IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2020, No. 4, pp. 39–70. DOI:10.46586/tosc.v2020.i4.39-70

# Highly Secure Nonce-based MACs from the Sum of Tweakable Block Ciphers

Wonseok Choi[1], Akiko Inoue[2], Byeonghak Lee[1], Jooyoung Lee[1], Eik List[3], Kazuhiko Minematsu[2] and Yusuke Naito[4]

[1] Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea
{krwioh,lbh0307,hicalf}@kaist.ac.kr,
[2] NEC Corporation, Kawasaki, Japan
{a_inoue,k-minematsu}@nec.com,
[3] Bauhaus-Universität Weimar, Weimar Germany
<firstname>.<lastname>@uni-weimar.de,
[4] Mitsubishi Electric Corporation, Kamakura, Kanagawa, Japan
Naito.Yusuke@ce.MitsubishiElectric.co.jp

**Abstract.** Tweakable block ciphers (TBCs) have proven highly useful to boost the security guarantees of authentication schemes. In 2017, Cogliati et al. proposed two MACs combining TBC and universal hash functions: a nonce-based MAC called NaT and a deterministic MAC called HaT. While both constructions provide high security, their properties are complementary: NaT is almost fully secure when nonces are respected (i.e., $n$-bit security, where $n$ is the block size of the TBC, and no security degradation in terms of the number of MAC queries when nonces are unique), while its security degrades gracefully to the birthday bound ($n/2$ bits) when nonces are misused. HaT has $n$-bit security and can be used naturally as a nonce-based MAC when a message contains a nonce. However, it does not have full security even if nonces are unique.
This work proposes two highly secure and efficient MACs to fill the gap: NaT2 and eHaT. Both provide (almost) full security if nonces are unique and more than $n/2$-bit security when nonces can repeat. Based on NaT and HaT, we aim at achieving these properties in a modular approach. Our first proposal, Nonce-as-Tweak2 (NaT2), is the sum of two NaT instances. Our second proposal, enhanced Hash-as-Tweak (eHaT), extends HaT by adding the output of an additional nonce-depending call to the TBC and prepending nonce to the message. Despite the conceptual simplicity, the security proofs are involved. For NaT2 in particular, we rely on the recent proof framework for Double-block Hash-then-Sum by Kim et al. from Eurocrypt 2020.

**Keywords:** Provable security · tweakable block cipher · message authentication code · authentication

## 1 Introduction

Message Authentication Codes (MACs) belong to the core algorithms in symmetric-key cryptography as they protect the authenticity and integrity of the communication between two parties that share a secret key. For this purpose, they provide keyed algorithms for the generation and verification of an authentication tag that is sent alongside the message. A variety of MACs exists in practice, many of which are based on block ciphers, such as the the NIST standard CMAC [Dwo05, Dwo16], the ISO/IEC 9797-1 constructions [ISO11], or the 3GPP standards [3GP99]. Nevertheless, the manyfold applications and security desiderata render research in this area still of high interest and progress.


Received: 2020-06-01 Revised: 2020-09-01 Accepted: 2020-11-01 Published: 2020-12-10

A widespread approach of constructing efficient MACs for variable-input-length messages is the combination of a universal hash function $H$ with a pseudorandom function (PRF) $F$. The former is used to reduce the potentially long input to a small hash value, and the latter is to process that hash value to produce a *tag*. Following Handschuh and Preneel [HP08], the classical options are $T \leftarrow F(H(M))$, $T \leftarrow H(M) \oplus F(N)$, or $T \leftarrow F(N \parallel H(M))$, where $M$ is a message, $N$ is a nonce (a value expected to never repeat, to protect against replay), and $T$ is a tag. The tuple $(N, M, T)$ will be sent to the verifier. For instance, the second option is well-known as Wegman-Carter(-Shoup) construction [WC81, Sho96], and employed, e.g., in the Poly1305-AES [Ber05b] or GMAC [Mor07] standards. The principles have been studied in depth [Ber05a]; n evertheless, the approaches suffer from two disadvantages: the security was limited to the birthday bound [Yuv79] (the application to MACs was studied by [PvO95]) of the primitive's block length, that is, an $n$-bit block cipher can provide up to $n/2$-bit security in terms of the data complexity. Moreover, the security can become void whenever a nonce is used for two authentication tags, which can happen by, e.g., a poor randomness source, or by misuse of the protocol.
Higher security guarantees can be desirable for many applications. A small block length, such as 64 bits, of the underlying primitive can render it a practical attack target when used in modes with birthday-bound security, as was illustrated by the recent attacks on popular communication protocols [BL16]. Even worse, Wegman-Carter(-Shoup) constructions risk that the hash-function key becomes known once a hash collision can be detected or a nonce is reused. Furthermore, robustness in the case of potential nonce repetitions can be of high interest to protect users from erroneous implementations.

Beyond-birthday-bound Security. The community has proposed a portfolio of MACs with higher security guarantees, namely the beyond-birthday-bound (BBB) security. The first such approach was probably suggested by [ISO99], which contained six CBC-like MACs. For higher security, it recommends to XOR two single-pass MACs under independent keys. Though, the analysis was given in [SW19]. Already at the beginning of the previous decade, Yasuda proposed and analyzed SUM-ECBC [Yas10]. Many works followed this direction, including but not limited to 3kf9 [ZWSW12], PMAC$^+$ [Yas11], LightMAC_Plus [Nai17], or 1k_PMAC$^+$ [DDN$^+$17]. Many of those double-block constructions were shown to be secure for up to at least $O(2^{2n/3})$ authentication queries. Datta et al. [DDNP18] coined the term Double-Block Hash-then-Sum (DbHtS) for this approach in general. Leurent et al. [LNS18] proposed generic attacks on DbHtS constructions with a query complexity of $O(2^{3n/4})$. Very recently, Kim et al. [KLL20] showed that the bound of $O(2^{3n/4})$ queries for DbHtS MACs is tight.

Tweakable block ciphers (TBCs) [LRW02]. Tweakable block ciphers (TBCs) enrich the domains of classical block ciphers by an additional public input called the tweak. Thus, they could effectively increase the security and/or to simplify the design of modes based on block ciphers by providing an input that cleanly separates several domains. While TBCs have originally been built from block ciphers [LRW02, Rog04], the increasing number of existing dedicated TBCs, such as Deoxys-BC [JNP14a] or Skinny [BJK$^+$16], allow efficient instantiations. Nowadays, these are attractive primitives for the construction of highly secure modes from TBCs. For example, deterministic (i.e. there is no nonce), $n$-bit secure MACs those solely based on $n$-bit block TBCs[1] have been studied in the literature [Nai15, IMPS17, LN17, Nai18].
In a different approach, Cogliati et al. [CLS17] presented compact designs for $n$-bit secure nonce-based/deterministic MACs that exploited $n$-bit block TBCs combined with a universal hash function. They used a (first) hash of the message with either a nonce or a second hash under an independent key as state and tweak inputs to a TBC. Accordingly, their nonce-based and deterministic constructions were called Nonce-as-Tweak (NaT) and

[1] The tweak length is also assumed to be $n$ bits.

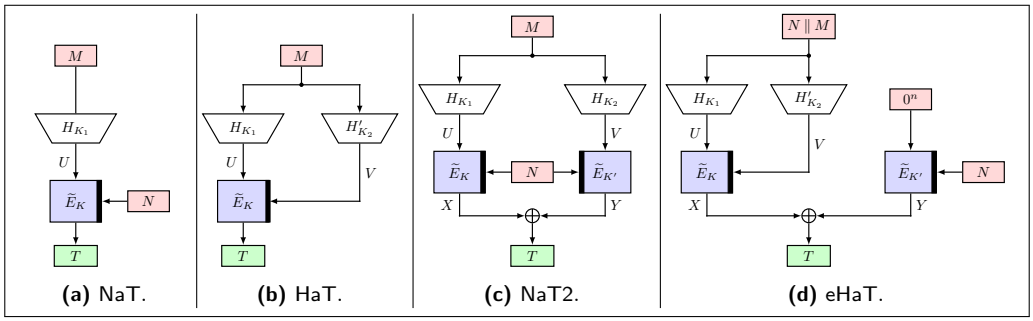



**Figure 1:** The previous constructions NaT and HaT [CLS17] and our proposals NaT2 and eHaT.

a Hash-as-Tweak (HaT), respectively.[2]

Due to their attractive simplicity, the constructions by Cogliati et al. pose an interesting research question: *How can we strengthen their security with (at least conceptually) minimal changes?* In more detail, NaT is almost fully secure when nonces are unique, i.e. under nonce-respecting (NR) adversaries, as its security bound is $O(v\delta)$, where $v$ denotes the maximal number of verification queries, and $\delta$ denotes the bound of the universal hash function for almost uniformity[3]. Thus, there is no contribution from MAC (tagging) queries in this setting. Note that this is the optimal security for a MAC scheme with $n$-bit tags. However, its security degrades gracefully to the birthday bound of $n/2$ bits when nonces repeat among MAC queries, i.e., under nonce-misusing (NM) adversaries.

HaT can be trivially converted into a nonce-based MAC by prepending a nonce to the message (e.g., by using $N \| M$ instead of $M$). In this case, its security in the NM setting is unchanged. On the downside, its security in the NR setting degrades concerning the number of MAC queries. Thus, it cannot achieve full security per se. Therefore, when used as nonce-based MACs, their security properties are complementary. Our goal is to achieve the best of both worlds by simple changes to the base constructions. We believe that this helps understand strong MAC constructions in general.

Our Contributions. We answer the aforementioned question by two novel constructions. Nonce-as-Tweak 2 (NaT2) is a sum of two instances of NaT. As a result, it is (almost) fully secure under nonce-respecting adversaries if the underlying TBCs are ideal – a property shared with NaT. However, in the nonce-misuse setting, its security degrades gracefully to $2n/3$ bits – instead of $n/2$ bits as NaT. If the number of verification queries is limited to $2^{n/2}$, NaT2 can effectively ensure $3n/4$-bit security, which is useful in some applications, e.g., that terminate communication when a number of verification failures are detected.

Our second construction, enhanced Hash-as-Tweak (eHaT), extends HaT by adding the result of a nonce-dependent call of the TBC to its output of HaT, in addition to attaching the nonce to the message. This simple and generic approach leads also to (almost) full NR-security while maintaining $n$-bit NM-security with graceful degradation. In general, eHaT offers stronger security than NaT2. However, this implies some costs, such as an increased input length to the universal hash functions, a non-static tweak input to TBCs, and a certain limitation on the maximal allowable number of MAC queries (see Section 6). On the other hand, the second TBC call of eHaT can be substituted by a call to a PRF if available. Both constructions are illustrated in Figure 1.

The security bounds in Table 1 reflect our explanations above. The only security terms in the nonce-respecting setting for NaT2 and eHaT depend on the number of verification queries and the properties of the hash-function. The bound NaT2 in the nonce-repeating

[2] They also considered counterparts based on ideal ciphers, which is beyond the scope of this paper.

[3] Assuming the universal hash function is $n$-bit polynomial hash function, we can set $\delta = \ell/2^n$ for $\ell$-block inputs, or $\ell/2^t$ if the output size is $t$ bits.

**Table 1:** Comparison of existing TBC- and nonce-based MACs with BBB security with our proposal assuming an $n$-bit, $O(\ell/2^n)$-AU polynomial hash function using $n$-bit hash keys (and $t$-bit, $O(\ell/2^t)$-AU polynomial hash function using $t$-bit keys when used for tweaks) for $\ell$-block inputs, and a TBC with $n$-bit state and $t$-bit tweaks. **NR** = nonce-respecting setting; **NM** = nonce-misusing setting. $\mu$ = #faulty nonces (Section 2), $q$ = #MAC queries, $v$ = #verification queries, $\sigma$ = #blocks in MAC queries, **#Mults.** = #field multiplications, **#TBC** = #TBC calls for $\ell$-block messages, **State size** in bits, $^{(*)}$ = deterministic MACs have no $\mu$ term, their mac bound is shown in the NM setting. $w$ = Size of the TBC.

| | **Security** ($O(\cdot)$) | | **Efficiency** | | | |
|---|---|---|---|---|---|---|
| **Construction** | **NR** | **NM** | **#Mults.** | **#TBC** | **#Key bits** | **State size** |
| EPWC [PS16] | $\frac{(q+v)}{2^n}$ | $\frac{(\mu^2+v)}{2^n}$ | – | $\ell+3$ | $k$ | $w+n$ |
| PMAC_TBC1k$^{(*)}$ [Nai15] | – | $\frac{q+v}{2^n}+\frac{q^2}{2^{2n}}$ | – | $\ell+2$ | $k$ | $w+2n$ |
| PMAC_TBC3k$^{(*)}$ [Nai15] | – | $\frac{q^2}{2^{2n}}+\frac{v}{2^n}$ | – | $\ell+2$ | $3k$ | $w+2n$ |
| ZMAC$^{(*)}$ [IMPS17] | – | $\frac{\sigma^2}{2^{n+\min(n,t)}}+(\frac{q}{2^n})^{1.5}+\frac{v}{2^n}$ | – | $\frac{\ell n}{(n+t)}+4$ | $k$ | $w+4n$ |
| ZMACb/ZMACt$^{(*)}$ [Nai18] | – | $\frac{(q^2+v)}{2^n}$ | – | $\frac{(\ell-1)n}{(n+t)}+1$ | $k$ | $w+4n$ |
| ZMAC1$^{(*)}$ [Nai18] | – | $\frac{\sigma^2}{2^{n+\min(n,t)}}+\frac{(q+v)}{2^n}$ | – | $\frac{\ell n}{(n+t)}+2$ | $k$ | $w+4n$ |
| NaT [CLS17] | $\frac{v\ell}{2^n}$ | $\frac{\mu(q+v)\ell}{2^n}$ | $\ell-1$ | 1 | $n+k$ | $\max\{w, n+t\}$ |
| NaT2 [**Sect.** 5] | $\frac{v\ell}{2^n}$ | $\frac{\mu^2v\ell^2}{2^{2n}}+\frac{\mu^2\ell^{1.5}}{2^{1.5n}}+\frac{v\ell}{2^n}$ | $2(\ell-1)$ | 2 | $2n+2k$ | $2\max\{w, n+t\}$ |
| HaT$^{(*)}$ [CLS17] | – | $\frac{(q^2+qv)\ell^2}{2^{n+t}}+\frac{v}{2^n}$ | $2(\ell-1)$ | 1 | $n+t+k$ | $\max\{w, 2n+2t\}$ |
| eHaT [**Sect.** 6] | $\frac{v}{2^n}+\frac{v\ell^2}{2^{n+t}}$ | $\frac{(\mu^2+\mu v+v)\ell^2}{2^{n+t}}+\frac{v}{2^n}$ | $2\ell$ | 2 | $n+t+2k$ | $w+\max\{w, 2n+2t\}$ |

setting is considerably stronger than that of NaT. The costs for the additional hash-function, TBC call, and the additional key for NaT2 are illustrated in the efficiency part. Similarly, the bound for eHaT is stronger in both settings, at the cost of an additional TBC call, two more multiplications in the hash function for the prepended nonce, and a second TBC key. We assumed that the underlying universal hash function is a polynomial hash function of $O(\ell/2^n)$-almost-universal for an $n$-bit output (or $\ell/2^t$-almost-universal for a $t$-bit output) for at most $\ell$ input blocks. Such a polynomial hash is well-known to need $(m-1)$ field multiplications over $\mathsf{GF}(2^n)$ for $m$-block inputs. We note that $\ell/2^n$ can be reduced by using a different universal hash function: an inner-product hash function achieves optimal $1/2^n$-almost-universality. As a disadvantage, the key length has to match that of the message. A better trade-off between collision probability and key size is possible, e.g., using the proposal by Sarkar [Sar11], who suggested multi-stage hash functions in the spirit of VHASH [Kro06]. For example, a two-stage hash construction with two polynomial hashes, a standard block length of $n = 128$ bits, and a maximal message length of $2^{32}$ words would be $2^{-96}$-almost-universal with only two 128-bit keys.

Computations vs. Security. Our constructions have a certain computational overhead, i.e., a second hash-function evaluation for NaT2 compared to NaT and the additional processing of the nonce in the hash function in eHaT compared to HaT, respectively. Plus, our constructions need one additional (parallelizable) call to the TBC each. Although the computational costs are increased, we believe that our constructions are close to the minimum for achieving our security goals. If a nonce is available, full nonce-respecting security can be obtained by using the nonce as tweak one TBC call. For security under repeating nonces with variable-length messages, a single $n$-bit hash can collide at the birthday bound. If the output size of the polynomial hash is fixed to $n$ bits, one needs a second hash call to produce more than $n$ bits of hash material.[4] Similarly, if the tweak space is limited to $n$ bits, two TBC calls are needed to process a $2n$-bit hash and an $n$-bit nonce. Thus, we consider our constructions minimal.

Use of Extended Mirror Theory. We note that our proof of NaT2 develops a variant of the Extended Mirror Theory [DDNY18] further, which itself advanced Patarin's

[4] Block-cipher-based hashes can produce longer hash outputs more efficiently. However, this holds also if they were used in our constructions. For comparison, we focus on polynomial hash functions hereafter.

famous Mirror Theory [Pat10, Pat17] by adding inequalities, which are necessary to address failed verification queries in the mac setting. Our approach can prove for the first time a security level of $3n/4$ bits for a system of equalities and inequalities, whereas earlier works [DDNY18, DNT19] showed at most $2n/3$-bit security.

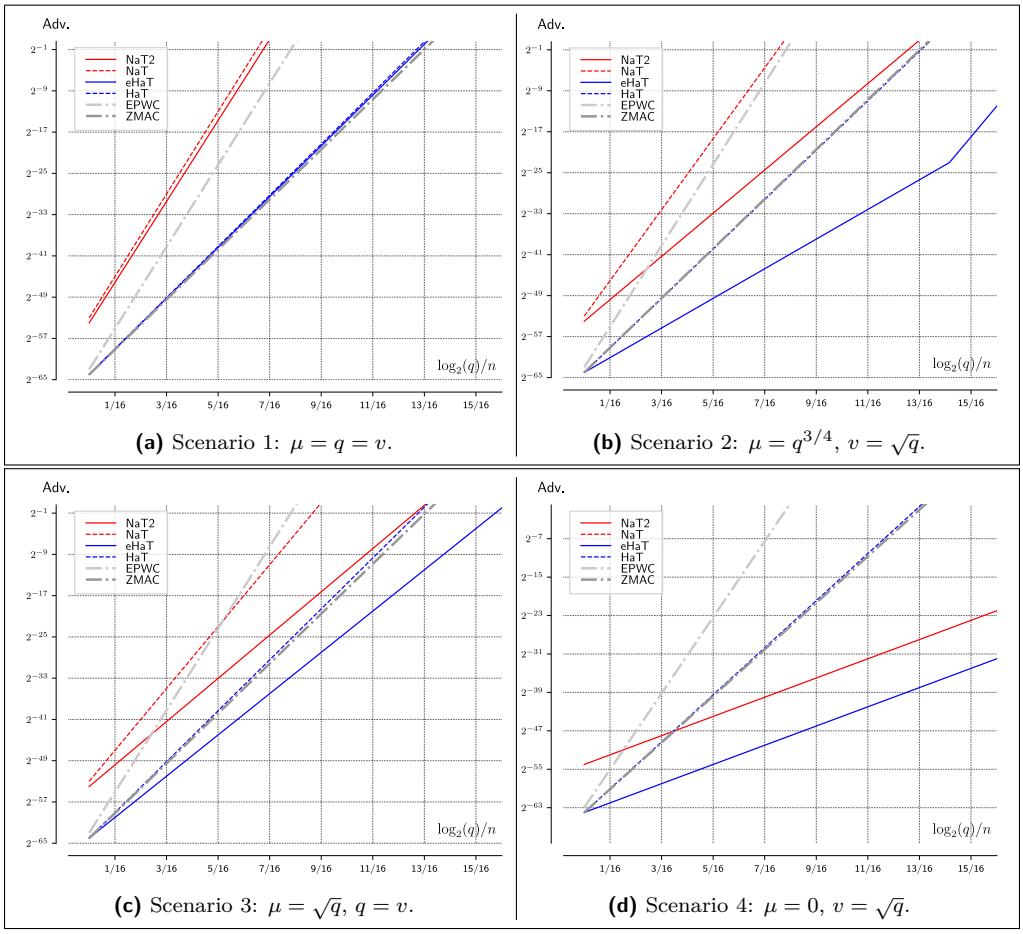
Generalized Tweak Lengths. NaT2 and eHaT provide security advantages compared to NaT and HaT, respectively, not only when $t = n$ but in a more general setting. For smaller tweaks, $t < n$, the bounds are comparable. Though, they are better if the tweak length can exceed the block length, $t > n$, as is possible in practice, e.g., with Deoxys-BC-128-384 [JNP14b] or Skinny-64-192 [BJK+16] (although, the TWEAKEY framework unifies key and tweak [JNP14b]). For a concrete example, assume $t = 2n$; in this case, NaT2 is secure for up to $q \in O(2^{2n})$ queries in the nonce-respecting and $\mu \in O(2^{3n/4})$ queries under nonce repetitions – assuming for simplicity that the adversary does not ask too many verification queries $v \ll 2^n$ and the hash functions are universal for all constructions. The security of NaT is capped at $q \in O(2^n)$ and $\mu \in O(2^{n/2})$. Similarly, the security of eHaT under nonce-respecting adversaries depends only on $v$, and scales up to $\mu \in O(2^{(n+t)/2})$ nonce-misusing queries. The security of HaT is limited by $O(2^{(n+t)/2})$ queries in both settings, and thus cannot benefit from nonces.
As shown in Figure 2d, NaT2 and eHaT offer higher security than deterministic MACs (e.g. ZMAC) when $\mu$ is small and $v \ll q$, which we think is a reasonable assumption. Compared to the nonce-based NaT, NaT2 is more secure when $0 < \mu \ll q^{3/4}$. eHaT is more secure for a broader range of $\mu$. Moreover, NaT2 and eHaT are the only constructions that are still secure for $q > 2^n$ queries when the tweak space is large enough.

Security Comparison. For better illustration, we compare the security of the constructions in four scenarios in Figure 2, for $n = t = 64$: (a) with many nonce repetitions $\mu = q = v$, (b) some repetitions, with $\mu = q^{3/4}$, $v = \sqrt{q}$, and (c) $\mu = \sqrt{q}$ and $v = q$, as well as (d) under nonce-respecting adversaries with $\mu = 0$ and $v = \sqrt{q}$. We comment that (d) mostly keeps its shape even when $\mu$ is a small positive constant. For comparability, we assumed a practical universal hash function with $\epsilon = \ell/2^n$ for $\ell = 2^{10}$ blocks as a practical standard size. We also included EPWC and ZMAC for comparison, whose TBC-based hash functions are close to optimally almost universal. The dashed lines red and blue curves represent NaT and HaT, the solid ones NaT2 and eHaT, respectively.
Note that we considered only constructions based on tweakable block ciphers. For example, while the DbHtS constructions [LNS18] are comparable in structure, they are built from classical block ciphers. Since those are weaker primitives, comparing with those constructions would be unfair to our advantage.

Outline. After Section 2 briefly recalls the necessary notations and definitions, Section 3 describes our proposed constructions NaT2 and eHaT. In Section 4, we provide what we call the extended mirror theory, which plays a key role in our analysis of NaT2, and several proofs of the lemmas. We start our analysis in Section 5 with NaT2, followed by that of eHaT in Section 6. Section 7 concludes this work.

## 2 Preliminaries

Notation. We fix a positive integer $n$ such that $n \geq 3$. We denote $0^n$ (i.e., $n$-bit string of all zeros) by $\mathbf{0}$. For positive integers $p \leq q$, we write $[q] = \{1, \ldots, q\}$ and $[p..q] = \{p, p+1, \ldots, q\}$. Given a finite non-empty set $\mathcal{X}$, $x \leftarrow_{\$} \mathcal{X}$ denotes that $x$ is chosen uniformly at random from $\mathcal{X}$. The set of all sequences that consist of $b$ pairwise distinct elements of $\mathcal{X}$ is denoted $\mathcal{X}^{*b}$. For integers $1 \leq b \leq a$, we will write $(a)_b = a(a-1)\cdots(a-b+1)$ and $(a)_0 = 1$ by convention. If $|\mathcal{X}| = a$, then $(a)_b$ becomes the size of $\mathcal{X}^{*b}$.
When two sets $\mathcal{X}$ and $\mathcal{Y}$ are disjoint, their (disjoint) union is denoted $\mathcal{X} \sqcup \mathcal{Y}$. For a




**(a)** Scenario 1: $\mu = q = v$. **(b)** Scenario 2: $\mu = q^{3/4}$, $v = \sqrt{q}$.

**(c)** Scenario 3: $\mu = \sqrt{q}$, $q = v$. **(d)** Scenario 4: $\mu = 0$, $v = \sqrt{q}$.

**Figure 2:** Security comparison of existing schemes with NaT2 and eHaT in four scenarios with different number of repeating-nonce queries $\mu$ and verification queries $v$.

set $\mathcal{X} \subset \{0,1\}^n$ and $\lambda \in \{0,1\}^n$, we will write $\mathcal{X} \oplus \lambda = \{x \oplus \lambda : x \in \mathcal{X}\}$. For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we will interchangeably write $|\mathcal{V}|$ and $|\mathcal{G}|$ for the number of vertices of $\mathcal{G}$.

Universal Hash Functions. Let $\delta > 0$, and let $H : \mathcal{K} \times \mathcal{M} \to \mathcal{X}$ be a keyed function for three non-empty sets $\mathcal{K}$, $\mathcal{M}$, and $\mathcal{X}$. $H$ is said to be *$\delta$-almost universal* (AU) if for any distinct $M$, $M' \in \mathcal{M}$, it holds that

$$\Pr\left[K \leftarrow_{\$} \mathcal{K} : H_K(M) = H_K(M')\right] \leq \delta\,.$$

For a positive integer $q$, fix $M_1, \ldots, M_q \in \mathcal{M}$. For a random key $K \in \mathcal{K}$, let $X_i = H_K(M_i)$ for $i = 1, \ldots, q$. Then we can define an equivalence relation $\sim$ on $[q]$: for $\alpha$, $\beta \in [q]$, $\alpha \sim \beta$ if and only if $X_\alpha = X_\beta$. For some nonnegative integer $r$, let $\mathcal{P}_1, \ldots, \mathcal{P}_r$ denote the equivalence classes of $[q]$ with respect to $\sim$ such that $p_i := |\mathcal{P}_i| \geq 2$ for $i = 1, \ldots, r$. Jha and Nandi [JN20] proved the following lemma, which is also useful in our security proof.

**Lemma 1.** Let $p_i$, $i = 1 \ldots, r$, be the random variables as defined above. Then we have

$$\mathbf{E}\left[\sum_{i=1}^{r} p_i^2\right] \leq 2q^2\delta\,,$$

where the expectation is taken over the uniform distribution of $K \in \mathcal{K}$.

*Proof.* Let $c$ denote the random variable that counts the number of "$X$-colliding" pairs. More precisely,

$$c \coloneqq \left|\left\{(i,j) \in [q]^2 : i < j \text{ and } X_i = X_j\right\}\right|.$$

Then it is easy to show that

$$\sum_{i=1}^{r} p_i^2 = 2c + \sum_{i=1}^{r} p_i \leq 4c\,.$$

Furthermore, we have $\mathbf{E}[c] \leq \binom{q}{2}\delta$, which completes the proof. □

Thus, Lemma 1 says that the number of collisions is limited by $2q^2\delta$ on expectation. Moreover, the corollary below yields an upper bound on the number of occurrences of any single hash value. The proof in [JN20] stems from Markov's inequality.

**Corollary 1** (Corollary 4.1 in [JN20])**.** Let $p_{\max} = \max\{p_i : i \in [r]\}$. Then, for $a \geq 1$, it holds that $\Pr[p_{\max} \geq a] \leq \frac{2q^2\delta}{a^2}$.

In our security proof, we also need to upper bound the probability of three hash collisions with two independent hash keys. With a smaller number of hash keys, the three collisions are not independent of each other anymore. To address this situation, one should carefully upper bound the number of colliding pairs made by a single key, and then use the randomness of the second hash key; Jha and Nandi [JN20] proved the following lemma.

**Lemma 2** (Alternating Collisions Lemma in [JN20])**.** Let $q$ be a positive integer, let $\delta > 0$, and let $H : \mathcal{K} \times \mathcal{M} \to \mathcal{X}$ be a $\delta$-almost universal hash function. Then, for any $(M_1, \ldots, M_q) \in \mathcal{M}^{*q}$, we have

$$\Pr\left[K_1, K_2 \leftarrow_{\$} \mathcal{K} : \exists (i,j,k,l) \in [q]^{*4}, H_{K_1}(M_i) = H_{K_1}(M_j) \right. \\ \left. \wedge H_{K_2}(M_j) = H_{K_2}(M_k) \wedge H_{K_1}(M_k) = H_{K_1}(M_l)\right] \leq q^2\delta^{3/2}\,.$$

Tweakable Block Ciphers. A tweakable permutation with tweak space $\mathcal{N}$ and message space $\mathcal{X}$ is a mapping $\widetilde{\pi} : \mathcal{N} \times \mathcal{X} \to \mathcal{X}$ such that, for any tweak $N \in \mathcal{N}$, $X \mapsto \widetilde{\pi}(N, X)$ is a permutation of $\mathcal{X}$.

A tweakable block cipher [LRW02] $\widetilde{E}$ with key space $\mathcal{K}$, tweak space $\mathcal{N}$ and message space $\mathcal{X}$ is a mapping $\widetilde{E} : \mathcal{K} \times \mathcal{N} \times \mathcal{X} \to \mathcal{X}$ such that for any key $K \in \mathcal{K}$, $(N, X) \mapsto \widetilde{E}(K, N, X)$ is a tweakable permutation with tweak space $\mathcal{N}$ and message space $\mathcal{X}$. Note that the tweak is public and can be chosen freely for every query by the adversary as long as a scheme does not restrict its usage otherwise. We will sometimes write $\widetilde{E}_K(N, X)$ to denote $\widetilde{E}(K, N, X)$. We also write $\mathsf{TPerm}(\mathcal{N}, \mathcal{X})$ to mean the set of all tweakable permutations with tweak space $\mathcal{N}$ and message space $\mathcal{X}$.

To analyze the security of a tweakable block cipher $\widetilde{E} : \mathcal{K} \times \mathcal{N} \times \{0,1\}^n \to \{0,1\}^n$, we consider a distinguisher $\mathbf{A}$ whose goal is to tell apart the *real* world and the *ideal* world; in the real world, $\mathbf{A}$ is given oracle access to $\widetilde{E}_K$ where a secret key $K \in \mathcal{K}$ is chosen uniformly at random. In the ideal world, $\mathbf{A}$ is given a random tweakable permutation $\widetilde{\pi} \in \mathsf{TPerm}(\mathcal{N}, \{0,1\}^n)$ instead of $\widetilde{E}_K$. In any world, the adversary is allowed to adaptively make forward and backward queries to the oracle. Formally, $\mathbf{A}$'s $\mathsf{tprp}$ (tweakable pseudorandom permutation) advantage is defined by

$$\mathsf{Adv}^{\mathsf{tprp}}_{\widetilde{E}}(\mathbf{A}) = \left|\Pr\left[\widetilde{\pi} \leftarrow_{\$} \mathsf{TPerm}(\mathcal{N}, \{0,1\}^n) : 1 \leftarrow \mathbf{A}^{\widetilde{\pi}}\right] - \Pr\left[K \leftarrow_{\$} \mathcal{K} : 1 \leftarrow \mathbf{A}^{\widetilde{E}_K}\right]\right|.$$

We define $\mathsf{Adv}^{\mathsf{tprp}}_{\widetilde{E}}(q,t)$ as the maximum of $\mathsf{Adv}^{\mathsf{tprp}}_{\widetilde{E}}(\mathbf{A})$ over all distinguishers $\mathbf{A}$ against $\widetilde{E}$ making at most $q$ encryption oracle queries and running in time at most $t$. When we consider information-theoretic security, we will drop the parameter $t$.

NONCE-BASED MACS. Given four non-empty sets $\mathcal{K}$, $\mathcal{N}$, $\mathcal{M}$, and $\mathcal{T}$, a nonce-based keyed function with key space $\mathcal{K}$, nonce space $\mathcal{N}$, message space $\mathcal{M}$ and tag space $\mathcal{T}$ is simply a function $F : \mathcal{K} \times \mathcal{N} \times \mathcal{M} \to \mathcal{T}$. Stated otherwise, it is a keyed function whose domain is a cartesian product $\mathcal{N} \times \mathcal{M}$. We denote $F_K(N, M)$ for $F(K, N, M)$.
For $K \in \mathcal{K}$, let $\mathsf{Auth}_K$ be the $\mathsf{mac}$ oracle which takes as input a pair $(N, M) \in \mathcal{N} \times \mathcal{M}$ and returns $F_K(N, M)$, and let $\mathsf{Ver}_K$ be the verification oracle which takes as input a triple $(N, M, T) \in \mathcal{N} \times \mathcal{M} \times \mathcal{T}$ and returns 1 ("accept") if $F_K(N, M) = T$, and 0 ("reject") otherwise. We assume that an adversary makes queries to the two oracles $\mathsf{Auth}_K$ and $\mathsf{Ver}_K$ for a secret key $K \in \mathcal{K}$. A MAC query $(N, M)$ made by an adversary is called a *faulty query* if the adversary has already queried to the $\mathsf{mac}$ oracle with the same nonce but with a different message (cf. [DNT19]). For example, if $i$-th query is denoted by $(N_i, M_i)$ and there are four distinct queries, $(N_i, M_i)$ for $i \in [4]$ such that $N_1 \neq N_2 = N_3 = N_4$, the third and the fourth queries are faulty and the number of faulty queries is two. We would like to emphasize that the term of faulty queries to provide consistency for readers familiar with [DNT19], where it characterized faulty implementations or environments that led to repeating nonces. It does not represent faults from processing errors or side-channel attacks.
A $(\mu, q, v, t)$-adversary against the nonce-based $\mathsf{mac}$ security of $F$ is an adversary $\mathbf{A}$ with oracle access to $\mathsf{Auth}_K$ and $\mathsf{Ver}_K$, making at most $q$ MAC queries to its first oracle with at most $\mu$ faulty queries and at most $v$ verification queries to its second oracle, and running in time at most $t$. We say that $\mathbf{A}$ forges if any of its queries to $\mathsf{Ver}_K$ returns 1. The advantage of $\mathbf{A}$ against the nonce-based $\mathsf{mac}$ security of $F$ is defined as

$$\mathsf{Adv}^{\mathsf{mac}}_F(\mathbf{A}) = \Pr\left[K \leftarrow_{\$} \mathcal{K} : \mathbf{A}^{\mathsf{Auth}_K, \mathsf{Ver}_K} \text{ forges}\right] ,$$

where the probability is also taken over the random coins of $\mathbf{A}$, if any. The adversary is not allowed to ask a verification query $(N, M, T)$ if a previous MAC query $(N, M)$ to $\mathsf{Auth}_K$ returned $T$. However, it is still possible that a verification query $(N, M, T)$ is first made, possibly rejected, and a MAC query $(N, M)$ is subsequently made.
When $\mu = 0$, we say that $\mathbf{A}$ is nonce-respecting, that is, all nonces in the MAC queries are unique. Otherwise, $\mathbf{A}$ is called nonce-misusing. However, the adversary is always allowed to repeat nonces in its verification queries and reuse a nonce from a previous MAC query.
We define $\mathsf{Adv}^{\mathsf{mac}}_F(\mu, q, v, t)$ as the maximum of $\mathsf{Adv}^{\mathsf{mac}}_F(\mathbf{A})$ over all $(\mu, q, v, t)$-adversaries. When we consider information-theoretic security, we will drop the parameter $t$.
This work shows the $\mathsf{mac}$ security of $\mathsf{NaT2}$ and $\mathsf{eHaT}$ using the advantage of an adversary trying to distinguish the real world $(\mathsf{Auth}_K, \mathsf{Ver}_K)$ and the ideal world. The ideal world oracles are $(\mathsf{Rand}, \mathsf{Rej})$, where $\mathsf{Rand}$ returns an independent random value (instantiating a truly random function) and $\mathsf{Rej}$ returns 0 for every verification query. Then,

$$\mathsf{Adv}^{\mathsf{mac}}_F(\mu, q, v, t) \leq \max_{\mathbf{A}} \left( \Pr\left[K \leftarrow_{\$} \mathcal{K} : 1 \leftarrow \mathbf{A}^{\mathsf{Auth}_K, \mathsf{Ver}_K}\right] - \Pr\left[1 \leftarrow \mathbf{A}^{\mathsf{Rand}, \mathsf{Rej}}\right] \right) ,$$

whereas for $\mathsf{mac}$ security, an adversary makes at most $q$ MAC queries to its first oracle with at most $\mu$ faulty queries and at most $v$ verification queries to its second oracle, runs in time at most $t$, and returns a decision bit. The detail of obtaining the bound is given in Section 2.3 of [CLS17].

EXPECTATION METHOD. Consider a construction $F$ based on a universal hash function $H$ and a tweakable block cipher $\widetilde{E}$ using keys $(K_h, K)$. Suppose, a distinguisher $\mathbf{A}$ adaptively makes $q$ MAC queries and $v$ verification queries to either $(\mathsf{Auth}_{K_h,K}, \mathsf{Ver}_{K_h,K})$ for a random secret key $(K_h, K) \in \mathcal{K}_h \times \mathcal{K}$ (in the real world) or $(\mathsf{Rand}, \mathsf{Rej})$ (in the ideal world), where $\mathsf{Rand}$ returns an independent random value (instantiating a truly random function) and $\mathsf{Rej}$ return 0 for every verification query. Moreover, $\mathbf{A}$ records all the queries in

$$\tau_m \stackrel{\text{def}}{=} ((N_1, M_1, T_1), \ldots, (N_q, M_q, T_q)) ,$$

$$\tau_v \stackrel{\text{def}}{=} ((N'_1, M'_1, T'_1, b'_1), \ldots, (N'_v, M'_v, T'_v, b'_v)) \ ,$$

where either $\mathsf{Auth}_{K_h,K}(N_i, M_i) = T_i$ or $\mathsf{Rand}(N_i, M_i) = T_i$ for $i = 1, \ldots, q$, and either $\mathsf{Ver}_{K_h,K}(N'_i, M'_i, T'_i) = b'_i$ or $\mathsf{Rej}(N'_i, M'_i, T'_i) = b'_i (= 0)$ for $i = 1, \ldots, v$, according to the world that $\mathbf{A}$ interacts with.

As a common means to alleviate the proof, we will provide the distinguisher $\mathbf{A}$ with additional information $\tau_a$ (e.g. hash key $K_h$) for free after $\mathbf{A}$ has finished its interaction with the oracles, but before it releases its output decision bit. Thus, $\mathbf{A}$ can compute all inputs to the internal primitives itself. In the ideal world, that information will be selected uniformly at random from the appropriate domain and given to $\mathbf{A}$. This will not degrade the adversarial distinguishing advantage since the distinguisher is free to ignore this additional information. We will call

$$\tau = (\tau_a, \tau_m, \tau_v)$$

the *transcript* of the attack; it contains all information that $\mathbf{A}$ has obtained at the end of the attack. When we consider an information-theoretic distinguisher, we can assume that the distinguisher is deterministic without making any redundant query.

A transcript $\tau$ is called *attainable* if the probability to obtain this transcript in the ideal world is non-zero. Note that any key $K_h \in \mathcal{K}_h$ and any sequence of tags $(T_1, \ldots, T_q) \in (\{0,1\}^n)^q$ uniquely determine an attainable transcript containing them, and each attainable transcript appears in the ideal world with the same probability, namely $1/(2^n)^q$. We denote $\Gamma$ the set of attainable transcripts. We also denote $\mathsf{T}_{\mathsf{re}}$ (resp. $\mathsf{T}_{\mathsf{id}}$) the probability distribution of the transcript $\tau$ induced by the real world (resp. the ideal world). By extension, we use the same notation to denote a random variable distributed according to each distribution.

In this setting, it is obvious that $\mathbf{A}$'s distinguishing advantage upper bounds $\mathbf{A}$'s forging probability. To upper bound the distinguishing advantage, we will use Patarin's H-coefficient technique; we partition the set of attainable transcripts $\Gamma$ into a set of "good" transcripts $\Gamma_{\mathsf{good}}$ such that the probabilities to obtain some transcript $\tau \in \Gamma_{\mathsf{good}}$ are close in the real world and the ideal world, and a set $\Gamma_{\mathsf{bad}}$ of "bad" transcripts such that the probability to obtain any $\tau \in \Gamma_{\mathsf{bad}}$ is small in the ideal world. The lower bound in the probability ratio for obtaining a good transcript in both worlds will be given as a function of $\tau$, and we will take its expectation. This refinement is called the *expectation method*, first introduced in [HT16], summarized in the following theorem.

**Lemma 3.** Fix a distinguisher $\mathbf{A}$. Let $\Gamma = \Gamma_{\mathsf{good}} \sqcup \Gamma_{\mathsf{bad}}$ be a partition of the set of attainable transcripts, and there exists a non-negative function $\varepsilon_1(\tau)$ s. t. for any $\tau \in \Gamma_{\mathsf{good}}$,

$$\frac{\Pr\left[\mathsf{T}_{\mathsf{re}} = \tau\right]}{\Pr\left[\mathsf{T}_{\mathsf{id}} = \tau\right]} \geq 1 - \varepsilon_1(\tau) \, ,$$

and there exists $\varepsilon_2$ such that $\Pr[\mathsf{T}_{\mathsf{id}} \in \Gamma_{\mathsf{bad}}] \leq \varepsilon_2$. Then, one has

$$\mathsf{Adv}^{\mathsf{mac}}_F(\mathbf{A}) \leq \mathbf{E}\left[\varepsilon_1(\tau)\right] + \varepsilon_2 \, , \tag{1}$$

where the expectation is taken over the distribution $\mathsf{T}_{\mathsf{id}}$ in the ideal world.

*Proof.* Since the distinguisher's output is a (deterministic) function of the transcript, its distinguishing advantage[5] is upper bounded by the statistical distance between $\mathsf{T}_{\mathsf{id}}$ and $\mathsf{T}_{\mathsf{re}}$. Thus we have

$$\mathsf{Adv}^{\mathsf{mac}}_F(\mathbf{A}) \leq \|\mathsf{T}_{\mathsf{re}} - \mathsf{T}_{\mathsf{id}}\| \coloneqq \frac{1}{2} \sum_{\tau \in \Gamma} \left|\Pr[\mathsf{T}_{\mathsf{re}} = \tau] - \Pr[\mathsf{T}_{\mathsf{id}} = \tau]\right| \, .$$

[5] For simplicity we took the specific notion ($\mathsf{Adv}^{\mathsf{mac}}$). However, the framework holds for the general distinguishing advantage.

**Algorithm 1** NaT2

101: **function** NaT2$[H_{K_1}, H_{K_2}, \widetilde{E}_K, \widetilde{E}_{K'}](N, M)$
102: $U \leftarrow H_{K_1}(M)$
103: $V \leftarrow H_{K_2}(M)$
104: $X \leftarrow \widetilde{E}_K(N, U)$
105: $Y \leftarrow \widetilde{E}_{K'}(N, V)$
106: $T \leftarrow X \oplus Y$
107: **return** $T$

**Algorithm 2** eHaT

201: **function** eHaT$[H_{K_1}, H'_{K_2}, \widetilde{E}_K, \widetilde{E}_{K'}](N, M)$
202: $U \leftarrow H_{K_1}(N \,\|\, M)$
203: $V \leftarrow H'_{K_2}(N \,\|\, M)$
204: $X \leftarrow \widetilde{E}_K(V, U)$
205: $Y \leftarrow \widetilde{E}_{K'}(N, 0^n)$
206: $T \leftarrow X \oplus Y$
207: **return** $T$

**Figure 3:** Our proposals, NaT2 and eHaT.

Moreover we have:

$$\begin{aligned}
\|\mathsf{T}_{\mathsf{re}} - \mathsf{T}_{\mathsf{id}}\| &= \sum_{\substack{\tau\in\Gamma \\ \Pr[\mathsf{T}_{\mathsf{id}}=\tau]>\Pr[\mathsf{T}_{\mathsf{re}}=\tau]}} (\Pr[\mathsf{T}_{\mathsf{id}} = \tau] - \Pr[\mathsf{T}_{\mathsf{re}} = \tau]) \\
&= \sum_{\substack{\tau\in\Gamma \\ \Pr[\mathsf{T}_{\mathsf{id}}=\tau]>\Pr[\mathsf{T}_{\mathsf{re}}=\tau]}} \Pr[\mathsf{T}_{\mathsf{id}} = \tau] \left(1 - \frac{\Pr[\mathsf{T}_{\mathsf{re}} = \tau]}{\Pr[\mathsf{T}_{\mathsf{id}} = \tau]}\right) \\
&\le \sum_{\tau\in\Gamma_{\mathsf{good}}} \Pr[\mathsf{T}_{\mathsf{id}} = \tau]\varepsilon_1(\tau) + \sum_{\tau\in\Gamma_{\mathsf{bad}}} \Pr[\mathsf{T}_{\mathsf{id}} = \tau] \le \mathbf{E}\left[\varepsilon_1(\tau)\right] + \varepsilon_2\,. \qquad \square
\end{aligned}$$

*Remark* 1. The standard H-coefficient technique [Pat08, CS14] correspond to a special case of the expectation method that requires $\varepsilon_1(\tau)$ is independent of (good) $\tau$. Thus it reduces to $\varepsilon_1$ and the distinguishing advantage is at most $\varepsilon_1 + \varepsilon_2$.

# 3 The NaT2 and eHaT Constructions

This section describes our proposals and discusses their efficiency.

## 3.1 Descriptions

NaT2. Let $H : \mathcal{K}_h \times \mathcal{M} \to \{0,1\}^n$ be a keyed function and let $\widetilde{E} : \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ be a TBC, where $\mathcal{M} = \{0,1\}^*$ denotes the message space and $\mathcal{T} = \{0,1\}^t$ denotes the tweak space. NaT2 is based on them using $\mathcal{T}$ as the nonce space. Specifically, for an input tuple $(N, M) \in \mathcal{T} \times \mathcal{M}$, the $n$-bit tag $T$ is computed as

$$T = \widetilde{E}_K(N, H_{K_1}(M)) \oplus \widetilde{E}_{K'}(N, H_{K_2}(M))\,.$$

This is exactly a sum of two independent instances of NaT. An illustration is given in Figure 1c. The security bounds of NaT2 will be given in Section 5.

eHaT. For eHaT, we extend HaT. Let $H' : \mathcal{K}'_h \times \mathcal{M} \to \mathcal{T}$ be a keyed function. The values $U$ and $V$ are hash values from hash instances of $H$ and $H'$, respectively. In this case, the keyed hash functions take the concatenation $N \,\|\, M$ instead of $M$ as in HaT. The two hash values $V$ and $U$ are given to a TBC to produce $X$, which corresponds to the output of HaT taking $N \,\|\, M$ as the message (recall that HaT is a deterministic MAC). Moreover, there is an additional TBC to process the nonce (as a tweak, the block input is set to a constant) to produce the output $Y$. The sum of $X$ and $Y$ is used as the tag output:

$$T = \widetilde{E}_K(V, U) \oplus \widetilde{E}_{K'}(N, 0^n), \text{ where } U = H_{K_1}(N \,\|\, M), V = H'_{K_2}(N \,\|\, M)\,.$$

An illustration is given in Figure 1d. The security bounds will be given in Section 6.

Both schemes use two keys of a TBC, however it is trivial to reduce to the single key by assuming one additional tweak bit. For example, we can use $[\widetilde{E}_K(0 \,\|\, *, *), \widetilde{E}_K(1 \,\|\, *, *)]$ instead of $[\widetilde{E}_K(*, *), \widetilde{E}_{K'}(*, *)]$.

### 3.2 Brief Comparison

Both constructions have two keyed hash functions and two TBC calls. In this sense, their respective efficiency values are close in general. Still, there are some differences: the keyed hash function under NaT2 takes the message, while those under eHaT takes the concatenation of the nonce and the message. Therefore, eHaT is more costly, and this can be non-trivial when the messages are short. For $t = n$, the hash functions $H$ and $H'$ for eHaT can be reduced to a single one under two independent keys. For $t \neq n$, both can also use the same core operation, e.g., a polynomial hash, that operates in different fields. In the case that a polynomial hash function over $\mathsf{GF}(2^n)$, this implies that the number of multiplications is increased by two (See Table 1). We note that omitting $N$ in the hash computations in eHaT (but keeping the nonce encryption by the second TBC) results in a nonce-based MAC without NM security. This corresponds to the classical hash-then-mask MAC, with the underlying keyed hash function being the whole HaT.
Another difference between NaT2 and eHaT is their tweak usage: NaT2 takes the nonce as a tweak for two TBCs, while eHaT takes a nonce for one TBC and a hash output for the other TBC. We observe that dedicated TBCs often employ a tweak schedule together with a key (dubbed tweakey schedule in Skinny for example) to derive the round keys. If the tweak is a nonce, most typically a counter, the tweak schedule can be pre-computed or incrementally computed to save the total computation. This implies that NaT2 is advantageous over eHaT in terms of tweak processing.
The difference in security is more involved. Despite the conceptual simplicity, the security analyses of both constructions are surprisingly complex, in particular for NaT2. It can be seen as a nonce-based variant of DbHtS MAC and adopt the security proof framework for DbHtS recently introduced by Kim et al. [KLL20].

## 4 Extended Mirror Theory

The goal of this section is to lower bound the number of solutions to a certain type of system of equalities and inequalities. This will be the foundation that we can thereupon prove the security of NaT2 in Section 5. For simplicity, we will denote $Z = 2^n$ throughout this section.

Transcript Graph. We will represent a system of inequalities and inequalities by a "bipartite" graph. The vertices in the graph are divided into two parts; $\mathcal{P}$ and $\mathcal{Q}$ are the two disjoint and independent vertex sets such that every edge connects a vertex in $\mathcal{P}$ to one in $\mathcal{Q}$. For both $\mathcal{P}$ and $\mathcal{Q}$, the vertices correspond to $n$-bit *distinct* unknowns. We will assume that the number of vertices is at most $Z/2$, and by abuse of notation, identify the vertices with the values assigned to them. We distinguish two types of edges, namely, $=$-labeled edges and $\neq$-labeled edges that correspond to inequalities and inequalities, respectively. Each edge is additionally labeled by an element in $\{0,1\}^n$. So, if two vertices $P$ and $Q$ are adjacent by an edge with label $(\lambda, =)$ (respectively $(\lambda, \neq)$) for some $\lambda \in \{0,1\}^n$, then it would mean that $P \oplus Q = \lambda$ (respectively $P \oplus Q \neq \lambda$).
Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}^{=} \sqcup \mathcal{E}^{\neq})$, where $\mathcal{E}^{=}$ and $\mathcal{E}^{\neq}$ denote the set of $=$-labeled edges and the set of $\neq$-labeled edges, respectively. Then $\mathcal{G}$ can be seen as a superposition of two subgraphs $\mathcal{G}^{=} := (\mathcal{V}, \mathcal{E}^{=})$ and $\mathcal{G}^{\neq} := (\mathcal{V}, \mathcal{E}^{\neq})$. Let $P \overset{\lambda}{-} Q$ denote a $(\lambda, =)$-labeled edge in

$\mathcal{G}^{=}$. For $\ell > 0$ and a trail[6]

$$\mathcal{L} : P_0 \overset{\lambda_1}{-} P_1 \overset{\lambda_2}{-} \cdots \overset{\lambda_\ell}{-} P_\ell$$

in $\mathcal{G}^{=}$, its label is defined as

$$\lambda(\mathcal{L}) \overset{\text{def}}{=} \lambda_1 \oplus \lambda_2 \oplus \cdots \oplus \lambda_\ell \,.$$

Nice Graphs. In this work, we will focus on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}^{=} \sqcup \mathcal{E}^{\neq})$ with certain properties, as listed below.

1. $\mathcal{G}^{=}$ contains no cycle.
2. $\lambda(\mathcal{L}) \neq \mathbf{0}$ for any trail $\mathcal{L}$ in $\mathcal{G}^{=}$.
3. If $P$ and $Q$ are connected with a $(\lambda, \neq)$-labeled edge, then they are not connected by a $\lambda$-labeled trail in $\mathcal{G}^{=}$.

Any graph $\mathcal{G}$ satisfying the above properties will be called a *nice* graph. Given a nice graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}^{=} \sqcup \mathcal{E}^{\neq})$, an assignment of *distinct* values to the vertices in $\mathcal{P}$ and $\mathcal{Q}$ satisfying all the inequalities in $\mathcal{E}^{=}$ and all the inequalities in $\mathcal{E}^{\neq}$ is called a *solution* to $\mathcal{G}$. We remark that if we assign any value to a vertex $P$, then $=$-labeled edges determine the values of all the other vertices in the component containing $P$ in $\mathcal{G}^{=}$, where the assignment is unique since $\mathcal{G}^{=}$ contains no cycle, and the values in the same component are all distinct since $\lambda(\mathcal{L}) \neq \mathbf{0}$ for any trail $\mathcal{L}$. Furthermore, any inequation between two vertices in the same component will be redundant due to the third property above.
The number of possible assignments of distinct values to the vertices in $\mathcal{V}$ is $(Z)_{|\mathcal{V}|}$. One might expect that when such an assignment is chosen uniformly at random, it would satisfy all the inequalities and inequalities in $\mathcal{G}$ with probability close to $1/Z^q$, where $q$ denotes the number of $=$-labeled edges (i.e., inequalities) in $\mathcal{G}^{=}$. Indeed, we can prove that the number of solutions to $\mathcal{G}$ is close to $\frac{(Z)_{|\mathcal{V}|}}{Z^q}$ up to a certain error (that can be negligible according to the parameters).

Proof Idea. Given an arbitrary nice graph $\mathcal{G}$, we will decompose $\mathcal{G}^{=}$ into three subgraphs, denoted $\mathcal{G}_1^{=}$, $\mathcal{G}_2^{=}$ and $\mathcal{G}_3^{=}$, respectively, where

- $\mathcal{G}_1^{=} = (\mathcal{V}_1, \mathcal{E}_1^{=})$ is the union of components containing at least one trail of length two;
- $\mathcal{G}_2^{=} = (\mathcal{V}_2, \mathcal{E}_2^{=})$ is the union of components of size two (i.e., trails of length one);
- $\mathcal{G}_3^{=} = (\mathcal{V}_3, \mathcal{E}_3^{=})$ is the set of isolated vertices.

For $i = 1, 2, 3$, let $\mathcal{E}_i^{\neq}$ denote the set of $\neq$-labeled edges connecting a vertex in $\mathcal{V}_i$ and one in $\bigsqcup_{j=1}^{i} \mathcal{V}_j$, and let

$$\mathcal{G}_i = \left( \bigsqcup_{j=1}^{i} \mathcal{V}_j, \bigsqcup_{j=1}^{i} \mathcal{E}_j^{=} \sqcup \bigsqcup_{j=1}^{i} \mathcal{E}_j^{\neq} \right).$$

In order to lower bound the number of solutions to $\mathcal{G}$, we will first lower bound the number of solutions to $\mathcal{G}_1$ using Lemma 4, and then $\mathcal{G}_2$ and $\mathcal{G}_3$ $(= \mathcal{G})$ using Lemma 5 and Lemma 6, respectively.

**Theorem 1.** For positive integers $q$ and $v$, let $\mathcal{G} = (\mathcal{V}, \mathcal{E}^{=} \sqcup \mathcal{E}^{\neq})$ be a nice graph such that $|\mathcal{E}^{=}| = q$ and $|\mathcal{E}^{\neq}| = v$. With the notation defined as above, assume that $\mathcal{G}_1^{=}$ is decomposed

[6] A trail is a walk wherein all edges are distinct.

into $k$ components $\mathcal{C}_1, \ldots, \mathcal{C}_k$ for some $k$. Then, the number of solutions to $\mathcal{G}$, denoted $h^*(\mathcal{G})$, satisfies

$$\frac{h^*(\mathcal{G})2^{nq}}{(2^n)_{|\mathcal{P}|}(2^n)_{|\mathcal{Q}|}} \geq 1 - \frac{|\mathcal{G}_1^=|^2}{2^{2n}} \sum_{i=1}^{k} |\mathcal{C}_i|^2 - \frac{|\mathcal{G}_1^=|q^2}{2^{2n}} - \frac{q^2}{2^{2n}} - \frac{4q^4}{2^{3n}} - \frac{2v}{2^n} - \frac{4qv}{2^{2n}},$$

provided that $q \leq 2^{n-3}$.

*Proof.* For $i = 1, 2, 3$, let $\mathcal{P}_i = \mathcal{V}_i \cap \mathcal{P}$, $\mathcal{Q}_i = \mathcal{V}_i \cap \mathcal{Q}$, $q_i = |\mathcal{E}_i^=|$ and $v_i = |\mathcal{E}_i^{\neq}|$. Then we have $q = q_1 + q_2$ (with $q_3 = 0$) and $v = v_1 + v_2 + v_3$. Note that we interchangeably write $|\mathcal{G}_i^=|$ and $|\mathcal{V}_i|$.
By Lemma 4, the number of solutions to $\mathcal{G}_1$, denoted $h(\mathcal{G}_1)$, satisfies

$$\frac{h(\mathcal{G}_1)Z^{q_1}}{(Z)_{|\mathcal{P}_1|}(Z)_{|\mathcal{Q}_1|}} \geq 1 - \frac{|\mathcal{V}_1|^2}{Z^2} \sum_{i=1}^{k} |\mathcal{C}_i|^2 - \frac{2v_1}{Z}. \tag{2}$$

By Lemma 5, for a fixed solution to $\mathcal{G}_1$, the number of solutions to $\mathcal{G}_2$, denoted $h(\mathcal{G}_2)$, satisfies

$$\begin{aligned}\frac{h(\mathcal{G}_2)Z^{q_2}}{(Z-|\mathcal{P}_1|)_{|\mathcal{P}_2|}(Z-|\mathcal{Q}_1|)_{|\mathcal{Q}_2|}} &\geq 1 - \frac{|\mathcal{V}_1|^2 q_2}{2Z^2} - \frac{|\mathcal{V}_1| q_2^2}{Z^2} - \frac{8|\mathcal{V}_1| q_2^3}{3Z^3} \\ &\quad - \frac{q_2^2}{Z^2} - \frac{4q_2^4}{Z^3} - \frac{2v_2}{Z} - \frac{4q_2v_2}{Z^2} \\ &\geq 1 - \frac{|\mathcal{V}_1| q^2}{Z^2} - \frac{q^2}{Z^2} - \frac{4q^4}{Z^3} - \frac{2v_2}{Z} - \frac{4qv}{Z^2}\end{aligned} \tag{3}$$

since $\frac{2}{3}|\mathcal{V}_1| + q_2 \leq q_1 + q_2 \leq q$. By Lemma 6, for a fixed solution to $\mathcal{G}_2$, the number of solutions to $\mathcal{G}_3$, denoted $h(\mathcal{G}_3)$, satisfies

$$\frac{h(\mathcal{G}_3)}{(Z-|\mathcal{P}_1|-|\mathcal{P}_2|)_{|\mathcal{P}_3|}(Z-|\mathcal{Q}_1|-|\mathcal{Q}_2|)_{|\mathcal{Q}_3|}} \geq 1 - \frac{2v_3}{Z}. \tag{4}$$

By (2), (3), (4), we have

$$\begin{aligned}\frac{h^*(\mathcal{G})Z^q}{(Z)_{|\mathcal{P}|}(Z)_{|\mathcal{Q}|}} &= \frac{h(\mathcal{G}_1)Z^{q_1}}{(Z)_{|\mathcal{P}_1|}(Z)_{|\mathcal{Q}_1|}} \cdot \frac{h(\mathcal{G}_2)Z^{q_2}}{(Z-|\mathcal{P}_1|)_{|\mathcal{P}_2|}(Z-|\mathcal{Q}_1|)_{|\mathcal{Q}_2|}} \\ &\quad \times \frac{h(\mathcal{G}_3)}{(Z-|\mathcal{P}_1|-|\mathcal{P}_2|)_{|\mathcal{P}_3|}(Z-|\mathcal{Q}_1|-|\mathcal{Q}_2|)_{|\mathcal{Q}_3|}} \\ &\geq 1 - \frac{|\mathcal{V}_1|^2}{Z^2} \sum_{i=1}^{k} |\mathcal{C}_i|^2 - \frac{|\mathcal{V}_1| q^2}{Z^2} - \frac{q^2}{Z^2} - \frac{4q^4}{Z^3} - \frac{4qv}{Z^2} \\ &\quad - \frac{2v_1}{Z} - \frac{2v_2}{Z} - \frac{2v_3}{Z} \\ &\geq 1 - \frac{|\mathcal{V}_1|^2}{Z^2} \sum_{i=1}^{k} |\mathcal{C}_i|^2 - \frac{|\mathcal{V}_1| q^2}{Z^2} - \frac{q^2}{Z^2} - \frac{4q^4}{Z^3} - \frac{2v}{Z} - \frac{4qv}{Z^2}.\end{aligned}$$

□

The proof is based on three lemmas:

- Lemma 4 will study the number of solutions of $\mathcal{G}_1^=$, that is, for a graph that contains exactly the components with a trail of length two or larger.
- Lemma 5 considers the number of solutions of $\mathcal{G}_2^=$, that is, for a graph that contains exactly the components with a trail of length one.

- Finally, Lemma 6 considers $\mathcal{G}_3^{=}$, that is the set of isolated vertices.

In the proofs of those lemmas, we will proceed stepwise. However, the other graphs may already have fixed some values before. For this purpose, we partition the set of vertices $\mathcal{V}$ into two disjoint sets, denoted $\mathcal{V}_{\mathsf{k}}$ and $\mathcal{V}_{\mathsf{u}}$, respectively.
The vertices in $\mathcal{V}_{\mathsf{k}}$ represent those distinct values that have been fixed already by *other* graphs. Then, the number of possible assignments of distinct values to the vertices in $\mathcal{V}_{\mathsf{u}}$ can be lower bounded in a way that the entire assignment becomes a solution to $\mathcal{G}$.

**Lemma 4.** For a positive integer $q$ and a nonnegative integer $v$, let $\mathcal{G} = (\mathcal{V}, \mathcal{E}^{=} \sqcup \mathcal{E}^{\neq})$ be a nice graph such that $|\mathcal{E}^{=}| = q$ and $|\mathcal{E}^{\neq}| = v$. Suppose that

1. $\mathcal{V}$ is partitioned into two subsets, denoted $\mathcal{V}_{\mathsf{k}}$ and $\mathcal{V}_{\mathsf{u}}$;
2. $\mathcal{P}$ (resp. $\mathcal{Q}$) is partitioned into two subsets, denoted $\mathcal{P}_{\mathsf{k}} = \mathcal{P} \cap \mathcal{V}_{\mathsf{k}}$ and $\mathcal{P}_{\mathsf{u}} = \mathcal{P} \cap \mathcal{V}_{\mathsf{u}}$ (resp. $\mathcal{Q}_{\mathsf{k}} = \mathcal{P} \cap \mathcal{Q}_{\mathsf{k}}$ and $\mathcal{Q}_{\mathsf{u}} = \mathcal{Q} \cap \mathcal{V}_{\mathsf{u}}$);
3. there is no $=$-labeled edge that is incident to a vertex in $\mathcal{V}_{\mathsf{k}}$;
4. there is no $\neq$-labeled edge connecting two vertices in $\mathcal{V}_{\mathsf{k}}$.

Suppose that $\mathcal{G}_{\mathsf{uk}}^{=} = (\mathcal{V}_{\mathsf{u}}, \mathcal{E}^{=})$ is decomposed into $k$ components $\mathcal{C}_1, \ldots, \mathcal{C}_k$ for some $k$. Given a fixed assignment of distinct values to the vertices in $\mathcal{V}_{\mathsf{k}}$, the number of solutions to $\mathcal{G}$, denoted $h(\mathcal{G})$, satisfies

$$\frac{h(\mathcal{G})Z^q}{(Z-|\mathcal{P}_{\mathsf{k}}|)_{|\mathcal{P}_{\mathsf{u}}|}(Z-|\mathcal{Q}_{\mathsf{k}}|)_{|\mathcal{Q}_{\mathsf{u}}|}} \geq 1 - \frac{|\mathcal{V}|^2}{Z^2}\sum_{i=1}^{k}|\mathcal{C}_i|^2 - \frac{2v}{Z}\,.$$

*Proof.* For $i = 1, \ldots, k$,

- let $\mathcal{C}_i = \mathcal{P}_i \sqcup \mathcal{Q}_i$ where $\mathcal{P}_i \in \mathcal{P}$ and $\mathcal{Q}_i \in \mathcal{Q}$;
- let $r_i = |\mathcal{P}_i|$, $s_i = |\mathcal{Q}_i|$ and $c_i = |\mathcal{C}_i| = r_i + s_i$;
- let $\alpha_i = |\mathcal{P}_{\mathsf{k}}| + \sum_{j=1}^{i} r_j$ and $\beta_i = |\mathcal{Q}_{\mathsf{k}}| + \sum_{j=1}^{i} s_j$;
- let $\sigma_i = |\mathcal{V}_{\mathsf{k}}| + \sum_{j=1}^{i} c_j = \alpha_i + \beta_i$;
- let $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ be the graph obtained from $\mathcal{V}_{\mathsf{k}} \sqcup \mathcal{C}_1 \sqcup \mathcal{C}_2 \sqcup \cdots \sqcup \mathcal{C}_i$ by adding all the $\neq$-labeled edges connecting the vertices in $\mathcal{V}_{\mathsf{k}} \sqcup \mathcal{C}_1 \sqcup \mathcal{C}_2 \sqcup \cdots \sqcup \mathcal{C}_i$;
- let $v_i$ be the number of $\neq$-labeled edges that connect a vertex in $\mathcal{C}_i$ and one in $\mathcal{G}_{i-1}$;
- let $h(i)$ be the number of solutions to $\mathcal{G}_i$.

Let $h(0) = 1$ and let $\sigma_0 = |\mathcal{V}_{\mathsf{k}}|$. Then we have $\mathcal{G}_k = \mathcal{G}$, and hence $h(k) = h(\mathcal{G})$. If there exists $i$ such that $\sigma_i c_{i+1} \geq Z$, we have

$$|\mathcal{V}|^2 \sum_{i=1}^{k} |\mathcal{C}_i|^2 \geq \sigma_i^2 c_{i+1}^2 \geq Z^2\,.$$

Thus, the lemma trivially holds. Therefore, we can assume that for $i = 0, \ldots, k-1$, $\sigma_i c_{i+1} \leq Z$. In order to find a relation between $h(i)$ and $h(i+1)$, we fix a solution to $\mathcal{G}_i$. If we fix a vertex $V^* \in \mathcal{P}_{i+1}$ and assign any value to $V^*$, then the other unknowns in $\mathcal{P}_{i+1}$ are uniquely determined, since there is a unique trail from $V^*$ to any other vertices in $\mathcal{P}_{i+1}$. In order to make all assigned values distinct (for each $\mathcal{P}$ and $\mathcal{Q}$), it is sufficient that

$$V^* \notin \bigcup_{\substack{1 \leq j \leq i \\ P \in \mathcal{P}_{i+1}}} ((\mathcal{P}_{\mathsf{k}} \sqcup \mathcal{P}_j) \oplus \lambda_P) \cup \bigcup_{\substack{1 \leq j \leq i \\ Q \in \mathcal{Q}_{i+1}}} ((\mathcal{Q}_{\mathsf{k}} \sqcup \mathcal{Q}_j) \oplus \lambda_Q)\,,$$

where $\lambda_V$ denotes the label of the unique trail from $V^*$ to $V$ if $V \neq V^*$ and $\lambda_{V^*} = \mathbf{0}$. Moreover, $V^*$ should satisfy $v_{i+1}$ inequalities. The number of choices satisfying these conditions is at least $Z - \alpha_i r_{i+1} - \beta_i s_{i+1} - v_{i+1}$, which means

$$h(i+1) \geq (Z - \alpha_i r_{i+1} - \beta_i s_{i+1} - v_{i+1}) h(i)\,.$$

Then, for $0 \leq i \leq q-1$, we have

$$\begin{aligned}
&\frac{h(i+1)Z^{c_{i+1}-1}}{h(i)(Z-\alpha_i)_{r_{i+1}}(Z-\beta_i)_{s_{i+1}}} \\
&\geq \frac{h(i+1)}{h(i)} \cdot \frac{1}{Z} \left(\frac{Z}{Z-\alpha_i}\right)^{r_{i+1}} \left(\frac{Z}{Z-\beta_i}\right)^{s_{i+1}} \\
&\geq \frac{h(i+1)}{h(i)} \cdot \frac{1}{Z} \left(1 + \frac{\alpha_i r_{i+1}}{Z-\alpha_i}\right) \left(1 + \frac{\beta_i s_{i+1}}{Z-\beta_i}\right) \\
&\geq \left(1 - \frac{\alpha_i r_{i+1} + \beta_i s_{i+1} + v_{i+1}}{Z}\right) \left(1 + \frac{\alpha_i r_{i+1} + \beta_i s_{i+1}}{Z}\right) \\
&\geq 1 - \left(\frac{\alpha_i r_{i+1} + \beta_i s_{i+1}}{Z}\right)^2 - \frac{(Z + \alpha_i r_{i+1} + \beta_i s_{i+1}) v_{i+1}}{Z^2} \\
&\geq 1 - \frac{\sigma_i^2 c_{i+1}^2}{Z^2} - \frac{2v_{i+1}}{Z}
\end{aligned}$$

since $\sigma_i c_{i+1} \leq Z$. From $\sum_{i=1}^{k} v_i = v$, we obtain

$$\begin{aligned}
\frac{h(\mathcal{G})Z^q}{(Z-|\mathcal{P}_{\mathsf{k}}|)_{|\mathcal{P}_{\mathsf{u}}|}(Z-|\mathcal{Q}_{\mathsf{k}}|)_{|\mathcal{Q}_{\mathsf{u}}|}} &= \prod_{i=0}^{k-1} \frac{h(i+1)Z^{c_{i+1}-1}}{h(i)(Z-\alpha_i)_{r_{i+1}}(Z-\beta_i)_{s_{i+1}}} \\
&\geq \prod_{i=0}^{k-1} \left(1 - \frac{\sigma_i^2 c_{i+1}^2}{Z^2} - \frac{2v_{i+1}}{Z}\right) \\
&\geq 1 - \sum_{i=0}^{k-1} \left(\frac{\sigma_i^2 c_{i+1}^2}{Z^2} + \frac{2v_{i+1}}{Z}\right) \\
&\geq 1 - \frac{\sigma_k^2}{Z^2} \sum_{i=1}^{k} c_i^2 - \frac{2v}{Z}\,.
\end{aligned}$$

The next lemma considers the case that every component of the graph contains exactly two vertices.

**Lemma 5.** For a positive integer $q$ and a nonnegative integer $v$, let $\mathcal{G} = (\mathcal{V}, \mathcal{E}^{=} \sqcup \mathcal{E}^{\neq})$ be a nice graph such that $|\mathcal{E}^{=}| = q$ and $|\mathcal{E}^{\neq}| = v$. Suppose that

1. $\mathcal{V}$ is partitioned into two subsets, denoted $\mathcal{V}_{\mathsf{k}}$ and $\mathcal{V}_{\mathsf{u}}$;
2. $\mathcal{P}$ (resp. $\mathcal{Q}$) is partitioned into two subsets, denoted $\mathcal{P}_{\mathsf{k}} = \mathcal{P} \cap \mathcal{V}_{\mathsf{k}}$ and $\mathcal{P}_{\mathsf{u}} = \mathcal{P} \cap \mathcal{V}_{\mathsf{u}}$ (resp. $\mathcal{Q}_{\mathsf{k}} = \mathcal{P} \cap \mathcal{Q}_{\mathsf{k}}$ and $\mathcal{Q}_{\mathsf{u}} = \mathcal{Q} \cap \mathcal{V}_{\mathsf{u}}$);
3. there is no $=$-labeled edge that is incident to a vertex in $\mathcal{V}_{\mathsf{k}}$;
4. there is no $\neq$-labeled edge connecting two vertices in $\mathcal{V}_{\mathsf{k}}$.

Suppose that $\mathcal{G}^{=}_{\mathsf{uk}} = (\mathcal{V}_{\mathsf{u}}, \mathcal{E}^{=})$ is decomposed into $q$ components of size two. Given a fixed assignment of distinct values to the vertices in $\mathcal{V}_{\mathsf{k}}$, the number of solutions to $\mathcal{G}$, denoted $h(\mathcal{G})$, satisfies

$$\frac{h(\mathcal{G})Z^q}{(Z-|\mathcal{P}_{\mathsf{k}}|)_{|\mathcal{P}_{\mathsf{u}}|}(Z-|\mathcal{Q}_{\mathsf{k}}|)_{|\mathcal{Q}_{\mathsf{u}}|}} \geq 1 - \frac{|\mathcal{V}_{\mathsf{k}}|^2 q}{2Z^2} - \frac{|\mathcal{V}_{\mathsf{k}}| q^2}{Z^2} - \frac{8|\mathcal{V}_{\mathsf{k}}| q^3}{3Z^3}$$

$$- \frac{q^2}{Z^2} - \frac{4q^4}{Z^3} - \frac{2v}{Z} - \frac{4qv}{Z^2} .$$

*Proof.* We will write the connected components of $\mathcal{G}^{=}_{\mathsf{unknown}}$ as follows:

$$\mathcal{C}_i : P_i \overset{\lambda_i}{-} Q_i ,$$

for $i = 1, \ldots, q$, where $P_i \in \mathcal{P}_{\mathsf{u}}$, $Q_i \in \mathcal{Q}_{\mathsf{u}}$ and $\lambda_i \in \{0,1\}^n$. For $i = 1, \ldots, k$,

- let $\alpha_i = |\mathcal{P}_{\mathsf{k}}| + i$ and $\beta_i = |\mathcal{Q}_{\mathsf{k}}| + i$;
- let $\sigma_i = |\mathcal{V}_{\mathsf{k}}| + 2i = \alpha_i + \beta_i$;
- let $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ be the graph obtained from $\mathcal{V}_{\mathsf{k}} \sqcup \mathcal{C}_1 \sqcup \mathcal{C}_2 \sqcup \cdots \sqcup \mathcal{C}_i$ by adding all the $\neq$-labeled edges connecting the vertices in $\mathcal{V}_{\mathsf{k}} \sqcup \mathcal{C}_1 \sqcup \mathcal{C}_2 \sqcup \cdots \sqcup \mathcal{C}_i$;
- let $v_i$ be the number of $\neq$-labeled edges that connect a vertex in $\mathcal{C}_i$ and one in $\mathcal{G}_{i-1}$;
- let $h(i)$ be the number of solutions to $\mathcal{G}_i$.

Let $h(0) = 1$, $\alpha_0 = |\mathcal{P}_{\mathsf{k}}|$, $\beta_0 = |\mathcal{Q}_{\mathsf{k}}|$ and $\sigma_0 = |\mathcal{V}_{\mathsf{k}}|$. Then we have $\mathcal{G}_k = \mathcal{G}$, and hence $h(k) = h(\mathcal{G})$.
In order to find a relation between $h(i)$ and $h(i+1)$, we fix a solution to $\mathcal{G}_i$. Then we can choose $P_{i+1}$ from $\{0,1\}^n \setminus (\mathcal{X}_i \cup \mathcal{Y}_i)$, where

$$\mathcal{X}_i \overset{\text{def}}{=} \mathcal{P}_{\mathsf{k}} \sqcup \{P_1, P_2, \ldots, P_i\} ,$$
$$\mathcal{Y}_i \overset{\text{def}}{=} (\mathcal{Q}_{\mathsf{k}} \sqcup \{Q_1, Q_2, \ldots, Q_i\}) \oplus \lambda_{i+1} .$$

Since $|\mathcal{X}_i| + |\mathcal{Y}_i| = \sigma_i$, we have

$$\begin{aligned} h(i+1) &\geq \sum_{\text{solutions to } \mathcal{G}_i} (Z - |\mathcal{X}_i \cup \mathcal{Y}_i| - v_{i+1}) \\ &= \sum_{\text{solutions to } \mathcal{G}_i} (Z - \sigma_i - v_{i+1} + |\mathcal{X}_i \cap \mathcal{Y}_i|) \\ &= (Z - \sigma_i - v_{i+1}) h(i) + \sum_{\text{solutions to } \mathcal{G}_i} |\mathcal{X}_i \cap \mathcal{Y}_i| . \end{aligned} \tag{5}$$

For $X \in \mathcal{X}_i$ and $Y \in \mathcal{Y}_i \oplus \lambda_{i+1}$, let $h'(X,Y)$ denote the number of solutions to $\mathcal{G}_i$ such that $X \oplus Y = \lambda_{i+1}$. Then we have

$$\sum_{\text{solutions to } \mathcal{G}_i} |\mathcal{X}_i \cap \mathcal{Y}_i| = \sum_{\substack{X \in \mathcal{X}_i \\ Y \in \mathcal{Y}_i \oplus \lambda_{i+1}}} h'(X,Y) \geq \sum_{\substack{X \in \{P_1, \ldots, P_i\} \\ Y \in \{Q_1, \ldots, Q_i\}}} h'(X,Y) . \tag{6}$$

We observe that

1. if $X$ and $Y$ are connected with a $(\lambda_{i+1}, =)$-labeled edge, then the additional equation $X \oplus Y = \lambda_{i+1}$ is redundant, and hence $h'(X,Y) = h(i)$;
2. if $X$ and $Y$ are connected with either a $(\lambda, =)$-labeled edge such that $\lambda \neq \lambda_{i+1}$ or a $(\lambda_{i+1}, \neq)$-labeled edge, then the system of equations and inequalities (with the additional equation) has no solution.

Let $i \geq 2$. Suppose that $X = P_j$ and $Y = Q_{j'}$ for distinct $j$ and $j'$, $X$ and $Y$ are not connected with any $\neq$-labeled edge, and $\lambda_{i+1} \notin \{\lambda_j, \lambda_{j'}\}$, then we have

$$h'(X,Y) \geq \frac{h(i)}{Z} \left(1 - \frac{4\sigma_i}{Z}\right) \tag{7}$$

since

$$h'(X,Y) \geq (Z-4\sigma_{i-2})\,h(i-2) \geq (Z-4\sigma_i)\,h(i-2),$$
$$h(i-2)Z^2 \geq h(i-2)\,(Z-2\sigma_{i-2})\,(Z-2\sigma_{i-1}) \geq h(i)\,.$$

Let

$$\mathcal{S}_1 = \left\{(j,j') \in [i]^{*2} : \text{there is a } \neq\text{-labeled edge between } P_j \text{ and } Q_{j'}\right\},$$
$$\mathcal{S}_2 = \left\{(j,j') \in [i]^{*2} : \lambda_j = \lambda_{i+1} \vee \lambda_{j'} = \lambda_{i+1}\right\},$$

and let

$$G = |\{1 \leq j \leq i : \lambda_j = \lambda_{i+1}\}|$$
$$H = \left|[i]^{*2} \setminus (\mathcal{S}_1 \cup \mathcal{S}_2)\right|.$$

Since $|\mathcal{S}_1| \leq 2v$ and $|\mathcal{S}_2| \leq 2iG$, we have

$$H \geq i(i-1) - 2v - 2iG\,. \tag{8}$$

By (6), (7), (8), and since $2i \leq 2q \leq Z$, we have

$$\sum_{\text{solutions to } \mathcal{G}_i} |\mathcal{X}_i \cap \mathcal{Y}_i| \geq \left(G + \frac{i^2-i-2v-2iG}{Z}\left(1-\frac{4\sigma_i}{Z}\right)\right) h(i)$$
$$\geq \frac{i^2-i-2v}{Z}\left(1-\frac{4\sigma_i}{Z}\right) h(i)\,,$$

and by (5),

$$h(i+1) \geq (Z-\sigma_i-v_{i+1})h(i) + \frac{i^2-i-2v}{Z}\left(1-\frac{4\sigma_i}{Z}\right) h(i)\,.$$

Since $\sigma_i \leq Z/2$, we have

$$\begin{aligned}
&\frac{h(i+1)Z}{h(i)(Z-\alpha_i)(Z-\beta_i)} \\
&\geq \frac{Z^2-\sigma_i Z - v_{i+1}Z + (i^2-i-2v)\left(1-\frac{4\sigma_i}{Z}\right)}{Z^2-\sigma_i Z+\alpha_i\beta_i} \\
&\geq 1 - \frac{\alpha_i\beta_i + v_{i+1}Z - (i^2-i-2v)\left(1-\frac{4\sigma_i}{Z}\right)}{Z^2-\sigma_i Z+\alpha_i\beta_i} \\
&\geq 1 - \frac{(\alpha_0+i)(\beta_0+i) + v_{i+1}Z - (i^2-i-2v)\left(1-\frac{4\sigma_i}{Z}\right)}{Z^2/2} \\
&\geq 1 - \frac{\alpha_0\beta_0 + \sigma_0 i + v_{i+1}Z + i + 2v + \frac{4\sigma_0 i^2}{Z} + \frac{8i^3}{Z}}{Z^2/2} \\
&\geq 1 - \frac{2\alpha_0\beta_0}{Z^2} - \frac{2\sigma_0 i}{Z^2} - \frac{8\sigma_0 i^2}{Z^3} - \frac{2i}{Z^2} - \frac{16i^3}{Z^3} - \frac{2v_{i+1}}{Z} - \frac{4v}{Z^2}\,.
\end{aligned}$$

Finally, with $|\mathcal{P}_{\mathsf{k}}||\mathcal{Q}_{\mathsf{k}}| \leq |\mathcal{V}_{\mathsf{k}}|^2/4$ we have

$$\begin{aligned}
&\frac{h(\mathcal{G})Z^q}{(Z-|\mathcal{P}_{\mathsf{k}}|)_{|\mathcal{P}_{\mathsf{u}}|}(Z-|\mathcal{Q}_{\mathsf{k}}|)_{|\mathcal{Q}_{\mathsf{u}}|}} \\
&= \prod_{i=0}^{q-1} \frac{h(i+1)Z}{h(i)(Z-\alpha_i)(Z-\beta_i)}
\end{aligned}$$

$$\geq \prod_{i=0}^{q-1} \left(1 - \frac{2\alpha_0\beta_0}{Z^2} - \frac{2\sigma_0 i}{Z^2} - \frac{8\sigma_0 i^2}{Z^3} - \frac{2i}{Z^2} - \frac{16i^3}{Z^3} - \frac{2v_{i+1}}{Z} - \frac{4v}{Z^2}\right)$$
$$\geq 1 - \sum_{i=0}^{q-1} \left(\frac{2\alpha_0\beta_0}{Z^2} + \frac{2\sigma_0 i}{Z^2} + \frac{8\sigma_0 i^2}{Z^3} + \frac{2i}{Z^2} + \frac{16i^3}{Z^3} + \frac{2v_{i+1}}{Z} + \frac{4v}{Z^2}\right)$$
$$\geq 1 - \frac{|\mathcal{V}_{\mathsf{k}}|^2 q}{2Z^2} - \frac{|\mathcal{V}_{\mathsf{k}}| q^2}{Z^2} - \frac{8|\mathcal{V}_{\mathsf{k}}| q^3}{3Z^3} - \frac{q^2}{Z^2} - \frac{4q^4}{Z^3} - \frac{2v}{Z} - \frac{4qv}{Z^2}\,.$$

Finally, we consider a graph containing no $=$-labeled edges. So $\mathcal{G}^{=}$ consists only of isolated vertices.

**Lemma 6.** For a nonnegative integer $v$, let $\mathcal{G} = (\mathcal{V}, \mathcal{E}^{\neq})$ be a nice graph such that $|\mathcal{E}^{\neq}| = v$. Suppose that

1. $\mathcal{V}$ is partitioned into two subsets, denoted $\mathcal{V}_{\mathsf{k}}$ and $\mathcal{V}_{\mathsf{u}}$;
2. $\mathcal{P}$ (resp. $\mathcal{Q}$) is partitioned into two subsets, denoted $\mathcal{P}_{\mathsf{k}} = \mathcal{P} \cap \mathcal{V}_{\mathsf{k}}$ and $\mathcal{P}_{\mathsf{u}} = \mathcal{P} \cap \mathcal{V}_{\mathsf{u}}$ (resp. $\mathcal{Q}_{\mathsf{k}} = \mathcal{P} \cap \mathcal{Q}_{\mathsf{k}}$ and $\mathcal{Q}_{\mathsf{u}} = \mathcal{Q} \cap \mathcal{V}_{\mathsf{u}}$);
3. there is no $\neq$-labeled edge connecting two vertices in $\mathcal{V}_{\mathsf{k}}$.

Given a fixed assignment of distinct values to the vertices in $\mathcal{V}_{\mathsf{k}}$, the number of solutions to $\mathcal{G}$, denoted $h(\mathcal{G})$, satisfies

$$\frac{h(\mathcal{G})}{(Z - |\mathcal{P}_{\mathsf{k}}|)_{|\mathcal{P}_{\mathsf{u}}|}(Z - |\mathcal{Q}_{\mathsf{k}}|)_{|\mathcal{Q}_{\mathsf{u}}|}} \geq 1 - \frac{2v}{Z}\,.$$

Since the proof is short, we can list it here:

*Proof.* The number of possible assignments of distinct values outside $\mathcal{V}_{\mathsf{k}}$ to the vertices in $\mathcal{V}_{\mathsf{u}}$ is $(Z - |\mathcal{P}_{\mathsf{k}}|)_{|\mathcal{P}_{\mathsf{u}}|}(Z - |\mathcal{Q}_{\mathsf{k}}|)_{|\mathcal{Q}_{\mathsf{u}}|}$. Among these assignments, at most $\frac{1}{Z - |\mathcal{V}_{\mathsf{k}}|}(Z - |\mathcal{P}_{\mathsf{k}}|)_{|\mathcal{P}_{\mathsf{u}}|}(Z - |\mathcal{Q}_{\mathsf{k}}|)_{|\mathcal{Q}_{\mathsf{u}}|}$ assignments violate any fixed $\neq$-labeled edge. Therefore, we have

$$h(\mathcal{G}) \geq (Z - |\mathcal{P}_{\mathsf{k}}|)_{|\mathcal{P}_{\mathsf{u}}|}(Z - |\mathcal{Q}_{\mathsf{k}}|)_{|\mathcal{Q}_{\mathsf{u}}|} - \frac{v}{Z - |\mathcal{V}_{\mathsf{k}}|}(Z - |\mathcal{P}_{\mathsf{k}}|)_{|\mathcal{P}_{\mathsf{u}}|}(Z - |\mathcal{Q}_{\mathsf{k}}|)_{|\mathcal{Q}_{\mathsf{u}}|}\,,$$

which means

$$\frac{h(\mathcal{G})}{(Z - |\mathcal{P}_{\mathsf{k}}|)_{|\mathcal{P}_{\mathsf{u}}|}(Z - |\mathcal{Q}_{\mathsf{k}}|)_{|\mathcal{Q}_{\mathsf{u}}|}} \geq 1 - \frac{2v}{Z}\,. \qquad \square$$

As a special case of interest of Theorem 1, we can also consider Theorem 2. In fact, it considers a nice transcript graph of a transcript that contains only a single MAC query, and $v$ verification queries.

**Theorem 2.** For a nonnegative integer $v$, let $\mathcal{G} = (\mathcal{V}, \mathcal{E}^{=} \sqcup \mathcal{E}^{\neq})$ be a nice graph such that $|\mathcal{E}^{=}| = 1$ and $|\mathcal{E}^{\neq}| = v$. The number of solutions to $\mathcal{G}$, denoted $h^*(\mathcal{G})$, satisfies

$$\frac{h^*(\mathcal{G})}{(2^n)_{|\mathcal{P}|}(2^n)_{|\mathcal{Q}|}} \geq 1 - \frac{2v}{2^n}\,.$$

*Proof.* Let $\mathcal{G}^{=} = (\mathcal{V}^{=}, \mathcal{E}^{=})$ be a unique component of size two. The number of solutions to $\mathcal{G}^{=}$, denoted $h(\mathcal{G}^{=})$, is exactly $Z$. By Lemma 6, for a fixed solution to $\mathcal{G}^{=}$, the number of solutions to $\mathcal{G}$, denoted $h(\mathcal{G})$, satisfies

$$\frac{h(\mathcal{G})}{(Z-1)_{|\mathcal{P}|-1}(Z-1)_{|\mathcal{Q}|-1}} \geq 1 - \frac{2v}{Z}\,. \tag{9}$$

By $h(\mathcal{G}^{=}) = Z$ and Equation (9), we have

$$\frac{h^*(\mathcal{G})Z}{(Z)_{|\mathcal{P}|}(Z)_{|\mathcal{Q}|}} = \frac{h(\mathcal{G}^{=})Z}{Z^2} \cdot \frac{h(\mathcal{G})}{(Z-1)_{|\mathcal{P}|-1}(Z-1)_{|\mathcal{Q}_2|-1}} \geq 1 - \frac{2v}{Z} \,. \qquad \square$$

# 5 Security Analysis of NaT2

Recall that NaT2 computes a tag $T$ for a tuple $(N, M)$ following Algorithm 1 and Figure 1c. Up to the tprp-security of $\widetilde{E}$, the keyed tweakable permutation $\widetilde{E}_K$ (resp. $\widetilde{E}_{K'}$) can be replaced by a truly tweakable random permutation $\widetilde{\pi}$ (resp. $\widetilde{\pi}'$). The core task will be to show the following Theorem.

**Theorem 3.** Let $\delta > 0$, and let $H : \mathcal{K} \times \mathcal{M} \to \{0,1\}^n$ be a $\delta$-almost universal hash function. For positive integers $\mu$, $q$, $v$, such that $\mu + v \leq 2^{n-3}$, we have

$$\mathsf{Adv}^{\mathsf{mac}}_{\mathsf{NaT2}}(\mu, q, v) \leq \frac{16\mu^2\delta}{2^n} + 8\mu^2\delta^{3/2} + \frac{24\mu^2\delta}{2^{n/2}} + \frac{16\mu^3\delta^2}{2^n} + \frac{4\mu^2}{2^{3n/2}} + \frac{4\mu^2}{2^{2n}} + \frac{64\mu^4}{2^{3n}} + \frac{2v}{2^n} + \frac{8\mu v}{2^{2n}} + 4(v+1)\mu^2\delta^2 + v\delta + \frac{3}{2^{n/2}} \,.$$

The remaining part of this section will be devoted to the proof of Theorem 3.

## 5.1 Graph Representation of Transcripts

Suppose that an adversary $\mathbf{A}$ makes $q$ MAC queries using at most $\mu$ faulty nonces, and makes $v$ verification queries. Let

$$\tau_m = (N_i, M_i, T_i)_{1 \leq i \leq q} \text{ and}$$
$$\tau_v = \left(N'_j, M'_j, T'_j, b'_j\right)_{1 \leq j \leq v}$$

denote the list of MAC queries and the list of verification queries, respectively. For a nonce $w$, we also define

$$\tau_m(w) = \{(N, M, T) \in \tau_m : N = w\} \text{ and}$$
$$\tau_v(w) = \{(N', M', T', b') \in \tau_v : N' = w\}$$

and let $q_w = |\tau_m(w)|$ and $v_w = |\tau_v(w)|$. Note that $\mathbf{A}$ is given $K_1$ and $K_2$ for free at the end of the attack. Then, from the transcript

$$\tau = (K_1, K_2, \tau_m, \tau_v) \,,$$

one can fix $U_i \coloneqq H_{K_1}(M_i)$ (resp. $V_i \coloneqq H_{K_2}(M_i)$), for $i \in [q]$, and $U'_j \coloneqq H_{K_1}(M'_j)$ (resp. $V'_j \coloneqq H_{K_2}(M'_j)$) for $j \in [v]$.
The core of the security proof is to estimate the number of possible ways of fixing evaluations of $\widetilde{\pi}$ and $\widetilde{\pi}'$ in a way that $\widetilde{\pi}(N_i, U_i) \oplus \widetilde{\pi}'(N_i, V_i) = T_i$ for $i \in [q]$, and $\widetilde{\pi}(N'_j, U'_j) \oplus \widetilde{\pi}'(N'_j, V'_j) \neq T'_j$ for $j \in [v]$. For a fixed $w$, we will identify $\{\widetilde{\pi}(N_i, U_i) : N_i = w\} \cup \left\{\widetilde{\pi}(N'_j, U'_j) : N'_j = w\right\}$ with a set of unknowns (by an abuse of notation)

$$\mathcal{P}_w = \{P_1, \ldots, P_{r_w}\} \,,$$

where $r_w \leq q_w + v_w$ since there can be hash collisions. Similarly, we will identify $\{\widetilde{\pi}'(N_i, V_i) : N_i = w\} \cup \left\{\widetilde{\pi}'(N'_j, V'_j) : N'_j = w\right\}$ with a set of unknowns

$$\mathcal{Q}_w = \{Q_1, \ldots, Q_{s_w}\} \,,$$

where $s_w \leq q_w + v_w$.
For $i \in [q]$ where $N_i = w$, let $\widetilde{\pi}(w, U_i) = P_j \in \mathcal{P}_w$ and let $\widetilde{\pi}'(w, V_i) = Q_k \in \mathcal{Q}_w$. Then $P_j$ and $Q_k$ are connected with a $(T_i, =)$-labeled edge. Similarly, for $i \in [v]$ where $N'_i = w$, $P_j$ and $Q_k$ are connected with a $(T'_i, \neq)$-labeled edge if $\widetilde{\pi}(w, U'_i) = P_j$ and $\widetilde{\pi}'(w, V'_i) = Q_k$. In this way, we obtain a graph $\mathcal{G}_w = (\mathcal{V}_w, \mathcal{E}_w)$ on $\mathcal{V}_w \coloneqq \mathcal{P}_w \sqcup \mathcal{Q}_w$, and call the union of graphs $\mathcal{G}_w$ for all nonces as the *transcript graph* of $\tau$ and denoted $\mathcal{G}_\tau$. By definition, $\mathcal{G}_\tau$ has no isolated vertices. Furthermore, $\mathcal{G}_\tau$ is a bipartite graph with independent sets $\bigsqcup_w \mathcal{P}_w$ and $\bigsqcup_w \mathcal{Q}_w$, and contains no edge between $\mathcal{P}_w$ and $\mathcal{Q}_{w'}$ for $w \neq w'$.

## 5.2 Bad Transcripts

For a fixed positive integer $L$ (to be optimized later), a transcript $\tau = (K_1, K_2, \tau_m, \tau_v)$ is defined as *bad* if one of the following conditions holds.

- $\mathsf{bad}_1 \coloneqq \mathsf{bad}_{1a} \vee \mathsf{bad}_{1b} \vee \mathsf{bad}_{1c}$ where
  - $\mathsf{bad}_{1a}$: there exist $(i, j) \in [q]^{*2}$ such that $N_i = N_j$, $U_i = U_j$, and $V_i = V_j$;
  - $\mathsf{bad}_{1b}$: there exist $(i, j, k, l) \in [q]^{*4}$ such that $N_i = N_j = N_k = N_l$, $U_i = U_j$, $V_j = V_k$, and $U_k = U_l$;
  - $\mathsf{bad}_{1c}$: there exist $(i, j, k, l) \in [q]^{*4}$ such that $N_i = N_j = N_k = N_l$, $V_i = V_j$, $U_j = U_k$, and $V_k = V_l$;
- $\mathsf{bad}_2 \coloneqq \mathsf{bad}_{2a} \vee \mathsf{bad}_{2b}$, where
  - $\mathsf{bad}_{2a}$: there exist $(i, j) \in [q]^{*2}$ such that $N_i = N_j$, $U_i = U_j$, and $T_i = T_j$;
  - $\mathsf{bad}_{2b}$: there exist $(i, j) \in [q]^{*2}$ such that $N_i = N_j$, $V_i = V_j$, and $T_i = T_j$;
- $\mathsf{bad}_3 \coloneqq \mathsf{bad}_{3a} \vee \mathsf{bad}_{3b}$, where
  - $\mathsf{bad}_{3a}$: there exist $i \in [q]$ and $j \in [v]$ such that $N_i = N'_j$, $U_i = U'_j$, $V_i = V'_j$, and $T_i = T'_j$;
  - $\mathsf{bad}_{3b}$: there exist $(i, j) \in [q]^{*2}$ and $k \in [v]$ such that $N_i = N_j = N'_k$, $U_i = U'_k$, and $V'_k = V_j$;
- $\mathsf{bad}_4 \coloneqq \mathsf{bad}_{4a} \vee \mathsf{bad}_{4b}$, where
  - $\mathsf{bad}_{4a}$: $|\{i \in [q] : N_i = N_j \wedge U_i = U_j \text{ for some } j \text{ such that } j \neq i\}| \geq L$;
  - $\mathsf{bad}_{4b}$: $|\{i \in [q] : N_i = N_j \wedge V_i = V_j \text{ for some } j \text{ such that } j \neq i\}| \geq L$.

If a transcript $\tau$ is not bad, then it will be called a *good* transcript. For a good transcript $\tau$ and for a $w$ such that $q_w + v_w > 0$, we observe that

1. $\mathcal{G}_w^{=}$, being a bipartite graph, contains no cycle without $\mathsf{bad}_1$;
2. $\mathcal{G}_w^{=}$ contains no even length trail $\mathcal{L}$ such that $\lambda(\mathcal{L}) = \mathbf{0}$ without $\mathsf{bad}_1 \vee \mathsf{bad}_2$;
3. if two vertices are connected by a $\lambda$-labeled trail in $\mathcal{G}_w^{=}$, then they cannot be connected with a $(\lambda, \neq)$-labeled edge without $\mathsf{bad}_1 \vee \mathsf{bad}_3$.

Furthermore, we see that $\mathcal{G}_\tau^{=}$ contains no trail of length 4 without $\mathsf{bad}_1$. With this observation, we conclude that for any $w$ and a good transcript $\tau$, it holds that

1. $\mathcal{G}_w$ is nice (as defined in Section 4);
2. $|\mathcal{G}_w| \leq 2(2\mu + v) \leq 2^{n-2}$.

These properties allow us to apply Theorem 1 later.
In the following, we upper bound the probabilities of the individual bad events in the ideal world.

$\mathsf{bad}_1$. The number of queries using any repeated nonce is at most $2\mu$. So the number of pairs $(i,j) \in [q]^{*2}$ such that $N_i = N_j$ is at most $4\mu^2$. For each of such pairs, say $(i,j)$, the probability that $U_i = U_j$ and $V_i = V_j$ is at most $\delta^2$. Therefore, we have

$$\Pr[\mathsf{bad}_{1a}] \leq 4\mu^2\delta^2 .$$

$\mathsf{bad}_{2a}$ AND $\mathsf{bad}_{2b}$ can be upper bounded from a similar argument: Since $\Pr[T_i = T_j] = 2^{-n}$ in the ideal world, we have

$$\Pr[\mathsf{bad}_{2a}] \leq \frac{4\mu^2\delta}{2^n} \quad \text{and} \quad \Pr[\mathsf{bad}_{2b}] \leq \frac{4\mu^2\delta}{2^n} .$$

$\mathsf{bad}_{1b}$. Since the number of queries using any repeated nonce is at most $2\mu$ and by Lemma 2, we have

$$\Pr[\mathsf{bad}_{1b}] \leq 4\mu^2\delta^{3/2} .$$

$\mathsf{bad}_{1c}$ can be upper bounded again from a similar argument:

$$\Pr[\mathsf{bad}_{1c}] \leq 4\mu^2\delta^{3/2} .$$

$\mathsf{bad}_{3a}$. When an adversary makes a verification query $(N'_j, M'_j, T'_j)$, there is at most one MAC query $(N_i, M_i, T_i)$ such that $N_i = N'_j$, $U_i = U'_j$, and $T_i = T'_j$ without $\mathsf{bad}_{2a}$.[7] For this pair of indices, the probability that $V_i = V'_j$ is upper bounded by $v\delta$. Therefore, we have

$$\Pr[\mathsf{bad}_{3a} \mid \neg\mathsf{bad}_{2a}] \leq v\delta .$$

$\mathsf{bad}_{3b}$. The number of pairs $(i,j)$ such that $(i,j) \in [q]^{*2}$ and $N_i = N_j$ is at most $4\mu^2$. For each of such pairs and $k \in [v]$, the probability that $U_i = U'_k$ and $V'_k = V_j$ is at most $\delta^2$.

$$\Pr[\mathsf{bad}_{3b}] \leq 4\mu^2 v\delta^2 .$$

$\mathsf{bad}_{4a}$ AND $\mathsf{bad}_{4b}$. The number of pairs $(i,j)$ such that $(i,j) \in [q]^{*2}$ and $N_i = N_j$ is at most $4\mu^2$. For a fixed $i \in [q]$, the probability that $U_i = U_j$ is at most $\delta$. By the Markov inequality, we have

$$\Pr[\mathsf{bad}_{4a}] \leq \frac{4\mu^2\delta}{L} \quad \text{and similarly} \quad \Pr[\mathsf{bad}_{4b}] \leq \frac{4\mu^2\delta}{L} .$$

All in all, we have

$$\begin{aligned} \Pr[\mathsf{T}_{\mathsf{id}} \in \Gamma_{\mathsf{bad}}] &\leq \Pr[\mathsf{bad}_1 \vee \mathsf{bad}_2 \vee \mathsf{bad}_3 \vee \mathsf{bad}_4] \\ &\leq \frac{8\mu^2\delta}{2^n} + 8\mu^2\delta^{3/2} + \frac{8\mu^2\delta}{L} + 4(v+1)\mu^2\delta^2 + v\delta . \end{aligned} \tag{10}$$

[7] For simplicity of analysis, one can assume that an adversary begins making verification queries after it makes all the MAC queries.

## 5.3 Concluding the Proof Using the Extended Mirror Theory

For any good transcript $\tau$ and nonce $w$, let $\mathcal{G}_w^{=}$ denote the graph obtained by deleting all $\neq$-labeled edges from $\mathcal{G}_w$. We can decompose $\mathcal{G}_w^{=}$ into three subgraphs as follows.

$$\mathcal{G}_w^{=} = \mathcal{G}_{w,1}^{=} \sqcup \mathcal{G}_{w,2}^{=} \sqcup \mathcal{G}_{w,3}^{=} ,$$

where $\mathcal{G}_{w,1}^{=}$ is the union of the components containing at least one trail of length two, $\mathcal{G}_{w,2}^{=}$ is the set of isolated edges, and $\mathcal{G}_{w,3}^{=}$ is the set of isolated vertices. We also decompose $\mathcal{G}_{w,1}^{=}$ into connected components as follows.

$$\mathcal{G}_{w,1}^{=} = (\mathcal{V}_{w,1}, \mathcal{E}_{w,1}^{=}) = \mathcal{C}_{w,1} \sqcup \cdots \sqcup \mathcal{C}_{w,k_w} ,$$

for some $k_w$. Let $c_{w,i} = |\mathcal{C}_{w,i}|$ for $i \in [k_w]$. We will also write $c_w = \left|\mathcal{G}_{w,1}^{=}\right| (= \sum_{i=1}^{k_w} c_{w,i})$ and $c = \sum_w c_w$.

The probability of obtaining $\tau$ in the real world is computed over the randomness of $\widetilde{\pi}$ and $\widetilde{\pi}'$. For a fixed nonce $w$, let $\pi(\cdot) = \widetilde{\pi}(w, \cdot)$ and $\pi'(\cdot) = \widetilde{\pi}'(w, \cdot)$. By Theorem 1 and Theorem 2, the number of possible ways of evaluating $\pi$ and $\pi'$ at the unknowns in $\mathcal{V}_w = \mathcal{P}_w \sqcup \mathcal{Q}_w$ (i.e., $h^*(\mathcal{G}_w)$) is lower bounded by

$$\frac{(2^n)_{|\mathcal{P}_w|}(2^n)_{|\mathcal{Q}_w|}}{2^{nq_w}} \left(1 - \varepsilon_1(\tau, w)\right) ,$$

where

$$\varepsilon_1(\tau, w) \coloneqq \frac{c_w^2}{2^{2n}} \sum_{i=1}^{k_w} {c_{w,i}}^2 + \frac{c_w q_w^2}{2^{2n}} + \frac{{q_w}^2}{2^{2n}} + \frac{4{q_w}^4}{2^{3n}} + \frac{2v_w}{2^n} + \frac{4q_w v_w}{2^{2n}} ,$$

for $w$ such that $q_w \geq 2$, and

$$\varepsilon_1(\tau, w) \coloneqq \frac{2v_w}{2^n}$$

for $w$ such that $q_w = 1$. Since the probability that $\pi$ (resp. $\pi'$) realizes each assignment is exactly $1/(2^n)_{|\mathcal{P}_w|}$ (resp. $1/(2^n)_{|\mathcal{Q}_w|}$) and

$$\Pr\left[\mathsf{T}_{\mathsf{id}} = \tau\right] = \frac{1}{|K_h|^2 \cdot 2^{nq}} = \frac{1}{|K_h|^2} \prod_w \frac{1}{2^{nq_w}} ,$$

we have

$$\frac{\Pr\left[\mathsf{T}_{\mathsf{re}} = \tau\right]}{\Pr\left[\mathsf{T}_{\mathsf{id}} = \tau\right]} \geq 1 - \sum_w \varepsilon_1(\tau, w) \geq 1 - \varepsilon_1(\tau) , \tag{11}$$

where

$$\varepsilon_1(\tau) \coloneqq \frac{c^2}{2^{2n}} \sum_w \sum_{i=1}^{k_w} {c_{w,i}}^2 + \frac{4c\mu^2}{2^{2n}} + \frac{4\mu^2}{2^{2n}} + \frac{64\mu^4}{2^{3n}} + \frac{2v}{2^n} + \frac{8\mu v}{2^{2n}} \tag{12}$$

since the sum of all $q_w \geq 2$ is at most $2\mu$.

Upper Bounding $c$. We observe that each edge of $\mathcal{E}_{w,1}^{=}$ corresponds to a collision on $U$ or $V$. Therefore, we have

$$c = \sum_w \left(k_w + \left|\mathcal{E}_{w,1}^{=}\right|\right) \leq 2L + \sum_w k_w \leq 3L . \tag{13}$$

Taking the Expectation of $\varepsilon_1(\tau, w)$. Let us define following three helpful random variables,

$$\mathsf{NC}_1 = \left|\left\{(i,j) \in [q]^{*2} : N_i = N_j, \text{ and } U_i = U_j\right\}\right| ,$$

$$\mathsf{NC}_2 = \left|\left\{(i,j) \in [q]^{*2} : N_i = N_j, \text{ and } V_i = V_j\right\}\right| ,$$
$$\mathsf{NC}_3 = \left|\left\{(i,j,k) \in [q]^{*3} : N_i = N_j = N_k,\ U_i = U_j, \text{ and } V_j = V_k\right\}\right| .$$

Moreover, for each $w$ and $i \in [k_w]$, let $r_{w,i} = |\mathcal{C}_{w,i} \cap \mathcal{P}|$ and $s_{w,i} = |\mathcal{C}_{w,i} \cap \mathcal{Q}|$. Then,

$$\begin{aligned}\sum_w \sum_{i=1}^{k_w} c_{w,i}^2 &= \sum_w \sum_{i=1}^{k_w} (r_{w,i}^2 + 2r_{w,i}s_{w,i} + s_{w,i}^2) \\ &\le \sum_w \sum_{i=1}^{k_w} ((r_{w,i})_2 + (s_{w,i})_2 + 2(r_{w,i}-1)(s_{w,i}-1) + 3c_{w,i}) \\ &\le \mathsf{NC}_1 + \mathsf{NC}_2 + 2\mathsf{NC}_3 + 9L\,.\end{aligned}$$

Since

$$\mathbf{E}\left[\mathsf{NC}_1\right] \le 4\mu^2\delta, \qquad \mathbf{E}\left[\mathsf{NC}_2\right] \le 4\mu^2\delta, \text{ and} \qquad \mathbf{E}\left[\mathsf{NC}_3\right] \le 8\mu^3\delta^2\,,$$

we obtain

$$\mathbf{E}\left[\varepsilon_1(\tau)\right] \le \frac{9L^2(8\mu^2\delta + 16\mu^3\delta^2 + 9L)}{2^{2n}} + \frac{12L\mu^2}{2^{2n}} + \frac{4\mu^2}{2^{2n}} + \frac{64\mu^4}{2^{3n}} + \frac{2v}{2^n} + \frac{8\mu v}{2^{2n}}\,. \tag{14}$$

We can set $L = \frac{2^{n/2}}{3}$. Our bound in Theorem 3 follows then from (10), (11), (14) and by applying Lemma 3.

# 6 Security Analysis of eHaT

Recall that eHaT computes a tag $T$ for a tuple $(N, M)$ following Algorithm 2 and Figure 1d. Up to the tprp-security of $\widetilde{E}$, the keyed tweakable permutation $\widetilde{E}_K$ (resp. $\widetilde{E}_{K'}$) can be replaced by a truly tweakable random permutation $\widetilde{\pi}$ (resp. $\widetilde{\pi}'$). For the $i$-th MAC query $(N_i, M_i)$, we define

$$U_i \coloneqq H_{K_1}(N_i \,\|\, M_i), V_i \coloneqq H'_{K_2}(N_i \,\|\, M_i), X_i \coloneqq \widetilde{\pi}(V_i, U_i), \text{ and } Y_i \coloneqq \widetilde{\pi}'(N_i, 0^n)\,.$$

For the $i$-th verification query $(N'_i, M'_i, T'_i)$, we define

$$U'_i \coloneqq H_{K_1}(N'_i \,\|\, M'_i), V'_i \coloneqq H'_{K_2}(N'_i \,\|\, M'_i), X'_i \coloneqq \widetilde{\pi}(V'_i, U'_i), \text{ and } Y'_i \coloneqq \widetilde{\pi}'(N'_i, 0^n)\,.$$

As a further step of simplifying our task, we introduce $\mu'$ for the number of MAC queries whose nonce repeats in other MAC queries. That is, $\mu'$ includes the number of queries with faulty nonces and the MAC queries with the initial occurrence of their respective nonces. If $\mu$ is the number of faulty queries, it is easy to see that

$$\mu < \mu' \le 2\mu\,, \tag{15}$$

where the equality $\mu' = 2\mu$ holds if every faulty nonce repeats exactly once and is strictly lower if any nonce repeats twice or more times. We will call those *nonce-repeating* MAC queries. The core task will be then to show the following theorem.

**Theorem 4.** Let $\delta > 0$, $H : \mathcal{K} \times \mathcal{M} \to \{0,1\}^n$ be a $\delta$-almost-universal hash function, $H' : \mathcal{K} \times \mathcal{M} \to \{0,1\}^t$ be a $\delta'$-almost-universal hash function, $\widetilde{\pi}, \widetilde{\pi}' \leftarrow_\$ \mathsf{TPerm}(\mathbb{F}_2^t, \mathbb{F}_2^n)$, and $K_1 \leftarrow_\$ \mathcal{K}$ and $K_2 \leftarrow_\$ \mathcal{K}'$. For non-negative integers $\mu$, $q$, and $v$, if $\mu = 0$, adversaries are nonce-respecting, we have

$$\mathsf{Adv}^{\mathsf{mac}}_{\mathsf{eHaT}[H_{K_1}, H'_{K_2}, \widetilde{\pi}, \widetilde{\pi}']}(0, q, v) \le \frac{v}{2^n - v} + v\delta\delta'\,,$$

and if $\mu > 0$, i.e., adversaries are nonce-misusing, we have

$$\mathsf{Adv}^{\mathsf{mac}}_{\mathsf{eHaT}[H_{K_1}, H'_{K_2}, \widetilde{\pi}, \widetilde{\pi}']}(\mu, q, v) \leq 2\mu^2\delta\delta' + \frac{2\mu^2\delta'}{2^n} + (3\mu+1)v\delta\delta' + \frac{v}{2^n} + 2q^{\frac{2}{3}}\delta' + \frac{3v}{2^n - (2q^{\frac{2}{3}} + v)} .$$

*Proof of Theorem 4 for $\mu = 0$.* First, we consider the nonce-respecting case i.e., $\mu = 0$. This proof uses the mac security of the Wegman-Carter construction [WC81], because eHaT represents an XOR sum of an $n$-bit random function $\widetilde{\pi}'(N, 0^n)$ and a hash function $\widetilde{\pi}(H'_{K_2}(N\|M), H_{K_1}(N\|M))$. Following the analysis of the Wegman-Carter construction, the mac security advantage is bounded by $v$ times

$$\max_{W \neq W', Z} \Pr\left[\widetilde{\pi}(H'_{K_2}(W), H_{K_1}(W)) \oplus \widetilde{\pi}(H'_{K_2}(W'), H_{K_1}(W')) = Z\right] .$$

Since $H_{K_1}$ and $H'_{K_2}$ are $\delta$- and $\delta'$-almost universal, respectively, we have

$$\Pr\left[H_{K_1}(W) = H_{K_1}(W') \wedge H'_{K_2}(W) = H'_{K_2}(W')\right] \leq \delta\delta' .$$

If such collisions of $H_{K_1}$ and $H'_{K_2}$ do not occur, as the number of solutions of $\widetilde{\pi}$ is at least $2^n - v$, the maximum is at most $1/(2^n - v)$. Thus we obtain the bound for $\mu = 0$ given in Theorem 4. □

*Proof of Theorem 4 for $\mu > 0$.* We consider the nonce-misuse case, i.e., $\mu > 0$. We reorder the queries into three disjoint parts: $\tau_n \cup \tau_{\mu'} \cup \tau_v$. The former, $\tau_n$, contains the nonce-respecting MAC queries such that their nonce does not repeat over other MAC queries. $\tau_{\mu'}$ contains all $\mu'$ nonce-repeating MAC queries, i.e., MAC queries with and including those whose nonces repeat among the $q$ MAC queries in total. $\tau_v$ contains all verification queries. After $\mathbf{A}$ has finished the interactions with its oracles but before it outputs its decision bit, it is provided with the hash keys $\tau_h = \{K_1, K_2\}$. Moreover, we employ a trick to simplify the proof: for all nonce-repeating MAC queries $i \in [q - \mu' + 1 \,..\, q]$, we provide $\mathbf{A}$ with the outputs of $Y_i = \widetilde{\pi}'(N_i, 0^n)$ at the same point of time as it is given the hash keys. In the ideal world, $Y_i \leftarrow_{\$} \mathbb{F}_2^n$ is sampled uniformly at random once for each new nonce. If the nonce had occurred before, its corresponding old $Y_i$ is used. Moreover, we define $b'_i \in \{0, 1\}$ as the responses corresponding to either accept or reject for the $i$-th verification query. So, the transcript looks like:

$$\begin{aligned} \tau_n &= (N_i, M_i, T_i)_{1 \leq i \leq q-\mu'} , \\ \tau_{\mu'} &= (N_i, M_i, T_i, Y_i)_{q-\mu'+1 \leq i \leq q} , \\ \tau_v &= (N'_i, M'_i, T'_i, b'_i)_{1 \leq i \leq v} , \\ \tau_h &= \{K_1, K_2\} . \end{aligned}$$

We further define $\tau_m = \tau_n \cup \tau_{\mu'}$ as the compound transcript of all MAC queries. Note that, given $T_i$ and $Y_i$, the adversary can compute $X_i$ itself for all nonce-repeating queries. Next, we partition the set of all attainable transcripts into two disjoint sets of good transcripts GoodT, and bad transcripts $\Gamma_{\mathsf{bad}}$. For fixed positive number $v_{\mathsf{max}}$ (to be optimized later), a transcript $\tau$ is defined as bad if at least one of the following so-called bad events occurs:

- $\mathsf{bad}_1$: There exist distinct MAC query indices $i, j \in [q - \mu' + 1 \,..\, q]$ such that $(U_i, V_i) = (U_j, V_j)$.
- $\mathsf{bad}_2$: There exist distinct MAC query indices $i, j \in [q - \mu' + 1 \,..\, q]$ such that $(V_i, X_i) = (V_j, X_j)$.

- $\mathsf{bad}_3$: There exists a MAC query index $i \in [q]$ and a verification query index $j \in [v]$ such that $M_i \neq M'_j$ and $(N_i, U_i, V_i) = (N'_j, U'_j, V'_j)$.
- $\mathsf{bad}_4$: There exists a MAC query index $i \in [q]$ and a verification query index $j \in [v]$ such that $(N_i, M_i, T_i) = (N'_j, M'_j, T'_j)$.
- $\mathsf{bad}_5$: There exists a MAC query index $i \in [q - \mu' + 1 \,..\, q]$ and a verification query index $j \in [v]$ such that $N_i \neq N'_j$ and $(U_i, V_i) = (U'_j, V'_j)$.
- $\mathsf{bad}_6$: There exists a value $V \in \mathbb{F}_2^t$ whose multiplicity among all MAC queries is at least $v_{\mathsf{max}}$.

The bound in Theorem 4 follows from Lemma 3, 7, and 8. Lemma 7 and the proof are given in Section 6.1, and Lemma 8 and the proof are given in Section 6.2. The bound in Lemma 7 contains $\mu'$, hence we substitute it by $2\mu'$ for the number of faulty queries $\mu$ from (15). and 8, so that these bounds are (closely) balanced. Our choice is $v_{\mathsf{max}} = \sqrt[3]{2q^2}$. Then, the term $2q^2\delta'/v_{\mathsf{max}}^2$ in Lemma 7 becomes $2q^{\frac{2}{3}}\delta'$, and the term $1 - v/(2^n - (v_{\mathsf{max}} - 1 + v))$ in Lemma 8 becomes $v/(2^n - (2q^{\frac{2}{3}} + v))$. Finally, summing there bounds, we have the bound in Theorem 4 for $\mu > 0$. □

## 6.1 Bad Transcripts

**Lemma 7.** It holds that

$$\Pr\left[\mathsf{T}_{\mathsf{id}} \in \Gamma_{\mathsf{bad}}\right] \leq \frac{\mu'^2\delta\delta'}{2} + \frac{\mu'^2\delta'}{2 \cdot 2^n} + (\mu + \mu' + 1)v\delta\delta' + \frac{v}{2^n} + \frac{2q^2\delta'}{v_{\mathsf{max}}^2}\,.$$

*Proof.* In the following, we upper bound the probabilities of the individual $\mathsf{bad}$ events in the ideal world.

$\mathsf{bad}_1$. In this event, two distinct MAC queries with repeating nonces collide in both hash outputs of $H_{K_1}(N_i, M_i) = H_{K_1}(N_j, M_j)$ as well as $H'_{K_2}(N_i, M_i) = H'_{K_2}(N_j, M_j)$. Thus, we have at most $\binom{\mu'}{2}$ combinations. Since both hash functions over the choice of $K_1$ and $K_2$ are $\delta$- and $\delta'$-almost-universal, respectively, it holds that

$$\Pr\left[\mathsf{bad}_1\right] \leq \frac{\mu'^2\delta\delta'}{2}\,.$$

$\mathsf{bad}_2$. Here, two distinct MAC queries with repeating nonces collide in the hash outputs of $H'_{K_2}$ as well as in $T_i \oplus Y_i = T_j \oplus Y_j$. Again, since $H'_{K_2}$ over the choice of $K_2$ is $\delta'$-almost-universal and the tags $T_i$ and $T_j$ are sampled uniformly and independently at random from all $n$-bit values with probability $2^{-n}$, it holds that

$$\Pr\left[\mathsf{bad}_2\right] \leq \frac{\mu'^2\delta'}{2 \cdot 2^n}\,.$$

$\mathsf{bad}_3$. In this event, MAC and verification queries with the same nonce collide in both hash outputs of $H_{K_1}(N_i, M_i) = H_{K_1}(N'_j, M'_j)$ as well as $H'_{K_2}(N_i, M_i) = H'_{K_2}(N'_j, M'_j)$. For each verification query $(N'_j, M'_j, T'_j)$, there can be at most one nonce-respecting query and $\mu$ faulty queries with $N_i = N'_j$. Thus, we have at most $(\mu + 1)v$ combinations. Since both hash functions over the choice of $K_1$ and $K_2$ are $\delta$- and $\delta'$-almost-universal, respectively, it holds that

$$\Pr\left[\mathsf{bad}_3\right] \leq (\mu + 1)v\delta\delta'\,.$$

$\mathsf{bad}_4$. This event considers the case of a verification query $(N'_j, M'_j, T'_j)$ that is rejected in the ideal world, but is equal to a later MAC query $(N_i, M_i, T_i) = (N'_j, M'_j, T'_j)$ that is

valid. Since MAC queries are no duplicates, there can exist at most one such MAC query $(N_i, M_i)$ that matches a prior verification query. Since $T_i$ is random and independently sampled from $T'_j$, their probability to match is given by $2^{-n}$. Thus, it holds that

$$\Pr[\mathsf{bad}_4] \leq \frac{v}{2^n}.$$

$\mathsf{bad}_5$. In this event, nonce-repeating MAC and verification queries with the distinct nonces collide in both hash outputs of $H_{K_1}(N_i, M_i) = H_{K_1}(N'_j, M'_j)$ as well as $H'_{K_2}(N_i, M_i) = H'_{K_2}(N'_j, M'_j)$. Thus, we have at most $\mu' v$ combinations. Since both hash functions over the choice of $K_1$ and $K_2$ are $\delta$- and $\delta'$-almost-universal, respectively, it holds that

$$\Pr[\mathsf{bad}_5] \leq \mu' v \delta \delta'.$$

$\mathsf{bad}_6$. In this case, we can apply Corollary 1 to upper bound the probability that any value $V$ occurs more than $v_{\mathsf{max}}$ times by

$$\Pr[\mathsf{bad}_6] \leq \frac{2q^2\delta'}{v_{\mathsf{max}}^2}.$$

Lemma 7 follows from the sum of probabilities of all $\mathsf{bad}$ events. □

## 6.2 Good Transcripts

It remains to study $\mathsf{good}$ transcripts.

**Lemma 8.** For an arbitrary attainable $\mathsf{good}$ transcript $\tau$, it holds that

$$\frac{\Pr[\mathsf{T}_{\mathsf{re}} = \tau]}{\Pr[\mathsf{T}_{\mathsf{id}} = \tau]} \geq 1 - \frac{3v}{2^n - (v_{\mathsf{max}} - 1 + v)}.$$

*Proof.* In the following, we study the probability to obtain a $\mathsf{good}$ transcript in the ideal and the real world, respectively. The difficult part will be to determine the probability in the real world.
We define $\mathsf{p}_H$ as the probability that the hash keys are compatible with $\tau$. We further define $\mathcal{N}_{\mu'} \coloneqq \{N_i : i \in [q - \mu' + 1 \,..\, q]\}$ as the set (not multi-set) of repeating nonces. Moreover, let $w \coloneqq |\mathcal{N}_{\mu'}|$ the number of repeating nonces in $\mathcal{N}_{\mu'}$.

### 6.2.1 Probability in the Ideal World

In the ideal world, the probability to obtain the transcript is simply given as

$$\Pr[\mathsf{T}_{\mathsf{id}} = \tau] = \frac{\mathsf{p}_H}{(2^n)^q \cdot (2^n)^w} \tag{16}$$

since the hash keys, and each out of $q$ tags and each of $w$ values $Y_i$ is sampled uniformly and independently at random.

### 6.2.2 Probability in the Real World

It remains to lower bound the probability of obtaining $\tau$ in the real world. We consider the partitioned transcript in the following. We define $\mathsf{T}_{\mathsf{re}} = \tau_i$ to refer to $\mathsf{T}_{\mathsf{re}}$ produced the (partial) transcript $\tau_i$.

Probability of $\tau_{\mu'}$. Firstly we evaluate the probability to obtain $Y_i$ for $i \in [q - \mu' + 1 \,..\, q]$ in $\tau_{\mu'}$. For each $N_i \in \mathcal{N}_{\mu'}$, the probability of $Y_i = \widetilde{\pi}'(N_i, 0)$ is $2^{-n}$ also in the real world. Since they are sampled independently and uniformly at random, we obtain

$$\Pr[Y_i : i \in [q - \mu' + 1 \,..\, q]] = \frac{1}{(2^n)^w}. \tag{17}$$

Note that, due to the absence of $\mathsf{bad}_5$, there is no contradiction from choosing $Y_i$. Then we evaluate the probability to obtain leftover values in $\tau_{\mu'}$. For $s \in \{0,1\}^t$, let

$$\mu'_s \coloneqq |\{i \in [q-\mu'+1\,..\,q] : V_i = s\}|\,.$$

Thus, $\mu'_s$ is the number of nonce-repeating queries whose tweak input to $\widetilde{\pi}$ is equal to $s$, i.e., $V = s$ and $\sum_{s\in\{0,1\}^t} \mu'_s = \mu'$. Note that by the absence of $\mathsf{bad}_1$, a $\mathsf{good}$ transcript avoids hash collisions in $(U_i, V_i) = (U_j, V_j)$ for distinct $i, j \in [q-\mu'+1\,..\,q]$. For each tweak $s$, the number of solutions of $\widetilde{\pi}$ with tweak inputs $s$ is $(2^n)_{\mu'_s}$. Note that the absence of $\mathsf{bad}_1$ and $\mathsf{bad}_2$ in $\mathsf{good}$ transcripts ensures that there is no hash value such that the probability would become 0. Thus, it holds that

$$\Pr\left[\mathsf{T}_{\mathsf{re}} = \tau_{\mu'} \mid Y_i : i \in [q-\mu'+1\,..\,q]\right] = \frac{1}{\prod_{s\in\{0,1\}^t}(2^n)_{\mu'_s}} \geq \frac{1}{\prod_{s\in\{0,1\}^t}(2^n)^{\mu'_s}} = \frac{1}{(2^n)^{\mu'}}\,. \tag{18}$$

Probability of $\tau_n$ and $\tau_v$. We define another useful partition of $\tau_n$ and $\tau_v$,

- $\tau_{n,0} \coloneqq \{(N_i, M_i, T_i) \in \tau_n : \text{exists } j \in [q-\mu'+1\,..\,q] \text{ s.t. } (U_i, V_i) = (U_j, V_j)\}$;
- $\tau_{n,1} \coloneqq \tau_n \setminus \tau_{n,0}$;
- $\tau_{v,0} \coloneqq \{(N'_i, M'_i, T'_i) \in \tau_v : \text{exists } j, k \in [q] \text{ s.t. } N'_i = N_j \text{ and } (U'_i, V'_i) = (U_k, V_k)\}$;
- $\tau_{v,1} \coloneqq \{(N'_i, M'_i, T'_i) \in \tau_v : \text{exists } j \in [q] \text{ s.t. } (U'_i, V'_i) = (U_j, V_j)\} \setminus \tau_{v,0}$;
- $\tau_{v,2} \coloneqq \tau_v \setminus (\tau_{v,0} \cup \tau_{v,1})$;
- $\mathcal{I}_{\mu'} \coloneqq \{j : (N_j, M_j, T_j) \in \tau_{\mu'}\}$;
- For $i \in \{0,1\}$, $\mathcal{I}_{n,i} \coloneqq \{j : (N_j, M_j, T_j) \in \tau_{n,i}\}$;
- For $i \in \{0,1,2\}$, $\mathcal{I}_{v,i} \coloneqq \{j : (N'_j, M'_j, T'_j) \in \tau_{v,i}\}$.

For each $(N_i, M_i, T_i) \in \tau_{n,0}$, the evaluation of $X_i$ is already fixed by $\tau_{\mu'}$, so the probability that $\Pr[Y_i = X_i \oplus T_i] = 2^{-n}$. It follows that

$$\Pr\left[\mathsf{T}_{\mathsf{re}} = \tau_{n,0} \mid \mathsf{T}_{\mathsf{re}} = \tau_{\mu'}\right] = \frac{1}{(2^n)^{|\tau_{n,0}|}}\,. \tag{19}$$

For each $(N_i, M_i, T_i) \in \tau_{n,1}$, both $X_i$ and $Y_i$ were not defined by $\tau_{\mu'}$ and $\tau_{n,0}$. Let $\mathcal{W}$ be the hash outputs (or namely, the inputs to $\widetilde{\pi}$) in $\tau_{n,1}$, i.e.,

$$\mathcal{W} = \{(U_j, V_j) : j \in \mathcal{I}_{n,1}\}$$

and we introduce an order on $\mathcal{W}$, so $\mathcal{W} = (W_1, \ldots, W_x)$ where $x = |\mathcal{W}| \leq |\tau_{n,1}|$. For $i \leq x$, let

$$r_i \coloneqq |\{j \in \mathcal{I}_{v,0} : \exists k \in \mathcal{I}_{n,1} \text{ s.t. } W_i = (U_k, V_k) \text{ and } N'_j = N_k\} \cup \{j \in \mathcal{I}_{v,0} : W_i = (U'_j, V'_j)\}|,$$
$$s_i \coloneqq |\{V_j : j \in (\mathcal{I}_{\mu'} \cup \mathcal{I}_{n,0}) \text{ or } \exists k < i, W_k = (U_j, V_j)\}|\,,$$

where $r_i$ counts the number of inequalities that $\widetilde{\pi}(W_i)$ should be satisfied and $s_i$ counts the number of available evaluations of $\widetilde{\pi}(W_i)$ assuming that we fixed $\widetilde{\pi}$ on $\tau_{\mu'}$, $\tau_{n,0}$ and $W_j$ where $j < i$. Then,

$$\Pr\left[\mathsf{T}_{\mathsf{re}} = \tau_{n,1}, \tau_{v,0} \mid \mathsf{T}_{\mathsf{re}} = \tau_{\mu'}, \tau_{n,0}\right] \geq \frac{1}{2^{n|\tau_{n,1}|}} \prod_{i\in[x]} \frac{2^n - r_i - s_i}{2^n - s_i} \geq \frac{1}{2^{n|\tau_{n,1}|}} \left(1 - \frac{2v}{2^n - v_{\max}}\right), \tag{20}$$

where the last inequality comes from the inequalities $\sum_{i\in[x]} r_i \le 2v$ and $s_i \le v_{\max}$.
For each $i \in \mathcal{I}_{v,1}$, the $X'_i$ is already fixed, so the probability that $\Pr[Y'_i = X'_i \oplus T'_i] = 2^{-n}$. Also, for each $i \in \mathcal{I}_{v,2}$, the real world fixes $Y'_i = \widetilde{\pi}'(N'_i, 0^n)$ if it is not defined. For each such $i$, the number of solutions for $\widetilde{\pi}(V_i, U_i)$ is at least $2^n - (v_{\mathsf{max}} - 1 + v)$ due to the absence of $\mathsf{bad}_6$. Therefore, for $i \in \mathcal{I}_{v,1} \cup \mathcal{I}_{v,2}$,

$$\Pr[X'_i = Y'_i \oplus T'_i] \le \frac{1}{2^n - (v_{\mathsf{max}} - 1 + v)}.$$

It follows that

$$\begin{aligned}\Pr[\mathsf{T}_{\mathsf{re}} = \tau_{v,1}, \tau_{v,2} \mid \mathsf{T}_{\mathsf{re}} = \tau_{\mu'}, \tau_n, \tau_{v,0}] &\ge \left(1 - \frac{1}{2^n - (v_{\mathsf{max}} - 1 + v)}\right)^{|\tau_{v,1}|+|\tau_{v,2}|} \\ &\ge 1 - \frac{v}{2^n - (v_{\mathsf{max}} - 1 + v)}. \end{aligned} \tag{21}$$

Note that for a verification query $(N'_i, M'_i, T'_i)$, if there exists a MAC query $(N_j, M_j)$ such that $(N'_i, M'_i) = (N_j, M_j)$, $\widetilde{\pi}(V_j, U_j)$ $(= \widetilde{\pi}(V'_i, U'_i))$ was defined so that it is not equal to $Y'_i \oplus T'_i$ due to the absence of $\mathsf{bad}_4$.

Summing up. Multiplying all bounds from Equations (17) through (21) yields

$$\begin{aligned}\Pr[\mathsf{T}_{\mathsf{re}} = \tau] &\ge \frac{\mathsf{p}_H \cdot (1 - \frac{2v}{2^n - v_{\mathsf{max}}}) \cdot (1 - \frac{v}{2^n - (v_{\mathsf{max}} - 1 + v)})}{(2^n)^w \cdot (2^n)^{\mu'} \cdot (2^n)^{|\tau_{n,0}|+|\tau_{n,1}|}} \\ &\ge \frac{1}{(2^n)^{q+w}} \cdot \left(1 - \frac{3v}{2^n - (v_{\mathsf{max}} - 1 + v)}\right) \cdot \mathsf{p}_H.\end{aligned}$$

Together with Equation (16), we obtain that

$$\frac{\Pr[\mathsf{T}_{\mathsf{re}} = \tau]}{\Pr[\mathsf{T}_{\mathsf{id}} = \tau]} \ge \frac{\frac{1}{(2^n)^{q+w}} \cdot \left(1 - \frac{3v}{2^n - (v_{\mathsf{max}} - 1 + v)}\right) \cdot \mathsf{p}_H}{\frac{1}{(2^n)^{q+w}} \cdot \mathsf{p}_H} = 1 - \frac{3v}{2^n - (v_{\mathsf{max}} - 1 + v)}.$$

The bound in Lemma 8 follows. □

## 6.3 Bound for $\mu > 0$ without $\mathsf{bad}_6$

For a maximum message length in blocks $\ell$, if $\delta' = \ell/2^t$, the term $2q^{\frac{2}{3}}\delta'$ depends on the message length. For $\ell < 2^{t/3}$, the security is not endangered before $q$ reaches $2^t$. The term $2q^{\frac{2}{3}}\delta'$ is introduced by the bad event $\mathsf{bad}_6$ that defines the maximum multiplicity of $V$ for MAC queries. Thus, removing $\mathsf{bad}_6$ from the bad events in the proof of Theorem 4 for $\mu > 0$ would allow us to remove the $\ell$-dependent term. In this case, the number of solutions for $\widetilde{\pi}(V_i, U_i)$ in the analysis of $\tau_v$ in Subsection 6.2 changes – more precisely, $v_{\mathsf{max}}$ is replaced with $q$. By the replacement, the lower bound in Eq. (21) becomes $1 - 3v/(2^n - (q + v))$. We thus get the $\ell$-free term $3v/(2^n - (q + v))$, which is valid as long as $q < 2^n$. Adding the term to the bound in Theorem 4 would yield the following corollary.

**Corollary 2.** Let $\delta, \delta' > 0$, $H : \mathcal{K} \times \mathcal{M} \to \{0,1\}^n$ be a $\delta$-almost-universal hash function and $H' : \mathcal{K}' \times \mathcal{M} \to \{0,1\}^t$ be a $\delta'$-almost universal hash function, $\widetilde{\pi}, \widetilde{\pi}' \leftarrow_{\$} \mathsf{TPerm}(\mathbb{F}_2^t, \mathbb{F}_2^n)$, and $K_1 \leftarrow_{\$} \mathcal{K}$ as well as $K_2 \leftarrow_{\$} \mathcal{K}'$. For non-negative integers $\mu$, $q$, and $v$, if $\mu > 0$, i.e., adversaries are nonce-misusing, we have

$$\begin{aligned}\mathsf{Adv}^{\mathsf{mac}}_{\mathsf{eHaT}[H_{K_1}, H'_{K_2}, \widetilde{\pi}, \widetilde{\pi}']}(\mu, q, v) \le 2\mu^2\delta\delta' + \frac{2\mu^2\delta'}{2^n} + (3\mu + 1)v\delta\delta' + \frac{v}{2^n} \\ + \min\left\{2q^{\frac{2}{3}}\delta' + \frac{3v}{2^n - (2q^{\frac{2}{3}} + v)}, \frac{3v}{2^n - (q + v)}\right\}.\end{aligned}$$

# 7 Conclusion

This work proposed NaT2 and eHaT, two highly secure nonce-based MACs. Taking NaT and HaT proposed by Cogliati et al. [CLS17] as a baseline, we derive NaT2 and eHaT with conceptually simple changes. Our proposals possess almost full security in the nonce-respecting and beyond-birthday-bound security in the nonce-misusing setting. Since neither NaT nor (a simple nonce-based variant of) HaT could achieve both properties simultaneously, our constructions enhance their security guarantees well.
Our constructions NaT2 and eHaT provide the same level of security in the nonce-respecting setting as NaT and HaT. However, in the nonce-misuse setting, NaT2 and eHaT provide stronger security in terms of the threshold number of verification queries and MAC queries, respectively. Few more possible future directions exist, most notably, studying the tightness of the bounds or related MAC designs in the (ideal-)block cipher setting.

# Acknowledgments

We are highly thankful for the fruitful comments from the reviewers at ToSC as well as the organizers of the Asian Symmetric Key workshop at Kobe 2019. Jooyoung Lee was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (Ministry of Science and ICT), No. NRF-2017R1E1A1A03070248.

# References

[3GP99] 3GPP. Technical Specification 35.201. 3G Security: Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 1: f8 and f9 Specifications. Technical report, 3GPP, 1999.

[Ber05a] Daniel J. Bernstein. Stronger Security Bounds for Wegman-Carter-Shoup Authenticators. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes on Computer Science*, pages 164–180. Springer, 2005.

[Ber05b] Daniel J. Bernstein. The Poly1305-AES Message-Authentication Code. In Henri Gilbert and Helena Handschuh, editors, *FSE*, volume 3557 of *Lecture Notes on Computer Science*, pages 32–49. Springer, 2005.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO II*, volume 9815 of *Lecture Notes on Computer Science*, pages 123–153. Springer, 2016. Full version at https://eprint.iacr.org/2016/660.

[BL16] Karthikeyan Bhargavan and Gaëtan Leurent. On the Practical (In-)Security of 64-bit Block Ciphers: Collision Attacks on HTTP over TLS and OpenVPN. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS*, pages 456–467. ACM, 2016.

[CLS17] Benoít Cogliati, Jooyoung Lee, and Yannick Seurin. New Constructions of MACs from (Tweakable) Block Ciphers. *IACR Trans. Symmetric Cryptol.*, 2017(2):27–58, 2017.

[CS14] Shan Chen and John P. Steinberger. Tight Security Bounds for Key-Alternating Ciphers. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT*,

volume 8441 of *Lecture Notes on Computer Science*, pages 327–350. Springer, 2014. Full version at https://eprint.iacr.org/2013/222.

[DDN+17] Nilanjan Datta, Avijit Dutta, Mridul Nandi, Goutam Paul, and Liting Zhang. Single Key Variant of PMAC_Plus. *IACR Trans. Symmetric Cryptol.*, 2017(4):268–305, 2017.

[DDNP18] Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Goutam Paul. Double-block Hash-then-Sum: A Paradigm for Constructing BBB Secure PRF. *IACR Trans. Symmetric Cryptol.*, 2018(3):36–92, 2018.

[DDNY18] Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Kan Yasuda. Encrypt or Decrypt? To Make a Single-Key Beyond Birthday Secure Nonce-Based MAC. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO I*, volume 10991 of *Lecture Notes on Computer Science*, pages 631–661. Springer, 2018.

[DNT19] Avijit Dutta, Mridul Nandi, and Suprita Talnikar. Beyond Birthday Bound Secure MAC in Faulty Nonce Model. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT I*, volume 11476 of *Lecture Notes on Computer Science*, pages 437–466. Springer, 2019.

[Dwo05] Morris J Dworkin. SP 800-38B. Recommendation for block cipher modes of operation: The CMAC mode for authentication, 2005.

[Dwo16] Morris J Dworkin. Recommendation for block cipher modes of operation: The cmac mode for authentication. Technical report, NIST, 2016. Supersedes SP 800-38B (https://www.nist.gov/node/562931).

[HP08] Helena Handschuh and Bart Preneel. Key-Recovery Attacks on Universal Hash Function Based MAC Algorithms. In David A. Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes on Computer Science*, pages 144–161. Springer, 2008.

[HT16] Viet Tung Hoang and Stefano Tessaro. Key-Alternating Ciphers and Key-Length Extension: Exact Bounds and Multi-user Security. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO I*, volume 9814 of *Lecture Notes on Computer Science*, pages 3–32. Springer, 2016.

[IMPS17] Tetsu Iwata, Kazuhiko Minematsu, Thomas Peyrin, and Yannick Seurin. ZMAC: A Fast Tweakable Block Cipher Mode for Highly Secure Message Authentication. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO, Part III*, volume 10403 of *Lecture Notes on Computer Science*, pages 34–65. Springer, 2017. Full version at https://eprint.iacr.org/2017/535.

[ISO99] ISO/IEC. Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher. Technical report, ISO/IEC, 1999.

[ISO11] ISO/IEC. Information Technology – Security Techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms Using a Block Cipher. Technical report, ISO/IEC, 2011.

[JN20] Ashwin Jha and Mridul Nandi. Tight Security of Cascaded LRW2. *J. Cryptology*, pages 1378–1432, 2020.

[JNP14a] Jérémy Jean, Ivica Nikolić, and Thomas Peyrin. Tweaks and Keys for Block Ciphers: The TWEAKEY Framework. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT II*, volume 8874 of *Lecture Notes on Computer Science*, pages 274–288. Springer, 2014.

[JNP14b] Jérémy Jean, Ivica Nikolić, and Thomas Peyrin. Tweaks and Keys for Block Ciphers: The TWEAKEY Framework. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT II*, volume 8874 of *Lecture Notes on Computer Science*, pages 274–288. Springer, 2014.

[KLL20] Seongkwang Kim, Byeonghak Lee, and Jooyoung Lee. Tight Security Bounds for Double-Block Hash-then-Sum MACs. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT I*, volume 12105 of *Lecture Notes on Computer Science*, pages 435–465. Springer, 2020.

[Kro06] Ted Krovetz. Message Authentication on 64-Bit Architectures. In Eli Biham and Amr M. Youssef, editors, *SAC*, volume 4356 of *Lecture Notes on Computer Science*, pages 327–341. Springer, 2006.

[LN17] Eik List and Mridul Nandi. ZMAC+ - An Efficient Variable-output-length Variant of ZMAC. *IACR Transactions of Symmetric Cryptology*, 2017(4):306–325, 2017.

[LNS18] Gaëtan Leurent, Mridul Nandi, and Ferdinand Sibleyras. Generic Attacks Against Beyond-Birthday-Bound MACs. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO I*, volume 10991 of *Lecture Notes on Computer Science*, pages 306–336. Springer, 2018.

[LRW02] Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable Block Ciphers. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes on Computer Science*, pages 31–46. Springer, 2002.

[Mor07] Morris J. Dworkin. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. *NIST Special Publication*, 800-38D, Nov 28 2007.

[Nai15] Yusuke Naito. Full PRF-Secure Message Authentication Code Based on Tweakable Block Cipher. In Man Ho Au and Atsuko Miyaji, editors, *ProvSec*, volume 9451 of *Lecture Notes on Computer Science*, pages 167–182. Springer, 2015.

[Nai17] Yusuke Naito. Blockcipher-Based MACs: Beyond the Birthday Bound Without Message Length. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT III*, volume 10626 of *Lecture Notes on Computer Science*, pages 446–470. Springer, 2017.

[Nai18] Yusuke Naito. On the Efficiency of ZMAC-Type Modes. In Jan Camenisch and Panos Papadimitratos, editors, *CANS*, volume 11124 of *Lecture Notes on Computer Science*, pages 190–210. Springer, 2018.

[Pat08] Jacques Patarin. The "Coefficients H" Technique. In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *SAC*, volume 5381 of *Lecture Notes on Computer Science*, pages 328–345. Springer, 2008.

[Pat10] Jacques Patarin. Introduction to Mirror Theory: Analysis of Systems of Linear Equalities and Linear Non Equalities for Cryptography. *IACR Cryptology ePrint Archive*, 2010:287, 2010.

[Pat17] Jacques Patarin. Mirror theory and cryptography. *Appl. Algebra Eng. Commun. Comput.*, 28(4):321–338, 2017.

[PS16] Thomas Peyrin and Yannick Seurin. Counter-in-Tweak: Authenticated Encryption Modes for Tweakable Block Ciphers. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO I*, volume 9814 of *Lecture Notes on Computer Science*, pages 33–63. Springer, 2016.

[PvO95] Bart Preneel and Paul C. van Oorschot. MDx-MAC and Building Fast MACs from Hash Functions. In Don Coppersmith, editor, *CRYPTO*, volume 963 of *Lecture Notes on Computer Science*, pages 1–14. Springer, 1995.

[Rog04] Phillip Rogaway. Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In Pil Joong Lee, editor, *ASIACRYPT*, volume 3329 of *Lecture Notes on Computer Science*, pages 16–31. Springer, 2004.

[Sar11] Palash Sarkar. A trade-off between collision probability and key size in universal hashing using polynomials. *Des. Codes Cryptogr.*, 58(3):271–278, 2011.

[Sho96] Victor Shoup. On Fast and Provably Secure Message Authentication Based on Universal Hashing. In Neal Koblitz, editor, *CRYPTO*, volume 1109 of *Lecture Notes on Computer Science*, pages 313–328. Springer, 1996.

[SW19] Yaobin Shen and Lei Wang. On Beyond-Birthday-Bound Security: Revisiting the Development of ISO/IEC 9797-1 MACs. *IACR Trans. Symmetric Cryptol.*, 2019(2):146–168, 2019.

[WC81] 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.

[Yas10] Kan Yasuda. The Sum of CBC MACs Is a Secure PRF. In Josef Pieprzyk, editor, *CT-RSA*, volume 5985 of *Lecture Notes on Computer Science*, pages 366–381. Springer, 2010.

[Yas11] Kan Yasuda. A New Variant of PMAC: Beyond the Birthday Bound. In Phillip Rogaway, editor, *CRYPTO*, volume 6841 of *Lecture Notes on Computer Science*, pages 596–609. Springer, 2011.

[Yuv79] Gideon Yuval. How to Swindle Rabin. *Cryptologia*, 3(3):187–191, 1979.

[ZWSW12] Liting Zhang, Wenling Wu, Han Sui, and Peng Wang. 3kf9: Enhancing 3GPP-MAC beyond the Birthday Bound. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT*, volume 7658 of *Lecture Notes on Computer Science*, pages 296–312. Springer, 2012.