BBB Secure Nonce Based MAC Using Public Permutations

Avijit Dutta and Mridul Nandi
Indian Statistical Institute, Kolkata.
avirocks.dutta13@gmail.com, mridul.nandi@gmail.com

Abstract. In the recent trend of CAESAR competition and NIST lightweight competition, cryptographic community have witnessed the submissions of several cryptographic schemes that are build on public random permutations. Recently, in CRYPTO 2019, Chen et al. have initiated an interesting research direction in designing beyond birthday bound PRFs from public random permutations and they proposed two instances of such PRFs. In this work, we extend this research direction by proposing a nonce-based MAC build from public random permutations. We show that our proposed MAC achieves $2^{n/3}$ bit security (with respect to the state size of the permutation) and the bound is essentially tight. Moreover, the security of the MAC degrades gracefully with the repetition of the nonce.

Keywords: Faulty Nonce, Mirror Theory, Public Permutation, Expectation Method.

1 Introduction

Nonce-Based MAC. Message Authentication Code (or in short MAC) is an important cryptographic primitive to authenticate any digital message or packet transmitted over an insecure communication channel. When a sender wants to send a message $m$, she computes a MAC function with input $m$, the shared secret key $k$, and possibly an auxiliary input variable $\nu$ (called nonce), and obtains a tag $t$. Then she sends $(\nu, m, t)$ to the receiver. Upon receiving, receiver verifies the authenticity of $(\nu, m, t)$ by computing the MAC using $(\nu, m, k)$ and checks whether the computed tag $t'$ matches with $t$.

Wegman-Carter (WC) MAC [25] is the first example of a nonce-based MAC which masks the hash value of the message with an encrypted nonce to generate the tag. WC MAC gives optimal security when the nonce is unique for every authenticated messages. However, its security is compromised if the nonce repeats even once. Wegman-Cater MAC, when instantiated with a polynomial hash, then the repetition of the nonce reveals the hash key of the polynomial hash. However, maintaining the uniqueness of the nonce for every authenticated messages is a challenging task in practical contexts. For example, it is difficult to maintain the uniqueness of the nonce while implementing the cipher in a
stateless device or in cases where the nonce is chosen randomly from a small set. The nonce may also accidentally repeats due to a faulty implementation of the cipher or due to the fault occurred by resetting of the nonce itself [4]. Therefore, the guard from the nonce repetition attack is much desired from a nonce-based MAC.

As a remedy of this, *Encrypted Wegman-Carter-Shoup* (EWCS) [11] MAC was proposed that guarantees the security even when the nonce repeats. But its security is limited only up to the birthday bound even when nonce is unique. To this end, *Encrypted Wegman-Carter with Davies-Meyer* [11] (or EWCDM) and *Decrypted Wegman-Carter with Davies-Meyer* [13] (or DWCDM) have been proposed that gives beyond the birthday bound security when nonce is unique \(^1\) and birthday bound security when nonce repeats \(^2\). However, the security of both these constructions fall to the birthday bound with a single repetition of the nonce, i.e., if the nonce ever repeats accidentally, security of both the constructions immediately drops to the birthday bound.

**Nonce Based Enhanced Hash-then-Mask.** In FSE 2010 [21], Minematsu proposed \(\text{EHtM}\), a beyond birthday bound secure probabilistic MAC. It is build upon two independent \(n\)-bit keyed functions \(F_{k_1}\) and \(F_{k_2}\) and an \(n\)-bit axu hash function \(H_{k_h}\), defined as follows:

\[
\text{EHtM}(m) \triangleq (r \leftarrow \{0, 1\}^n, F_{k_1}(r) \oplus F_{k_2}(r \oplus H_{k_h}(m))).
\]

This construction has been further analyzed in [15] for improving its security bound. In Eurocrypt 2019, Dutta et al. [16] proposed a nonce-based variant of \(\text{EHtM}\), called \(n\text{EHtM}\) MAC, where the random salt \(r\) is replaced by an \(n-1\) bit nonce value \(\nu\) and an \(n\)-bit block cipher \(E_k\) is used as an internal primitive instead of two independent \(n\)-bit keyed functions. Schematic diagram of \(n\text{EHtM}\) is shown in Fig. 1.1. Similar to EWCDM and DWCDM, \(n\text{EHtM}\) gives beyond the (birthday bound) security in nonce-respecting (resp. nonce misuse) setting. But, unlike these two constructions, security of \(n\text{EHtM}\) MAC degrades gracefully with the repetition of the nonce. In other words, security of \(n\text{EHtM}\) remains beyond the birthday bound with a single repetition of the nonce (which is not true for EWCDM and DWCDM). That is, one can get adequate security from \(n\text{EHtM}\) if the repetition of the nonce occurs in a controlled way, a feature which is not present in EWCDM or DWCDM. This phenomena is formally captured by a notion, called **faulty nonce model** [16]. Informally, it says that a nonce is faulty if it appears in a previous signing query. It has been stated in [16] that faulty nonce model is a weaker notion than *multicollision* of nonces – a natural and a popular metric to measure the misuses of nonce. Under the notion of faulty nonce model, Dutta et al. have shown that \(n\text{EHtM}\) is secured roughly upto \(2^{2n/3}\) queries.

We would like to mention here that this construction was also analyzed by Moch and List [22] in parallel to [16] in the name of HPxNP, where two independent \(n\)-bit block ciphers have been used (as they did not use the domain separation

\(^1\) We call this notion *nonce-respecting* setting

\(^2\) We call this notion *nonce-misuse* setting
technique). However, Moch and List analyzed its security under the condition of the uniqueness of the nonce, whereas Dutta et al. [16] proved its graceful security with respect to the repetition of the nonce.

1.1 Permutation Based Cryptography

All the above discussed nonce-based MACs are build on block ciphers as their underlying primitives and even stronger, these primitives are evaluated only in the forward direction. As most of the block ciphers are designed to be efficient in both the forward and the inverse direction, block ciphers are over-hyped primitives for such purpose [10]. On the other extreme, cryptographic permutations are particularly designed with the motive to be fast in the forward direction, but not necessarily in the inverse direction. Examples of such permutation includes Keccak [2], Gimli [1], SPONGENT [5]. Moreover, in most of the cases evaluating an unkeyed public permutation is faster than evaluating a keyed block cipher, as the latter involves in evaluating the underlying key scheduling algorithm each time the block cipher is invoked in the design. With the advancement of public permutation-based designs and the efficiencies of evaluating it in the forward direction, numerous public permutation-based inverse-free hash and authenticated encryption designs have been proposed. The use of cryptographic permutation gained the momentum during SHA-3 competition [24]. Furthermore, the selection of the permutation-based Keccak sponge function as the SHA-3 standard has given a high level of confidence on using this primitive in the community. Today, permutation-based sponge construction has become a successful and a full-fledged alternative to the block cipher-based modes. In fact, in the first round of the ongoing NIST light-weight competition [23], 24 out of 57 submissions are based on cryptographic permutations, and out of 24, 16 permutation based proposals have been qualified for the second round. This statistics, beyond any doubt, clearly depicts the wide adoption of permutation based designs [7, 1, 3, 8, 12, 14] in the community. In another direction, a long line of research work has been carried out in the study of designing block ciphers and tweakable block ciphers out of public random permutations. Even Mansour (EM) [17] and Iterated Even Mansour (IEM) cipher [6] are the notable approaches in this direction.

Nonce-based MAC build from Public Permutations. Nonce-based MACs using public permutations are mostly designed with sponge type of constructions. But the drawback of such designs are: (i) they do not use the full size of the permutation for guarantting security and (ii) they attain only the birthday bound security in the size of its capacity $c$, i.e., $c/2$ bit security (except Bettle [7], whose security bound is roughly the size of its capacity). Now, it is an admissible fact that the sponge type designs, which offer $c/2$-bit security, are good in practice when they are instantiated with large size permutations such as Keccak [2], whose state size is 1600 bits. But such large size permutations are not suitable for use in resource constrained environment. In such scenario, instead of using such large size permutations, one aims to use light-weight permutations such as SPONGENT [5] and PHOTON [18], whose state size go as
low as 88 and 100 bits respectively. If we use these light-weight permutations as underlying primitives in birthday bound secure sponge type constructions, then it practically offers inadequate security. As a result, sponge type constructions instantiated with light-weight permutations are not suitable for deploying in resource constrained environment. Thus, it is natural to ask

*Can we design a public permutation-based nonce-based MAC that gives an adequate security when instantiated with light-weight permutation?*

This question hinted us to think of designing a MAC whose security depends on the entire size of the underlying permutation (unlike sponge type constructions whose security depends on only a part of the entire size of the underlying permutation) and the security must cross the birthday barrier. Coming up with such a design is the goal of this paper. In this direction, Chen et al. [10] have shown two instances of public permutation-based pseudo random functions that give beyond the birthday bound security with respect to the size of the permutation. We extend this line of research work by designing a public permutation-based nonce-based MAC that gives beyond the birthday bound security with respect to the size of the permutation.

**OUR CONTRIBUTION.** The sole contribution of this paper is to design a beyond birthday bound secure nonce-based MAC using public random permutations. To this end we propose $nEHTM_p$, a nonce based MAC desiged using public permutations. As depicted in Fig. 1.1, our construction structurally resembles to the $nEHTM$ MAC [16] where we replace its block cipher with a public random permutation and an appropriate masking of the key.

![Diagram](image)

**Fig. 1.1.** (Left): $nEHTM$ MAC based on block cipher $E_k$; (Middle): $nEHTM_p$ MAC based on single public random permutation $\pi$; (Right): 2-round iterated even mansour cipher.

Note that, by instantiating the underlying block cipher of $nEHTM$ MAC with 2-round iterated Even-Mansour cipher (as shown in Fig 1.1), one can easily make
the public permutation variant of nEHtM MAC, which becomes secure beyond
the birthday bound (in faulty nonce model). However such transformation
requires 4 permutation calls, 7 xor operations and one hash evaluation. Compared
to this, nEHtM_p requires only 2 permutation calls, 3 xor operations and one
hash evaluation. We have shown that nEHtM_p is secured roughly up to $2^{2n/3}$
queries in the nonce-respecting setting. Moreover, this security bound degrades
in a graceful manner under the faulty nonce model [16]. We show the unforge-
ability of this construction through an extended distinguishing game and apply
the expectation method to bound its distinguishing advantage. We also show
that our proven security bound is tight by giving a matching attack on it with
roughly $2^{2n/3}$ query complexity and $2^{2n-4}$ time complexity.

2 Preliminaries

General Notations: For $n \in \mathbb{N}$, we denote the set of all binary strings of
length $n$ and the set of all binary strings of finite arbitrary length by \{0, 1\}^n
and \{0, 1\}^* respectively. We often refer the elements of \{0, 1\}^n as block. For an
n-bit binary string $x = (x_{n-1} \ldots x_0)$, msb$(x)$ denotes the first bit of $x$ in left
to right ordering, i.e. msb$(x) = x_{n-1}$. Moreover, chop_{msb}(x) \triangleq (x_{n-2} \ldots x_0),
i.e., chop_{msb}(x) returns the string $x$ by dropping just its msb. For any element
$x \in \{0, 1\}^*, |x|$ denotes the number of bits in $x$ and for $x, y \in \{0, 1\}^*$, $x \parallel y$
denotes the concatenation of $x$ followed by $y$. We denote the bitwise xor operation of
$x, y \in \{0, 1\}^n$ by $x \oplus y$. We parse $x \in \{0, 1\}^*$ as $x = x_1 \parallel x_2 \parallel \ldots \parallel x_l$
where for each $i = 1, \ldots, l-1$, $x_i$ is a block and $1 \leq |x_i| \leq n$. For a sequence of
elements $(x^1, x^2, \ldots, x^r) \in \{0, 1\}^*$, $x^a_i$ denotes the $a$-th block of $i$-th element $x^i$. For a value $s$, we denote by $t \leftarrow s$ the assignment of $s$ to variable $t$. For any
natural number $j \in \mathbb{N}$, (j)$_s$ denotes the $s$ bit binary representation of integer
$j$. For $i \in \{0, 1\}^n$, left$k(i)$ represents the leftmost $k$ bits of $i$. Similarly, right$k(i)$
represents the rightmost $k$ bits of $i$. For any finite set $\mathcal{X}$, $\mathcal{X} \leftarrow \mathcal{X}$ denotes that
$\mathcal{X}$ is sampled uniformly at random from $\mathcal{X}$ and $X_1, \ldots, X_s \leftarrow \mathcal{X}$ denotes that
$X_i$’s are sampled uniformly and independently from $\mathcal{X}$. $\mathcal{F}(\mathcal{X})$ denotes the set of
all functions from $\mathcal{X}$ to $\{0, 1\}^n$. We often write $\mathcal{F}(\mathcal{n})$ when the domain is clear
from the context. We denote the set of all permutations over $\{0, 1\}^n$ by $\mathcal{P}(\mathcal{n})$.
For integers $1 \leq b \leq a$, (a)$_b$ denotes the product $a(a-1) \ldots (a-b+1)$, where
(a)$_0 = 1$ by convention and for $q \in \mathbb{N}$, $[q]$ refers to the set $\{1, \ldots, q\}$.

2.1 Public Permutation Based Nonce Based MAC

Let $F : \mathcal{K} \times \mathcal{N} \times \mathcal{M} \rightarrow \mathcal{T}$ be a keyed function where \mathcal{K}, \mathcal{N}, \mathcal{M}
and \mathcal{T} are the key space, nonce space, message space and the tag space respectively.
We assume that $F$ makes internal calls to the public random permutations $\pi = (\pi_1, \ldots, \pi_d)$ for
d \geq 1, where all of the $d$ permutations are independent and uniformly sampled

$^3$ time complexity does not refer to the evaluation of permutations, but only refers to the
time required to find a suitable matching pair.
To obtain an upper bound for $K$ of the nonce based MAC verification algorithm returns 1. The advantage of $A$ of its verification queries (not obtained through a previous signing query), the $p$ and $d$ with the signing and the verification queries. Moreover, the primitive queries are interleaved respecting (resp. nonce misuse) if $\eta$ that it makes at most $\eta$ faulty signing queries out of $q_m$ signing, $q_v$ verification and $p$ primitive queries with running time of $A$ at most $t$. $A$ is said to be nonce respecting (resp. nonce misuse) if $\eta = 0$ (resp. $\eta \geq 1$). However, $A$ may repeats nonces in its verification queries. Moreover, the primitive queries are interleaved with the signing and the verification queries, $A$ is said to forge $I$ if for any of its verification queries (not obtained through a previous signing query), the verification algorithm returns 1. The advantage of $A$ against the unforgeability of the nonce based MAC $I$ is defined as

$$\text{Adv}^{n\text{MAC}}_I(A) \overset{\Delta}{=} \Pr \left[ A[I,\text{Sign},I,\text{Ver}]_k,\pi,\pi^{-1} \Rightarrow \text{forge} \right],$$

where the randomness is defined over $k \leftarrow \mathcal{K}$, $\pi_1, \ldots, \pi_d \leftarrow \mathcal{P}(n)$ and the randomness of the adversary (if any). We write

$$\text{Adv}^{n\text{MAC}}_I(\eta,q_m,q_v,p,t) \overset{\Delta}{=} \max_{A} \text{Adv}^{n\text{MAC}}_I(A),$$

where the maximum is taken over all $(\eta,q_m,q_v,p,t)$-adversaries $A$. In this paper, we skip the time parameter of the adversary as we will assume throughout the paper that the adversary is computationally unbounded. This will render us to assume that the adversary is deterministic.

**Upper bound on $\text{Adv}^{n\text{MAC}}_I(A)$ ([15]).** To obtain an upper bound for $\text{Adv}^{n\text{MAC}}_I(A)$, we consider a random oracle $\mathcal{O}$ that samples the tag $t$ independently and uniformly at random from $\{0,1\}^n$ for every nonce message pair $(\nu,m)$ and the $\text{Rej}$ oracle always returns $\bot$ for any $(\nu,m,t)$. Then, $\text{Adv}^{n\text{MAC}}_I(A)$ is upper bounded by

$$\max_{A} \left| \Pr \left[ A[I,\text{Sign},I,\text{Ver}]_k,\pi,\pi^{-1} \Rightarrow 1 \right] - \Pr \left[ A[\mathcal{O},\text{Rej}]_k,\pi,\pi^{-1} \Rightarrow 1 \right] \right|.$$

where $A[\mathcal{O}] \Rightarrow 1$ denotes that adversary $A$ outputs 1 after interacting with its oracle $\mathcal{O}$ (which could be a multiple of oracles).

### 2.2 Almost Xor Universal and Almost Regular Hash Function

Let $\mathcal{K}_h$ and $\mathcal{X}$ be two non-empty finite sets and $H$ be a keyed function $H : \mathcal{K}_h \times \mathcal{X} \rightarrow \{0,1\}^n$. Then, $H$ is said to be an $\epsilon_{\text{axu}}$-almost xor universal (axu) hash function $H : \mathcal{K}_h \times \mathcal{X} \rightarrow \{0,1\}^n$. Then, $H$ is said to be an $\epsilon_{\text{axu}}$-almost xor universal (axu)
function, if for any distinct \(x, x' \in \mathcal{X}\) and for any \(\Delta \in \{0, 1\}^n\),
\[
\Pr[K_h \leftarrow K_h : H_{K_h}(x) \oplus H_{K_h}(x') = \Delta] \leq \varepsilon_{\text{acc}}.
\]
Moreover, \(H\) is said to be an \(\varepsilon_{\text{reg}}\)-almost regular (ar) hash function, if for any \(x \in \mathcal{X}\) and for any \(\Delta \in \{0, 1\}^n\),
\[
\Pr[K_h \leftarrow K_h : H_{K_h}(x) = \Delta] \leq \varepsilon_{\text{reg}}.
\]

2.3 Expectation Method

The Expectation Method of Hoang and Tessaro [19] was used to derive a tight multi-user security bound of the key-alternating cipher. This technique has subsequently been used in [20, 16]. Let \(A\) be a computationally unbounded deterministic distinguisher that interacts with either of the two worlds: \(O_{re}\) or \(O_{id}\), where these oracles are possibly randomized stateful systems. After the interaction, \(A\) returns a single bit. This interaction between \(A\) and the system results in an ordered sequence of queries and responses which is summarized in \(\tau = ((x_1, y_1), (x_2, y_2), \ldots, (x_q, y_q))\), called a transcript, where \(x_i\) is the \(i\)-th query of \(A\) and \(y_i\) is the corresponding response of the system to which \(A\) interacts with.

Let \(D_{re}\) (resp. \(D_{id}\)) be the random variable that takes a transcript resulting from the interaction between \(A\) and \(O_{re}\) (resp. \(O_{id}\)). A transcript \(\tau\) is said to be attainable if \(\Pr[D_{id} = \tau] > 0\). Let \(\Theta\) denotes the set of all attainable transcripts.

Let \(\Phi : \Theta \rightarrow [0, \infty)\) be a non-negative function which maps any attainable transcript to a non-negative real value. Suppose there is a set of good transcripts \(\text{GoodT} \subseteq \Theta\) such that for any \(\tau \in \text{GoodT}\),
\[
\frac{\Pr[D_{re} = \tau]}{\Pr[D_{id} = \tau]} \geq 1 - \Phi(\tau). \tag{2}
\]

Then, the statistical distance between \(D_{re}\) and \(D_{id}\) can be bounded as
\[
\Delta(D_{re}, D_{id}) \leq \mathbb{E}[\Phi(D_{id})] + \Pr[D_{id} \in \text{BadT}], \tag{3}
\]
where \(\text{BadT} \triangleq \Theta \setminus \text{GoodT}\) is the set of all bad transcripts. In other words, the advantage of \(A\) in distinguishing \(O_{re}\) from \(O_{id}\) is bounded by \(\mathbb{E}[\Phi(D_{id})] + \Pr[D_{id} \in \text{BadT}]\). In the rest of the paper, we write \(\Theta, \text{GoodT}\) and \(\text{BadT}\) to denote the set of attainable, set of good and set of bad transcripts respectively.

2.4 Sum-Capture Lemma

We use the sum capture lemma by Chen et al. [9]. Informally, the result states that for a random subset \(S\) of \(\{0, 1\}^n\) of size \(q\) and for any two arbitrary subsets \(\mathcal{A}\) and \(\mathcal{B}\) of \(\{0, 1\}^n\), the size of the set \(\{(s, a, b) \in S \times \mathcal{A} \times \mathcal{B} : s = a \oplus b\}\) is at most \(q|\mathcal{A}||\mathcal{B}|2^n\), except with negligible probability. In our setting, \(S\) is the set of tag values \(t_i\), which are sampled with replacement from \(\{0, 1\}^n\).
Lemma 1 (Sum-Capture Lemma). Let $n, q \in \mathbb{N}$ such that $9n \leq q \leq 2^{n-1}$. Let $S = \{t_1, \ldots, t_q\} \subseteq \{0,1\}^n$ such that $t_i$’s are with replacement sample of $\{0,1\}^n$. Then, for any two subsets $A$ and $B$ of $\{0,1\}^n$, we have

$$\Pr[|\{(t, a, b) \in S \times A \times B : t = a \oplus b\}| \geq q|A||B|/2^n + 3\sqrt{nq|A||B|} \leq \frac{2}{2^n}, \quad (4)$$

where the randomness is defined over the set $S$.

3 Solving a System of Affine (Non-)Equations

In this section, we present a lower bound on the number of solutions of a system of bi-variate affine equations and bi-variate affine non-equations over a finite number of unknown variables which are without replacement samples of $\{0,1\}^n$. This result will become handy for analysing the security of our proposed construction.

**Initial Setup:** Consider an undirected edge-labelled acyclic graph $G = (V, E, \mathcal{F}, \mathcal{F}', \mathcal{L})$ with edge labelling function $\mathcal{L} : \mathcal{F} \cup \mathcal{F}' \to \{0,1\}^n$, where the edge set is partitioned into two disjoint sets $\mathcal{F}$ and $\mathcal{F}'$. For an edge $(y_i, y_j) \in \mathcal{F}$, we write $\mathcal{L}(y_i, y_j) = \lambda_{ij}$ (and so $\lambda_{ij} = \lambda_{ji}$) and $\mathcal{L}(y_i, y_j) = \lambda'_{ij}$ for all $(y_i, y_j) \in \mathcal{F}'$. Let $G^u = (V, \mathcal{F}, \mathcal{L}_{\mathcal{F}^u})$ denotes the subgraph of $G$, where $\mathcal{L}_{\mathcal{F}^u}$ is the function $\mathcal{L}$ restricted over the set $\mathcal{F}$. We say $G$ is **good** if it satisfies the following two conditions: (i) for all paths $P_{st}$ in graph $G^u$, $\mathcal{L}(P_{st}) \neq 0$, where $\mathcal{L}(P_{st}) \overset{\Delta}{=} \sum_{e \in P_{st}} \mathcal{L}(e) = y_s \oplus y_t$ and $P_{st}$ is a path of $G^u$ between vertex $s$ and $t$ and (ii) for all cycles $C$ in $G$ such that the edge set of $C$ contains exactly one non-equation edge $e' \in \mathcal{F}'$, $\mathcal{L}(C) \neq 0$, where $\mathcal{L}(C) \overset{\Delta}{=} \sum_{e \in C} \mathcal{L}(e)$. For such a good graph $G$, the induced system of equations and non-equations is defined as:

$$\mathcal{E}_G = \begin{cases} Y_i \oplus Y_j = \lambda_{ij} \forall \{y_i, y_j\} \in \mathcal{F}, \\ Y_i \oplus Y_j \neq \lambda'_{ij} \forall \{y_i, y_j\} \in \mathcal{F}'. \end{cases}$$

The set of components in $G$ is denoted by $\text{comp}(G) = (C_1, \ldots, C_k)$, $\mu_i$ denotes the size of (i.e. the number of vertices in) the $i$-th component $C_i$ and $\mu_{\text{max}} = \max\{\mu_1, \ldots, \mu_k\}$ is the size of the largest component of $G$. $\rho_i$ the total number of vertices upto the $i$-th component with the convention that $\rho_0 = 0$.

**Definition 1 (Injective Solution).** With respect to the system of equations and non-equations $\mathcal{E}_G$ (as defined above), an injective function $\Phi : V \to R$, where $R \subseteq \{0,1\}^n$, is said to be an injective solution if $\Phi(Y_i) \oplus \Phi(Y_j) = \lambda_{ij}$ for all $\{y_i, y_j\} \in \mathcal{F}$ and $\Phi(Y_i) \oplus \Phi(Y_j) \neq \lambda'_{ij}$ for all $\{y_i, y_j\} \in \mathcal{F}'$.

**Theorem 1.** Let $U = \{u_1, \ldots, u_{\sigma}\}$ be a non-empty finite subset of $\{0,1\}^n$, for some $\sigma \geq 0$. Let $G = (V, \mathcal{F} \cup \mathcal{F}', \mathcal{L})$ be a good graph with $\alpha$ vertices such that $|\mathcal{F}| = q_n, |\mathcal{F}'| = q_n$. Let $\text{comp}(G^u) = (C_1, \ldots, C_k)$ and $|C_i| = \mu_i$, $\rho_i = (\mu_1 + \cdots + \mu_i)$. Then the total number of injective solutions, chosen from a set
$Z = \{0, 1\}^n \setminus U$ of size $2^n - \sigma$, for the induced system of equations and non-equations $E_G$ is at least:

$$\frac{(2^n - \sigma)_\alpha}{2^{mqm}} \left( 1 - \sum_{i=1}^{k} \frac{6(\rho'_i - 1)^2 (\mu_i)}{2^{2n}} - \frac{2q_h}{2^n} \right),$$

provided $\rho'_k \mu_{\text{max}} \leq 2^n / 4$ where $\rho'_i = \rho_i + \sigma$.

**Proof.** We proceed the proof by counting the number of solutions in each of the $k$ components. Let $\tilde{\mu}_{ij}$ denotes the number of edges from $F'$ connecting vertices between $i$-th and $j$-th component of $G = (V \setminus G(V))$. For the first component, the number of solutions is at least exactly $(2^n - \mu_1 \sigma)$. We fix such a solution and count the number of solutions for the second component, which is $(2^n - \mu_1 \mu_2 - \tilde{\mu}_{1,2} - \mu_2 \sigma)$. This is because, let $Y_{\mu_1+1}$ be an arbitrary vertex of the second component and let $y_{\mu_1+1}$ be a solution of it. This solution is valid if the following conditions hold:

- $y_{\mu_1+1} \notin U$.
- $y_{\mu_1+1}$ does not take $\mu_1$ values $(y_{\mu_1}, \ldots, y_{\mu_1})$ from the first component.
- It must discard $\mu_1(\mu_2 - 1)$ values $(y_{\mu_1} \oplus L(P_j), \ldots, y_{\mu_1} \oplus L(P_j))$ for all possible paths $P_j$ from a fixed vertex to any other vertex in the second component.
- It must discard $\rho(\mu_2 - 1)$ values as $(y_{\mu_1+1} \oplus L(P_j)) \notin Y$ for all possible paths $P_j$ from $Y_{\mu_1+1}$ to any other vertices in the second component.
- $y_{\mu_1+1}$ does not take $\tilde{\mu}_{1,2}$ values to compensate for the fact that the set of values is no longer a group.

Summing up all the conditions, the number of solutions for the second component is at least $(2^n - \mu_1 \mu_2 - \mu_2 \sigma - \tilde{\mu}_{1,2})$. In general, the total number of solutions for the $i$-th component is at least $\prod_{j=1}^{k} \left( 2^n - \rho_{i-1} \mu_i - \mu_i \sigma - \sum_{j=1}^{i-1} \tilde{\mu}_{ij} \right)$. Suppose there are $k'$ vertices that do not belong to the set of vertices of the subgraph $G'$. Fix such a vertex $Y_{\rho_{k'+1}}$ and let us assume that $\rho'_{k'+1}$ blue dashed edges are incident on it. If $y_{\rho_{k'+1}}$ is a valid solution to the variable $Y_{\rho_{k'+1}}$, then we must have (a)
$y_{ρ_{k+i}}$ should be distinct from the previous $ρ_k$ assigned values, (b) $y_{ρ_{k+i}}$ should be distinct from the $(i−1)$ values assigned to the variables that do not belong to the set of vertices of the subgraph $G^m(V)$, (c) $y_{ρ_{k+i}}$ should be distinct from the values of $μ$, and (d) $y_{ρ_{k+i}}$ should not take those $μ'_{ρ_{k+i}}$ values. Therefore, the total number of solutions is at least

$$h_α \geq \prod_{i=1}^{k} \left(2^n - ρ_{i-1}µ_i - µ_iσ - \sum_{j=1}^{i-1} \tilde{µ}_{i,j}\right) \cdot \prod_{i \in [k']} \left(2^n - ρ_k - σ - i + 1 - ρ'_{ρ_k+i}\right).$$

Let $χ_i = (ρ_{i+1} + ... + ρ_{i+k-1})q''_v$, and $ρ'_i = ρ_i + σ$. After a simple algebraic calculation on Eqn. (6), we obtain

$$h_α \frac{2^αq_m}{(2^n - σ)_α} \geq \prod_{i=1}^{k} \frac{(2^n - ρ'_{i-1}µ_i - χ_i)2^{α(µ_i-1)}}{(2^n - ρ'_{i-1}µ_i)} \cdot \prod_{i=1}^{k'} \frac{(2^n - ρ'_k - i + 1 - ρ'_{ρ_k+i})}{(2^n - ρ'_k - i + 1)}.$$  

By expanding $(2^n - ρ'_{i-1}µ_i)$ we have $(2^n - ρ'_{i-1}µ_i) ≤ 2^n µ_i - 2^n(µ_i-1)(ρ'_{i-1}µ_i + \binom{µ_i}{2}) + 2^n(µ_i-2)A_i$, where $A_i = \binom{µ_i}{2}(ρ'_{i-1})^2 + \binom{µ_i}{2}(µ_i-1)ρ'_{i-1} + \binom{µ_i}{2}(µ_i-2)(3µ_i-1)2$. 

**Bounding D.1.** With a simplification on the expression of D.1, we have

$$D.1 \geq \prod_{i=1}^{k} \left(1 - \frac{A_i}{2^{2n} - 2^n(ρ'_{i-1}µ_i + \binom{µ_i}{2}) + A_i} - \frac{2^nχ_i}{2^{2n} - 2^n(ρ'_{i-1}µ_i + \binom{µ_i}{2}) + A_i}\right)$$

$$\geq \prod_{i=1}^{k} \left(1 - \frac{2A_i}{2^n} - \frac{2χ_i}{2^n}\right)^{(4)} \geq \left(1 - \sum_{i=1}^{k} \frac{6(ρ'_{i-1})^2(µ_i)}{2^{2n}} - \frac{2q'_v}{2^n}\right),$$

where (4) follows from the fact that $2^n(ρ'_{i-1}µ_i + \binom{µ_i}{2}) - A_i ≤ 2^{2n}/2$, which holds true when $ρ'_{k}µ_{max} ≤ 2^n/4$, (5) holds true due to the fact that $A_i ≤ 3(ρ'_{i-1})^2(µ_i/2)$ and $(µ_1 + ... + µ_k) = q'_v$, the total number of blue dashed edges across the components of $G^m$ and $µ_1 + ... + µ_k ≤ α$.

**Bounding D.2.** For bounding D.2, we have

$$D.2 \geq \prod_{i=1}^{k'} \left(1 - \frac{ρ'_{ρ_k+i}}{(2^n - ρ'_k - i + 1)}\right)^{(6)} \geq \left(1 - \sum_{i=1}^{k'} \frac{ρ'_{ρ_k+i}}{2^n}\right)^{(7)} \geq \left(1 - \frac{2q''_v}{2^n}\right),$$

where (6) follows due to the fact that $(ρ'_k + i - 1) ≤ 2^n/2$ and (7) follows as we denote $(ρ'_{ρ_k+i} + ... + ρ'_{ρ_k+k'}) = q''_v$, the total number of blue dashed edges incident on the vertices outside of the set $G^m(V)$. 

...
Combining D.1 and D.2. Finally, by combining the expression of D.1 and D.2, we have

\[
\frac{2^{\eta m}}{(2^n - \sigma)_{\eta m}} \geq \left(1 - \sum_{i=1}^{k} \frac{6(n'_i - 1)^2}{2^{2n}} - \frac{2q_v}{2^n}\right),
\]

where \( q_v = q'_v + q''_v \), the total number of non-equation edges in \( G \).

\[\square\]

4 Security of nEHtM in Public Permutation Model

In this section, we first state that \( \text{nEHtM}_p \) achieves \( 2n/3 \)-bit security in public permutation model in the faulty nonce model. Followed by this, we demonstrate a matching attack in subsect. 4.2 to show the security bound is tight.

4.1 Security of nEHtM_p

We show that \( \text{nEHtM}_p \) is secure against all adversaries that makes roughly \( 2^{2n/3} \) queries in the faulty nonce model. However, similar to \( \text{nEHtM} \), the construction possesses a birthday bound forging attack when the number of faulty nonces reaches an order of \( 2^{n/2} \) [16].

**Theorem 2.** Let \( M \) and \( K_h \) be two finite and non-empty sets. Let \( \pi \leftarrow \mathcal{P}(n) \) be an \( n \)-bit public random permutation and \( H : K_h \times M \rightarrow \{0,1\}^{n-1} \) be an \((n-1)\)-bit \( \epsilon_{\text{axu}} \)-almost xor universal and \( \epsilon_{\text{reg}} \)-almost regular hash function. Moreover, \( K \leftarrow \{0,1\}^{n-1} \) be an \( n-1 \) bit random key and \( \eta \) be a fixed parameter. Then the forgery advantage for any \((\eta,q_m,q_v,2p)\)-adversary against the construction \( \text{nEHtM}_p[\pi,H,K] \) that makes at most \( \eta \) faulty queries out of \( q_m \) signing, \( q_v \) verification and altogether \( 2p \) primitive queries, is given by

\[
\begin{align*}
\text{Adv}_{\text{nEHtM}_p}^{\text{MAC}}(\eta,q_m,q_v,2p) \leq & 12\eta^2 \left( q_m + 2p \right) / 2^{2n} + (p + q_m) \left( 192pq_m / 2^{2n} + 48pq_m^2 \epsilon_{\text{axu}} / 2^{2n} \right) \\
&+ 48q_m^3 / 2^{2n} + 12q_m^4 \epsilon_{\text{axu}} / 2^{2n} + 2q_v / 2^n + p^2 \epsilon_{\text{reg}} (3q_m + 2q_v) / 2^n + q_m / 2^n \\
&+ \epsilon_{\text{axu}} \left( 4q_m^3 / 2^n + 2\eta q_m + pq_m^2 / 2^n + q_m^3 / 2^{n+1} + (\eta + 1)q_v \right) \\
&+ \epsilon_{\text{reg}} (2p + p^2 \sqrt{3nq_m}) / 2^{2n} + 2p^2 q_m / 2^n + 2 + 2\eta / 2^n + 2p^2 \sqrt{3nq_m} / 2^{2n}.
\end{align*}
\]

By assuming \( \epsilon_{\text{axu}} \approx 2/2^n \) and \( \epsilon_{\text{reg}} \approx 2/2^n \), the above bound is simplified to

\[
\begin{align*}
\text{Adv}_{\text{nEHtM}_p}^{\text{MAC}}(\eta,q_m,q_v,2p) \leq & 80q_m^3 / 2^{2n} + 4(q_m + q_v) / 2^n + 4p \sqrt{3nq_m} / 2^n + 12\eta^2 / 2^{2n} \left( q_m + 2p \right) / 2^n \\
&+ (p + q_m) \left( 200pq_m / 2^{2n} + 96pq_m^2 / 2^{3n} + 4\eta / 2^n \right) + 2q_v / 2^n + 4p^2 q_v / 2^{2n} \\
&+ 2 / 2^n + 2\eta / 2^n,
\end{align*}
\]
We defer the proof of this theorem in Sect. 5. The forging advantage of nEHTM$_p$ for $\eta \leq 2^{n/3}$, $q_m \leq 2^{2n/3}$ and $p \leq 2^{2n/3}$ is thus given by

$$\text{Adv}^{\text{MAC}}_{\text{nEHTM}}(q_m, q_v, 2p) \leq \left( \frac{29q_m}{2^{2n/3}} + \frac{6q_v}{2^{2n/3}} + \frac{28p}{2^{2n/3}} \right) + \frac{296p^2q_m}{2^{2n}} + \frac{296p^2q_v^2}{2^{2n}} + \frac{4p^2q_v}{2^{2n}} + \frac{4}{2^{2n/3}}.$$ 

4.2 Matching Attack on nEHTM$_p$

In this section we show a matching attack on nEHTM$_p$ with $2^{2n/3}$ signing queries and total $2^{2n/3} + 2$ primitive queries. For carrying out the attack, we consider the following version of Polyhash function, a specific instantiation of an axu and ar hash function: for a message $m$, if the size of $m$ is not a multiple of $n$, where $n$ is the key size of the hash function, then we first apply an injective padding (e.g., $10^n$) on it to generate a padded message $m'$. Then the output of the hash function for $m'$ is computed as follows:

$$\text{Poly}_h(m') = k_{h}^{l+1} \oplus k_{h}^{l} \cdot m'_{l} \oplus k_{h}^{l-1} \cdot m'_{l-1} \oplus \ldots \oplus k_{h} \cdot m'_{1},$$

where $l$ denotes the number of message blocks of $m'$ and $m'_{i}$ denotes the $i$-th message block of $m'$. Now, it is easy to see that the hash function is $(l_{\text{max}}+1)/2^n$-secure axu and ar hash function, where $l_{\text{max}}$ is the maximum number of message blocks allowed. With this instance of the hash function of nEHTM$_p$, we mount the following attack. To begin with, we exploit bad event B.1 to mount the attack on the construction. We construct a deterministic adversary $A$ that forges nEHTM$_p$ by making $2^{2n/3}$ signing queries and total $2^{2n/3} + 2$ many primitive queries to $\pi$ as follows:

**Attack Algorithm:**

1. A first chooses a single block message $m$ consisting of all zeros, i.e., $m = 0^n$.
2. Then $A$ makes $2^{2n/3}$ signing queries with $(\nu_j, m)$ and obtains the tag $t_j$ for $j \in [2^{2n/3}]$, where $\nu_j = 0^{n/3-1}|| (j)_{2n/3}$.
3. $A$ makes $2^{2n/3-1}$ forward primitive queries to $\pi$ with $x^1_j$ and obtains the output $y^1_j$ for $j \in [2^{2n/3-1}]$, where $x^1_j = 0|| (j)_{2n/3-1}|| 0^{n/3}$.
4. $A$ makes again $2^{2n/3-1}$ forward primitive queries to $\pi$ with $x^2_j$ and obtains the output $y^2_j$ for $j \in [2^{2n/3-1}]$, where $x^2_j = 1|| \text{left}_{n/3-1}(j)_{2n/3-1}|| 0^{n/3}|| \text{right}_{n/3}(j)_{2n/3-1}$.
5. Then, $A$ finds a triplet $(i, j, l) \in [2^{2n/3}] \times [2^{2n/3-1}] \times [2^{2n/3-1}]$ such that $t_i = y^1_j \oplus y^2_j$.
6. $A$ makes two additional forward primitive queries to $\pi$ with $x^1_j = x^1_i \oplus 0|| 1^{n-1}$ and $x^2_j = x^2_i \oplus 0|| 1^{n-1}$. Let the received response be $y^1_j$ and $y^2_j$ respectively.
7. Finally, $A$ forges with $(\nu_l \oplus 1^{n-1}, m, y^1_j \oplus y^2_j)$.

**Analysis of the Forging Advantage.** We first note that the structure of $\nu_j, x^1_j$ and $x^2_j$ are as follows:

$\nu = \begin{cases} 0 & n/3-1 \\ 0 & n/3 \end{cases} \oplus \begin{cases} 0 & n/3-1 \\ 0 & n/3 \end{cases} \oplus \begin{cases} 0 & n/3 \end{cases} \oplus \begin{cases} 0 & n/3 \end{cases} \oplus \begin{cases} 0 & n/3 \end{cases}, \quad x^1_j = \begin{cases} 0 & n/3 \end{cases} \oplus \begin{cases} 0 & n/3 \end{cases} \oplus \begin{cases} 0 & n/3 \end{cases} \oplus \begin{cases} 0 & n/3 \end{cases}$.
Note that, the number of elements \( \langle \nu_i, x_i \rangle \) that satisfy the relation \( 0 \parallel (\nu_i \oplus k) = x_i \) is exactly \( 2^{n/3} \). As a result, the expected number of triplets \( (i,j,k) \) that satisfy \( 0 \parallel (\nu_i \oplus k) = x_i \) and \( 1 \parallel (\nu_i \oplus k^2) = x_i^2 \) is exactly 1. For this particular triplet \( (i,j,k) \) that satisfies the relation, \( A \) makes two additional forward primitive queries to \( \pi \) with \( x_i^1 = x_i \oplus i \) and \( x_i^2 = x_i^2 \oplus i \), where \( i = 0 \parallel n-1 \). Thus, if \( A \) makes a forging query with \( \nu \oplus 1 \parallel n-1 \) (which is distinct from all other nonces that belong to the signing queries) and with the same message \( m = 0^n \), then we have

\[
\pi(0 \parallel (\nu \oplus 1^{n-1} \oplus i) \parallel (\nu \oplus 1^{n-1} \oplus i^2)) \\
= \pi(0 \parallel (\nu \oplus k) \parallel (\nu \oplus k^2)) = \pi(x_i^1 \parallel x_i^2) = y_i^1 \parallel y_i^2
\]

which makes \( (\nu \oplus 1^{n-1}, m, y_i^1 \parallel y_i^2) \) a valid and successful forging attempt. Note that, the number of signing queries required is \( 2^{2n/3} \) and the total number of primitive queries required is \( 2^{2n/3} + 2 \). However, the time complexity of this attack is \( 2^{2n-2} \).

5 Proof of Theorem 2: MAC Security of nEHTM

Due to Eqn. (1), we bound the distinguishing advantage instead of bounding the forging advantage of nEHTMp. For this, we consider any information theoretic deterministic distinguisher \( A \) that has access to the following oracles in either the real world or in the ideal world: in the real world it has access to \( (\text{nEHTM}_p, \text{Sig}_{(k,h)}, \text{nEHTM}_p, \text{Ver}_{(k,h)}, \pi, \pi^{-1}) \); in the ideal world it has access to \( (\text{RF}, \text{Rej}, \pi, \pi^{-1}) \). We summarize the interactions of the distinguisher with its oracle in a transcript \( \tau = \tau_m \cup \tau_v \), where \( \tau_m = \langle \nu_1, m_1, t_1 \rangle, \ldots, \langle \nu_m, m_m, t_m \rangle \) is the MAC transcript and \( \tau_v = \langle \nu_1', m_1', t_1' \rangle, \ldots, \langle \nu_v, m_v, t_v \rangle \) is the verification transcript. Primitives queries to \( \pi \) are summarized in two lists in the form of \( \tau^{(1)}_p = \{ (x_1^1, y_1^1), \ldots, (x_p^1, y_p^1) \} \) and \( \tau^{(2)}_p = \{ (x_1^2, y_1^2), \ldots, (x_p^2, y_p^2) \} \), where \( \text{msb}(x_i^1) = 0 \) and \( \text{msb}(x_i^2) = 1 \). We assume that none of the transcripts contain any duplicate elements and after the interaction, we reveal the keys \( k, h \) to the distinguisher (before it output its decision), which happens to be the keys used in the construction for the real world and uniformly sampled dummy keys for the ideal world. The complete view is denoted by \( \tau' = (\tau_m, \tau_v, \tau^{(1)}_p, \tau^{(2)}_p, k, h) \).

5.1 Definition and Probability of Bad Transcripts

For the notational simplicity, we denote \( \text{H}_{k_b}(m_b) = H_b \). \( \hat{x}_b^0 \) denotes \( \text{chop}_{\text{msb}}(x_b^0) \) for \( b = 1, 2 \). We also define three sets: (a) \( T \triangleq \{ t \parallel (\nu_i, m_i, t_i) \in \tau_m \} \), (b) \( \mathcal{Y}_1 \triangleq \{ y_i^1 : (x_i^1, y_i^1) \in \tau^{(1)}_p \} \) and (c) \( \mathcal{Y}_2 \triangleq \{ y_i^2 : (x_i^2, y_i^2) \in \tau^{(2)}_p \} \). The main idea of identifying bad events is to avoid the input collision of the permutation with
primitive queries as that will determine the corresponding tag; hence losing the randomness of the tag, which in turn, will help the adversary to distinguish the output from random.

Definition 2 (Bad Transcript for nEHTM_p). Given a parameter ξ ∈ N, where ξ ≥ η, an attainable transcript τ' = (τ_m, τ_v, τ_p^{(1)}, τ_p^{(2)}, k_h, k) is called a bad transcript if any one of the following holds:

- B.1 : ∃ i ∈ [q_m], j, ℓ ∈ [p] such that ν_i ⊕ k = ˆx^i_j, ν_i ⊕ H_i = ˆx^i_j.
- B.2 : ∃ i, j, ℓ ∈ [q_m], i ≠ j, i ≠ ℓ such that ν_i = ν_j and ν_i ⊕ H_j = ν_j ⊕ H_j.
- B.3 : ∃ i ≠ j ∈ [q_m], ℓ ∈ [p] such that ν_i ⊕ k = ˆx^i_j and ν_i ⊕ H_i = ν_j ⊕ H_j.
- B.4 : ∃ i ≠ j ∈ [q_m], ℓ ∈ [p] such that ν_i = ν_j and ν_i ⊕ H_i = ˆx^i_j.
- B.5 : ∃ i ≠ j ∈ [q_m] such that ν_i = ν_j and t_i = t_j.
- B.6 : ∃ i ≠ j ∈ [q_m] such that ν_i ⊕ H_i = ν_j ⊕ H_j and t_i = t_j.
- B.7 : # (τ) × T × Y : t_i = y_i^j ⊕ y_j^i) ≥ p^2 q_m / 2^n + p √3 q_m n.
- B.8 : ∃ i ∈ [q_m], j, ℓ ∈ [p] such that ν_i ⊕ k = ˆx^i_j, y_i^j ⊕ t_i = y_j^i.
- B.9 : ∃ i ∈ [q_m], j, ℓ ∈ [p] such that ν_i ⊕ H_i = ˆx^i_j, y_i^j ⊕ t_i = y_j^i.
- B.10 : (∃ (t_1, ..., t_{ξ+1}) ∈ [q_m] such that ν_i ⊕ H_i = ν_j ⊕ H_j = ... = ν_k ⊕ H_k (the optimal value of ξ shall be determined later in the proof).
- B.11 : ∃ a ∈ [q_m], i ∈ [q_m] such that ν_i = ν_a, ν_i ⊕ H_i = ν_a ⊕ H_a and t_i = t_a.
- B.12 : a ∈ [q_m], i, j, ℓ ∈ [p] such that ν_i ⊕ k = ˆx^i_j, ν_i ⊕ H_i = ˆx^i_j and t_i = y_i^j.
- B.13 : ∃ i ∈ [q_m] such that t_i = 0^n.

Lemma 2. Let D_id and BadT be defined as in Sect. 2.3. Then

\[ \Pr[D_id ∈ BadT] ≤ \frac{2^2 \epsilon_{reg}}{2^n} (3 q_m + 2 q_m) + \epsilon_{aux}(\frac{q_m^2}{2^n} + 2 η q_m + \frac{p q_m}{2^n} + \frac{q_m}{2^n} + (\eta + 1) q_p) + \epsilon_{reg}(2 n p + p √3 q_m n) + \frac{2^2 q_m}{2^n} + 2 + 2 n + \frac{2 q_m}{2^n} + \frac{q_m}{2^n} \]

Proof of the lemma can be found in Sect. 6.

5.2 Analysis of Good Transcripts

For a good transcript τ' = (τ_m, τ_v, τ_p^{(1)}, τ_p^{(2)}, k_h, k), the ideal interpolation probability is

\[ p_{id}(τ') = \Pr[D_id = τ'] = \frac{1}{|K_h|} \cdot \frac{1}{2^{n-1}} \cdot \frac{1}{2^{q_m}} \cdot \frac{1}{(2^n)^{q_p}}. \]  (7)

Computing Real Interpolation Probability. To compute the real interpolation probability, we regroup the elements of τ_m, τ_p^{(1)} and τ_p^{(2)} into three new transcripts ˆτ_m, ˆτ_p^{(1)} and ˆτ_p^{(2)} in the following way: initially the new transcripts are set to the old one. Now, for each (ν_i, m_i, t_i) ∈ τ_m, if (a) ν_i ⊕ k = ˆx^i_j, then ˆτ_m ← τ_m \{(ν_i, m_i, t_i)\} and ˆτ_p^{(2)} ← ˆτ_p^{(2)} ∪ \{(ν_i ⊕ H_i), t_i ⊕ y_j\}; if (b) ν_i ⊕ H_i = ˆx^i_j,
then $\hat{\tau}_m \leftarrow \tau_m \setminus \{(\nu_i, m_i, t_i)\}$ and $\hat{\tau}_p^{(1)} \leftarrow \hat{\tau}_p^{(1)} \cup \{0\}[(\nu_i \oplus k), t_i \oplus y_i^2]$. Since $\tau'$ is a good transcript, it does not meet any of the bad conditions listed in Defn. 2. We know that if $\nu_i \oplus k = \hat{x}_j$, then $\nu_i \oplus H_i$ cannot collide with $\hat{x}^{\tau}_k$ (due to $\neg B.1$) and $y_i^1 \oplus t_i$ cannot collide with $y_i^2$ (due to $\neg B.8$). Similarly for $\hat{\tau}_p^{(2)}$. This way, we will end up with soundly defined $\hat{\tau}_p^{(1)}$ and $\hat{\tau}_p^{(2)}$ and a set of signing queries $\hat{\tau}_m$ that does not collide with any tuple in $\hat{\tau}_p^{(1)}$ or $\hat{\tau}_p^{(2)}$.

Let $s_1, s_2 \leq p$ be the number of signing queries that collides with any element of $\tau_p^{(1)}$ and $\tau_p^{(2)}$ respectively. Therefore, $p_1 \triangleq |\hat{\tau}_p^{(1)}| = p + s_2$, $p_2 \triangleq |\hat{\tau}_p^{(2)}| = p + s_1$ and $q_m' \triangleq |\hat{\tau}_m| = q_m - s_1 - s_2$. We denote $q_p' = p_1 + p_2 = 2p + s_1 + s_2$. We say that a permutation $\pi$ is compatible with $\hat{\tau} \triangleq \hat{\tau}_m \cup \tau_n \cup \hat{\tau}_p^{(1)} \cup \hat{\tau}_p^{(2)}$ if the following holds:

- for all $(\nu_i, m_i, t_i) \in \hat{\tau}_m$, $\pi(0||\nu_i \oplus k) \oplus \pi(1||\nu_i \oplus H_i) = t_i$
- for all $a \in \{q_m', p_1, p_2\}$, $\pi(0||\nu_a' \oplus k) \oplus \pi(1||\nu_a' \oplus H_a') \neq t_a'$
- for all $(x_i^1, y_i^1) \in \hat{\tau}_p^{(1)}$, $\pi(x_i^1) = y_i^1$
- for all $(x_i^2, y_i^2) \in \hat{\tau}_p^{(2)}$, $\pi(x_i^2) = y_i^2$.

Therefore, the remaining part is to count the number of compatible permutations $\pi$. As a result, we have

$$\rho_{\pi}(\tau') \triangleq \Pr[D_{\pi} = \hat{\tau}] = \frac{1}{|K_h|} \cdot \frac{1}{2^{n+1}} \cdot \frac{h_\alpha}{(2^n)_{p_1 + p_2 + \alpha}},$$

(8)

where $h_\alpha$ denotes the number of injective solutions to the following system of equations and non-equations ($E^= \cup E^\neq$), with $\alpha$ many distinct variables. For notational simplicity, we denote $\pi(0||\nu_i \oplus k)$ as $U_i$ and $\pi(1||\nu_i \oplus H_i)$ as $V_i$.

$$E^= = \begin{align*}
U_1 \oplus V_1 &= t_1 \\
U_2 \oplus V_2 &= t_2 \\
& \vdots \\
U_{q_m'} \oplus V_{q_m'} &= t_{q_m'}
\end{align*}$$

$$E^\neq = \begin{align*}
U_1' \oplus V_1' &\neq t_1' \\
U_2' \oplus V_2' &\neq t_2' \\
& \vdots \\
U_{q_m'}' \oplus V_{q_m'}' &\neq t_{q_m'}
\end{align*}$$

where $q_m' = q_m - s_1 - s_2$. It is to be noted here that $E^= \cup E^\neq$ is defined over $\alpha$ many distinct variables. Therefore, some variables in $E^= \cup E^\neq$ may collide to each other. Thus, from Eqn. (7) and Eqn. (8), we have,

$$\frac{\rho_{\pi}(\tau')}{\rho_{\id}(\tau')} = \frac{h_\alpha \cdot 2^{q_m}}{A_1 \cdot 2^{2n} \cdot A_2 \cdot 2^{2n} \cdot A_3 \cdot 2^{n+1}}.$$

(9)

Note that, $A_1 \geq 1$ and $A_2 \geq 1$. Therefore, we are left to bound $A_3$. Note that, the induced graph $G$ of $E^= \cup E^\neq$ has $\alpha$ many vertices. Moreover, $|F| = q_m$ and $|F'| = q_v$. It is easy to verify that as $\tau'$ is a good transcript, $G$ is a good graph.
Therefore, by putting \( \sigma = q'_p \) in Theorem 1, we have

\[
h_n \geq \frac{(2^m - 2p - s_1 - s_2)}{2^n q_m} \cdot \left( 1 - \sum_{i=1}^{k} \frac{6(\rho'_{i-1})^2}{2^n} - \frac{2q_v}{2^n} \right).
\]  \( (10) \)

From Eqn. (8) and Eqn. (10), we have

\[
p_{\text{re}}(\tau') \geq \left( 1 - \sum_{i=1}^{k} \frac{6(\rho'_{i-1})^2}{2^n} - \frac{2q_v}{2^n} \right) \geq 1 - \sum_{i=1}^{k} 24(q'_m + q'_p)(\mu_i^2) - \frac{2q_v}{2^n},
\]  \( (1) \)

where the simplification for (1) follows from the fact

\[
\rho'_{i-1} \leq \alpha + q'_p \leq 2(q'_m + q'_p).
\]

Now, from Sect. 6.2 of [16] we have

\[
E\left[ \sum_{i=1}^{k} \left( \frac{\mu_i}{2} \right)^2 \right] \leq \frac{(q'_m)^2}{2} + \frac{\eta^2}{2} + 2q'_m.
\]  \( (11) \)

By applying the expectation method of Sect. 2.3 on Eqn. (11), we have

\[
E[\Phi(D_{id})] \leq \frac{12(q'_m + q'_p)^2}{2^n} \left( \frac{(q'_m)^2}{2} + \frac{\eta^2}{2} + 4q'_m \right) + \frac{2q_v}{2^n}.
\]  \( (12) \)

By doing a simple algebra on Eqn. (12) and by assuming \( q'_m \leq q_m, q'_p \leq 4p \), we have

\[
E[\Phi(D_{id})] \leq \frac{12q'_m q_m}{2^n} + \frac{12q'_p q_m}{2^n} + \frac{48q'^3_m}{2^n} + \frac{48pq'_m q_m}{2^n} + \frac{48\eta^2 q'_m}{2^n} + \frac{192p^2 q'_m}{2^n}.
\]  \( (13) \)

**Finalizing the proof:** We have assumed that \( \xi \geq \eta \) and from the condition of Theorem 1, we have \( \xi \leq 2^n / (8q'_m + 2q'_p) \leq 2^n / 8q'_m \). By assuming \( \eta \leq 2^n / 8q'_m \) (otherwise the bound becomes vacuously true) we choose \( \xi = 2^n / 8q'_m \). Hence, the result follows by applying Eqn. (3), Lemma 2, Eqn. (13) and \( \xi = 2^n / 8q'_m \).

**6 Proof of Lemma 2**

By the union bound,

\[
Pr[D_{id} \in \text{BadT}] \leq \sum_{i=1}^{7} Pr[B.i] + Pr[B.8 | B.7] + Pr[B.9 | B.7] + \sum_{i=10}^{13} Pr[B.i].
\]  \( (14) \)
In the following, we bound the probabilities of all the bad events individually. The lemma will follow by adding the individual bounds.

Bounding B.1. For any possible signing query \((\nu_i, m_i, t_i) \in \tau_m\) and a pair of any possible primitive queries \((x_j^1, y_j^1) \in \tau_p^{(1)}\) and \((x_j^2, y_j^2) \in \tau_p^{(2)}\), the only randomness in the equation \(\nu_i \oplus k = x_j^1\) is \(k\) and the randomness in the equation \(\nu_i \oplus H_j = x_j^2\) is \(k_h\). In the ideal world, \(k\) and \(k_h\) are dummy keys, sampled uniformly and independently from their respective space. Therefore, for a fixed choice of \(i, j \) and \(\ell\), the probability of the event is \(\epsilon_{reg}/2^{n-1}\), where \(\epsilon_{reg}\) is the regular advantage of the underlying hash function. Summing over all possible choices of \(i, j \) and \(\ell\) we have

\[
\Pr[B.1] \leq \frac{2p^2q_m\epsilon_{reg}}{2^n}.
\]

Bounding B.2. Let \(\mathcal{N}\) be the set of all query indices \(i\) for which there is a \(j \neq i\) such that \(\nu_i = \nu_j\). It is easy to see that \(|\mathcal{N}| \leq 2\eta\). Event B.2 occurs if for some \(j \in \mathcal{N}\), \(\nu_i \oplus H_j = \nu_i \oplus H_\ell\) for some \(\ell \neq j\). For any such fixed \(i, j, \ell\), the probability of the event is at most \(\epsilon_{axu}\), where \(\epsilon_{axu}\) is the almost xor universal advantage of the underlying hash function. The number of such choices of \((i, j, \ell)\) is at most \(2\eta q_m\). Hence,

\[
\Pr[B.2] \leq 2\eta q_m \epsilon_{axu}.
\]

Bounding B.3. For any two signing queries \((\nu_i, m_i, t_i), (\nu_j, m_j, t_j) \in \tau_m\) and a primitive query \((x_l^1, y_l^1) \in \tau_p^{(1)}\), the only randomness in the equation \(\nu_i \oplus k = x_l^1\) is \(k\) and the randomness in the equation \(H_i \oplus H_j = \nu_i \oplus \nu_j\) is \(k_h\). In the ideal world, \(k\) and \(k_h\) are dummy keys, sampled uniformly and independently from their respective space. Therefore, for a fixed choice of \(i, j \) and \(\ell\), the probability of the event is \(\epsilon_{axu}/2^{n-1}\), where \(\epsilon_{axu}\) is the almost xor universal advantage of the underlying hash function. Summing over all possible choices of \(i, j \) and \(\ell\) we have

\[
\Pr[B.3] \leq \frac{pq_m^2 \epsilon_{axu}}{2^n}.
\]

Bounding B.4. For any two signing queries \((\nu_i, m_i, t_i), (\nu_j, m_j, t_j) \in \tau_m\) and a primitive query \((x_l^2, y_l^2) \in \tau_p^{(2)}\), the only randomness in the equation \(\nu_i \oplus H_i = x_l^2\) is \(k_h\). In the ideal world, \(k_h\) is sampled uniformly from \(\mathcal{K}_h\). Therefore, for a fixed choice of \(i, j \) and \(\ell\), the probability of the event is \(\epsilon_{reg}\). The number of choices of \(i \neq j \in q_m\) such that \(\nu_i = \nu_j\) is at most \(2\eta\) and the number of choices of \(\ell\) is at most \(p\). Summing over all possible choices of \(i, j \) and \(\ell\) we have

\[
\Pr[B.4] \leq 2\eta p \epsilon_{reg}.
\]

Bounding B.5. For a fixed choice of indices \(i\) and \(j\), the probability of the event is at most \(1/2^n\). Number of choices of \(i\) and \(j\) such that \(\nu_i = \nu_j\) is at most \(2\eta\). Summing over all possible choices of \(i\) and \(j\) we have

\[
\Pr[B.5] \leq \frac{2\eta}{2^n}.
\]
Bounding B.6. Similar to B.5, for a fixed choice of indices \( i \) and \( j \), the probability of the event is at most \( \epsilon_{\text{axu}}/2^n \), as the event \( \nu_i \oplus H_i = \nu_j \oplus H_j \) is independent over \( t_i = t_j \). Summing over all possible choices of \( i \) and \( j \) we have

\[
\Pr[B.6] \leq \frac{q_m^2 \epsilon_{\text{axu}}}{2^{n+1}}.
\]  

Bounding B.7. Event B.7 is bounded by Lemma 1, where we take \( A = \mathcal{Y}_1 \) and \( B = \mathcal{Y}_2 \).

\[
\Pr[B.7] \leq \frac{2}{2^n}.
\]

Bounding B.8 | B.7. Let \( C \triangleq p^2 q_m / 2^n + p \sqrt{3n q_m} \). As we are bounding the event \( B.8 | B.7 \), number of \( i, j \) and \( \ell \) that satisfies \( t_i = y_i^j \oplus y_i^\ell \) is at most \( C \). For a fixed choice of indices \( i, j \) and \( \ell \), the probability of the event is at most \( 1/2^{n-1} \). Hence, by summing over all possible choices of \( i, j \) and \( \ell \), we have

\[
\Pr[B.8 | B.7] \leq \frac{2p^2 q_m}{2^n} + \frac{2p \sqrt{3n q_m}}{2^n}.
\]

Bounding B.9 | B.7. Bounding B.9 | B.7 is identical to that of B.8 | B.7. For a fixed choice of indices \( i, j \) and \( \ell \), the probability of the event is at most \( \epsilon_{\text{reg}} \). Summing over all possible choices of \( i, j \) and \( \ell \) we have

\[
\Pr[B.9 | B.7] \leq \frac{p^2 q_m \epsilon_{\text{reg}}}{2^n} + \frac{p \sqrt{3n q_m} \epsilon_{\text{reg}}}{2^n}.
\]

Bounding B.10. Event B.10 occurs if there exist \( \xi + 1 \) distinct signing query indices \( \{i_1, \ldots, i_{\xi+1}\} \subseteq [q_m] \) such that \( \nu_i \oplus H_i = \ldots = \nu_{i_{\xi+1}} \oplus H_{i_{\xi+1}} \). This event is thus a \((\xi + 1)\)-multicollision on the \( \epsilon_{\text{univ}}\)-universal hash function \(^4\) mapping \((\nu, m)\) to \( \nu \oplus H_{h_k}(m) \) (as \( H_{h_k} \) is an \( \epsilon_{\text{axu}}\)-almost-xor universal). Therefore, by applying the multicollision theorem of universal hash function (Theorem 1) of [16], we have

\[
\Pr[B.10] \leq q_m^2 \epsilon_{\text{axu}}/2^\xi.
\]

Bounding B.11. For some \( a \in [q_k] \) and \( i \in [q_m] \), if \( \nu_i = \nu'_a \), \( \nu_i \oplus H_i = \nu'_a \oplus H'_a \) and \( t_i = t'_a \), then \( m_i \neq m'_a \) (as the distinguisher is non-trivial). Hence the probability that \( \nu_i \oplus H_i = \nu'_a \oplus H'_a \) holds is at most \( \epsilon_{\text{axu}} \), due to the axu property of the hash function. Now, for any choice of \( a \in [q_k] \), there can be at most \( \eta + 1 \) indices \( i \) such that \( \nu_i = \nu'_a \). Hence, the required probability is bounded as

\[
\Pr[B.11] \leq (\eta + 1) q_k \epsilon_{\text{axu}}.
\]

Bounding B.12. For any possible verification query \((\nu'_a, m'_a, t'_a) \in \tau_v\) and a pair of any possible primitive queries \((x'_j, y'_j) \in \tau_p^{(1)}\) and \((x'_2, y'_2) \in \tau_p^{(2)}\), the only randomness in the equation \( \nu_a^j \oplus k = x'_j \) is \( k \) and the randomness in the equation

\[^4\] A hash function \( H_{h_k} \) is said to be an \( \epsilon_{\text{univ}}\)-universal hash function if for all \( x \neq x' \),

\[
\Pr[H_{h_k}(x) = H_{h_k}(x')] \leq \epsilon_{\text{univ}}.
\]
ν′ \oplus H'_a = x^2_\ell is k_h. In the ideal world, k and k_h are dummy keys, sampled uniformly and independently from their respective spaces. Therefore, for a fixed choice of a, j and \ell, the probability of the event is \epsilon_{reg}/2^{2n-1}. Summing over all possible choices of a, j and \ell we have

$$\Pr[B.12] \leq \frac{2q_p^2 \epsilon_{reg}}{2^n}.$$  \hspace{1cm} (26)

**Bounding B.13.** For a fixed choice of i, the probability that t_i = 0^n is exactly 2^{-n}. Summing over all possible choices of i we have

$$\Pr[B.13] \leq \frac{q_m}{2^n}.$$  \hspace{1cm} (27)

The proof follows from Eqn. (14)-Eqn. (27).

**Acknowledgement:** We would like to thank all the anonymous reviewers of Africacrypt 2020. Mridul Nandi is supported by NTRO Project.

**References**

1. Daniel J. Bernstein, Stefan Kölbl, Stefan Lucks, Pedro Maat Costa Massolino, Florian Mendel, Kashif Nawaz, Tobias Schneider, Peter Schwabe, François-Xavier Standaert, Yosuke Todo, and Benoît Viguier. Gimli : A cross-platform permutation. In *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, pages 299–320, 2017.

2. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 313–314, 2013.

3. Tim Beyne, Yu Long Chen, Christoph Dobraunig, and Bart Mennink. Elephant. *NIST LWC*, 2019.

4. Hanno Böck, Aaron Zauner, Sean Devlin, Juraj Somorovsky, and Philipp Jovanovic. Nonce-disrespecting adversaries: Practical forgery attacks on GCM in TLS. In *10th USENIX Workshop on Offensive Technologies, WOOT 16, Austin, TX, USA, August 8-9, 2016.*, 2016.

5. Andrey Bogdanov, Miroslav Knezevic, Gregor Leander, Deniz Toz, Kerem Varici, and Ingrid Verbauwhede. SPONGENT: the design space of lightweight cryptographic hashing. *IEEE Trans. Computers*, 62(10):2041–2053, 2013.

6. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Francois-Xavier Standaert, John Steinberger, and Elmar Tischhauser. Key-alternating ciphers in a provable setting: Encryption using a small number of public permutations. In *Advances in Cryptology – EUROCRYPT 2012*, pages 45–62, 2012.

7. Avik Chakraborti, Nilanjan Datta, Mridul Nandi, and Kan Yasuda. Beetle family of lightweight and secure authenticated encryption ciphers. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):218–241, 2018.

8. Bishwajit Chakraborty and Mridul Nandi. Orange. *NIST LWC*, 2019.

9. Shan Chen, Rodolphe Lampe, Jooyoung Lee, Yannick Seurin, and John P. Steinberger. Minimizing the two-round even-mansour cipher. In *Advances in Cryptology - CRYPTO 2014*, pages 39–56, 2014.
10. Yu Long Chen, Eran Lambooij, and Bart Mennink. How to build pseudorandom functions from public random permutations. In Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I, pages 266–293, 2019.
11. Benoît Cogliati and Yannick Seurin. EWCMD: an efficient, beyond-birthday secure, nonce-misuse resistant MAC. In CRYPTO 2016, Proceedings, Part I, pages 121–149, 2016.
12. Joan Daemen, Seth Hoffert, Michal Peeters, Gilles Van Assche, and Ronny Van Keer. Xoodyak, a lightweight cryptographic scheme. NIST LWC, 2019.
13. Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Kan Yasuda. Encrypt or decrypt? to make a single-key beyond birthday secure nonce-based MAC. In Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I, pages 631–661, 2018.
14. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2. NIST LWC, 2019.
15. Avijit Dutta, Ashwin Jha, and Mridul Nandi. Tight security analysis of ehtm MAC. IACR Trans. Symmetric Cryptol., 2017(3):130–150, 2017.
16. Avijit Dutta, Mridul Nandi, and Suprita Talnikar. Beyond birthday bound secure MAC in faulty nonce model. In Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I, pages 437–466, 2019.
17. Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. J. Cryptology, 10(3):151–162, 1997.
18. Jian Guo, Thomas Peyrin, and Axel Poschmann. The PHOTON family of lightweight hash functions. In Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings, pages 222–239, 2011.
19. Viet Tung Hoang and Stefano Tessaro. Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I, pages 3–32, 2016.
20. Viet Tung Hoang and Stefano Tessaro. The multi-user security of double encryption. In Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II, pages 381–411, 2017.
21. Kazuhiko Minematsu. How to thwart birthday attacks against macs via small randomness. In Fast Software Encryption, FSE 2010, pages 230–249, 2010.
22. Alexander Moch and Eik List. Parallelizable macs based on the sum of prps with security beyond the birthday bound. In Applied Cryptography and Network Security - 17th International Conference, ACNS 2019, Bogota, Colombia, June 5-7, 2019, Proceedings, pages 131–151, 2019.
23. NIST. Lightweight cryptography, 2018. Online: https://csrc.nist.gov/Projects/Lightweight-Cryptography. Accessed: August 01, 2019.
24. Phillip Rogaway, Mihir Bellare, and John Black. Sha-3 standard. ACM Transactions on Information and System Security (TISSEC), 6(3):365–403, 2003.
25. Mark N. Wegman and Larry Carter. New hash functions and their use in authentication and set equality. J. Comput. Syst. Sci., 22(3):265–279, 1981.