New Constructions of MACs from (Tweakable) Block Ciphers

Benoît Cogliati, Jooyoung Lee and Yannick Seurin

University of Luxembourg
KAIST, Korea
ANSSI, Paris, France
benoitcogliati@hotmail.fr
hicall@kaist.ac.kr
yannick.seurin@m4x.org

Abstract. We propose new constructions of Message Authentication Codes (MACs) from tweakable or conventional block ciphers. Our new schemes are either stateless and deterministic, nonce-based, or randomized, and provably secure either in the standard model for tweakable block cipher-based ones, or in the ideal cipher model for block cipher-based ones. All our constructions are very efficient, requiring only one call to the underlying (tweakable) block cipher in addition to universally hashing the message. Moreover, the security bounds we obtain are quite strong: they are beyond the birthday bound, and nonce-based/randomized variants provide graceful security degradation in case of misuse, i.e., the security bound degrades linearly with the maximal number of repetitions of nonces/random values.

Keywords: MAC · tweakable block cipher · nonce-misuse resistance · graceful security degradation

1 Introduction

MACs. A Message Authentication Code (MAC) is a fundamental symmetric primitive allowing two entities sharing a secret key to verify that a received message originates from one of the two parties and was not modified by an attacker. Most existing MACs are built from a block cipher, e.g., CBC-MAC [BKR00] or OMAC [IK03], or from a cryptographic hash function, e.g., HMAC [BCK96]. At a high level, many of these constructions follow the well-established UHF-then-PRF design paradigm: the message $M$ is first mapped onto a short string through a universal hash function (UHF), and then “encrypted” through a fixed-input-length PRF to obtain a short tag.\footnote{Actually, this kind of construction yields a variable-input-length PRF rather than a mere MAC.} This method is simple (in particular, it is deterministic and stateless), yet its security caps at the so-called birthday-bound since any collision at the output of the UHF, which translate into a tag collision, is usually enough to break the security of the scheme. Better security bounds can be obtained by incorporating in the tag computation a nonce (a value that never repeats), e.g., in Wegman-Carter type MACs [WC81, Sho96, Ber05, CS16] or a random value [BGK99, JJV02, JL04, Min10].

Our Contribution. We propose new MAC constructions, which are either nonce-based/randomized or stateless and deterministic, and which are based on a universal hash function and either on a (conventional) block cipher or a tweakable block cipher. Hence, in total, we propose four new constructions, two of which can be analyzed in two slightly different (but related) security models (namely nonce-based or randomized). Tweakable block ciphers (TBCs) are a generalization of conventional block ciphers which, in addition to a message and a cryptographic key, take another (public, or even controlled by the adversary) input called a tweak. This tweak should provide inherent variability to the block cipher and plays a similar role to an IV or a nonce in an encryption scheme. The
security notion for this primitive was first formalized in [LRW02], where it was pointed out that tweakable block ciphers are very useful for building various higher level cryptographic schemes.

Our two TBC-based constructions follow the traditional UHF-then-PRF approach, the PRF being “instantiated” from the TBC $E$. The starting point of our nonce-based construction, called NaT (Nonce-as-Tweak) and depicted on Figure 1 (top left), is the simple remark that, as long as tweaks do not repeat, a tweakable block cipher behaves as a random function. Hence, if the hash of each message is encrypted with a fresh nonce as tweak, collisions among hash values don’t matter since the hashes are encrypted by “independent” random functions $\tilde{E}_N^N$. Even if tweaks (i.e., nonces) repeat, the security loss is negligible as long as the number of repetitions is small. The provable security bound for the NaT construction is dominated by terms of the form $\mu q \varepsilon$, where $\mu$ denotes the maximal number of repetitions of any nonce, $q$ denotes the number of adversarial (MAC or verification) queries, and $\varepsilon$ is the parameter characterizing the collision probability of the UHF. A typical value (e.g., for polynomial-based hashing [Sho96, Ber07]) for $\varepsilon$ is $\ell/2^n$, where $n$ is the output length of the UHF (which is also the block length of the TBC) and $\ell$ is the maximal length of messages in $n$-bit blocks. Hence, in the nonce-respecting case (i.e., $\mu = 1$) the adversary’s advantage is of the form $q\ell/2^n$, whereas in the nonce-misusing case (where $\mu$ might be as large as $q$), it becomes $q^2\ell/2^n$, i.e., a birthday-type bound. The security bound degrades linearly with $\mu$, the maximal number of repetitions of nonces. We note that the NaT construction is used in version 1.41 of the authenticated encryption scheme Deoxys [JNPS16], a third round candidate of the CAESAR competition.

To obtain a stateless deterministic TBC-based construction, one simply replaces the nonce by an independent hash of the message. The resulting construction is called HaT (Hash-as-Tweak), see Figure 1 (top right). This construction is secure beyond the birthday bound. Both NaT and HaT are provably secure in the standard model, assuming only that the TBC is a secure pseudorandom tweakable permutation.

Our two block cipher-based constructions, on the other hand, depart from the standard UHF-then-PRF approach, since the output transformation is unkeyed. Actually, they can be seen as block cipher-based instantiations of a new paradigm (which, to the best of our knowledge, has not been formally explored yet), which could be dubbed UHF-then-RO: the tag is computed as $T = G(H_K(M))$, where $H$ is a (keyed) uniform and universal hash function, and $G$ is a (keyless) cryptographic hash function. It is easy to prove (and we do so in Appendix B) that this construction is a secure MAC (in fact, a variable-input-length PRF) in the random oracle (RO) model for $G$.

Obviously, the output transformation must be hard to invert (as otherwise the adversary can compute the output of the UHF from the tag), which for the nonce-based construction implies that we must use the block cipher in Davies-Meyer mode. The resulting variants of NaT and HaT, called respectively NaK (Nonce-as-Key) and HaK (Hash-as-Key), are depicted in Figure 1 (bottom). They are provably secure in the ideal cipher model [BRS02].

We provide a comparison of our new constructions with existing UHF-based MAC constructions in Table 1.

### Proof Technique.

Our proofs rely on the H-coefficients technique, which has been introduced by Patarin [Pat08b], and has recently been highlighted by Chen and Steinberger for analyzing the iterated Even-Mansour cipher [CS14]. This method is typically used to prove information-theoretic pseudorandomness of constructions such as Feistel networks [Pat90, Pat91, Pat10], the XOR of permutations [Pat08a, Pat13] or Even-Mansour constructions [CLS15, CS15b, CS15a, Men15, HT16]. The use of the H-coefficients technique to study the security of MAC constructions (in particular, to directly handle

---

2 A natural question is whether standard security assumptions on $G$ are sufficient to prove MAC/PRF security.
verification queries rather than appealing to generic results resulting in looser bounds) was previously introduced by Cogliati and Seurin [CS16].

**More Related Work.** Several other MAC constructions based on tweakable block ciphers have been proposed. For example, TBC-MAC [LRW02] and TBC-MAC2 [LST12] are two such constructions. They are similar in design to CBC-MAC, however the chaining in these constructions is done through the tweak. Because of their structure, these two constructions require as many calls to the tweakable block cipher as the number of blocks in the message, whereas our constructions only require one call to the TBC and one to the universal hash function, which can be much more efficient depending on the choice of the UHF. Moreover, the proven security of TBC-MAC is still birthday bound and, while TBC-MAC2 has a security bound comparable to our bounds, it requires the underlying TBC to have a tweak length much larger than the block length, which is not the case in our constructions.

Black and Cochran [BC09] also proposed a nonce-based MAC construction, called WMAC, which is based on a PRF and a universal hash function. Our NaT construction can actually be seen as a particular case of WMAC, where the PRF is instantiated by a tweakable block cipher. However, our security bound is actually tighter than what would be achieved by simply applying [BC09, Theorem 6] to our construction.

Naito [Nai15] and more recently List and Nandi [LN17] have proposed TBC-based constructions of stateless deterministic MACs that do not use a generic UHF, but instead construct the UHF from the underlying TBC (so that the resulting constructions are entirely TBC-based). In principle, the two UHFs of our construction HaT can be instantiated for example with PMAC1 [Rog04] (the TBC-based generalization of PMAC [BR02]) to obtain
Table 1: Comparison of our new constructions with prominent existing MAC constructions based on an arbitrary (xor-)universal hash function. WC stands for Wegman-Carter; SD stands for stateless deterministic; “prim.” indicates which primitive is used in addition to the UHF; “# calls” gives the number of calls to the underlying primitive (in addition to the UHF call); “BBB” indicates whether the construction is secure beyond the birthday bound (when nonces are not repeated for nonce-based ones); “NMR” indicates whether nonce-based constructions are nonce-misuse resistant; “proof” indicates whether the security proof is in the standard model (SM) or the ideal cipher model (ICM).

<table>
<thead>
<tr>
<th>construction</th>
<th>type</th>
<th>prim.</th>
<th># calls</th>
<th>BBB</th>
<th>NMR</th>
<th>proof</th>
<th>ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>UHF-then-BC</td>
<td>SD</td>
<td>BC</td>
<td>1</td>
<td>×</td>
<td>—</td>
<td>SM</td>
<td>BS, Section 7</td>
</tr>
<tr>
<td>NaT</td>
<td>SD</td>
<td>TBC</td>
<td>1</td>
<td>✓</td>
<td>—</td>
<td>SM</td>
<td>this paper</td>
</tr>
<tr>
<td>NaK</td>
<td>SD</td>
<td>BC</td>
<td>1</td>
<td>✓</td>
<td>—</td>
<td>ICM</td>
<td>this paper</td>
</tr>
<tr>
<td>PRF-based WC</td>
<td>nonce</td>
<td>PRF</td>
<td>1</td>
<td>✓</td>
<td>×</td>
<td>SM</td>
<td>[WC81]</td>
</tr>
<tr>
<td>BC-based WC</td>
<td>nonce</td>
<td>BC</td>
<td>1</td>
<td>×</td>
<td>×</td>
<td>SM</td>
<td>[Sho96, Ber05]</td>
</tr>
<tr>
<td>EWCDM</td>
<td>nonce</td>
<td>BC</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>SM</td>
<td>[CS16]</td>
</tr>
<tr>
<td>NaT</td>
<td>nonce</td>
<td>TBC</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>SM</td>
<td>this paper</td>
</tr>
<tr>
<td>NaK</td>
<td>nonce</td>
<td>BC</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>ICM</td>
<td>this paper</td>
</tr>
<tr>
<td>EHTM</td>
<td>rand.</td>
<td>PRF</td>
<td>2</td>
<td>✓</td>
<td>—</td>
<td>SM</td>
<td>[Min10]</td>
</tr>
<tr>
<td>NaT</td>
<td>rand.</td>
<td>TBC</td>
<td>1</td>
<td>✓</td>
<td>—</td>
<td>SM</td>
<td>this paper</td>
</tr>
<tr>
<td>NaK</td>
<td>rand.</td>
<td>BC</td>
<td>1</td>
<td>✓</td>
<td>—</td>
<td>ICM</td>
<td>this paper</td>
</tr>
</tbody>
</table>

a similarly “purely” TBC-based construction, however the resulting construction makes two TBC calls per message block, whereas Naito’s and List-Nandi’s constructions only make one, so that we do not claim to compete with them in terms of efficiency.

**Organization.** We first establish the notation and recall standard security definitions in Section 2. We also give a general lemma allowing to translate any security bound for a nonce-based MAC that provides graceful security degradation with respect to the number of nonce repetitions to the corresponding randomized scheme (where “nonces” are chosen uniformly at random). We then describe and prove the security of our TBC-based constructions in Section 3 and of our block cipher-based constructions in Section 4. In Section 5, we give a simple generic result about the security loss induced by tag truncation that might be of independent interest.

## 2 Preliminaries

### 2.1 General Definitions

**Basic Notation.** Given a non-empty set $X$, we let $X \leftarrow_{\$} X$ denote the draw of an element $X$ from $X$ uniformly at random. The set of all functions from $X$ to $Y$ is denoted $\text{Func}(X, Y)$, and the set of all permutations of $X$ is denoted $\text{Perm}(X)$. The set of binary strings of length $n$ is denoted $\{0, 1\}^n$. The set of all functions from $\{0, 1\}^n$ to $\{0, 1\}^n$ is simply denoted $\text{Func}(n)$, and the set of all permutations of $\{0, 1\}^n$ is simply denoted $\text{Perm}(n)$. 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. Remark that, using our notation, the probability that a random permutation $P \leftarrow_{\$} \text{Perm}(n)$ satisfies $q$ equations $P(X_i) = Y_i$ for distinct $X_i$’s and distinct $Y_i$’s is exactly $1/(2^n)^q$. 
PRFs. A keyed function with key space $\mathcal{K}$, domain $\mathcal{X}$, and range $\mathcal{Y}$ is a function $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$. We write $F_K(X)$ for $F(K, X)$. A $(q,t)$-adversary against $F$ is an algorithm $A$ with oracle access to a function from $\mathcal{X}$ to $\mathcal{Y}$, making at most $q$ oracle queries, running in time at most $t$, and outputting a single bit. The advantage of $A$ in breaking the PRF-security of $F$, i.e., in distinguishing $F$ from a uniformly randomly chosen function $R \leftarrow \mathcal{R}$, is defined as

$$\text{Adv}_F^\text{PRF}(A) = \left| \Pr [K \leftarrow \mathcal{R} : A^{F_K} = 1] - \Pr [R \leftarrow \mathcal{R} : A^R = 1] \right|.$$ 

**Block Ciphers and Tweakeable Block Ciphers.** A block cipher with key space $\mathcal{K}$ and message space $\mathcal{X}$ is a mapping $E : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$ such that for any key $K \in \mathcal{K}$, $X \mapsto E(K, X)$ is a permutation of $\mathcal{X}$. We write $E_K(X)$ for $E(K, X)$. The security proofs for block cipher-based constructions studied in this paper will be done in the ideal cipher model (ICM); this means that a block cipher $E$ is drawn uniformly at random from the set of all block ciphers with key space $\mathcal{K}$ and message space $\mathcal{X}$, and given as an oracle (both in the encryption and decryption directions) to the adversary.

A tweakable permutation with tweak space $W$ and message space $X$ is a mapping $\tilde{E} : W \times X \to X$ such that, for any tweak $W \in W$, $X \mapsto \tilde{E}(W, X)$ is a permutation of $X$. We let $\text{Perm}(W, X)$ denote the set of all tweakable permutations with tweak space $W$ and message space $X$. As in the case of simple permutations, we let $\text{Perm}(W, n)$ denote the set of all tweakable permutations with tweak space $W$ and message space $\{0, 1\}^n$.

A tweakable block cipher $\tilde{E}$ with key space $\mathcal{K}$, tweak space $W$ and message space $X$ is a mapping $\tilde{E} : \mathcal{K} \times W \times X \to X$ such that for any key $K \in \mathcal{K}$, $(W, X) \mapsto \tilde{E}(K, W, X)$ is a tweakable permutation with tweak space $W$ and message space $X$. We write $E_K(T, X)$ or $E_{\tilde{E}}(X)$ for $E(K, T, X)$. A $(q,t)$-adversary against the security of $\tilde{E}$ as a tweakable pseudorandom permutation (TPRP-security) is an algorithm $A$ with oracle access to a tweakable permutation with tweak space $W$ and message space $X$, making at most $q$ oracle queries, running in time at most $t$, and outputting a single bit. The advantage of $A$ in breaking the TPRP-security of $\tilde{E}$ is defined as

$$\text{Adv}_{\tilde{E}}^\text{TPRP}(A) = \left| \Pr [K \leftarrow \mathcal{R} : \tilde{E}^{E_K} = 1] - \Pr [\tilde{E} \leftarrow \mathcal{R} : \tilde{E}^\tilde{F} = 1] \right|.$$ 

Note that we do not require the “strong TPRP”-security for $\tilde{E}$, i.e., when the adversary is allowed to adaptively query an encryption and a decryption oracle, since the underlying tweakable block cipher in our construction will only be queried in one direction.

**MACs.** We define three security notions for MACs: stateless and deterministic MACs (SD-MACs), nonce-based MACs, and randomized MACs.

**Definition 1** (SD-MAC). Let $\mathcal{K}, \mathcal{M},$ and $\mathcal{T}$ be non-empty sets. Let $F : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ be a keyed function. For $K \in \mathcal{K}$, let $\text{Ver}_K$ be the verification oracle which takes as input a pair $(M, T) \in \mathcal{M} \times \mathcal{T}$ and returns 1 (“accept”) if $F_K(M, T) = T$, and 0 (“reject”) otherwise. A $(q_m, q_v, t)$-adversary against the sdMAC-security of $F$ is an adversary $A$ with oracle access to the two oracles $F_K$ and $\text{Ver}_K$ for $K \in \mathcal{K}$, making at most $q_m$ “MAC” queries to its first oracle and at most $q_v$ “verification” queries to its second oracle, and running in time at most $t$. We say that $A$ forges if any of its queries to $\text{Ver}_K$ returns 1. The advantage of $A$ against the sdMAC-security of $F$ is defined as

$$\text{Adv}_F^{\text{sdMAC}}(A) = \Pr [K \leftarrow \mathcal{R} : A^{F_K, \text{Ver}_K \text{ forg}es}] ,$$

where the probability is also taken over the random coins of $A$, if any. The adversary is not allowed to ask a verification query $(M, T)$ if a previous query $M$ to $F_K$ returned $T$. 


Given four non-empty sets $K$, $N$, $M$, and $T$, a nonce-based keyed function with key space $K$, nonce space $N$, message space $M$ and range $T$ is simply a function $F : K \times N \times M \to T$. Stated otherwise, it is a keyed function whose domain is a cartesian product $N \times M$. We write $F_K(N, M)$ for $F(K, N, M)$. Given an adversary with oracle access to $F_K$ for some key $K$, the multiplicity $\mu$ of a nonce $N$ in an attack is the number of times it is used in oracle queries to $F_K$ (e.g., $\mu = 1$ for all nonces for a nonce-respecting adversary).

**Definition 2** (Nonce-Based/Randomized MAC). Let $K$, $N$, $M$, and $T$ be non-empty sets. Let $F : K \times N \times M \to T$ be a nonce-based keyed function. For $K \in K$, let $\text{Ver}_K$ be the verification oracle which takes as input a triple $(N, M, T) \in N \times M \times T$ and returns 1 (“accept”) if $F_K(N, M) = T$, and 0 (“reject”) otherwise.

- $A(\mu, q_m, q_v, t)$-adversary against the nonce-based MAC-security of $F$ is an adversary $A$ with oracle access to the two oracles $F_K$ and $\text{Ver}_K$ for $K \in K$, making at most $q_m$ MAC queries to its first oracle with maximal nonce multiplicity at most $\mu$ and at most $q_v$ verification queries to its second oracle, and running in time at most $t$. We say that $A$ forges if any of its queries to $\text{Ver}_K$ returns 1. The advantage of $A$ against the nonce-based MAC-security of $F$ is defined as
  
  $$\text{Adv}^\text{MAC}_F(A) = \Pr[K \leftarrow K : A^{F_K, \text{Ver}_K} \text{ forges}],$$
  
  where the probability is also taken over the random coins of $A$, if any. The adversary is not allowed to ask a verification query $(N, M, T)$ if a previous query $(N, M)$ to $F_K$ returned $T$. When $\mu = 1$, we say that $A$ is nonce-respecting, otherwise $A$ is said nonce-misusing.

- For $K \in K$, let $F^S_K$ be the probabilistic algorithm which takes as input $M \in M$, internally generates a uniformly random $N \leftarrow \mathcal{N}$, computes $T = F_K(N, M)$, and outputs $(N, T)$. $A(\mu, q_m, q_v, t)$-adversary against the randomized MAC-security of $F$ is an adversary $A$ with oracle access to the two oracles $F^S_K$ and $\text{Ver}_K$ for $K \in K$, making at most $q_m$ MAC queries to its first oracle and at most $q_v$ verification queries to its second oracle, and running in time at most $t$. We say that $A$ forges if any of its queries to $\text{Ver}_K$ returns 1. The advantage of $A$ against the randomized MAC-security of $F$ is defined as
  
  $$\text{Adv}^\text{MAC}_F(A) = \Pr[K \leftarrow K : A^{F^S_K, \text{Ver}_K} \text{ forges}],$$
  
  where the probability is also taken over the random coins of $F^S_K$ and of $A$, if any. The adversary is not allowed to ask a verification query $(N, M, T)$ if a previous query $M$ to $F^S_K$ returned $(N, T)$.

For the three notions above, in case the function $F$ is built from a block cipher and the security proof is done in the ideal cipher model, the advantage additionally depends on the number $q_e$ of ideal cipher queries made by the adversary. The notation is modified in the natural way (e.g., we will talk of a $(\mu, q_e, q_m, q_v, t)$-adversary against the nonce-based MAC security of $F$).

**Almost Uniform and AU Hash Functions.** We will need the following definitions of almost uniform and almost universal (AU) hash functions.

**Definition 3** (Almost Uniform and AU Hash Functions). Let $\varepsilon > 0$, and let $H : K_h \times \mathcal{X} \to \mathcal{Y}$ be a keyed hash function for three non-empty sets $K_h$, $\mathcal{X}$, and $\mathcal{Y}$.

- $H$ is said to be $\varepsilon$-almost uniform if for any $X \in \mathcal{X}$ and any $Y \in \mathcal{Y}$,
  
  $$\Pr[K_h \leftarrow K_h : H_{K_h}(X) = Y] \leq \varepsilon;$$

- $H$ is said to be $\varepsilon$-almost universal if for any $X, Y \in \mathcal{X} \cap \mathcal{Y}$
  
  $$\Pr[K_h \leftarrow K_h : H_{K_h}(X) = H_{K_h}(Y)] \leq \varepsilon.$$
• $H$ is said to be $\varepsilon$-almost universal ($\varepsilon$-AU) if for any distinct $X$ and $X' \in \mathcal{X}$,

$$\Pr [K_h \leftarrow \$ : H_{K_h}(X) = H_{K_h}(X')] \leq \varepsilon.$$  

**Remark 1.** Recall that for an $\varepsilon$-AU hash function with $n$-bit outputs one has $\varepsilon \gtrapprox 2^{-n}$ [Sti96]. (In fact, $\varepsilon$ can be slightly less than $2^{-n}$, but when the domain $\mathcal{X}$ is much larger than the range $\mathcal{Y}$ it can only be negligibly smaller.) In order to simplify our bounds, we will always assume that the $\varepsilon$-AU hash functions used in our constructions are such that $\varepsilon \geq 2^{-n}$.

### 2.2 From Nonce-Based to Randomized MACs

Let $F$ be a nonce-based MAC with nonce space $\mathcal{N}$. In some situations, it can be cumbersome to maintain a state on the MAC generation side to avoid repeating nonces. However, as suggested by Definition 2, any nonce-based MAC can easily be turned into a randomized MAC by letting the MAC generation algorithm choose "nonces" uniformly at random. Of course, these are no longer real nonces since they will start to repeat after roughly $|\mathcal{N}|^{1/2}$ queries. For some schemes (e.g., Wegman-Carter MACs), security might completely collapse as soon as a single nonce is repeated. However, if the original nonce-based scheme is sufficiently resilient to nonce repetition (in particular, if security only degrades linearly with the maximal nonce multiplicity), the resulting randomized scheme will still enjoy good security bounds. This is captured by the following lemma, which holds for MACs provably secure in the standard or ideal cipher model. (Note that for $\mu_0 = 1$, the first term is exactly a birthday term.)

**Lemma 1.** Let $F : \mathcal{K} \times \mathcal{N} \times \mathcal{M} \rightarrow \mathcal{T}$ be a nonce-based keyed function (potentially constructed from some underlying block cipher $E$). Then, for any $((q_e), q_m, q_v, t)$-adversary $A$ against the rMAC security of $F$, and for any integer $\mu_0 \leq q_m$, one has

$$\text{Adv}^{\text{rMAC}}_{F}(A) \leq \frac{(2|\mathcal{N}|)^{\mu_0}}{2^{q_m}} + \max_{\mathcal{A}'} \{ \text{Adv}^{\text{nMAC}}_{F}(\mathcal{A}') \},$$

where the maximum is taken over all $(\mu_0, (q_e), q_m, q_v, t)$-adversaries against the nMAC security of $F$.

**Proof.** Let us fix a $((q_e), q_m, q_v, t)$-adversary $A$ against the rMAC-security of $F$. We make the randomness of the MAC oracle $F_{K}$ explicit through a random function $R : \{1, \ldots, q_m\} \rightarrow \mathcal{N}$. Let $\mathcal{F}(q_m, \mathcal{N})$ denote the set of every such function. For every function $R \in \mathcal{F}(q_m, \mathcal{N})$, let

$$\mu(R) \overset{\text{def}}{=} \max_{i \in \{1, \ldots, q_m\}} |\{j \in \{1, \ldots, q_m\} : R(j) = R(i)\}|$$

be the maximal multiplicity of any element in the image of $R$.

We define a $(\mu(R), (q_e), q_m, q_v, t)$-adversary $A_{R}$ against the nMAC security of $F$ as follows: $A_{R}$ runs $A$, answering its verification queries (and potentially its ideal cipher queries) using its own oracles, and answering $A$'s $i$-th MAC query by querying his own MAC oracle $F_{K}$ using the same message and nonce $R(i)$, for $i = 1, \ldots, q_m$. Then, one has

$$\text{Adv}^{\text{rMAC}}_{F}(A) = \sum_{R' \in \mathcal{F}(q_m, \mathcal{N})} \Pr \left[ R \leftarrow \$ : F(q_m, \mathcal{N}) : R = R' \text{ and } A_{R'}^{(E), F_{K}, \text{Ver} K \text{ forgives}} \right]$$

$$= \sum_{R' \in \mathcal{F}(q_m, \mathcal{N})} \Pr [R = R'] \cdot \Pr \left[ A_{R'}^{(E), F_{K}, \text{Ver} K \text{ forgives}} \right]$$

$$\leq \Pr [\mu(R) \geq \mu_0 + 1] + \frac{1}{|\mathcal{N}|^{q_m}} \sum_{R' \in \mathcal{F}(q_m, \mathcal{N})} \Pr \left[ A_{R'}^{(E), F_{K}, \text{Ver} K \text{ forgives}} \right]$$
The details will be slightly different depending on whether the construction is proven secure in the ideal cipher model or the standard model. We start by describing the formalism for block cipher-based constructions proven secure in the ideal cipher model.

Let \( \text{MAC}[E] \) denote a SD or nonce-based MAC construction based on a block cipher \( E \in \text{Perm}(\kappa, n) \). In all the following, nonces \( N \) and multiplicity \( \mu \) will be written in parenthesis to indicate that they are omitted for a SD-MAC. Let \( \kappa' \) denote the key space for \( \text{MAC}[E] \), and let \( \text{Ver}[E]_{\kappa'} \) be the verification oracle for key \( \kappa' \in \kappa' \). Let \( A \) be a \((\mu, q_e, q_m, q_v, t)\)-adversary against the sd/nMAC security of \( \text{MAC}[E] \) and recall that

\[
\text{Adv}_{\text{MAC}[E]}^{(sd/n)\text{MAC}}(A) = \text{Pr}[E \leftarrow \text{Perm}(\kappa, n), K' \leftarrow \kappa' : A^{E, \text{MAC}[E]_{|K'}, \text{Ver}[E]_{|K'}} \text{ forges}].
\]

It will be more convenient to express this quantity as the advantage of a 
\textit{distinguisher} trying to distinguish the real world \((E, \text{MAC}[E]_{|K'}, \text{Ver}[E]_{|K'})\) and an ideal world defined as follows. Let \( \text{Rand} \) denote a perfectly random oracle with the same domain and range as \( \text{MAC}[E]_{|K'} \), and let \( \text{Rej} \) denote an oracle with the same domain as \( \text{Ver}[E]_{|K'} \) which always returns \( 0 \) (“reject”). Since the adversary cannot have the right oracle return 1 in the ideal world (i.e., when interacting with \((E, \text{Rand}, \text{Rej})\)), we have

\[
\text{Adv}_{\text{MAC}[E]}^{(sd/n)\text{MAC}}(A) = \text{Pr}[A^{E, \text{MAC}[E]_{|K'}, \text{Ver}[E]_{|K'}} \text{ forges}] - \text{Pr}[A^{E, \text{Rand}, \text{Rej}} \text{ forges}].
\]

Consider now an adversary \( D \) which queries a triplet of oracles \((O_1, O_2, O_3)\) and outputs a bit \( \beta \), which we write \( D^{\beta} \). \( O_2, O_3 = \beta \). (We will refer to such an adversary as a 
\textit{distinguisher}.) Say that such an adversary is \textit{non-trivial} if it never makes a query \(((N), M, T)\) to its right (verification) oracle if a previous query \(((N), M)\) to its middle (MAC) oracle returned \( T \). Then

\[
\text{Adv}_{\text{MAC}[E]}^{(sd/n)\text{MAC}}(A) \leq \max_{\beta} \left\{ \text{Pr}[D^{E, \text{MAC}[E]_{|K'}, \text{Ver}[E]_{|K'}} = 1] - \text{Pr}[D^{E, \text{Rand}, \text{Rej}} = 1] \right\},
\]

(1)

where the maximum is taken over all non-trivial distinguishers making \( q_e \) queries to \( O_1 \), \( q_m \) queries to \( O_2 \) (with maximal nonce multiplicity \( \mu \) in the case of a nonce-based MAC), and \( q_v \) queries to \( O_3 \). (This follows easily by considering the particular \( D \) which runs \( A \) and
outputs 1 iff A successfully forges.) This formulation of the problem now allows us to use the H-coefficients technique [Pat08b, CS14], as we explain in more details below.

We assume that D is computationally unbounded (and hence wlog deterministic) and that it never repeats a query. Let

\[ \tau_e = \left( (K_1, X_1, Y_1), \ldots, (K_{q_e}, X_{q_e}, Y_{q_e}) \right) \]

be the list of ideal cipher queries of D and corresponding answers (i.e., for \( 1 \leq i \leq q_e \), D either made a query \( E(K_i, X_i) \) and received answer \( Y_i \), or a query \( E^{-1}(K_i, Y_i) \) and received answer \( X_i \)). Let

\[ \tau_m = \left( ((N_1), M_1, T_1), \ldots, ((N_{q_m}), M_{q_m}, T_{q_m}) \right) \]

be the list of MAC queries of D and corresponding answers. Let also

\[ \tau_v = \left( ((N'_1), M'_1, T'_1, b_1), \ldots, ((N'_{q_v}), M'_{q_v}, T'_{q_v}, b_{q_v}) \right) \]

be the list of verification queries of D and the corresponding answers (with \( b_i \in \{0,1\} \)). The triple \((\tau_e, \tau_m, \tau_v)\) constitutes the queries transcript of the attack. In order to have a simple description of bad transcripts, we slightly modify the security experiment by revealing to the distinguisher (after the interaction but before it outputs its decision bit) the secret key \( K' \) if we are in the real world, or a uniformly random “dummy” key \( K' \) if we are in the ideal world (this is obviously wlog since the distinguisher can ignore this additional piece of information). All in all, the transcript of the attack is the tuple \( \tau = (\tau_e, \tau_m, \tau_v, K') \).

A transcript \( \tau \) is said attainable (with respect to distinguisher D) if the probability to obtain this transcript in the ideal world is non-zero. Let \( \Theta \) denote the set of attainable transcripts. We also let \( X_{\text{rec}}, \text{resp. } X_{\text{id}}, \) denote the probability distribution of the transcript \( \tau \) induced by the real world, resp. the ideal world. Then the main lemma of the H-coefficients technique allows one to upper bound D’s distinguishing advantage as follows (see for example [CS14] or [CLL+14] for the proof).

**Lemma 2** ([Pat08b]). Fix a distinguisher D. Let \( \Theta = \Theta_{\text{good}} \cup \Theta_{\text{bad}} \) be a partition of the set of attainable transcripts. Assume that there exists \( \varepsilon_1 \) such that for any \( \tau \in \Theta_{\text{good}} \), one has\(^3\)

\[
Pr[X_{\text{rec}} = \tau] \geq 1 - \varepsilon_1,
\]

and that there exists \( \varepsilon_2 \) such that \( Pr[X_{\text{id}} \in \Theta_{\text{bad}}] \leq \varepsilon_2 \). Then \( \text{Adv}(D) \leq \varepsilon_1 + \varepsilon_2 \).

Note that for an attainable transcript \( \tau = (\tau_e, \tau_m, \tau_v, K') \), any verification query \( ((N'_i), M'_i, T'_i, b_i) \in \tau_v \) is such that \( b_i = 0 \). Hence, some transcripts are attainable in the real world but not in the ideal world, which is unusual as, in most H-coefficients-based proofs, the set of transcripts attainable in the real world is a subset of those attainable in the ideal world. However, the standard proof of Lemma 2 can be trivially extended to handle this peculiarity (see Appendix A). In order to simplify the notation, in all the following, since we only deal with attainable transcripts, we omit decision bits \( b_i \) from the verification queries transcript and simply write

\[ \tau_v = \left( ((N'_1), M'_1, T'_1), \ldots, ((N'_{q_v}), M'_{q_v}, T'_{q_v}) \right). \]

Since security in the ideal cipher model for block cipher-based constructions hold against computationally unbounded adversaries, we omit parameter \( t \) from theorem statements. For TBC-based constructions proven secure in the standard model, the formalism is identical, except that there is no ideal cipher queries transcript \( \tau_e \) and hence parameter \( q_e \) is irrelevant.

\(^3\)Recall that for an attainable transcript, one has \( Pr[X_{\text{id}} = \tau] > 0 \).
Note that in our proofs we see the forgery game for a MAC construction as a particular case of a distinguishing game against the construction. Then, we lower bound the advantage of any distinguisher against the construction. Hence, when the number \( q_e \) of verification queries is fixed to 0, our security bounds correspond to the PRF security of our constructions.

### 2.4 Permutation (In)equalities List

Fix any non-empty set \( S = \{s_1, \ldots, s_r\} \). At some point in our security proofs, we will need to evaluate the probability that a certain family \( (P_s)_{s \in S} \in \text{Perm}(n)^S \) of uniformly random and independent permutations\(^4\) satisfies some sets of equalities and inequalities. To this end, we introduce the notion of permutation equalities list and permutation inequalities list.

A permutation equalities list is a set \( \lambda_{eq} \) of triples \((s, x, y)\) such that, for any pair of distinct triple \((s, x, y), (s', x', y')\) in \( \lambda_{eq} \), if \( s = s' \) then \( x \neq x' \) and \( y \neq y' \). A permutation inequalities list is a set \( \lambda_{ineq} \) of triples \((s', x', y')\) such that, for any pair of distinct triple \((s, x, y), (s', x', y')\) in \( \lambda_{ineq} \), if \( s = s' \) then \( x \neq x' \) and \( y \neq y' \). A permutation equalities list is a set \( \lambda_{eq} \) of triples \((s, x, y)\) such that, for any pair of distinct triple \((s, x, y), (s', x', y')\) in \( \lambda_{eq} \), if \( s = s' \) then \( x \neq x' \) and \( y \neq y' \). A permutation inequalities list is a set \( \lambda_{ineq} \) of triples \((s', x', y')\) such that, for any pair of distinct triple \((s, x, y), (s', x', y')\) in \( \lambda_{ineq} \), if \( s = s' \) then \( x \neq x' \) and \( y \neq y' \).

Fix any permutation equalities list \( \lambda_{eq} \) and any permutation inequalities list \( \lambda_{ineq} \). A family of permutations \((P_s)_{s \in S}\) of uniformly random and independent permutations\(^4\) is said compatible with \( \lambda_{eq} \) if, for any \( (s, x, y) \in \lambda_{eq} \), one has \( P_s(x) = y \). It is said compatible with \( \lambda_{ineq} \) if, for every \( (s, x, y) \in \lambda_{ineq} \), one has \( P_s(x) \neq y \). Finally, we say that \((P_s)_{s \in S}\) is compatible with \( \lambda = (\lambda_{eq}, \lambda_{ineq}) \) if it is both compatible with \( \lambda_{eq} \) and \( \lambda_{ineq} \). We let \( \text{Comp}(\lambda_{eq}) \) and \( \text{Comp}(\lambda_{ineq}) \) denote the set of families of permutations that are compatible with respectively \( \lambda_{eq} \), \( \lambda_{ineq} \) and \( \lambda \). Then one has the following lemma.

**Lemma 3.** Let \( S = \{s_1, \ldots, s_r\} \), \( \lambda_{eq} \) be a permutation equalities list, and \( \lambda_{ineq} \) be a permutation inequalities list compatible with \( \lambda_{eq} \). Let \( q = |\lambda_{eq}| \) and \( q' = |\lambda_{ineq}| \). Assume that \( q < 2^n \) and \( q' < 2^n \). For \( i = 1, \ldots, r \), let \( q_i \) be the number of \((s, x, y) \in \lambda_{eq} \) such that \( s = s_i \). Then, one has

\[
\Pr \left[ (P_s)_{s \in S} \leftarrow \text{Perm}(n)^S : (P_s) \in \text{Comp}(\lambda) \right] \geq \frac{1}{\prod_{i=1}^r (2^n)_{q_i}} \left( 1 - \frac{q'}{2^n - \max(q_1, \ldots, q_r)} \right).
\]

**Proof.** We are going to consider the permutation equalities list and the permutation inequalities list in turn.

First, we lower bound the probability that a random family \((P_s)_{s \in S}\) of permutations satisfies

\[
\forall (s, x, y) \in \lambda_{eq}, P_s(x) = y.
\]

Since \( \lambda_{eq} \) is a permutation equalities list, each permutation \( P_{s_i} \) must satisfy exactly \( q_i \) equalities. Thus one has

\[
\Pr \left[ (P_s)_{s \in S} \in \text{Comp}(\lambda_{eq}) \right] = \frac{1}{\prod_{i=1}^r (2^n)_{q_i}}.
\] (2)

We will now lower bound the probability that a random family \((P_s)_{s \in S}\) of permutations is compatible with \( \lambda_{ineq} \), conditioned on \((P_s)_{s \in S}\) being compatible with \( \lambda_{eq} \). It will actually be easier to upper bound the probability that \((P_s)_{s \in S}\) is not compatible with \( \lambda_{ineq} \), i.e., that there exists \((s', x', y') \in \lambda_{ineq} \) such that

\[
P_{s'}(x') = y'.
\] (3)

Fix any permutation inequality \((s', x', y') \in \lambda_{ineq} \). We consider two possible cases:

\(^4\)For block cipher-based constructions, \((P_s)_{s \in S}\) will be a block cipher, and for TBC-based constructions, \((P_s)_{s \in S}\) will be a tweakable permutation.
1. if there exists \((s, x, y) \in \lambda_{\text{eq}}\) such that \(s = s'\) and \(x = x'\) or \(y = y'\), then, since by definition of \(\lambda_{\text{ineq}}\) being compatible with \(\lambda_{\text{eq}}\) one has \((s, x, y) \neq (s', x', y')\), Equation (3) cannot hold;

2. otherwise, Equation (3) holds with probability \(1/(2^n - m)\), where \(m\) is the number of times \(s'\) appears in \(\lambda_{\text{eq}}\), which cannot be larger than \(\max\{q_1, \ldots, q_r\}\).

Hence, we see that for any \((s', x', y') \in \lambda_{\text{ineq}}\), Equation (3) is satisfied with probability at most \(1/(2^n - \max\{q_1, \ldots, q_r\})\). By a union bound over the \(q'\) permutation inequalities, we obtain that

\[
\Pr [(P_s)_{s \in S} \in \Comp(\lambda_{\text{ineq}}) | (P_s)_{s \in S} \in \Comp(\lambda_{\text{eq}})] \geq 1 - \frac{q'}{2^n - \max\{q_1, \ldots, q_r\}}. \tag{4}
\]

Using Equation (2) and Equation (4), we have

\[
\Pr [(P_s)_{s \in S} \in \Comp(\lambda)] \geq \frac{1}{\prod_{i=1}^{r}(2^n)_{q_i}} \left(1 - \frac{q'}{2^n - \max\{q_1, \ldots, q_r\}}\right). \tag{\text{\square}}
\]

### 3 Tweakable Block Cipher-Based Constructions

In this section, we describe and analyze two TBC-based MAC constructions: a nonce-based one called \(\text{NaT}\) and a stateless deterministic one called \(\text{HaT}\). The security proofs are done in the standard model (i.e., they do not require to idealize the underlying TBC).

#### 3.1 The Nonce-Based Construction NaT

We start with a nonce-based construction named \(\text{NaT}\) (Nonce-as-Tweak). Given a TBC \(\tilde{E}\) with key space \(K\), tweak space \(W\), and message space \(\{0, 1\}^n\) and a keyed hash function \(H\) with key space \(K_h\), domain \(M\), and range \(\{0, 1\}^n\), we define a MAC with key space \(K \times K_h\), nonce space \(W\), and message space \(M\) as

\[
\text{NaT}[\tilde{E}, H]_{K,K_h}(N, M) = \tilde{E}^N_{K_h}(H_{K_h}(M)).
\]

Our security result is the following one.

**Theorem 1.** Let \(M, K, W\) and \(K_h\) be non-empty sets. Let \(\tilde{E} : K \times W \times \{0, 1\}^n \to \{0, 1\}^n\) be a tweakable block cipher and \(H : K_h \times M \to \{0, 1\}^n\) be an \(\varepsilon\)-AU hash function. Let \(\mu, q_m, q_v\), and \(t\) be integers such that \(q_m, q_v \leq 2^n\) and \(\mu \leq \min\{q_m, 2^n - 1\}\). Then for any \((\mu, q_m, q_v, t)\)-adversary \(A\) against the nMAC-security of \(\text{NaT}[\tilde{E}, H]\), there exists a \((q_m + q_v, t')\)-adversary \(A'\) against the TPRP-security of \(\tilde{E}\), where \(t' = O(t + (q_m + q_v)t_H)\) and \(t_H\) is an upper bound on the time to compute \(H\) on any message, such that

\[
\text{Adv}_{\text{NaT}[\tilde{E}, H]}^{\text{MAC}}(A) \leq \text{Adv}_{\tilde{E}}^{\text{TPRP}}(A') + 2(\mu - 1)q_m \varepsilon + \frac{q_v}{2^n - \mu} + \mu q_v \varepsilon.
\]

Recall that for an \(\varepsilon\)-AU hash function with \(n\)-bit outputs one has \(\varepsilon \geq 2^{-n}\) [Sti96] (see Remark 1). Hence, in the nonce-respecting case (i.e., \(\mu = 1\)), the \(\text{NaT}\) construction is secure up to roughly \(\varepsilon^{-1}\) verification queries, irrespectively of the number of MAC queries (neglecting the effect of the TPRP-advantage term). When \(A\) can freely choose nonces (i.e., \(\mu = q_m\)), then the \(\text{NaT}\) construction is secure up to the birthday bound. The security bound degrades linearly with the maximal multiplicity \(\mu\) of nonces.

As a corollary, we obtain the following for the security of the \(\text{NaT}\) construction as a randomized MAC, which shows that it is secure up to roughly \(\varepsilon^{-1}/n\) MAC and verification queries under the additional assumption that \(|W| \geq 2^n\).
Corollary 1. Let $\mathcal{M}$, $\mathcal{K}$, $\mathcal{W}$ and $\mathcal{K}_h$ be non-empty sets. Let $E: \mathcal{K} \times \mathcal{W} \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a tweakable block cipher and $H: \mathcal{K}_h \times \mathcal{M} \rightarrow \{0,1\}^n$ be an $\varepsilon$-AU hash function. Let $q_m$, $q_v$, and $t$ be integers such that $q_m, q_v \leq 2^n$. Then, for any $(q_m, q_v, t)$-adversary $A$ against the rMAC-security of $\text{NaT}[E, H]$, there exists a $(q_m + q_v, t')$-adversary $A'$ against the TPRP-security of $\tilde{E}$, where $t' = O(t + (q_m + q_v)t_H)$ and $t_H$ is an upper bound on the time to compute $H$ on any message, such that

$$\text{Adv}_{\text{NaT}[E, H]}^{\text{MAC}}(A) \leq \text{Adv}_{E}^{\text{TPRP}}(A') + 2(n-1)q_m\varepsilon + \frac{q_v}{2^n-n} + nq_v\varepsilon + \frac{q_{m+1} n^2}{(2|\mathcal{W}|)^n}.$$  

Proof. This follows by combining Lemma 1 with $\mu_0 = n$ and Theorem 1.  

The remaining of this section is devoted to the proof of Theorem 1. Let us fix a $(\mu, q_m, q_v, t)$-adversary $A$ against the $\text{MAC}$-security of $\text{NaT}[E, H]$. The first step of the proof is standard and consists in replacing $\tilde{E}_K$ by a uniformly random tweakable permutation $\tilde{P}$, both in the MAC and in the verification oracles (in other words, we replace the tweakable block cipher $\tilde{E}$ by the perfect tweakable block cipher $E^*$ whose key space is the set of all tweakable permutations of $\{0,1\}^n$ with tweak space $\mathcal{W}$). Let $\text{NaT}[E^*, H]$ denote the resulting construction. It is easy to show that there exists an adversary $A'$ against the TPRP-security of $\tilde{E}$, making at most $q_m + q_v$ oracle queries and running in time at most $O(t + (q_m + q_v)t_H)$, such that

$$\text{Adv}_{\text{NaT}[E^*, H]}^{\text{MAC}}(A) \leq \text{Adv}_{E}^{\text{TPRP}}(A') + \text{Adv}_{\text{NaT}[E^*, H]}^{\text{MAC}}(A).$$  

(5)

The next step is to find an upper bound for

$$\text{Adv}_{\text{NaT}[E^*, H]}^{\text{MAC}}(A) = \text{Pr} \left[ \tilde{P} \leftarrow \text{Rand}(\mathcal{W}, n), K_h \leftarrow \text{Rand}(\mathcal{K}_h), A^{\text{NaT}[\tilde{P}, H], K_h, \text{Ver} \tilde{P}, H, K_h} \text{ forges} \right] ,$$

where, overloading the notation, $\text{NaT}[\tilde{P}, H, K_h]$ denotes construction $\text{NaT}[\tilde{E}^*, H]$ instantiated with tweakable permutation $\tilde{P}$ and hashing key $K_h$ and $\text{Ver}[\tilde{P}, H, K_h]$ denotes the corresponding verification oracle. This is now a purely information-theoretic problem, and we can follow the H-coefficients technique as explained in Section 2.3.

Let us fix a non-trivial $(\mu, q_m, q_v)$- distinguisher $D$ interacting either with the real world $(\text{NaT}[\tilde{P}, H], \text{Ver}[\tilde{P}, H])$ or with the ideal world $(\text{Rand}, \text{Rej})$. We let

$$\text{Adv}(D) = \text{Pr} \left[ D^{\text{NaT}[\tilde{P}, H], K_h, \text{Ver}[\tilde{P}, H, K_h]} = 1 \right] - \text{Pr} \left[ D^{\text{Rand}, \text{Rej}} = 1 \right].$$

Let $\tau = (\tau_m, \tau_v, K_h)$ denote the transcript of the attack, with

$$\tau_m = ((N_1, M_1, T_1), \ldots, (N_{q_m}, M_{q_m}, T_{q_m}))$$

and

$$\tau_v = ((N'_1, M'_1, T'_1), \ldots, (N'_{q_v}, M'_{q_v}, T'_{q_v})).$$

Recall that $\Theta$ denotes the set of attainable transcripts and $X_{\text{re}}$, resp. $X_{\text{id}}$, the probability distribution of the transcript $\tau$ induced by the real world, resp. the ideal world.

The remaining of the proof of Theorem 1 is structured as follows. First, we define bad transcripts and upper bound their probability in the ideal world (Lemma 4). Then, we analyze good transcripts and prove that they are almost as likely in the real and the ideal world (Lemma 5). Combining Lemma 2 from Section 2.3 with Lemma 4 and Lemma 5 gives us an upper bound on $D$’s advantage, which by Equation (1) from Section 2.3 gives us an upper bound on

$$\text{Adv}_{\text{NaT}[E^*, H]}^{\text{MAC}}(A).$$

Theorem 1 follows easily by combining this upper bound with Equation (5).

We start by defining bad transcripts.
Definition 4. We say that an attainable transcript $\tau = (\tau_m, \tau_v, K_h)$ is bad if one of these conditions is fulfilled:

(C-1) there exists two distinct MAC queries $(N_i, M_i, T_i)$ and $(N_j, M_j, T_j)$ such that $N_i = N_j$ and either $H_{K_h}(M_i) = H_{K_h}(M_j)$ or $T_i = T_j$;

(C-2) there exists a MAC query $(N_i, M_i, T_i) \in \tau_m$ and a verification query $(N'_j, M'_j, T'_j) \in \tau_v$ such that

$$\begin{cases} N_i = N'_j \\ H_{K_h}(M_i) = H_{K_h}(M'_j) \\ T_i = T'_j. \end{cases}$$

We let $\Theta_{\text{bad}}$, resp. $\Theta_{\text{good}}$ denote the set of bad, respectively good transcripts.

Note that the second condition can only happen in the ideal world since in the real world, if $N_i = N'_j$, $H_{K_h}(M_i) = H_{K_h}(M'_j)$, and $T_i = T'_j$, the verification oracle should return 1 on query $(N'_j, M'_j, T'_j)$ (which is impossible for an attainable transcript).

Lemma 4. For any integers $q_m$ and $q_v$, one has

$$\Pr[X_{id} \in \Theta_{\text{bad}}] \leq 2(\mu - 1)q_m\varepsilon + \mu q_v\varepsilon.$$  

Proof. We let $\Theta_1$ denote the set of attainable transcripts satisfying condition (C-1). Recall that, in the ideal world, $K_h$ is drawn independently from the queries transcript. We are going to consider both conditions in turn.

Condition (C-1). Fix a MAC query $(N_i, M_i, T_i)$. There are exactly $q_m$ possible choices for this query. Then we fix another MAC query $(N_j, M_j, T_j)$ such that $N_i = N_j$ (there are at most $\mu - 1$ possible choices). The probability, over the random draw of $T_i$ and $T_j$ that $T_i = T_j$ is $2^{-n}$, and the probability, over the random draw of $K_h$, that $H_{K_h}(M_i) = H_{K_h}(M_j)$, is lower than $\varepsilon$. Summing over every possible choice of $(N_i, M_i, T_i)$ and $(N_j, M_j, T_j)$, we get

$$\Pr[X_{id} \in \Theta_1] \leq \frac{(\mu - 1)q_m}{2^n} + (\mu - 1)q_m\varepsilon \leq 2(\mu - 1)q_m\varepsilon,$$

where we used that $\varepsilon \geq 2^{-n}$ (see Remark 1).

Condition (C-2). We consider any verification query $(N'_j, M'_j, T'_j) \in \tau_v$ and upper bound the probability that this condition is satisfied for this particular query. By definition of the multiplicity, there are at most $\mu$ MAC queries $(N_i, M_i, T_i)$ such that $N_i = N'_j$. Fix any of these queries. We distinguish two cases:

- If the verification query comes after the MAC query, then since the distinguisher is non-trivial, either $T_i \neq T'_j$, or $M_i \neq M'_j$. In the former case, the condition cannot be satisfied, while in the latter case, the probability over the random draw of $K_h$ that $H_{K_h}(M_i) = H_{K_h}(M'_j)$ is at most $\varepsilon$.

- If the MAC query comes after the verification query, then $T_i$ is random and independent from $T'_j$ and the probability that $T_i = T'_j$ is $2^{-n}$.

Since $\varepsilon \geq 2^{-n}$ (see Remark 1), we see that in all cases the condition is met with probability at most $\varepsilon$. Thus, by summing over every verification query, and every MAC query using the same nonce as the verification query, one has

$$\Pr[X_{id} \in \Theta_2] \leq \mu q_v\varepsilon.$$

The result follows by a union bound over these conditions.
We now analyze good transcripts and prove the following lemma.

**Lemma 5.** For any good transcript $\tau$, one has

$$\frac{\Pr[X_{re} = \tau]}{\Pr[X_{id} = \tau]} \geq 1 - \frac{q_v}{2^n - \mu}.$$  

**Proof.** Let $\tau = (\tau_m, \tau_v, K_h)$ be a good transcript. Let $\mathcal{L} = \{N_1, \ldots, N_{q_m}\}$ be the set of all nonces used in MAC queries. Using any arbitrary order, we rewrite the set $\mathcal{L}$ as

$$\mathcal{L} = \{L_1, \ldots, L_r\},$$

where $r$ is the total number of distinct values in $\mathcal{L}$. For $i = 1, \ldots, r$, we let $q_i$ denote the multiplicity of nonce $N_i$ in $\tau_m$. Note that $q_i \leq \mu$ for $i = 1, \ldots, r$.

Since in the ideal world the MAC oracle is perfectly random and the verification always rejects, one simply has

$$\Pr[X_{id} = \tau] = \frac{1}{|K_h| \cdot (2^n)^{q_m}}.$$  

We must now lower bound the probability of getting $\tau$ in the real world. We say that a tweakable permutation $\tilde{P}$ is compatible with $\tau_m$ if

$$\forall i \in \{1, \ldots, q_m\}, \text{NaT}[\tilde{P}, H]_{K_h}(N_i, M_i) = T_i,$$

and compatible with $\tau_v$ if

$$\forall i \in \{1, \ldots, q_v\}, \text{NaT}[\tilde{P}, H]_{K_h}(N'_i, M'_i) \neq T'_i.$$  

We simply say that $\tilde{P}$ is compatible with $\tau$ if it is compatible with $\tau_m$ and $\tau_v$. We let $\text{Comp}(\tau_m)$, $\text{Comp}(\tau_v)$, and $\text{Comp}(\tau)$ denote the set of tweakable permutations that are compatible with respectively $\tau_m$, $\tau_v$, and $\tau$. Then one can easily check (see for example [CS14] for a detailed explanation) that

$$\Pr[X_{re} = \tau] = \frac{1}{|K_h|} \cdot \Pr[\tilde{P} \leftarrow \text{Perm}(W, n) : \tilde{P} \in \text{Comp}(\tau)].$$  

We now define

$$\lambda_{eq} = \{(N_1, H_{K_h}(M_1), T_1), \ldots, (N_{q_m}, H_{K_h}(M_{q_m}), T_{q_m})\}$$

and

$$\lambda_{ineq} = \{(N'_1, H_{K_h}(M'_1), T'_1), \ldots, (N'_{q_v}, H_{K_h}(M'_{q_v}), T'_{q_v})\}.$$  

Then, since $\tau$ is a good transcript, $\lambda_{eq}$ is a permutation equalities list\(^5\) (otherwise condition (C-1) defining bad transcripts would be met), and $\lambda_{ineq}$ is a permutation inequalities list which is compatible with $\lambda_{eq}$ (otherwise condition (C-2) would be met). Moreover, $|\lambda_{eq}| = q_m$ and $|\lambda_{ineq}| = q_v$. Note that the event $\tilde{P} \in \text{Comp}(\tau)$ is actually equivalent to the event $\tilde{P} \in \text{Comp}(\lambda)$ where $\lambda = (\lambda_{eq}, \lambda_{ineq})$. Using Lemma 3, one has

$$\Pr[\tilde{P} \in \text{Comp}(\tau)] \geq \frac{1}{\prod_{i=1}^{r} (2^n)^{q_i}} \left(1 - \frac{q_v}{2^n - \mu}\right).$$

Combining this equation with Equation (6) and Equation (7), and using the fact that $q_m = \sum_{i=1}^{r} q_i$, we get

$$\frac{\Pr[X_{re} = \tau]}{\Pr[X_{id} = \tau]} \geq \left(1 - \frac{q_v}{2^n - \mu}\right) \cdot \prod_{i=1}^{r} \frac{(2^n)^{q_i}}{(2^n)^{q_i}} \geq 1 - \frac{q_v}{2^n - \mu}.$$  

\(^5\)Refer to Section 2.4 for the definition of permutation (in)equalities lists.
3.2 The Stateless Deterministic Construction HaT

Our second TBC-based construction is a stateless deterministic MAC called HaT (Hash-as-Tweak). Given a TBC $\bar{E} : K \times W \rightarrow \{0,1\}^n$ and two keyed hash functions $H : K_h \times M \rightarrow \{0,1\}^n$ and $H' : K'_h \times M \rightarrow W$, we define a MAC with key space $K \times K_h \times K'_h$ and message space $M$ as

$$ HaT[\bar{E}, H, H']_{K,K_h,K'_h}(M) = \bar{E}^{H'_{K_h}}_{K_h}((H_{K_h}(M)). $$

Then the following result holds.

**Theorem 2.** Let $\mathcal{M}, \mathcal{W}, K, K_h$, and $K'_h$ be non-empty sets. Let $\bar{E} : K \times W \rightarrow \{0,1\}^n$ be a tweakable block cipher and let $H : K_h \times M \rightarrow \{0,1\}^n$ and $H' : K'_h \times M \rightarrow W$ be two $\varepsilon$-AU hash functions. Let $q_m$, $q_v$, and $t$ be integers such that $q_m < 2^n$. Then for any $(q_m, q_v, t)$-adversary $A$ against the sdMAC-security of $HaT[\bar{E}, H, H']$, there exists a $(q_m + q_v, t')$-adversary $A'$ against the TPRP-security of $\bar{E}$, where $t' = O(t + (q_m + q_v)t_H)$ and $t_H$ is an upper bound on the time to compute $H$ or $H'$ on any message, such that

$$ \operatorname{Adv}_{\text{sdMAC}}^{\text{sdMAC}}_{\text{HaT}[\bar{E}, H, H']}(A) \leq \operatorname{Adv}_{\text{TPRP}}^{\text{TPRP}}_{\bar{E}}(A') + q_m^2\varepsilon^2 + q_mq_v\varepsilon^2 + \frac{q_v + q_m}{2^n}.$$

Hence, the HaT construction is secure up to roughly $q_m \simeq \varepsilon^{-1}$ MAC queries and $q_v \simeq \min\{2^n, \varepsilon^{-2}/q_m\}$ verification queries.

The remaining of this section is devoted to the proof of Theorem 2. Let us fix a $(q_m, q_v, t)$-adversary $A$ against the MAC-security of $HaT[\bar{E}, H, H']$.

The first step of the proof is standard and consists in replacing $\bar{E}_K$ by a tweakable permutation $\tilde{P}$ both in the MAC and in the verification oracles (in other words, we replace the tweakable block cipher $\bar{E}$ by the perfect tweakable block cipher $\bar{E}^*$ whose key space is the set of all tweakable permutations of $\{0,1\}^n$ with tweak space $N$). Let $HaT[\bar{E}^*, H, H']$ denote the resulting construction. It is easy to show that there exists an adversary $A'$ against the TPRP-security of $\bar{E}$, making at most $q_m + q_v$ oracle queries and running in time at most $O(t + (q_m + q_v)t_H)$, such that

$$ \operatorname{Adv}_{\text{sdMAC}}^{\text{sdMAC}}_{\text{HaT}[\bar{E}, H, H']}(A) \leq \operatorname{Adv}_{\text{TPRP}}^{\text{TPRP}}_{\bar{E}}(A') + \operatorname{Adv}_{\text{sdMAC}}^{\text{sdMAC}}_{\text{HaT}[\bar{E}^*, H, H']}(A). $$

The next step is to find an upper bound for

$$ \operatorname{Adv}_{\text{sdMAC}}^{\text{sdMAC}}_{\text{HaT}[\bar{E}, H, H']}(A) = \Pr \left[ \tilde{P} \leftarrow \Permute(N, n), (K_h, K'_h) \leftarrow \mathcal{R}K_h \times K'_h : 
\begin{align*}
A^H_{\text{HaT}[\tilde{P}, H, H']_{K_h,K'_h}} &\text{Ver}[\tilde{P}, H, H']_{K_h,K'_h} \text{ forges,}
A^H_{\text{HaT}[\bar{E}, H, H']_{K_h,K'_h}} &\text{Ver}[\bar{E}, H, H']_{K_h,K'_h} \text{ forges,}
\end{align*}
\right] = 1 \right] \Pr \left[ D_{\text{Rand,Rej}} = 1 \right]. $$

where, slightly overloading the notation, $HaT[\tilde{P}, H, H']_{K_h,K'_h}$ denotes the construction $HaT[\bar{E}^*, H, H']$ instantiated with tweakable permutation $\tilde{P}$ and hashing keys $(K_h, K'_h)$ and $\text{Ver}[\tilde{P}, H, H']_{K_h,K'_h}$ denotes the corresponding verification oracle. This is now a purely information-theoretic problem, and we can follow the H-coefficients technique as explained in Section 2.3.

Let us fix a non-trivial $(q_m, q_v)$-distinguisher $D$ interacting either with the real world $(\text{HaT}[\tilde{P}, H, H']_{K_h,K'_h}, \text{Ver}[\tilde{P}, H, H']_{K_h,K'_h})$ or with the ideal world $(\text{Rand, Rej})$. We let

$$ \operatorname{Adv}(D) = \Pr \left[ D_{\text{HaT}[\tilde{P}, H, H']_{K_h,K'_h}}^{\text{Ver}[\tilde{P}, H, H']_{K_h,K'_h}} = 1 \right] = \Pr \left[ D_{\text{Rand,Rej}} = 1 \right]. $$

Let $\tau = (\tau_m, \tau_v, K_h, K'_h)$ denote the transcript of the attack, with

$$ \tau_m = ((M_1, T_1), \ldots, (M_{q_m}, T_{q_m})) $$
$$ \tau_v = ((M'_1, T'_1), \ldots, (M'_{q_v}, T'_{q_v})). $$
Recall that $\Theta$ denotes the set of attainable transcripts and $X_{\text{re}}$, resp. $X_{\text{id}}$, the probability distribution of the transcript $\tau$ induced by the real world, resp. the ideal world.

The remaining of the proof of Theorem 2 is structured as follows. First, we define bad transcripts and upper bound their probability in the ideal world (Lemma 6). Then, we analyze good transcripts and prove that they are almost as likely in the real and the ideal world (Lemma 7). Combining Lemma 2 from Section 2.3 with Lemma 6 and Lemma 7 gives us an upper bound on $D$’s advantage, which by Equation (1) from Section 2.3 gives us an upper bound on

$$\text{Adv}_{\text{sdMAC}_{\text{H} \text{aT}[E^*, H', H]}}(A).$$

Theorem 2 follows easily by combining this upper bound with Equation (8).

We start by defining bad transcripts.

**Definition 5.** We say that an attainable transcript $\tau = (\tau_m, \tau_v, K_h, K_h')$ is bad if one of these conditions is fulfilled:

(C-1) there exists two distinct MAC queries $(M_i, T_i)$ and $(M_j, T_j)$ such that $H'_{K_h'}(M_i) = H'_{K_h}(M_j)$ and either $H_{K_h}(M_i) = H_{K_h}(M_j)$ or $T_i = T_j$;

(C-2) there exist a MAC query $(M_i, T_i) \in \tau_m$ and a verification query $(M_j', T_j') \in \tau_v$ such that

$$\begin{cases} 
    H'_{K_h'}(M_i) = H'_{K_h}(M'_j) \\
    H_{K_h}(M_i) = H_{K_h}(M'_j) \\
    T_i = T_j. 
\end{cases}$$

We let $\Theta_{\text{bad}}$, resp. $\Theta_{\text{good}}$ denote the set of bad, respectively good transcripts.

We now upper bound the probability to get a bad transcript in the ideal world.

**Lemma 6.** For any integers $q_m$ and $q_v$, one has

$$\Pr[X_{\text{id}} \in \Theta_{\text{bad}}] \leq q_m^2 \varepsilon^2 + q_m q_v \varepsilon^2.$$

**Proof.** Let $\Theta_i$ denote the set of attainable transcripts satisfying condition (C-i). Recall that, in the ideal world, $(K_h, K_h')$ is drawn independently from the queries transcript. We are going to consider both conditions in turn.

**Condition (C-1).** Fix two distinct MAC queries $(M_i, T_i)$ and $(M_j, T_j)$. Then the probability that $H'_{K_h'}(M_i) = H'_{K_h}(M_j)$ (over the draw of $K_h'$) is at most $\varepsilon$, the probability that $H_{K_h}(M_i) = H_{K_h}(M_j)$ (over the draw of $K_h$) is at most $\varepsilon$, and the probability that $T_i = T_j$ is at most $2^{-n}$. Summing over all pairs of distinct MAC queries,

$$\Pr[X_{\text{id}} \in \Theta_i] \leq \frac{q_m^2 \varepsilon}{2 n} + \frac{q_m q_v \varepsilon^2}{2} \leq q_m^2 \varepsilon^2,$$

where we used that $\varepsilon \geq 2^{-n}$ (see Remark 1).

**Condition (C-2).** In order to upper bound the probability of obtaining bad transcripts satisfying condition (C-2) in the ideal world, fix a MAC query $(M_i, T_i) \in \tau_m$ and a verification query $(M_j', T_j') \in \tau_v$. Since $K_h'$ is drawn independently from the queries transcript and $H'$ is $\varepsilon$-AU, the probability that $H'_{K_h'}(M_i) = H'_{K_h'}(M_j')$ is upper bounded by $\varepsilon$. We now distinguish two cases:

- If the verification query comes after the MAC query, then since the distinguisher is non-trivial, either $T_i \neq T_j'$, or $M_i \neq M_j'$. In the former case, the condition cannot be satisfied, while in the latter case, the probability over the random draw of $K_h$ that $H_{K_h}(M_i) = H_{K_h}(M_j')$ is at most $\varepsilon$. 

---

42 New Constructions of MACs from (Tweakable) Block Ciphers
• If the MAC query comes after the verification query, then \( T_i \) is random and independent from \( T_j \) and the probability that \( T_i = T_j \) is \( 2^{-n} \).

Since \( \varepsilon \geq 2^{-n} \) (see Remark 1), we see that in all cases the condition is met with probability at most \( \varepsilon^2 \). Thus, by summing over every verification query, and every MAC query using the same nonce as the verification query, one has

\[
\Pr[X_{id} \in \Theta_2] \leq q_m q_v \varepsilon^2.
\]

The result follows by a union bound over the two conditions.

We now analyze good transcripts and prove the following lemma.

**Lemma 7.** For any good transcript \( \tau \), one has

\[
\frac{\Pr[X_{id} = \tau]}{\Pr[X_{id}]} \geq 1 - \frac{q_v}{2^n - q_m}.
\]

**Proof.** Let \( \tau = (\tau_m, \tau_v, K_h, K_h') \) be a good transcript. Let \( \mathcal{L} = \{H_{K_h'}(M_1), \ldots, H_{K_h'}(M_{q_m})\} \) be the set of all the tweaks used in the MAC queries. Using an arbitrary order, we rewrite the set \( \mathcal{L} \) as

\[
\mathcal{L} = \{L_1, \ldots, L_r\},
\]

where \( r \) is the total number of distinct values of \( \mathcal{L} \). For \( i = 1, \ldots, r \), we let \( q_i \) denote the number of MAC queries \((M, T)\) in \( \tau_m \) such that \( H_{K_h'}(M) = L_i \).

Since in the ideal world the MAC oracle is perfectly random and the verification always rejects, one simply has

\[
\Pr[X_{id} = \tau] = \frac{1}{|K_h| \cdot |K_h'| \cdot (2^n)^{q_m}}.
\]  

(9)

We must now lower bound the probability of getting \( \tau \) in the real world. We say that a tweakable permutation \( \tilde{P} \) is compatible with \( \tau_m \) if

\[
\forall i \in \{1, \ldots, q_m\}, \quad \tilde{P}(H_{K_h'}(M_i), H_{K_h}(M_i)) = T_i,
\]

and compatible with \( \tau_v \) if

\[
\forall i \in \{1, \ldots, q_v\}, \quad \tilde{P}(H_{K_h'}(M'_i), H_{K_h}(M'_i)) \neq T'_i.
\]

We simply say that \( \tilde{P} \) is compatible with \( \tau \) if it is compatible with \( \tau_m \) and \( \tau_v \). We let \( \text{Comp}(\tau_m) \), \( \text{Comp}(\tau_v) \), and \( \text{Comp}(\tau) \) denote the set of tweakable permutations that are compatible with respectively \( \tau_m \), \( \tau_v \), and \( \tau \). Then one can easily check (see for example [CS14] for a detailed explanation) that

\[
\Pr[X_{re} = \tau] = \frac{1}{|K_h| \cdot |K_h'|} \cdot \Pr[\tilde{P} \leftarrow \text{Perm}(\mathcal{W}, n) : \tilde{P} \in \text{Comp}(\tau)].
\]  

(10)

We now define

\[
\lambda_{eq} = \{(H_{K_h'}(M_1), H_{K_h}(M_1), T_1), \ldots, (H_{K_h'}(M_{q_m}), H_{K_h}(M_{q_m}), T_{q_m})\}
\]

\[
\lambda_{ineq} = \{(H_{K_h'}(M'_1), H_{K_h}(M'_1), T'_1), \ldots, (H_{K_h'}(M'_{q_v}), H_{K_h}(M'_{q_v}), T'_{q_v})\}.
\]

Then, since \( \tau \) is a good transcript, \( \lambda_{eq} \) is a permutation equalities list (otherwise condition (C-1) defining bad transcripts would be met), and \( \lambda_{ineq} \) is a permutation inequalities list which is compatible with \( \lambda_{eq} \) (otherwise condition (C-2) would be met). Moreover,
\[ |\lambda_{eq}| = q_m \text{ and } |\lambda_{ineq}| = q_v. \] Note that the event \( \tilde{P} \in \text{Comp}(\tau) \) is actually equivalent to the event \( \tilde{P} \in \text{Comp}(\lambda) \) where \( \lambda = (\lambda_{eq}, \lambda_{ineq}) \). Using Lemma 3, one has
\[ \Pr \left[ \tilde{P} \in \text{Comp}(\tau) \right] \geq \frac{1}{\prod_{i=1}^{r} (2^n)q_i} \left( 1 - \frac{q_v}{2^n - q_m} \right). \]
Combining this equation with Equation (9) and Equation (10), and using the fact that \( q_m = \sum_{i=1}^{r} q_i \), we get
\[ \frac{\Pr [X_{re} = \tau]}{\Pr [X_{id} = \tau]} \geq \left( 1 - \frac{q_v}{2^n - q_m} \right) \prod_{i=1}^{r} \frac{(2^n)q_i}{(2^n)q_i} \geq 1 - \frac{q_v}{2^n - q_m}. \] \hfill \square

## 4 Block Cipher-Based Constructions

The two constructions NaT and HaT of Section 3 follow the traditional UHF-then-PRF paradigm: first, the message is hashed with a UHF, and then a keyed transformation based on a TBC is applied. In this section, we give variants of these constructions where the final transformation uses a keyless transformation based on a block cipher. As a result, the MAC key of these construction only consists of the key(s) of the underlying universal hash function(s). The keyless final mapping must obviously be pre-image resistant (otherwise the adversary can recover the output of the UHF from the tag, which might reveal the hashing key, e.g., for polynomial-based universal hashing). For the nonce-based construction NaK, since the nonce is known from the adversary, this implies that we need to use the block cipher in Davies-Meyer mode. Technically, we also need to require that the universal hash functions be almost uniform. The security proofs for these two variants are done in the ideal cipher model.

As a warm-up, the reader might want to check that the construction
\[ F_{K_h}(M) = G(H_{K_h}(M)), \]
where \( H : K_h \times M \rightarrow \{0,1\}^n \) is \( \varepsilon \)-AU and \( \varepsilon' \)-almost uniform and \( G : \{0,1\}^m \rightarrow \{0,1\}^n \) is some fixed function, is a PRF provably secure in the Random Oracle model for \( G \), as proved in Appendix B.

### 4.1 The Nonce-Based Construction NaK

We start with the block-cipher-based variant of NaT named NaK (Nonce-as-Key). Given a block cipher \( E \) with key space \( K \) and message space \( \{0,1\}^n \) and a keyed hash function \( H \) with key space \( K_h \), domain \( M \), and range \( \{0,1\}^n \), we define a MAC with key space \( K_h \), nonce space \( K \), and message space \( M \) as
\[ \text{NaK}[E,H](N,M) = E_N(H_{K_h}(M)) \oplus H_{K_h}(M). \]

Our security result is the following one.

**Theorem 3.** Let \( \mathcal{M} \), \( K \), and \( K_h \) be non-empty sets. Let \( E : K \times \{0,1\}^n \rightarrow \{0,1\}^n \) be a block cipher and \( H : K_h \times \mathcal{M} \rightarrow \{0,1\}^n \) be an \( \varepsilon \)-AU and \( \varepsilon' \)-almost uniform hash function. Let \( \mu, q_e, q_m, \text{and} q_v \) be integers such that \( q_e, q_m, q_v \leq 2^n \), and \( \mu \leq \min\{q_m, 2^n - 1 - q_e\} \). Then, in the ideal cipher model for \( E \), for any \((\mu, q_e, q_m, q_v)\)-adversary \( \mathcal{A} \) against the nMAC-security of \( \text{NaK}[E,H] \), one has
\[ \text{Adv}_{\text{NaK}[E,H]}^{\text{nMAC}}(\mathcal{A}) \leq 2(\mu - 1)q_m\varepsilon + \frac{q_v}{2^n - \mu - q_e} + \mu q_e \varepsilon + nq_e \varepsilon' + \left( \frac{q_e}{2^n - q_e + 1} \right)^{n+1} + 2\mu q_e \varepsilon'. \]
Proof. Deferred to Appendix C. □

Hence, in the nonce-respecting case (µ = 1), the NaK construction is secure up to \(q_e \leq \min\{\varepsilon^{-1}, (\varepsilon')^{-1}/n\}\) verification queries and \(q_e \leq (\varepsilon')^{-1}\) ideal cipher queries, irrespectively of the number of MAC queries. When A can freely choose nonces (i.e., \(\mu = q_m\)), then the NaK construction is secure up to the birthday bound. The security bound degrades linearly with the maximal multiplicity \(\mu\) of nonces.

As a corollary, we obtain the following for the security of NaK as a randomized MAC.

**Corollary 2.** Let \(M, K,\) and \(K_h\) be non-empty sets. Let \(E : K \times \{0,1\}^n \rightarrow \{0,1\}^n\) be a block cipher and \(H : K_h \times M \rightarrow \{0,1\}^n\) be an \(\varepsilon\)-almost uniform hash function. Let \(q_e, q_m,\) and \(q_v\) be integers such that \(q_e, q_m, q_v \leq 2^n\) and \(q_e < 2^n\). Then, in the ideal cipher model for \(E\), for any \((q_e, q_m, q_v)\)-adversary \(A\) against the rMAC-security of NaK\([E,H]\), one has

\[
\text{Adv}_{\text{MAC}}^{\text{NaK}[E,H]}(A) \leq 2(n-1)q_m\varepsilon + \frac{q_e}{2^n} - n - q_e + nq_v\varepsilon + nq_v\varepsilon' + \left(\frac{q_e}{2^n} + 1\right)^{n+1} + 2nq_v\varepsilon' + q_m^{n+1} + \frac{q_v}{2^n - q_m - q_e}.
\]

Proof. This follows by combining Lemma 1 with \(\mu_0 = n\) and Theorem 3. □

### 4.2 The Stateless Deterministic Construction HaK

The block cipher-based variant of HaT is a stateless deterministic MAC called HaK (Hash-as-Key). Given a block cipher \(E : K \times \{0,1\}^n \rightarrow \{0,1\}^n\) and two keyed hash functions \(H : K_h \times M \rightarrow \{0,1\}^n\) and \(H' : K'_h \times M \rightarrow K\), we define a MAC with key space \(K_h \times K'_h\) and message space \(M\) as

\[
\text{HaK}[E,H,H']_{K_h,K'_h}(M) = E_{H'_K}(H_{K_h}(M)).
\]

Our security result is the following one.

**Theorem 4.** Let \(M, K, K_h,\) and \(K'_h\) be non-empty sets. Let \(E : K \times \{0,1\}^n \rightarrow \{0,1\}^n\) be a block cipher and let \(H : K_h \times M \rightarrow \{0,1\}^n\) and \(H' : K'_h \times M \rightarrow K\) be two \(\varepsilon\)-almost uniform hash functions. Let \(q_e, q_m,\) and \(q_v\) be integers such that \(q_m + q_e < 2^n\). Then, in the ideal cipher model for \(E\), for any \((q_e, q_m, q_v)\)-adversary \(A\) against the sdMAC-security of HaK\([E,H,H']\), one has

\[
\text{Adv}_{\text{sdMAC}}^{\text{HaK}[E,H]}(A) \leq q_m^2\varepsilon^2 + q_mq_v\varepsilon^2 + q_mq_v(\varepsilon')^2 + \left(\frac{q_m}{2^n}\right)^{n+1} + nq_v\varepsilon' + q_vq_v(\varepsilon')^2 + \frac{q_v}{2^n - q_m - q_e}.
\]

Proof. Deferred to Appendix D. □

### 5 Security of Truncated MACs

In this section, we analyze how tag truncation affects the security of MACs. Let \(F : K \times (N \times)M \rightarrow \{0,1\}^n\) be a keyed function with key space \(K\), message space \(M\), range \(T = \{0,1\}^n\) and potentially nonce space \(N\) (the reasoning below applies both to SD-MACs and nonce-based MACs). For any \(1 \leq s \leq n-1\), let \(\text{trunc}_s : \{0,1\}^n \rightarrow \{0,1\}^s\) be a function that takes \(s\) bits of the input in any way (e.g., the leftmost \(s\) bits of an \(n\)-bit input). Let

\[
F_s \overset{\text{def}}{=} \text{trunc}_s \circ F
\]

declare a truncated variant of \(F\) that returns only \(s\) bits of the original tag.
Lemma 8. If there exists a function \( \delta \) of \( q_m, q_v, t \), and potentially \( \mu \), such that, for any \( ((\mu), q_m, q_v, t) \)-adversary \( A \) against \( F \),

\[
\text{Adv}_{F^{\mu/nMAC}}^s(A) \leq \delta((\mu), q_m, q_v, t),
\]

then, for any \( ((\mu), q_m, q_v, t) \)-adversary \( A' \) against \( F \), one has

\[
\text{Adv}_{F^{\mu/nMAC}}^s(A') \leq \delta((\mu), q_m, 2^{n-s} q_v, t).
\]

Proof. Given a \( ((\mu), q_m, q_v, t) \)-adversary \( A \) against \( F \), one can use it as a subroutine to construct a \( ((\mu), q_m, 2^{n-s} q_v, t) \)-adversary \( A \) against \( F \) as follows:

- A faithfully relays each MAC query made by \( A' \) to its MAC oracle; if \( A \) receives \( T \) from the oracle as the answer to this query, then \( A \) sends \( T' = \text{trunc}_s(T) \) to \( A' \);

- If \( A' \) makes a verification query \( ((N'), M', T') \), then \( A \) makes \( 2^{n-s} \) verification queries \( ((N'), M', T) \) for all \( n \)-bit \( T \) such that \( \text{trunc}_s(T) = T' \).

Clearly, \( A \) is successful at least as often as \( A \), hence one has

\[
\text{Adv}_{F^{\mu/nMAC}}^s(A') \leq \text{Adv}_{F^{\mu/nMAC}}^s(A).
\]

As an example, applying this analysis to \( \text{HaT} \), we obtain the following theorem.

Theorem 5. For any \( 1 \leq s \leq n - 1 \), let

\[
\text{HaT}_s[\tilde{E}, H, H'] = \text{trunc}_s \circ \text{HaT}[\tilde{E}, H, H']
\]

denote an \( s \)-bit truncated variant of \( \text{HaT}[\tilde{E}, H, H'] \), where \( \tilde{E} : \mathbb{K} \times W \times \{0, 1\}^n \rightarrow \{0, 1\}^n \) is a tweakable block cipher and \( H : \mathbb{K}_b \times \mathcal{M} \rightarrow \{0, 1\}^n \) and \( H' : \mathbb{K}_b' \times \mathcal{M} \rightarrow W \) are \( \varepsilon \)-AU hash functions. Let \( q_m, q_v, \) and \( t \) be integers such that \( q_m < 2^n \). Then for any \( (q_m, q_v, t) \)-adversary \( A \) against the sdMAC-security of \( \text{HaT}_s[\tilde{E}, H, H'] \), there exists a \( (q_m + q_v, t) \)-adversary \( A' \) against the TPRP-security of \( \tilde{E} \), where \( t' = O(t + (q_m + q_v) t_H) \) and \( t_H \) is an upper bound on the time to compute \( H \) or \( H' \) on any message, such that

\[
\text{Adv}_{\text{HaT}_s[\tilde{E}, H, H']^{\mu/E}}^s(A) \leq \text{Adv}_{\tilde{E}^{\mu/E}}^{\text{TPRP}}(A') + 2 q_m^2 \varepsilon^2 + 2^{n-s} q_v \varepsilon^2 + \frac{2^{n-s} q_v}{2^n - q_v}.
\]

Assuming \( \varepsilon \simeq t 2^{-n} \), where \( t \) is the maximal length of messages in \( n \)-bit blocks, the \( \text{HaT}_s \) construction is secure up to \( q_m t \simeq 2^n \) blocks in MAC queries and \( q_v \simeq 2^n \) verification queries, as long as \( q_m q_v t^2 \) is small compared to \( 2^{n+s} \). Similar results can be obtained for nonce-based, randomized and/or ideal cipher-based MACs.

Acknowledgments

We thank the anonymous reviewers of TOSC for their useful feedback and suggestions. The first author has been partially supported by the European Union’s H2020 Programme under grant agreement number ICT-644209 and the third author has been partially supported by the French Agence Nationale de la Recherche through the BRUTUS project under Contract ANR-14-CE28-0015.
References


A Proof of Lemma 2

Let $\Theta'$ be the set of all transcripts $\tau$ such that

$$\max\{\Pr[X_{id} = \tau], \Pr[X_{re} = \tau]\} > 0.$$ 

Remark that the set $\Theta$ of attainable transcripts is included in (and in our case, different from) $\Theta'$. Let $O_{id}$, resp. $O_{re}$ denote the oracle from the ideal, resp. the real world. Recall that

$$\text{Adv}(D) = |\Pr[D_{O_{id}} = 1] - \Pr[D_{O_{re}} = 1]|$$

Moreover, the distinguisher’s output is a deterministic function of the transcript. If we let $\Theta'_i$ denote the subset of $\Theta'$ such that $D$ outputs $i$, for $i = 0, 1$, it is easy to see that

$$\Pr[D_{O_{id}} = i] = \sum_{\tau \in \Theta'_i} \Pr[X_{id} = \tau]$$

and

$$\Pr[D_{O_{re}} = i] = \sum_{\tau \in \Theta'_i} \Pr[X_{re} = \tau]$$

for $i = 0, 1$. Thus

$$\text{Adv}(D) = \left| \sum_{\tau \in \Theta'_0} (\Pr[X_{re} = \tau] - \Pr[X_{id} = \tau]) \right|$$

$$\leq \sum_{\tau \in \Theta'_0} |\Pr[X_{re} = \tau] - \Pr[X_{id} = \tau]|.$$ 

Similarly,

$$\text{Adv}(D) \leq \sum_{\tau \in \Theta'_0} |\Pr[X_{re} = \tau] - \Pr[X_{id} = \tau]|,$$

which implies that

$$\text{Adv}(D) \leq \frac{1}{2} \sum_{\tau \in \Theta'} |\Pr[X_{re} = \tau] - \Pr[X_{id} = \tau]| = \|X_{re} - X_{id}\|$$

since $\Theta' = \Theta'_0 \cup \Theta'_1$. Moreover, one has

$$\|X_{re} - X_{id}\| = \sum_{\Pr[X_{id} = \tau] > \Pr[X_{re} = \tau]} (\Pr[X_{id} = \tau] - \Pr[X_{re} = \tau]).$$

For every transcript $\tau$ appearing in this sum, one has $\Pr[X_{id} = \tau] > \Pr[X_{re} = \tau]$, which means, in particular, that $\tau$ is an attainable transcript. Thus one has

$$\|X_{re} - X_{id}\| = \sum_{\Pr[X_{id} = \tau] > \Pr[X_{re} = \tau]} (\Pr[X_{id} = \tau] - \Pr[X_{re} = \tau])$$

$$= \sum_{\Pr[X_{id} = \tau]} \Pr[X_{id} = \tau] \left(1 - \frac{\Pr[X_{re} = \tau]}{\Pr[X_{id} = \tau]}\right)$$

$$\leq \sum_{\tau \in \Theta_{good}} \Pr[X_{id} = \tau] \epsilon_1 + \sum_{\tau \in \Theta_{bad}} \Pr[X_{id} = \tau]$$

$$\leq \epsilon_1 + \epsilon_2.$$
B The UHF-then-RO Construction

We prove the following theorem.

**Theorem 6.** Let $H : K_h \times M \rightarrow \{0, 1\}^m$ be a $\varepsilon$-AU and $\varepsilon'$-almost uniform hash function and $G : \{0, 1\}^m \rightarrow \{0, 1\}^n$. Let $F : K_h \times M \rightarrow \{0, 1\}^n$ be the keyed function defined as $F_{K_h}(M) = G(H_{K_h}(M))$.

Then, in the random oracle for $G$, for any adversary $A$ making at most $q$ queries to $F$ and $q'$ queries to $G$, one has

$$
\text{Adv}_{A}^{\text{PRF}}(A) \leq \frac{q^2 \varepsilon}{2} + qq' \varepsilon'.
$$

**Proof.** The adversary is trying to distinguish $F_{K_h}$ for a random key $K_h$ from a uniformly random function $\text{Rand} : M \rightarrow \{0, 1\}^n$. Let $\tau = (\tau_f, \tau_g, K_h)$ be the transcript of the attack, where

$$
\tau_f = ((M_1, T_1), \ldots, (M_q, T_q))
$$

$$
\tau_g = ((X_1, Y_1), \ldots, (X_{q'}, Y_{q'}))
$$

are respectively the queries of the adversary to $F$ and $G$. (As usual, we provide the real or a dummy key to the distinguisher at the end of the attack, depending on which oracle it is interacting with.)

We say that a transcript is bad if there exists two distinct queries $(M_i, T_i), (M_j, T_j) \in \tau_f$ such that $H_{K_h}(M_i) = H_{K_h}(M_j)$, or if there exists $(M, T) \in \tau_f$ and $(X, Y) \in \tau_g$ such that $H_{K_h}(M) = X$. By respectively the $\varepsilon$-AU and $\varepsilon'$-almost uniformity of $H$, and since in the ideal world $K_h$ is drawn independently from $(\tau_f, \tau_g)$, the probability to obtain a bad transcript in the ideal world is at most

$$
\frac{q^2 \varepsilon}{2} + qq' \varepsilon'.
$$

Fix now any good transcript $\tau = (\tau_f, \tau_g, K_h)$. The probability to obtain $\tau$ in the ideal world is

$$
\frac{1}{|K_h|} \cdot \Pr[F(M_i) = T_i, i \in \{1, \ldots, q\}] \cdot \Pr[G(X_j) = Y_j, j \in \{1, \ldots, q'\}]
$$

$$
= \frac{1}{|K_h|} \cdot \frac{1}{(2^n)^{q+q'}},
$$

while in the real world it is

$$
\frac{1}{|K_h|} \cdot \Pr[G(H_{K_h}(M_i)) = T_i, i \in \{1, \ldots, q\} \text{ and } G(X_j) = Y_j, j \in \{1, \ldots, q'\}]
$$

$$
= \frac{1}{|K_h|} \cdot \frac{1}{(2^n)^{q+q'}},
$$

since by definition of a good transcript, all values $H_{K_h}(M_i), i = 1, \ldots, q$ and $X_j, j = 1, \ldots, q'$ are distinct. Hence the ratio is 1 and Lemma 2 allows to conclude.

C Proof of Theorem 3

Following Section 2.3, let us fix a non-trivial $(\mu, q_e, q_m, q_r)$-distinguisher $D$ interacting either with the real world $(E, \text{NaK}[E, H]_{K_h}, \text{Ver}[E, H]_{K_h})$ for a uniformly random block cipher $E$ and a random hashing key $K_h$, or with the ideal world $(E, \text{Rand}, \text{Rej})$. We let

$$
\text{Adv}(D) = \Pr[D^{E, \text{NaK}[E, H]_{K_h}, \text{Ver}[E, H]_{K_h}} = 1] - \Pr[D^{E, \text{Rand}, \text{Rej}} = 1].
$$
Let $\tau = (\tau_e, \tau_m, \tau_v, K_h)$ denote the transcript of the attack, with

$$
\tau_e = ((K_1, X_1, Y_1), \ldots, (K_{q_e}, X_{q_e}, Y_{q_e}))
$$
$$
\tau_m = ((N_1, M_1, T_1), \ldots, (N_{q_m}, M_{q_m}, T_{q_m}))
$$
$$
\tau_v = ((N'_1, M'_1, T'_1), \ldots, (N'_{q_v}, M'_{q_v}, T'_{q_v})).
$$

Recall that $\Theta$ denotes the set of attainable transcripts and $X_{re}$, resp. $X_{id}$, the probability distribution of the transcript $\tau$ induced by the real world, resp. the ideal world.

The remaining of the proof of Theorem 3 is structured as follows. First, we define bad transcripts and upper bound their probability in the ideal world (Lemma 9). Then, we analyze good transcripts and prove that they are almost as likely in the real and the ideal transcripts and upper bound their probability in the ideal world (Lemma 10). Theorem 3 follows easily by combining Equation (1) and Lemma 2 from Section 2.3 with Lemma 9 and Lemma 10.

**Definition 6.** We say that an attainable transcript $\tau = (\tau_e, \tau_m, \tau_v, K_h)$ is **bad** if one of these conditions is fulfilled:

1. (C-1) there exists two distinct MAC queries $(N_i, M_i, T_i)$ and $(N_j, M_j, T_j)$ such that $N_i = N_j$ and either $H_{K_h}(M_i) = H_{K_h}(M_j)$ or $T_i \oplus H_{K_h}(M_i) = T_j \oplus H_{K_h}(M_j)$;

2. (C-2) there exists an IC query $(K_i, X_i, Y_i) \in \tau_e$ and a MAC query $(N_j, M_j, T_j) \in \tau_m$ such that $K_i = N_j$ and either $X_i = H_{K_h}(M_j)$ or $Y_i = T_j \oplus H_{K_h}(M_j)$;

3. (C-3) there exists a MAC query $(N_i, M_i, T_i) \in \tau_m$ and a verification query $(N'_j, M'_j, T'_j) \in \tau_v$ such that

   $$
   \begin{cases}
   N_i = N'_j \\
   H_{K_h}(M_i) = H_{K_h}(M'_j) \\
   T_i = T'_j,
   \end{cases}
   $$

4. (C-4) there exists an IC query $(K_i, X_i, Y_i) \in \tau_e$ and a verification query $(N'_j, M'_j, T'_j) \in \tau_v$ such that

   $$
   \begin{cases}
   K_i = N'_j \\
   X_i = H_{K_h}(M'_j) \\
   Y_i = T'_j \oplus H_{K_h}(M'_j).
   \end{cases}
   $$

We let $\Theta_{bad}$, resp. $\Theta_{good}$ denote the set of bad, respectively good transcripts.

Note that the third and fourth conditions can only happen in the ideal world since in the real world, if e.g., $N = N'$, $T = T'$, and $H_{K_h}(M) = H_{K_h}(M')$, the verification oracle should return 1 on query $(N', M', T')$ (which is impossible for any attainable transcript). We now upper bound the probability to get a bad transcript in the ideal world.

**Lemma 9.** For any integers $q_e$, $q_m$ and $q_v$ such that $q_e \leq 2^n$, one has

$$
\Pr[X_{id} \in \Theta_{bad}] \leq 2(\mu - 1)q_m\varepsilon + mq_{e}\varepsilon' + nq_{e}\varepsilon' + \left(\frac{q_e}{2^n - q_e + 1}\right)^{n+1} + 2\mu q_e\varepsilon'.
$$

**Proof.** We let $\Theta_i$ denote the set of attainable transcripts satisfying condition (C-i). Recall that, in the ideal world, $K_h$ is drawn independently from the queries transcript, and that $E$ is independent from $Rand$. We are going to consider the four conditions in turn.
**CONDITION (C-1).** Fix a MAC query \((N_i, M_i, T_i)\). There are exactly \(q_m\) possible choices for this query. Then we fix another MAC query \((N_j, M_j, T_j)\) such that \(N_i = N_j\) (there are at most \(\mu - 1\) possible choices). The probability, over the random draw of \(T_i\) and \(T_j\) that \(T_i \oplus H_K(M_i) = T_j \oplus H_K(M_j)\) is \(2^{-n}\), and the probability, over the random draw of \(K\), that \(H_K(M_i) = H_K(M_j)\), is at most \(\varepsilon\). Summing over every possible choice of \((N_i, M_i, T_i)\) and \((N_j, M_j, T_j)\), we get

\[
\Pr[X_{id} \in \Theta_1] \leq \left( \frac{\mu - 1}{2^n} \right) q_m + (\mu - 1)q_m \varepsilon \leq 2(\mu - 1)q_m \varepsilon,
\]

where we used that \(\varepsilon \geq 2^{-n}\) (see Remark 1).

**CONDITION (C-2).** Fix an ideal cipher query \((K_i, X_i, Y_i) \in \tau_e\). Then, since \(D\) cannot repeat a nonce in its MAC queries more than \(\mu\) times, there are at most \(\mu\) MAC queries \((N_j, M_j, T_j) \in \tau_m\) such that \(K_i = N_j\). Fix any of these queries. Then, the probability, over the random draw of \(K\), that either \(X_i = H_K(M_j)\) or \(Y_i = T_j \oplus H_K(M_j)\) is lower than \(2\varepsilon\) thanks to the \(\varepsilon'\)-almost uniformity of \(H\). Summing over every possible choice of queries, we get

\[
\Pr[X_{id} \in \Theta_2] \leq 2\mu q_e \varepsilon'.
\]

**CONDITION (C-3).** This condition is exactly the same as condition (C-2) in Lemma 4, hence by exactly the same proof one has

\[
\Pr[X_{id} \in \Theta_3] \leq \mu q_e \varepsilon.
\]

**CONDITION (C-4).** In order to upper bound the probability that condition (C-4) is fulfilled, we need to upper bound the number of ideal cipher queries \((K_i, X_i, Y_i) \in \tau_e\) satisfying \(K_i = N_j\) and \(H_K(M_j) = X_i = Y_i \oplus T_j\), for a verification query \((N_j, M_j, T_j) \in \tau_m\). In particular, this means that such a query must satisfy \(X_i \oplus Y_i = T_j\). The first step of our proof is to upper bound the probability, over the random draw of \(E\), that there exists \(n + 1\) distinct ideal cipher queries \((K_i, X_i, Y_i, i = 1, \ldots, n)\) such that \(X_i \oplus Y_i = T_j\) is constant for \(l = 1, \ldots, n + 1\). Let us define

\[
\alpha(E) = \max_{a \in \{0, 1\}^n} \left| \{i \in \{1, \ldots, q_e\} : X_i \oplus Y_i = a \} \right|.
\]

We are going to upper bound the probability, over the random choice of \(E\), that \(\alpha(E) \geq n + 1\). Fix any \(n + 1\)-tuple of indexes \((i_1, \ldots, i_{n+1})\) such that \(1 \leq i_1 < \cdots < i_{n+1} \leq q_e\). Then one has

\[
\Pr[X_{i_1} \oplus Y_{i_1} = \cdots = X_{i_{n+1}} \oplus Y_{i_{n+1}}] \leq \frac{1}{(2^n - q_e + 1)^{n+1}}.
\]

This can easily be seen as follows: if query \((K_{i_j}, X_{i_j}, Y_{i_j})\) is an encryption (resp. decryption) query, then \(Y_{i_j}\) (resp. \(X_{i_j}\)) is chosen uniformly at random in a set of size at least \(2^n - q_e + 1\), and the probability to have \(X_{i_j} \oplus Y_{i_j} = X_{i_j} \oplus Y_{i_j}\) is lower than \(1/(2^n - q_e + 1)\), for every \(j = 2, \ldots, n + 1\). Summing over every such possible tuple of queries, one has

\[
\Pr[\alpha(E) \geq n + 1] = \Pr[\exists 1 \leq i_1 < \cdots < i_{n+1} \leq q_e : X_{i_1} \oplus Y_{i_1} = \cdots = X_{i_{n+1}} \oplus Y_{i_{n+1}}] \leq \frac{(q_e)_{n+1}}{(n+1)!}(2^n - q_e + 1)^n \leq \left( \frac{q_e}{2^n - q_e + 1} \right)^{n+1},
\]

where we used that \((n+1)! \geq 2^n \geq 2^n - q_e + 1\). Now assume that \(\alpha(E) \leq n\) and fix any verification query \((N_j, M_j, T_j) \in \tau_m\). There are at most \(n\) ideal cipher queries \((K_i, X_i, Y_i) \in \tau_e\) satisfying \(X_i \oplus Y_i = T_j\). Fix any of these queries. The probability, over
the random choice of $K_h$, that $H_{K_h}(M'_i) = X_i$, is lower than $\varepsilon'$. Thus, by summing over every possible choice of queries, one has

$$\Pr[X_{id} \in \Theta_\delta] \leq \Pr[\alpha(E) \geq n + 1] + \Pr[\alpha(E) \leq n] \left(nq_e\varepsilon'\right) \leq \left(\frac{q_e}{2n-q_e+1}\right)^{n+1} + nq_e\varepsilon'.$$

Note that while this reasoning assumes that $q_e \geq n + 1$, the bound still holds when $q_e \leq n$.

The result follows by a union bound over these conditions. 

We now analyze good transcripts and prove the following lemma.

**Lemma 10.** For any good transcript $\tau$, one has

$$\frac{\Pr[X_{re} = \tau]}{\Pr[X_{id} = \tau]} \geq 1 - \frac{q_e}{2^n - \mu - q_e}.$$

**Proof.** Let $\tau = (\tau_e, \tau_m, \tau_v, K_h)$ be a good transcript. Let $\mathcal{L} = \{K_1, \ldots, K_{q_e}, N_1, \ldots, N_{q_m}\}$ be the set of every key or nonce used in the ideal cipher or MAC queries. Using an arbitrary order, we rewrite the set $\mathcal{L}$ as

$$\mathcal{L} = \{L_1, \ldots, L_r\},$$

where $r$ is the total number of distinct values in $\mathcal{L}$. For $i = 1, \ldots, r$, we let $q_i$ denote the number of ideal cipher queries in $\tau_e$ using $L_i$ as a key and $q'_i$ the number of MAC queries using $L_i$ as a nonce.

Since in the ideal world the ideal cipher is perfectly random and independent from the other oracles, the MAC oracle is perfectly random, and the verification oracle always rejects, one simply has

$$\Pr[X_{id} = \tau] = \frac{1}{|\mathcal{K}_h| \cdot \left(2^n\right)^{q_e} \cdot \prod_{i=1}^{r} \left(2^n\right)^{q_i}} = \frac{1}{|\mathcal{K}_h| \prod_{i=1}^{r} \left(2^n\right)^{q_i}},$$

since $q_e = \sum_{i=1}^{r} q_i$ and $q_m = \sum_{i=1}^{r} q'_i$. We must now lower bound the probability of getting $\tau$ in the real world. We say that a block cipher $E$ is compatible with $\tau_e$ if

$$\forall i \in \{1, \ldots, q_e\}, \ E_{K_i}(X_i) = Y_i,$$

compatible with $\tau_m$ if

$$\forall i \in \{1, \ldots, q_m\}, \ \text{NaK}[E, H]_{K_h}(N_i, M_i) = T_i,$$

and compatible with $\tau_v$ if

$$\forall i \in \{1, \ldots, q_v\}, \ \text{NaK}[E, H]_{K_h}(N_i', M_i') \neq T_i'.$$

We simply say that $E$ is compatible with $\tau$ if it is compatible with $\tau_e$, $\tau_m$, and $\tau_v$. We let $\text{Comp}(\tau_e, \tau_m)$, $\text{Comp}(\tau_v)$, and $\text{Comp}(\tau)$ denote the set of block ciphers that are compatible with respectively $\tau_e$, $\tau_m$, $\tau_v$, and $\tau$. Then one can easily check (see for example [CS14] for a detailed explanation) that

$$\Pr[X_{re} = \tau] = \frac{1}{|\mathcal{K}_h|} \cdot \Pr[E \leftarrow \$ \text{Perm}(\mathcal{K}, n) : E \in \text{Comp}(\tau)].$$

We now define

$$\lambda_{eq} = \{(K_1, X_1, Y_1), \ldots, (K_{q_e}, X_{q_e}, Y_{q_e})\} \cup \{(N_1, H_{K_h}(M_1), T_1 \oplus H_{K_h}(M_1)), \ldots, (N_{q_m}, H_{K_h}(M_{q_m}), T_{q_m} \oplus H_{K_h}(M_{q_m}))\},$$
and

\[ \lambda_{\text{ineq}} = \{(N'_1, H_{K_h}(M'_1), T'_1 \oplus H_{K_h}(M'_1)), \ldots, (N'_q, H_{K_h}(M'_q), T'_q \oplus H_{K_h}(M'_q))\}. \]

Then, since \( \tau \) is a good transcript, \( \lambda_{\text{eq}} \) is a permutation equalities list (as otherwise condition (C-1) or (C-2) would be fulfilled), and \( \lambda_{\text{ineq}} \) is a permutation inequalities list which is compatible with \( \lambda_{\text{eq}} \) (as otherwise condition (C-3) or (C-4) would be fulfilled). Moreover \( |\lambda_{\text{eq}}| = q_m + q_v, |\lambda_{\text{ineq}}| = q_e \), and for \( i = 1, \ldots, r \), \( L_i \) appears in \( \lambda_{\text{eq}} \) exactly \( q_i + q'_i \leq q_e + \mu \) times. Note that the event \( E \in \text{Comp}(\tau) \) is actually equivalent to the event \( E \in \text{Comp}(\lambda) \) where \( \lambda = (\lambda_{\text{eq}}, \lambda_{\text{ineq}}) \). Using Lemma 3, one has

\[ \Pr [E \in \text{Comp}(\tau)] \geq \frac{1}{\prod_{i=1}^{r} (2^n)^{q_i + q'_i}} \left( 1 - \frac{q_v}{2^n - \mu - q_e} \right). \]

Combining this equation with Equation (11) and Equation (12), we get

\[ \frac{\Pr [X_{\text{tv}} = \tau]}{\Pr [X_{\text{id}} = \tau]} \geq \left( 1 - \frac{q_v}{2^n - \mu - q_e} \right) \cdot \prod_{i=1}^{r} \left( \frac{(2^n)^{q_i}}{2^n} \right) \geq 1 - \frac{q_v}{2^n - \mu - q_e}. \]

\[ \square \]

D Proof of Theorem 4

Following Section 2.3, let us fix a non-trivial \((q_v, q_m, q_E)\)-distinguisher \( D \) interacting either with the real world \((E, \text{Hash}[E, H, H']_{K_h, K'_h}, \text{Ver}[E, H, H']_{K_h, K'_h})\) for a uniformly random block cipher \( E \) and independent random hashing keys \( K_h \) and \( K'_h \), or with the ideal world \((E, \text{Rand}, \text{Rej})\), making at most \( q_m \) queries to its left (ideal cipher) oracle, at most \( q_m \) queries to its middle (MAC) oracle and at most \( q_v \) queries to its right (verification) oracle, and outputting a single bit. We let

\[ \text{Adv}(D) = \Pr [E, \text{Hash}[E, H, H']_{K_h, K'_h}, \text{Ver}[E, H, H']_{K_h, K'_h} = 1] - \Pr [D, \text{Rand}, \text{Rej} = 1]. \]

Let \( \tau = (\tau_e, \tau_m, \tau_v, K_h, K'_h) \) be the transcript of the attack, where

\[ \tau_e = ((K_1, X_1, Y_1), \ldots, (K_q, X_q, Y_q)) \]
\[ \tau_m = ((M_1, T_1), \ldots, (M_q, T_q)) \]
\[ \tau_v = ((M'_1, T'_1), \ldots, (M'_q, T'_q)). \]

As usual, we let \( \Theta \) denote the set of attainable transcripts, and \( X_{\text{tv}}, \) resp. \( X_{\text{id}}, \) the probability distribution of the transcript \( \tau \) induced by the real world, resp. the ideal world. As in Appendix C, Theorem 4 follows easily by combining Equation (1) and Lemma 2 from Section 2.3 with Lemma 11 and Lemma 12 proven below.

We start by defining bad transcripts.

Definition 7. We say that an attainable transcript \( \tau = (\tau_e, \tau_m, \tau_v, K_h, K'_h) \) is bad if one of these conditions is fulfilled:

(C-1) there exists two distinct MAC queries \((M_i, T_i)\) and \((M_j, T_j)\) such that \( H'_{K'_h}(M_i) = H'_{K'_h}(M_j) \) and either \( H_{K_h}(M_i) = H_{K_h}(M_j) \) or \( T_i = T_j \);

(C-2) there exists an IC query \((K_i, X_i, Y_i) \in \tau_e \) and a MAC query \((M_j, T_j) \in \tau_m \) such that \( K_i = H'_{K'_h}(M_j) \) and either \( X_i = H_{K_h}(M_j) \) or \( Y_i = T_j \);
We let 

where we used that (C-4) there exist an IC query 

Hence, each such pair, 

\( nq \transcript, \text{i.e.,} \)

and

The probability that there exists an IC query 

\[ \Pr \left[ X_{\text{id}} \in \Theta_{\text{bad}} \right] \leq q_m^2 \varepsilon^2 + q_m q_e (\varepsilon')^2 + \left( \frac{q_m}{2^n} \right)^{n+1} + n q_e \varepsilon' + q_m q_e^2 + q_e q_e (\varepsilon')^2. \]

Proof. Let \( \Theta_i \) denote the set of attainable transcripts satisfying condition (C-1). Recall that, in the ideal world, \((K_h, K'_h)\) is drawn independently from the queries transcript. We are going to consider each condition in turn.

**Condition (C-1).** This condition is exactly the same as condition (C-1) in Lemma 6, hence by exactly the same proof one has

\[ \Pr \left[ X_{\text{id}} \in \Theta_1 \right] \leq q_m^2 \varepsilon^2. \]

**Condition (C-2).** The probability that there exists an IC query \((K_i, X_i, Y_i) \in \tau_e\) and a MAC query \((M_j, T_j) \in \tau_m\) such that \( K_i = H'_{K_h} (M_j) \) and \( X_i = H_{K_h} (M_j) \) (over the draw of \( K_h \) and \( K'_h \)) is at most \( q_m q_e (\varepsilon')^2 \). We now upper bound the probability that there exists an IC query \((K_i, X_i, Y_i) \in \tau_e\) and a MAC query \((M_j, T_j) \in \tau_m\) such that \( K_i = H'_{K_h} (M_j) \) and \( Y_i = T_j \). Let us denote \( \alpha(\tau_m) \) the maximal multiplicity of any tag in the MAC queries transcript, i.e.,

\[ \alpha(\tau_m) = \max_{T \in \{0,1\}^n} \left| \left\{ j \in \{1, \ldots, q_m \} : T_j = T \right\} \right|. \]

Then, over the random draw of the \( T_j \)'s, one has

\[ \Pr \left[ \alpha(\tau_m) \geq n + 1 \right] = \Pr \left[ \exists 1 \leq i_1 < \cdots < i_{n+1} \leq q_m : T_{i_1} = \cdots = T_{i_{n+1}} \right] \]

\[ \leq \frac{\left( \frac{q_m}{2^n} \right)^{n+1}}{(n+1)!(2^n)^n} \leq \left( \frac{q_m}{2^n} \right)^{n+1}, \]

where we used that \((n+1)! \geq 2^n\). Now assume that \( \alpha(\tau_m) \leq n \). Then there are at most \( n q_e \) pairs of ideal cipher/MAC queries \(((K_i, X_i, Y_i), (M_j, T_j))\) such that \( Y_i = T_j \) and for each such pair, \( K_i = H'_{K_h} (M_j) \) with probability at most \( \varepsilon' \) over the random choice of \( K_h \). Hence,

\[ \Pr \left[ X_{\text{id}} \in \Theta_2 \right] \leq q_m q_e (\varepsilon')^2 + \Pr \left[ \alpha(\tau_m) \geq n + 1 \right] + \Pr \left[ \alpha(\tau_m) \leq n \right] (n q_e \varepsilon') \]

\[ \leq q_m q_e (\varepsilon')^2 + \left( \frac{q_m}{2^n} \right)^{n+1} + n q_e \varepsilon'. \]
CONDITION (C-3). This condition is exactly the same as condition (C-2) in Lemma 6, hence by exactly the same proof one has
\[ \Pr[X_{id} \in \Theta_3] \leq q_m q_e \varepsilon^2. \]

CONDITION (C-4). Fix an ideal cipher query \((K_i, X_i, Y_i) \in \tau_e\) and a verification query \((M_i', T_i') \in \tau_v\). Since in the ideal world \(K_h\) and \(K_h'\) are drawn independently from the queries transcript and \(H\) and \(H'\) are \(\varepsilon'\)-almost uniform, the probability that \(K_i = H'_h(M_i')\) and \(X_i = H_h(K_i)(M_i')\) is upper bounded by \((\varepsilon')^2\) (just ignoring the condition \(Y_i = T_i'\)), and hence
\[ \Pr[X_{id} \in \Theta_4] \leq q_e q_v (\varepsilon')^2. \]

The result follows by a union bound over these conditions. \(\square\)

We now analyze good transcripts and prove the following lemma.

**Lemma 12.** For any good transcript \(\tau\), one has
\[ \frac{\Pr[X_{re} = \tau]}{\Pr[X_{id} = \tau]} = 1 - \frac{q_v}{2^n - q_m - q_e}. \]

**Proof.** Let \(\tau = (\tau_e, \tau_m, \tau_v, K_h, K_h')\) be a good transcript. Let
\[ \mathcal{L} = \{K_1, \ldots, K_{q_v}, H'_1(M_1), \ldots, H'_{q_m}(M_{q_m})\} \]
be the set of all the keys used in the ideal cipher or MAC queries. Using an arbitrary order, we rewrite the set \(\mathcal{L}\) as
\[ \mathcal{L} = \{L_1, \ldots, L_r\}, \]
where \(r\) is the total number of distinct values in \(\mathcal{L}\). For \(i = 1, \ldots, r\), we let \(q'_i\) denote the number of ideal cipher queries \((K, X)\) in \(\tau_e\) such that \(K = L_i\) and \(q'_{v_i}\) the number of MAC queries \((M, T)\) in \(\tau_v\) such that \(H'_h(M) = L_i\).

Since in the ideal world the ideal cipher is perfectly random and independent from the other oracles, the MAC oracle is perfectly random, and the verification always rejects, one simply has
\[ \Pr[X_{id} = \tau] = \frac{1}{|\mathcal{K}_h| \cdot |\mathcal{K}'_h| \cdot (2^n)^{q_m} \cdot \prod_{i=1}^{r} (2^n)_{q'_i}} = \frac{1}{|\mathcal{K}_h| \cdot |\mathcal{K}'_h| \cdot \prod_{i=1}^{r} (2^n)^{q'_i}}, \]  

(13)
since \(q_m = \sum_{i=1}^{r} q'_i\). We must now lower bound the probability of getting \(\tau\) in the real world. We say that a block cipher \(E\) is compatible with \(\tau_m\) if
\[ \forall i \in \{1, \ldots, q_m\}, \; \text{HaK}[E, H]_{K_h, K_h'}(M_i) = T_i, \]
compatible with \(\tau_e\) if
\[ \forall i \in \{1, \ldots, q_e\}, \; E_{K_i}(X_i) = Y_i, \]
and compatible with \(\tau_v\) if
\[ \forall i \in \{1, \ldots, q_v\}, \; \text{HaK}[E, H]_{K_h, K_h'}(M_i') \neq T_i'. \]

We simply say that \(E\) is compatible with \(\tau\) if it is compatible with \(\tau_e, \tau_m, \) and \(\tau_v\). We let \(\text{Comp}(\tau_e, \tau_m), \text{Comp}(\tau_v), \) and \(\text{Comp}(\tau)\) denote the set of block ciphers that are compatible with respectively \(\tau_e\) and \(\tau_m, \tau_v, \) and \(\tau\). Then one can easily check that
\[ \Pr[X_{re} = \tau] = \frac{1}{|\mathcal{K}_h| \cdot |\mathcal{K}'_h|} \cdot \Pr[E \leftarrow_{\$} \text{Perm}(\mathcal{K}, n) : E \in \text{Comp}(\tau)]. \]  

(14)
We now define
\[
\lambda_{eq} = \{(K_1, X_1, Y_1), \ldots, (K_{qe}, X_{qe}, Y_{qe})\} \cup \bigcup_{i=1}^{qm} \{(H_{K_i}^h(M_i), H_{K_i}(M_i), T_i)\},
\]
and
\[
\lambda_{ineq} = \bigcup_{i=1}^{qv} \{(H_{K_i}^h(M_i'), H_{K_i}(M_i'), T_{i}')\}.
\]
Then, since \(\tau\) is a good transcript, \(\lambda_{eq}\) is a permutation equalities list (as otherwise condition (C-1) or (C-2) would be fulfilled) and \(\lambda_{ineq}\) is a permutation inequalities list which is compatible with \(\lambda_{eq}\) (as otherwise condition (C-3) or (C-4) would be fulfilled). Moreover \(|\lambda_{eq}| = q_m + q_e\), \(|\lambda_{ineq}| = q_v\), and for \(i = 1, \ldots, r\), key \(L_i\) appears in \(\lambda_{eq}\) exactly \(q_i + q_i' \leq q_e + q_m\) times. Note that the event \(E \in \text{Comp}(\tau)\) is actually equivalent to the event \(E \in \text{Comp}(\lambda)\) where \(\lambda = (\lambda_{eq}, \lambda_{ineq})\). Using Lemma 3, one has
\[
\Pr[E \in \text{Comp}(\tau)] \geq \frac{1}{\prod_{i=1}^{r} (2^n)^{q_i + q_i'}} \left(1 - \frac{q_v}{2^n - q_m - q_e}\right).
\]
Combining this with Equation (13) and Equation (14), we get
\[
\frac{\Pr[X_{re} = \tau]}{\Pr[X_{id} = \tau]} \geq \left(1 - \frac{q_v}{2^n - q_m - q_e}\right) \prod_{i=1}^{r} \frac{(2^n)^{q_i}}{(2^n)^{q_i + q_i'}} \geq 1 - \frac{q_v}{2^n - q_m - q_e}.
\]