Ran(π_1);
2 Z_2 ← {0,1}^n \ Ran(π_2);
3 Choose a fair set F ⊂ Z_1 × Z_2;
4 (z_1, z_2) ←$ Z_1 × Z_2;
  if (z_1, z_2) ∉ F then
5 |   bad set to true;
  end
6 T ← z_1 ⊕ z_2;
7 Return T;
```

Algorithm 6: Subroutine for **Case A**

Therefore, in this case, **bad** flag set to true when (z_1, z_2) do not fall in the fair set F, as shown in Algorithm 6. In this case, the maximum number of consumed elements after q queries is given by $\sigma = lq$, because each of the q queries is at most l blocks long. Therefore, substituting $\sigma = lq$ in Theorem 1, we get

$$\Pr[\textbf{bad} \text{ set to true in } \textbf{Case A}] \leq \frac{q(lq+q)^2}{2^{2n}} \leq \frac{4l^2q^3}{2^{2n}}.$$

5.3 Analysis of Case B: $\Sigma \in \mathbf{Dom}(\pi_1)$ and $\Theta \notin \mathbf{Dom}(\pi_2)$

The subroutine for this case is presented in Algorithm 7. In this case, Σ has been collided with an element of domain of π_1 and Θ is fresh. Therefore, $z_1 = \pi_1(\Sigma)$ is already defined and sample the range point of $\pi_2(\Theta)$ as $z_2 \xleftarrow{\$} \{0,1\}^n$. Then the output $T = z_1 \oplus z_2$ will be uniformly distributed if $z_2 \in \{0,1\}^n \setminus \mathrm{Ran}(\pi_2)$. Therefore, T will not be uniformly distributed if $z_2 \in \mathrm{Ran}(\pi_2)$. Hence the **bad** flag set to true when $z_2 \in \mathrm{Ran}(\pi_2)$ which is shown in Algorithm 7. Note that the code without boxed statement simulates the random function ρ.

1 $z_1 \leftarrow \pi_1(\Sigma)$;
2 $z_2 \xleftarrow{\$} \{0,1\}^n$;
if $z_2 \in Ran(\pi_2)$ **then**
3 **bad** set to true; $\boxed{z_2 \xleftarrow{\$} \{0,1\}^n \setminus Ran(\pi_2)}$
end
4 $T \leftarrow z_1 \oplus z_2$;
5 Return T;

Algorithm 7: Subroutine for **Case B**

Note that

$$
\begin{aligned}
&\Pr[\textbf{bad} \text{ set to true in } \textbf{Case B}] \\
&\leq \sum_{i=2}^{q} \Pr[\Sigma^{(i)} \in Dom(\pi_1) \wedge z_2^{(i)} \in Ran(\pi_2) : \pi_1, \pi_2 \xleftarrow{\$} Perm(n)] \\
&\leq \sum_{i=2}^{q} \Pr[\Sigma^{(i)} \in Dom(\pi_1) : \pi_1 \xleftarrow{\$} Perm(n)] \cdot \Pr[z_2^{(i)} \in Ran(\pi_2) : \pi_2 \xleftarrow{\$} Perm(n)] \\
&\leq \sum_{i=2}^{q} \sum_{j=1}^{i-1} (\Pr[\Sigma^{(i)} = \Sigma^{(j)} : \pi_1 \xleftarrow{\$} Perm(n)] \\
&\quad + \Pr[\Sigma^{(i)} = X_\alpha : \pi_1 \xleftarrow{\$} Perm(n)]) \cdot \Pr[z_2^{(i)} \in Ran(\pi_2) : \pi_2 \xleftarrow{\$} Perm(n)] \\
&\leq \sum_{i=2}^{q} \sum_{j=1}^{i-1} \Pr[\Sigma^{(i)} = \Sigma^{(j)} : \pi_1 \xleftarrow{\$} Perm(n)] \cdot \Pr[z_2^{(i)} \in Ran(\pi_2) : \pi_2 \xleftarrow{\$} Perm(n)] \\
&+ \sum_{i=2}^{q} \sum_{j=1}^{i-1} \Pr[\Sigma^{(i)} = X_\alpha : \pi_1 \xleftarrow{\$} Perm(n)] \cdot \Pr[z_2^{(i)} \in Ran(\pi_2) : \pi_2 \xleftarrow{\$} Perm(n)],
\end{aligned}
$$

where $1 \leq \alpha \leq l_i + l_j$.

We can now prove the following result.

Lemma 3. *Let $M^{(1)}, M^{(2)}, \ldots, M^{(q)}$ are the distinct prefix-free message queries, each of length at most l blocks, asked by any non-adaptive adversary $\mathcal{A}$. Then,*

$$\Pr[\textbf{bad} \text{ set to true in } \textbf{Case B}] \leq \frac{32l^2q^3}{2^{2n}},$$

where $l \leq 2^{n/2}$.

Proof. The proof follows from Claim 1 and 2. From Equation (2) and Equation (5) we have,

$$\begin{aligned}
\Pr[\textbf{bad} \text{ set to true in } \textbf{Case B}] &\leq \sum_{i=2}^{q}\sum_{j=1}^{i-1} 2(\frac{8l}{2^n} + \frac{8l^4}{2^{2n}}) \cdot \frac{|Ran(\pi_2)|}{2^n} \\
&\leq \sum_{i=2}^{q}\sum_{j=1}^{i-1} 2(\frac{8l}{2^n} + \frac{8l^4}{2^{2n}}) \cdot \frac{2l}{2^n} \\
&\leq \sum_{i=2}^{q}\sum_{j=1}^{i-1} \frac{32l^3}{2^{2n}} \cdot \left(1 + \frac{l^2}{2^n}\right), \qquad \text{assuming } l \leq 2^{n/2} \\
&\leq \frac{q^2}{2} \cdot \frac{64l^3}{2^{2n}} \\
&\leq \frac{32q^2l^3}{2^{2n}}.
\end{aligned}$$

□

Now we state and prove the two claims used in the above Lemma.

Claim 1 *Fix any two distinct messages M and $M^{'}$ each of which is at most l blocks long such that M is not a prefix of $M^{'}$ and vice-versa. Then*

$$\Pr[\Sigma = \Sigma^{'}] \leq \frac{8l}{2^n} + \frac{8l^4}{2^{2n}} \tag{2}$$

Proof. We prove the claim using the structure graph. After fixing two messages M and $M^{'}$ and choosing a permutation π uniformly at random from the set of all permutations, we analyze the structure graph $G := G^{\pi}(M, M^{'})$. In particular, we analyze the probability of the event $\Sigma = \Sigma^{'}$ in view of the number of accidents occurred in the corresponding structure graph G. Therefore, we have,

$$\Pr[\Sigma = \Sigma^{'}] = \Pr[\Sigma = \Sigma^{'} \wedge |Acc(G)| = 1] + \Pr[\Sigma = \Sigma^{'} \wedge |Acc(G)| \geq 2]$$

Now, when the number of accident in the graph G is 1, then we have the following:

$$\Pr[\Sigma = \Sigma^{'} \wedge |Acc(G)| = 1] \leq \frac{8l}{2^n}$$

Now, when the number of accident in the graph G is 1, then we have the following:

$$\Pr[\Sigma = \Sigma^{'} \wedge |Acc(G)| = 1] \leq \frac{8l}{2^n}$$

Note that, according to Figure 3.1, $\Sigma = \Sigma^{'} \iff Y_l = Y_{l'}^{'}$, where l is the number of blocks of M and $l^{'}$ is the number of blocks of $M^{'}$. Therefore,

$$\begin{aligned}\Pr[\Sigma = \Sigma^{'} \wedge |Acc(G)| = 1] &= \Pr[Y_l = Y_{l'}^{'} \wedge |Acc(G)| = 1] \\ &= \Pr[CBC_\pi(M) = CBC_\pi(M^{'}) \wedge |Acc(G)| = 1] \quad (3)\end{aligned}$$

According to Equation (3), calculating $\Pr[\Sigma = \Sigma^{'} \wedge |Acc(G)| = 1]$ is the same as calculating the collision probability of CBC for a pair of prefix free messages when the number of accident in the structure graph is 1. Thus, using [6, Lemma 17] we have, $\Pr[CBC_\pi(M) = CBC_\pi(M^{'}) \wedge |Acc(G)| = 1] \leq \frac{8l}{2^n}$.
Again, when the number of accident in the graph G is at least 2, then using Proposition 2 we have the following:

$$\Pr[\Sigma = \Sigma^{'} \wedge |Acc(G)| \geq 2] \leq \Pr[|Acc(G)| \geq 2] \leq \frac{8l^4}{2^{2n}} \quad (4)$$

Therefore,

$$\Pr[\Sigma = \Sigma^{'}] \leq \left(\frac{8l}{2^n} + \frac{8l^4}{2^{2n}}\right)$$

□

Claim 2 *Fix any two distinct messages M and $M^{'}$ such that M is not a prefix of $M^{'}$ and vice-versa. Then*

$$\Pr[\Sigma = X] \leq \frac{8l}{2^n} + \frac{8l^4}{2^{2n}}, \quad (5)$$

where X is any intermediate input of CBC MAC computation for $\left\{M, M^{'}\right\}$.

Proof. Proof sketch of the Claim 2 is exactly same to the proof sketch of the Claim 1. □

5.4 Analysis of Case C: $\Sigma \notin \mathbf{Dom}(\pi_1)$ and $\Theta \in \mathbf{Dom}(\pi_2)$

This case is symmetric to Case B where the role of Σ and Θ has interchanged. Therefore,

$$\Pr[\textbf{bad} \text{ set to true in } \textbf{Case C}] \leq \frac{32q^2l^3}{2^{2n}},$$

where $l \leq 2^{n/2}$.

5.5 Analysis of Case D: $\Sigma \in \mathbf{Dom}(\pi_1)$ and $\Theta \in \mathbf{Dom}(\pi_2)$

We present the subroutine for this case in Algorithm 8.

bad set to true;
$z_1 \leftarrow \pi_1(\Sigma)$; $z_2 \leftarrow \pi_2(\Theta)$;
Return $T \leftarrow z_1 \oplus z_2$;

Algorithm 8: Subroutine for **Case D**

Since both Σ and Θ are not fresh, output T is defined and the **bad** flag set to true. Therefore, in **Case D bad** flag set to true when both Σ and Θ are not fresh.

We show the following lemma.

Lemma 4. *Let $M^{(1)}, M^{(2)}, \ldots, M^{(q)}$ are the distinct prefix-free message queries, each of length at most l blocks, asked by any non-adaptive adversary $\mathcal{A}$. Then,*

$$\Pr[\textbf{bad} \text{ set to true in } \textbf{Case D}] \leq \frac{128l^3q^3}{2^{2n}},$$

if $l \leq 2^{2n/5}$.

Proof. It is to be noted that,

$$\begin{aligned}
&\Pr[\textbf{bad} \text{ set to true in } \textbf{Case D}] \\
&\leq \sum_{i=2}^{q} \Pr[\Sigma^{(i)} \in Dom(\pi_1) \wedge \Theta^{(i)} \in Dom(\pi_2) : \pi_1, \pi_2 \xleftarrow{\$} Perm(n)] \\
&\leq \sum_{i=2}^{q} \Pr[\Sigma^{(i)} \in Dom(\pi_1) : \pi_1 \xleftarrow{\$} Perm(n)] \cdot \Pr[\Theta^{(i)} \in Dom(\pi_2) : \pi_2 \xleftarrow{\$} Perm(n)] \\
&\leq \sum_{i=2}^{q} \sum_{j=1}^{i-1} \sum_{k=1}^{i-1} (\Pr[\Sigma^{(i)} = \Sigma^{(j)} : \pi_1 \xleftarrow{\$} Perm(n)] + \Pr[\Sigma^{(i)} = X_\alpha : \pi_1 \xleftarrow{\$} Perm(n)]) \\
&\quad \cdot (\Pr[\Theta^{(i)} = \Theta^{(k)} : \pi_2 \xleftarrow{\$} Perm(n)] + \Pr[\Theta^{(i)} = X_\beta : \pi_2 \xleftarrow{\$} Perm(n)])
\end{aligned}$$

where $1 \leq \alpha, \beta \leq l_i + l_j + l_k$.

From Equation (2) and Equation (5) we obtain the following.

$$\begin{aligned}
\Pr[\textbf{bad} \text{ gets true in } \textbf{Case D}] &\leq \sum_{i=2}^{q}\sum_{j=1}^{i-1}\sum_{k=1}^{i-1}(\frac{8l}{2^n} + \frac{8l^4}{2^{2n}})^2 \\
&\leq \sum_{i=2}^{q}\sum_{j=1}^{i-1}\sum_{k=1}^{i-1}(\frac{64l^2}{2^{2n}} + \frac{128l^5}{2^{3n}} + \frac{64l^8}{2^{4n}}) \\
&\leq \sum_{i=2}^{q}\sum_{j=1}^{i-1}\sum_{k=1}^{i-1}\frac{l^3}{2^{2n}} \cdot (64 + \frac{128l^2}{2^n} + \frac{64l^5}{2^{2n}}) \\
&\leq \frac{q^3}{6} \cdot \frac{256l^3}{2^{2n}} \\
&\leq \frac{128q^3l^3}{2^{2n}}, \quad \text{if } l \leq 2^{2n/5}.
\end{aligned}$$

Note that, $\Pr[\Theta^{(i)} = \Theta^{(k)} : \pi_2 \xleftarrow{\$} Perm(n)]$, or $\Pr[\Theta^{(i)} = X_\beta : \pi_2 \xleftarrow{\$} Perm(n)] \leq \frac{8l}{2^n} + \frac{8l^4}{2^{2n}}$ as the role of Θ in the second lane of CBC-MAC$_{\pi_2}(M)$ is same as that of Σ in first lane of CBC-MAC$_{\pi_1}(M)$).

6 Security Analysis of NI2$^+$ MAC

Gazi et. al in [10] have shown that the advantage of distinguishing the output of NI-MAC from random output is bounded above by $\frac{q^2}{2^n}\left(l + \frac{64l^4}{2^n}\right)$ and that for NI2-MAC is $\frac{q^2}{2^n}\left(ld^{'}(l) + \frac{64l^4}{2^n}\right)$ where $d^{'}(l) = \max\limits_{l^{'} \in \{1,\ldots,l\}} |\left\{d \in \mathbb{N} : d|l^{'}\right\}|$. In this section we analyze the advantage of our construction NI2$^+$-MAC and show that the advantage of our construction achieves beyond birthday bound security; better than that of NI-MAC or NI2-MAC. Thus we have the following theorem.

Theorem 3. *Let $f : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^b \to \{0,1\}^n$ be a (ϵ_1, t, q) secure PRF and (ϵ_2, t, lq) secure PRF. Let $h : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a (ϵ_3, t, q) secure PRF. Then NI2$^+$ be a $\left(\epsilon^{'}, t^{'}, q, l\right)$ secure PRF, where*

$$\epsilon^{'} \leq \epsilon_1 + \epsilon_2 + \epsilon_3 + \frac{15q^2l^4}{2^{2n}},$$

such that $t = t^{'} + \tilde{O}(lq)$.

Proof. We below give the sketch of the proof of Theorem 3. Let $\mathcal{A}$ be a adaptive PRF-adversary against NI2$^+$ running in time t and asking at most q queries, each of length at most l blocks. NI2$^+$ uses three independent keyed functions f_1, f_2 and h_3. Now if we replace f_1, f_2 and h_3 by three different random functions r_1, r_2 and r_3 respectively such that $r_1, r_2 \xleftarrow{\$} Func(\{0,1\}^n \times \{0,1\}^n \times \{0,1\}^b, \{0,1\}^n)$ and $r_3 \xleftarrow{\$} Func(\{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n, \{0,1\}^n)$ and call the resulting construction NI2$_r^+$, then we have

$$\Delta^{\mathcal{A}}(\text{NI2}^+, R) \leq \epsilon_1 + \epsilon_2 + \epsilon_3 + \Delta^{\mathcal{A}}(\text{NI2}_r^+, R),$$

where ϵ_i is the PRF-advantage of f_i, $i = 1, 2$ and ϵ_3 is the PRF-advantage of h_3 and $R : \{0,1\}^* \to \{0,1\}^n$ be a uniform random function.

Therefore to prove Theorem 3, we only need to prove

$$\Delta^{\mathcal{A}}(\text{NI2}_r^+, R) \leq \frac{15q^2l^4}{2^{2n}}.$$

In the experiment where $\mathcal{A}$ interacts with NI2$_r^+$, let C_i denotes the event that during the first i queries, the inputs to r_3 i.e (Σ, Θ) for any two distinct queries $M^{(j)}$ and $M^{(k)}$ are also distinct. That means $(\Sigma^{(j)}, \Theta^{(j)}) \neq (\Sigma^{(k)}, \Theta^{(k)}),\ \forall 1 \leq j, k \leq i$. Therefore, as long as the monotone condition [16] $C = C_0, C_1, \ldots$ remains satisfied, the distribution of the responses of NI2$_r^+$ to distinct queries will be exactly identical to the distribution of the outputs of r_3 on distinct inputs and thus to independent uniform random values. In other words, we have

$$\text{NI2}_r^+ | C \equiv R.$$

Thus, using Lemma 1 in [10] we have, $\Delta^{\mathcal{A}}(\text{NI2}_r^+, R)$ is upper-bounded by the probability that a distinguisher $\mathcal{A}$ issuing q queries to NI2$_r^+$ makes the monotone condition C fail. This probability is denoted by $\Pr_{\mathcal{A}}[\text{NI2}_r^+; \overline{C}]$. Thus,

$$\Delta^{\mathcal{A}}(\text{NI2}_r^+, R) \leq \Pr_{\mathcal{A}}[\text{NI2}_r^+; \overline{C}]. \tag{6}$$

Now we explain how to construct a non-adaptive PRF adversary $\mathcal{A}_{na}$ from the above adaptive PRF adversary $\mathcal{A}$.

Construction of Non-adaptive PRF Adversary. Let $\mathcal{A}_{na}$ be the non adaptive PRF adversary that we want to construct from the adaptive PRF adversary $\mathcal{A}$. $\mathcal{A}_{na}$ will simulate the adaptive PRF adversary $\mathcal{A}$ in the following way. At the time of i^{th} query, $M^{(i)}$, where $1 \leq i \leq q$, asked by adversary $\mathcal{A}$, $\mathcal{A}_{na}$ will return random string in response of i^{th} query to $\mathcal{A}$. After all the q queries are over, $\mathcal{A}_{na}$ will (non-adaptively) ask all the queries that $\mathcal{A}$ asked during simulated interaction.

Therefore, we have the following

$$\Pr_{\mathcal{A}}[\mathrm{NI2}_r^+; \overline{C}] = \Pr_{\mathcal{A}_{na}}[\mathrm{NI2}_r^+; \overline{C}]. \tag{7}$$

The maximum probability over all such non-adaptive distinguishers $\mathcal{A}_{na}$ is given by

$$\Pr[\mathrm{NI2}_r^+; \overline{C}] = \max_{\mathcal{A}_{na}} \Pr_{\mathcal{A}_{na}}[\mathrm{NI2}_r^+; \overline{C}] \tag{8}$$

With respect to the $\mathrm{NI2}_r^+$ construction, let $Coll(l)$ denotes the probability that for random choice of the compression function f_1 and f_2, results in a collision in Σ and Θ maximized over the choice of two distinct inputs $M^{(i)}, M^{(j)}$, each of which is at most l blocks long.

More formally, for $f_1, f_2 \xleftarrow{\$} Func\left(\{0,1\}^n \times \{0,1\}^n \times \{0,1\}^b \to \{0,1\}^n\right)$ we define,

$$Coll(l) := \max_{M^{(i)} \neq M^{(j)} || |M^{(i)}|| , ||M^{(j)}|| \leq l} \Pr^{f_1,f_2}[\left(\Sigma^{(i)}, \Theta^{(i)}\right) = \left(\Sigma^{(j)}, \Theta^{(j)}\right)]$$

Note that, $(\Sigma^{(i)}, \Theta^{(i)}) = (\Sigma^{(j)}, \Theta^{(j)})$ implies $\Sigma^{(i)} = \Sigma^{(j)}$ and $\Theta^{(i)} = \Theta^{(j)}$. Therefore, to bound the probability of occurrence of a collision in the input of r_3 necessarily implies to bound the probability of occurrence of a collision in Σ and a collision in Θ. That means

$$\Pr^{f_1,f_2}[\left(\Sigma^{(i)}, \Theta^{(i)}\right) = \left(\Sigma^{(j)}, \Theta^{(j)}\right)] = \Pr^{f_1,f_2}[\Sigma^{(i)} = \Sigma^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)}] \tag{9}$$

Note that, $\mathcal{A}_{na}$ violates the monotone condition C only when the collision occurs at the input of r_3. Therefore from Equation (6), (7) and (8), and using union bound we obtain,

$$\Delta^{\mathcal{A}}(\mathrm{NI2}_r^+, R) \leq \Pr[\mathrm{NI2}_r^+; \overline{C}] \leq \frac{q^2}{2} Coll(l). \tag{10}$$

In Lemma 5 of Section 6.1, we show that $Coll(l) \leq \frac{30l^4}{2^{2n}}$. Therefore, plug-in the bound of $Coll(l)$ into Equation (10), we get the result. □

6.1 Computation of $Coll(l)$

Recall that, $Coll(l)$ was defined as $\Pr[\Sigma^{(i)} = \Sigma^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)}]$ maximized over the choice of pair of distinct inputs $M^{(i)}$ and $M^{(j)}$, each of length at most l blocks. Therefore, to establish the bound on $Coll(l)$, we derive the bound on $\Pr[\Sigma^{(i)} = \Sigma^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)}]$

Lemma 5. *Given two fixed distinct messages $M^{(i)}, M^{(j)}$, each of length is at most l blocks. Then*

$$\Pr[\Sigma^{(i)} = \Sigma^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)}] \leq \frac{30l^4}{2^{2n}}.$$

Proof. Let $Z^{(i)} = Y_{l_i}^{(i)}$ denote the input to the function r_2 for message $M^{(i)}$(refer to Fig.3.2). Similarly, we set $Z^{(j)} = Y_{l_j}^{(j)}$. So, we have,

$$\begin{aligned}
&\Pr[\Sigma^{(i)} = \Sigma^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)}] \\
&= \Pr[\Sigma^{(i)} = \Sigma^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)} \wedge Z^{(i)} = Z^{(j)}] + \\
&\quad \Pr[\Sigma^{(i)} = \Sigma^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)} \wedge Z^{(i)} \neq Z^{(j)}] \qquad (11) \\
&\leq \Pr[Z^{(i)} = Z^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)}] + \\
&\quad \Pr[\Sigma^{(i)} = \Sigma^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)} | Z^{(i)} \neq Z^{(j)}] \cdot \Pr[Z^{(i)} \neq Z^{(j)}] \qquad (12) \\
&\leq \Pr[Z^{(i)} = Z^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)}] + \\
&\quad \left(\Pr[\Sigma^{(i)} = \Sigma^{(j)} | Z^{(i)} \neq Z^{(j)}] \cdot \Pr[\Theta^{(i)} = \Theta^{(j)} | Z^{(i)} \neq Z^{(j)}]\right) \cdot \Pr[Z^{(i)} \neq Z^{(j)}]. \qquad (13)
\end{aligned}$$

Since the event $Z^{(i)} = Z^{(j)}$ is a subset of the event $\Sigma^{(i)} = \Sigma^{(j)}$, the first term of Equation (11) is equal to $\Pr[Z^{(i)} = Z^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)}]$. Further, the two events $\Sigma^{(i)} = \Sigma^{(j)}$ and $\Theta^{(i)} = \Theta^{(j)}$ are independent, conditioned on the event that there is no collision in the input of r_2. Therefore, the second term of Equation (12) is equal to $\Pr[\Sigma^{(i)} = \Sigma^{(j)} | Z^{(i)} \neq Z^{(j)}] \cdot \Pr[\Theta^{(i)} = \Theta^{(j)} | Z^{(i)} \neq Z^{(j)}]$.

According to Claim 3 we have, $\Pr[Z^{(i)} = Z^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)}] \leq \frac{ld'(l)}{2^{2n}} + \frac{8l^4}{2^{2n}}$ and according to Claim 4 we have, $\Pr[\Theta^{(i)} = \Theta^{(j)} | Z^{(i)} \neq Z^{(j)}] \leq \frac{4l^2+1}{2^n} + \frac{16l^4}{2^{2n}}$. Moreover, it is easy to see that, $\Pr[\Sigma^{(i)} = \Sigma^{(j)} | Z^{(i)} \neq Z^{(j)}] \leq \frac{1}{2^n}$, collision probability of a random function. Therefore,

$$\begin{aligned}
\Pr[\Sigma^{(i)} = \Sigma^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)}] &\leq \frac{ld'(l)}{2^{2n}} + \frac{8l^4}{2^{2n}} + \left[\frac{1}{2^n}\left(\frac{4l^2+1}{2^n} + \frac{16l^4}{2^{2n}}\right)(1-\epsilon)\right] \\
&\leq \frac{ld'(l)}{2^{2n}} + \frac{8l^4}{2^{2n}} + \left[\frac{4l^2+1}{2^{2n}} + \frac{16l^4}{2^{3n}}\right] \\
&\leq \frac{30l^4}{2^{2n}},
\end{aligned}$$

where $\epsilon = Pr[Z^{(i)} = Z^{(j)}]$, i.e., collision probability of $Casc^{r_1}$. In other words,

$$\epsilon = \Pr[Casc^{r_1}(M^i) = Casc^{r_1}(M^j)].$$

□

In the next two sections, we state and prove the two claims above.

7 Details of the Proof of Claim 3

Claim 3 *Fix two distinct messages $M^{(i)}, M^{(j)}$ each of length at most l blocks. Then,*

$$\Pr[Z^{(i)} = Z^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)}] \leq \frac{ld'(l)}{2^{2n}} + \frac{8l^4}{2^{2n}},$$

where $Z^{(i)} = Y^{(i)}_{l_i}, Z^{(j)} = Y^{(j)}_{l_j}$, and l_i, l_j are the number of blocks of $M^{(i)}, M^{(j)}$ respectively.

Proof. We prove the claim using the structure graph. After fixing two messages $M^{(i)}$ and $M^{(j)}$ and choosing a function f uniformly at random from the set of all functions over $\{0,1\}^b \times \{0,1\}^n \to \{0,1\}^n$, we analyze the structure graph $G := G^f(M^{(i)}, M^{(j)})$. In particular, we analyze the probability of the event $Z^{(i)} = Z^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)}$ in view of number of collisions (say, NCOL) occurred in the corresponding structure graph G. Therefore, we have,

$$\begin{aligned} Pr[Z^{(i)} = Z^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)}] = Pr[Z^{(i)} = Z^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)} \wedge NCOL = 1] \\ + Pr[Z^{(i)} = Z^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)} \wedge NCOL \geq 2]. \end{aligned}$$

In Section 7.1, we show that

$$\Pr[Z^{(i)} = Z^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)} \wedge NCOL = 1] \leq \frac{ld'(l)}{2^{2n}}, \tag{14}$$

where $d'(l)$ is the maximum number of positive divisors of the integer l' from $[1, l]$.

When NCOL in the graph is at least 2, then using Proposition 2 we have,

$$\Pr[Z^{(i)} = Z^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)} \wedge NCOL \geq 2] \leq \Pr[NCOL \geq 2] \leq \frac{8l^4}{2^{2n}}. \tag{15}$$

Therefore, combining Equations (14) and (15), we get the result. □

Now the only part of the proof that remains is to prove Equation (14).

7.1 Proof of Equation (14)

We can write

$$\begin{aligned} &Pr[Z^{(i)} = Z^{(j)} \wedge \Theta^{(i)} = \Theta^{(j)} \wedge NCOL = 1] \\ &= Pr[Z^{(i)} = Z^{(j)} \wedge NCOL = 1] \cdot Pr[\Theta^{(i)} = \Theta^{(j)} \mid Z^{(i)} = Z^{(j)} \wedge NCOL = 1]. \end{aligned} \tag{16}$$

In Equation (16), there are two probabilities that need to be computed. First, we compute $Pr[Z^{(i)} = Z^{(j)} \wedge NCOL = 1]$ by considering different structure graphs with $NCOL = 1$, corresponding to the construction NI2_r^+. Let G denote the set of all structure graphs with $NCOL = 1$ and $Z^{(i)} = Z^{(j)}$. Without loss of generality, let l_i and l_j be the lengths of the messages $M^{(i)}$ and $M^{(j)}$ respectively, with $l_i \geq l_j$. Let $G_1 \subset G$ be the set of all structure graphs such that the $M^{(i)}$-path does not contain any loop. The $G_2 = G \setminus G_1$ is the set of the remaining structure graphs. For the ease of understanding blue colored path represents the $M^{(i)}$ path and red colored path represents the $M^{(j)}$ path.

Analysis of G_1. If $M^{(j)}$ is a proper prefix of $M^{(i)}$, then $|G_1| = 0$, since in that case $Z^{(i)}$ won't be equal to $Z^{(j)}$. So without loss of generality, lets assume that $M^{(j)}$ is not a prefix of $M^{(i)}$. Suppose the first p blocks constitute the common prefix. Define $t^* = \min\{t > l_i + p : [[t]] \leq l_i\}$. Thus, the edge $([[t^* - 1]]', [[t^*]])$ in G creates the collision and from that point onwards, $M^{(j)}$ path will follow the rest of $M^{(i)}$ path which is nothing but the common suffix part of $M^{(i)}$ and $M^{(j)}$.

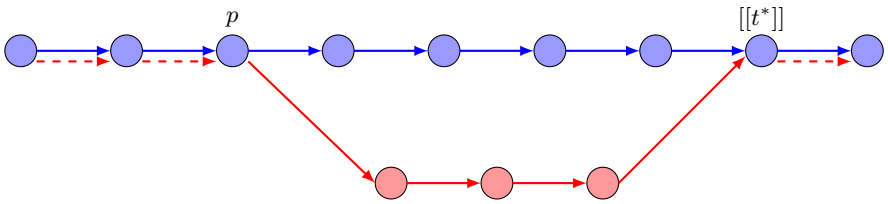

Fig. 7.1: Structure Graph of type G_1

The scenario is explained in Fig. 7.1. Since there are $\leq l$ choices for t^*, we have $|G_1| \leq l$.

Analysis of G_2. In graph G_2, $M^{(i)}$ path creates a collision by creating a self loop. We define $t^* = \min\{t : [[t]] \leq t\}$ and let $p^* = [[t^*]]$. Therefore, (t^*, p^*) denotes the collision in $M^{(i)}$ path. Now we can split $M^{(i)}$ into three mutual disjoint strings x, y, z such that $x := M_1^{(i)}||\ldots||M_{p^*}^{(i)}$, $y := M_{p^*+1}^{(i)}||\ldots M_{t^*}^{(i)}$ and some z chosen to be the smallest string so that we can write $M^{(i)} = x||y^a||z$ for some $a \geq 1$.

Note that to have $Z^{(i)} = Z^{(j)}$ and one collision has already been occurred in the loop, therefore, $M^{(j)}$-path must be a subpath of $M^{(i)}$-path and it cannot bifurcate from $M^{(i)}$ path and then collide with the last output block of $M^{(i)}$ as that would increases the number of collisions to 2. Thus, the $M^{(j)}$-path must be of the form

$x||y^b||z$, where $b < a$ (since $l_i > l_j$ in this case). Hence, the number of blocks in y, i.e., $t^* - p^*$, in the diagram must divide $l_i - l_j$. This scenario is explained in Fig. 7.2.

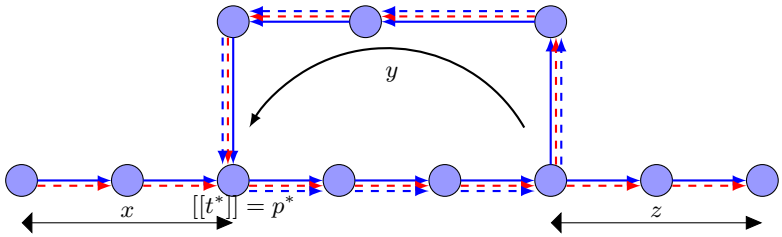

Fig. 7.2: Structure Graph of type G_2

There are at most l choices for such a t^* and $d'(l)$ choices for such a p^*. Hence, $|G_2| \leq ld'(l)$. In the special case, when $l_i = l_j$, then obviously, $|G_2| = 0$.

Therefore, considering G_1 and G_2 together, by Proposition 1, we have

$$\Pr[Z^{(i)} = Z^{(j)} \wedge NCOL = 1] \leq \frac{ld'(l)}{2^n}. \tag{17}$$

Now, we compute the second probability of Equation 16, i.e., $Pr[\Theta^{(i)} = \Theta^{(j)} \mid Z^{(i)} = Z^{(j)} \wedge NCOL = 1]$. Note that $\Theta^{(i)} = \Theta^{(j)}$ gives an equation of the form

$$2^{l_i}Y_1^{(i)} + 2^{l_i-1}Y_2^{(i)} + \cdots + 2Y_{l_i}^{(i)} = 2^{l_j}Y_1^{(j)} + 2^{l_i-1}Y_2^{(i)} + \cdots + 2Y_{l_j}^{(j)}. \tag{18}$$

Given the condition $Z^{(i)} = Z^{(j)}$, i.e., $Y_{l_i}^{(i)} = Y_{l_j}^{(j)}$, the above equation becomes

$$2^{l_i}Y_1^{(i)} + 2^{l_i-1}Y_2^{(i)} + \cdots + 2^2Y_{l_i-1}^{(i)} = 2^{l_j}Y_1^{(j)} + 2^{l_i-1}Y_2^{(i)} + \cdots + 2^2Y_{l_j-1}^{(j)}. \tag{19}$$

Now, for both the graphs G_1 and G_2, we will be able to find at least one Y variable belonging to the part between p and t^*, such that Equation (19) becomes non-trivial for such variable Y, giving a probability of $\frac{1}{2^n}$ for the second term of Equation (16). When this along with Equation (17) is plugged in Equation (16), the probability in Equation (16), i.e., in Equation (14),becomes bounded by $\frac{ld'(l)}{2^{2n}}$.

8 Details of the Proof of Claim 4

Claim 4 *Fix two distinct messages $M^{(i)}, M^{(j)}$ each of length at most l blocks. Then,*

$$\Pr[\Theta^{(i)} = \Theta^{(j)} | Z^{(i)} \neq Z^{(j)}] \leq \frac{4l^2 + 1}{2^n} + \frac{16l^4}{2^{2n}},$$

where $Z^{(i)} = Y_{l_i}^{(i)}, Z^{(j)} = Y_{l_j}^{(j)}$, l_i, l_j is the number of blocks of $M^{(i)}, M^{(j)}$ respectively.

Proof. It is to be noted that,

$$\begin{aligned}\Pr[\Theta^{(i)} = \Theta^{(j)} | Z^{(i)} \neq Z^{(j)}] &= \Pr[\Theta^{(i)} = \Theta^{(j)} \wedge NCOL = 0 \mid Z^{(i)} \neq Z^{(j)}] \\ &+ \Pr[\Theta^{(i)} = \Theta^{(j)} \wedge NCOL = 1 \mid Z^{(i)} \neq Z^{(j)}] \\ &+ \Pr[\Theta^{(i)} = \Theta^{(j)} \wedge NCOL \geq 2 \mid Z^{(i)} \neq Z^{(j)}]. \quad (20)\end{aligned}$$

Now,

$$\begin{aligned}&\Pr[\Theta^{(i)} = \Theta^{(j)} \wedge NCOL \geq 2 \mid Z^{(i)} \neq Z^{(j)}] \\ &= \frac{\Pr[\Theta^{(i)} = \Theta^{(j)} \wedge NCOL \geq 2 \wedge Z^{(i)} \neq Z^{(j)}]}{\Pr[Z^{(i)} \neq Z^{(j)}]} \\ &\leq \frac{\Pr[NCOL \geq 2]}{\Pr[Z^{(i)} \neq Z^{(j)}]} \\ &\leq \frac{8l^4}{2^{2n}(1-\epsilon)} \\ &\leq \frac{8l^4(1+2\epsilon)}{2^{2n}} \qquad (\text{Since } \epsilon < \frac{1}{2}) \\ &\leq \frac{16l^4}{2^{2n}}. \quad (21)\end{aligned}$$

Now, we need to bound $\Pr[\Theta^{(i)} = \Theta^{(j)} \wedge NCOL = 0 \mid Z^{(i)} \neq Z^{(j)}]$ and $\Pr[\Theta^{(i)} = \Theta^{(j)} \wedge NCOL = 1 \mid Z^{(i)} \neq Z^{(j)}]$ separately.

Again we consider two distinct messages $M^{(i)}$ and $M^{(j)}$ with lengths l_i and $l_j$respectively, with $l_i \geq l_j$. Since we are given the condition $Z^{(i)} \neq Z^{(j)}$, the structure graphs will have the common feature that the end-point $Y_{l_i}^{(i)}$ of $M^{(i)}$-path and the end-point $Y_{l_j}^{(j)}$ of $M^{(j)}$-path must be different, i.e., from Equation (18), we have $Y_{l_i}^{(i)} \oplus Y_{l_j}^{(j)} = c \neq 0$. Thus, Equation (18) becomes non-trivial, with probability $\frac{1}{2^n}$.

Now, we need to count the number of distinct structure graphs for each of the cases $NCOL = 0$ and $NCOL = 1$.

Clearly, when $NCOL = 0$, only such structure graph is possible, as shown in Fig. 8.1. Thus, we have

$$\Pr[\Theta^{(i)} = \Theta^{(j)} \wedge NCOL = 0 \mid Z^{(i)} \neq Z^{(j)}] \leq \frac{1}{2^n}. \quad (22)$$

Now, let us consider the case $NCOL = 1$. Let G be the set of all structure graphs with $NCOL = 1$ with $Z^{(i)} \neq Z^{(j)}$. Let $G_1 \subset G$ be the set of all structure graphs such that the $M^{(i)}$-path does not contain any loop. The $G_2 = G \setminus G_1$ is the set of remaining structure graphs.

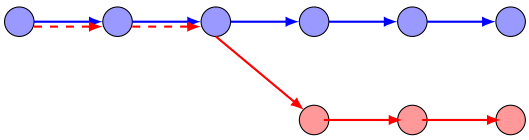

Fig. 8.1: Structure Graph of accident 0

Analysis of G_1. For G_1, the $M^{(j)}$ path can either intersect with $M^{(i)}$ exactly once or $M^{(j)}$ path does not intersect with $M^{(i)}$ but it creates a loop with itself. In the first case, $M^{(j)}$-path cannot have any loop as shown in Fig. 8.2 as that would increase the number of collision to 2, and in the second case, the $M^{(j)}$ path cannot intersect $M^{(i)}$-path at all as that would again increase the number of collision to 2 as shown in Fig. 8.3. In either case, the number of such graphs is at most l^2.

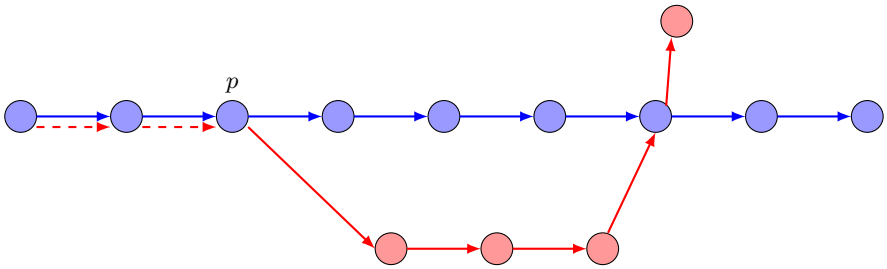

Fig. 8.2: Structure Graph of type G_1; $M^{(i)}$ path has no loop

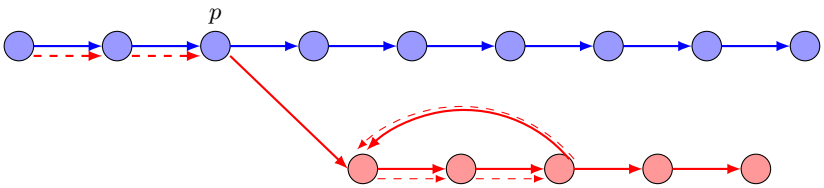

Fig. 8.3: Structure Graph of type G_1; $M^{(i)}$ path has no loop, $M^{(j)}$ path has loop

Analysis of G_2. For G_2, note that $M^{(i)}$ path contains a loop. Now the $M^{(j)}$ path may or may not intersects $M^{(i)}$ path. If it does, then it must follow the same loop as $M^{(i)}$ and then exit either from the loop or afterwards, as shown in Fig. 8.4. $M^{(j)}$ path may also bifurcate from $M^{(i)}$ path before the loop and then it should not intersect

with $M^{(i)}$ path again or it should not make any self loop with itself as both of the cases would increases the number of collision to 2. Note that $M^{(j)}$ path cannot intersect $M^{(i)}$ path before the loop as that would increase the number of collision to 2.

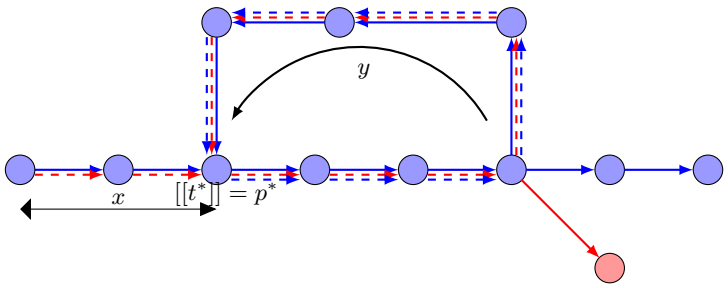

Fig. 8.4: Structure Graph of type G_2; $M^{(i)}, M^{(j)}$ both path contain a loop

If $M^{(j)}$ path does not intersect $M^{(i)}$ path, then $M^{(j)}$ path cannot make a loop with itself as that would increase the number of collision to 2. Therefore, again the case is similar to Fig. 8.3 where the blue colored path will then represent the $M^{(j)}$ path and red colored path will represent $M^{(i)}$ path. In either case, the number of such graphs is at most l^2.

Thus, for the above $4l^2$ graphs (combined G_1 and G_2),

$$\Pr[\Theta^{(i)} = \Theta^{(j)} \wedge NCOL = 1 \mid Z^{(i)} \neq Z^{(j)}] \leq \frac{4l^2}{2^n}. \tag{23}$$

Now, plugging in the probabilities from Equations (21), (22) and (23) into Equation (20), we get

$$\Pr[\Theta^{(i)} = \Theta^{(j)} | Z^{(i)} \neq Z^{(j)}] \leq \frac{1}{2^n} + \frac{4l^2}{2^n} + \frac{16l^4}{2^{2n}}.$$

□

9 Conclusion and Future Work

In this paper, we first show that the sum of two independent random permutations with restricted domain and range gives a security bound of beyond birthday. While we use this result in our subsequent proofs, this leave open the security of the sum of a single permutation with restricted domain and range set.

Next, we solve an open problem that the sum of two CBC MACs is a secure PRF with security bound beyond birthday. While ours is a better construction compared to Kan Yasuda's in terms of number of keys, a further research work in this direction is to reduce the number of key to 1 and achieves beyond birthday bound security.

Finally, we show a modified construction of NI2 MAC and prove its security to be beyond birthday. While we use we use an extra keyed function (f_{K_3}) in NI2$^+$, an interesting research problem would be to avoid the usage of this extra keyed function and achieves beyond birthday security.

References

1. Jee Hea An and Mihir Bellare. Constructing vil-macsfrom fil-macs: Message authentication under weakened assumptions. In Wiener [18], pages 252–269.
2. Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 1996.
3. Mihir Bellare, Oded Goldreich, and Hugo Krawczyk. Stateless evaluation of pseudorandom functions: Security beyond the birthday barrier. In Wiener [18], pages 270–287.
4. Mihir Bellare and Russell Impagliazzo. A tool for obtaining tighter security analyses of pseudorandom function based constructions, with applications to PRP to PRF conversion. *IACR Cryptology ePrint Archive*, 1999:24, 1999.
5. Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of cipher block chaining. In Yvo Desmedt, editor, *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings*, volume 839 of *Lecture Notes in Computer Science*, pages 341–358. Springer, 1994.
6. Mihir Bellare, Krzysztof Pietrzak, and Phillip Rogaway. Improved security analyses for CBC macs. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 527–545. Springer, 2005.
7. Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer, 2006.
8. John Black, Shai Halevi, Hugo Krawczyk, Ted Krovetz, and Phillip Rogaway. UMAC: fast and secure message authentication. In Wiener [18], pages 216–233.
9. John Black and Phillip Rogaway. A block-cipher mode of operation for parallelizable message authentication. In Knudsen [14], pages 384–397.
10. Peter Gazi, Krzysztof Pietrzak, and Michal Rybár. The exact prf-security of NMAC and HMAC. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 113–130. Springer, 2014.
11. Tetsu Iwata and Kaoru Kurosawa. OMAC: one-key CBC MAC. In Johansson [12], pages 129–153.
12. Thomas Johansson, editor. *Fast Software Encryption, 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003, Revised Papers*, volume 2887 of *Lecture Notes in Computer Science*. Springer, 2003.

13. Antoine Joux, Guillaume Poupard, and Jacques Stern. New attacks against standardized macs. In Johansson [12], pages 170–181.
14. Lars R. Knudsen, editor. *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*. Springer, 2002.
15. Stefan Lucks. The sum of prps is a secure PRF. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 470–484. Springer, 2000.
16. Ueli M. Maurer. Indistinguishability of random systems. In Knudsen [14], pages 110–132.
17. Erez Petrank and Charles Rackoff. CBC MAC for real-time data sources. *J. Cryptology*, 13(3):315–338, 2000.
18. Michael J. Wiener, editor. *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*. Springer, 1999.
19. Kan Yasuda. The sum of CBC macs is a secure PRF. In Josef Pieprzyk, editor, *Topics in Cryptology - CT-RSA 2010, The Cryptographers' Track at the RSA Conference 2010, San Francisco, CA, USA, March 1-5, 2010. Proceedings*, volume 5985 of *Lecture Notes in Computer Science*, pages 366–381. Springer, 2010.