

Misuse-Resistant Variants of the OMD Authenticated Encryption Mode

Reza Reyhanitabar, Serge Vaudenay, and Damian Vizár

EPFL, Lausanne, Switzerland

Abstract. We present two variants of OMD which are robust against nonce *misuse*. Security of OMD—a CAESAR candidate—relies on the assumption that implementations always ensure *correct* use of nonce (a.k.a. message number); namely that, the nonce never gets repeated. However, in some application environments, this non-repetitiveness requirement on nonce might be compromised or ignored, yielding to full collapse of the security guaranty. We aim to reach maximal possible level of robustness against repeated nonces, as defined by Rogaway and Shrimpton (FSE 2006) under the name misuse-resistant AE (MRAE). Our first scheme, called misuse-resistant OMD (MR-OMD), is designed to be substantially similar to OMD while achieving stronger security goals; hence, being able to reuse any existing common code/hardware. Our second scheme, called parallelizable misuse-resistant OMD (PMR-OMD), further deviates from the original OMD design in its encryption process, providing a parallelizable algorithm, in contrast with OMD and MR-OMD which have serial encryption/decryption processes. Both MR-OMD and PMR-OMD are single-key mode of operation. It is known that maximally robust MRAE schemes are necessarily two-pass, a price paid compared to a one-pass scheme such as OMD. Nevertheless, in MR-OMD and PMR-OMD, we combine the two passes in a way that minimizes the incurred additional cost: the overhead incurred by the second pass in our two-pass variants is about 50% of the encryption time for OMD.

Keywords: authenticated encryption, misuse-resistance, OMD, CAESAR competition.

1 Introduction

An authenticated encryption scheme (AE) is a symmetric-key scheme that guarantees confidentiality (privacy) and authenticity (integrity) of data at the same time. Classical authenticated encryption schemes were based on the *generic composition* paradigm: combining a traditional encryption scheme for privacy with a message authentication code (MAC) for integrity. Generic composition based schemes were formally analysed for the first time in [2], and more recently, further investigated in [19].

The syntax and security notions for authenticated encryption, as a primitive of its own right, were originally formalized in [2, 3, 16], and further developed to include different variations in [12, 21, 23, 24].

Once the topic started to be investigated more, it became clear that there is a need for *dedicated* authenticated encryption schemes—designs that would provide higher security levels, efficiency or other desired features, in particular, being easier to use and less prone to implementation errors/attacks, compared to the generic composition-based schemes. This is backed by the fact that the generic composition paradigm is neither the most efficient (it requires processing the input stream at least twice) nor the most robust to implementation errors [7, 19, 27].

In this line, one of the most commonly known schemes is the GCM algorithm, which was originally introduced in [18] and standardized by NIST [11] as a blockcipher mode of operation for AE. GCM is a representative example of a nonce-based, one-pass AE scheme which supports “associated data”—data that are logically bound to the plaintext, need to be authenticated, but not to be encrypted. Other prominent standard algorithms in this category include CCM [10], OCB [17, 22, 23], and EAX [4], which are specified in ISO/IEC 19772:2009.

Lately, authenticated encryption has received a lot of attention through the recent CAESAR competition [5]. There were 57 submissions to the first round of CAESAR, from which (at the time of writing this paper) 7 were withdrawn due to major attacks. The proposed CAESAR candidates cover a wide range of designs, advertising different features, such as, being super efficient, single-pass (online), fully or partially nonce-misuse-resistant; online misuse-resistant, and so on.

One of the CAESAR candidates is the Offset Merkle-Damgård (OMD)—a nonce-based, single-pass mode of operation for authenticated encryption with associated data that uses a compression function as its lower-level primitive. To the best of our knowledge, OMD is the only candidate that uses a compression function (in particular, those of SHA-256 and SHA-512). The majority of other candidates are (AES) blockcipher-based or permutation-based, and some use round functions of AES. OMD has some promising features, among them, are provable security in the standard model based on the well-known PRF assumption on the compression function and high bit-security level (127 bits and 255 bits for OMD-sha256 and OMD-sha512, respectively). Being able to take advantage of the Intel SHA instructions on next-generation processors [28] also seems to be quite interesting.

However, we notice that the security of OMD relies on the assumption that implementations always ensure *correct* use of nonce (a.k.a. message number); namely that, the nonce never gets repeated, otherwise security will fully collapse. While the nonce-based security is sufficient and desirable in many situations, it is not rare that in practice nonces are misused due to poor or erroneous implementations; e.g., a random IV with bad randomness generator might be used instead of the nonce, a counter with a short cycle of repetition can be used as a nonce, or the nonce can even be set to a constant.

Providing robustness against such nonce-misuse scenarios has motivated development of *nonce-misuse-resistant AE* schemes—an AE scheme, that retains

most of its security even if the nonces are not used properly. There are two different categories of such schemes with different levels of robustness.

The first is the category of two-pass schemes that can provide maximal security in the presence of nonce reuse. These schemes make a first pass over all data (message and authenticated data) to compute a tag (or IV) and then uses the result (IV) to parametrize a second pass for encryption. The first such (two-pass) scheme is the synthetic-IV (SIV) construction described in [24, 25]; other examples are HBS [15] and BTM [14]. When the nonce is reused, these two-pass schemes only leak minimal additional information compared to semantically secure encryption schemes—the leaked information being the fact that a plaintext together with its associated data are exactly repeated.

The second category are the one-pass (online) AE schemes that promise some limited level of misuse resistance; the first such scheme is McOE [13], followed by several other designs, such as those in [1, 9]. Being online is considered as an advantage in many applications, but it must be noted that such online AE schemes will reveal much more information compared to the two-pass scheme; namely, the ciphertext reveals to the adversary whether two messages share a common prefix when the nonce is reused. This is intrinsic to deterministic online encryption. To the best of our knowledge, there is yet no clear consensus in the cryptographic community (and no systematic study) on whether such an extra information leakage (namely, the longest common prefix) can be tolerated and considered safe in different applications of an AE scheme.

Aiming to keep the good features of OMD as far as possible and making it robust to nonce reuse, we introduce two variants of OMD, called misuse-resistant OMD (MR-OMD) and parallelizable misuse-resistant OMD (PMR-OMD). We aim to reach maximal possible level of robustness against repeated nonces, as defined by Rogaway and Shrimpton (FSE 2006) under the name misuse-resistant AE (MRAE), so similar to the previously known schemes in this category (e.g., SIV, HBS and BTM) our constructions are also two-pass. The main motives that influenced design of MR-OMD are the struggle to have a construction that is very similar to OMD (so that common code and hardware can be reused) and to have an efficient, provably secure MRAE scheme at the same time. The design of PMR-OMD further deviates from OMD, providing a fully parallelizable variant, in contrast with OMD and MR-OMD which have serial encryption process.

In MR-OMD and PMR-OMD, the two passes are combined in a way that minimizes the incurred additional cost: using a keyed compression function with $(n + m)$ -bit input and n -bit output, for processing a message M with associated data A , MR-OMD and PMR-OMD only need $|M|/(n + m)$ more calls to the compression function compared to OMD, where $|M|$ is the bit length of M . Noticing that the encryption pass in OMD requires $1 + |M|/m$ compression function calls, and considering $m = n$ (as suggested in OMD), the overhead incurred by the second pass in our two-pass variants is about 50% of the encryption time for OMD. We note that the *overhead* is independent of A as it is processed in the same way in all these algorithms.

Compared with SIV which requires two keys, MR-OMD and PMR-OMD only uses a single key (as is also the case for HBS and BTM). Compared to HBS and BTM which use polynomial-based hashing and need general finite field multiplications in their IV generation part, MR-OMD and PMR-OMD use compression function-based hashing process and only need doubling (multiplication by 2) operation in $\text{GF}(2^n)$ which can be easily and efficiently implemented as shown in Section 2. Avoiding polynomial based hashing seems to be an advisable practice due to the recent attacks and issues of such schemes as recently described in [20, 26]. We note that all these two-pass schemes have the same high-level generic structure (called “Scheme A4” in [19]); what differs is the design of the IV generation and encryption processes.

There is also another subtle difference between the design of our variants of OMD with those of SIV, HBS and BTM; namely, while the latter schemes are designed to be deterministic AE (DAE) and incorporate nonce (if used) and associated data as the header information, our schemes treat the nonce and associated data differently from the beginning. As stated by Rogaway and Shrimpton [24] “the MRAE goal is conceptually different from the DAE goal, the former employing an IV and gaining for this a stronger notion of security. The header and the IV are conceptually different, the one being user-supplied data that the user wants authenticated, the other being a mechanism-supplied value needed to obtain a strong notion of security.”

ORGANIZATION OF THE PAPER. Notations and preliminary concepts are presented in Section 2. Security notions are defined in Section 3. Section 4 provides the specification of the MR-OMD mode of operation. In Section 5, we provide the security analysis of MR-OMD. In Section 6 we describe PMR-OMD and provide its security analysis. The bulk parts of the proofs are provided in the appendices.

2 Preliminaries

NOTATIONS. For a finite set \mathcal{S} , by $x \xleftarrow{\$} S$ we denote that x is chosen from S uniformly at random. Any string is a binary string. Let $\{0, 1\}^n$ denote set of all binary strings of length n and let $\{0, 1\}^*$ denote the set of all finite-length strings. For two strings X and Y , $X||Y$ and XY denote the result of concatenating the two strings. For an n -bit binary string $X = X[n-1] \cdots X[0]$, let $X[i \cdots j] = X[i] \cdots X[j]$ denote a substring of X , for $0 \leq j \leq i \leq n-1$; let $\text{msb}(X) = X[n-1]$ and $\text{lsb}(X) = X[0]$. Let $1^n 0^m$ denote concatenation of n ones by m zeros. For a non-negative integer i let $\langle i \rangle_m$ denote binary representation of i by an m -bit string.

The special symbol \perp means that a variable is undefined and it also signifies an error. Let $|Z|$ denote the cardinality of Z if Z is a set, and the length of Z in bits if Z is a binary string. The empty string is denoted by ε and we let $|\varepsilon| = 0$. For $X \in \{0, 1\}^*$ let $X_1 || X_2 \cdots || X_m \xleftarrow{b} X$ denote partitioning X into blocks X_i such that $|X_i| = b$ for $1 \leq i \leq m-1$ and $|X_m| \leq b$. Let $|X|_b = \lceil |X|/b \rceil$ denote length of X in b -bit blocks and let $\|X\|_b = \max\{1, |X|_b\}$.

For two binary strings $X = X[m-1] \cdots X[0]$ and $Y = Y[n-1] \cdots Y[0]$, the notation $X \oplus Y$ denotes bitwise xor of $X[m-1] \cdots X[m-1-\ell]$ and $Y[n-1] \cdots Y[n-1-\ell]$ where $\ell = \min\{m-1, n-1\}$. That is, $X \oplus Y$ is the result of xoring first ℓ msb bits of X and Y and dropping the rest (if any) for the longer string. When $m = n$, this simply denotes the conventional bitwise xor of two strings. For any string X , define $X \oplus \varepsilon = \varepsilon \oplus X = \varepsilon$. The notation $X \oplus_{msb} Y$ stands for bitwise xor $X \parallel 0^{L-m} \oplus Y \parallel 0^{L-n}$, where $L = \max\{m, n\}$. In other words, we xor the shorter string to the longer one, aligning the strings by their leftmost bits.

For a binary string $X = X[m-1] \cdots X[0]$, let $X \ll n$ denote the left-shift operation, where the n left-most bits are discarded and the n vacated right bits are set to 0. We let $X \gg_s n$ denote the *signed* right-shift operation, where the n right-most bits are discarded and the n vacated left bits are filled with the left-most bit (which is considered as the sign bit); for example, $1001100 \gg_s 3 = 1111001$.

THE FINITE FIELD WITH 2^n POINTS. Let $(GF(2^n), \oplus, \cdot)$ denote the Galois Field with 2^n points. When considering a point α in $GF(2^n)$ it can be represented in any of the following equivalent ways: (1) as an integer between 0 and $2^n - 1$, (2) as a binary string $\alpha_{n-1} \cdots \alpha_0 \in \{0, 1\}^n$, or (3) as a formal polynomial $\alpha(X) = \alpha_{n-1}X^{n-1} + \cdots + \alpha_1X + \alpha_0$ with binary coefficients. The addition “ \oplus ” and multiplication “ \cdot ” of two field elements in $GF(2^n)$ are defined as usual (e.g. see [23]). For $GF(2^{256})$, we use $P_{256}(X) = X^{256} + X^{10} + X^5 + X^2 + 1$, and for $GF(2^{512})$ we use $P_{512}(X) = X^{512} + X^8 + X^5 + X^2 + 1$ as the irreducible polynomials used in the field multiplications. It is easy to multiply an arbitrary field element α by the element 2 (i.e. X). For example, in $GF(2^{256})$ using $P_{256}(X)$ the doubling operation can be described as follows:

$$2 \cdot \alpha = \begin{cases} \alpha \ll 1 & \text{if } \text{msb}(\alpha) = 0 \\ (\alpha \ll 1) \oplus 0^{245}10000100101 & \text{if } \text{msb}(\alpha) = 1 \end{cases} \quad (1)$$

$$= (\alpha \ll 1) \oplus ((\alpha \gg_s 255) \wedge 0^{245}10000100101) \quad (2)$$

We note that the results computed in (1) and (2) are the same but an implementation using (2) will not be susceptible to the timing attacks unlike one which uses (1).

3 Security definitions

As usual in the concrete-security definitions, we measure the insecurity of a scheme Π using the resource parametrized function $\mathbf{Adv}_{\Pi}^{\text{xxx}}(\mathbf{r})$, denoting the maximal value of the adversary’s advantage — $\mathbf{Adv}_{\Pi}^{\text{xxx}}(\mathbf{r}) = \max_{\mathbf{A}} \{\mathbf{Adv}_{\Pi}^{\text{xxx}}(\mathbf{A})\}$ — over all adversaries \mathbf{A} , against the xxx property of a primitive or scheme Π , that use resources bounded by \mathbf{r} . Let \mathbf{A} be an adversary that returns a binary value; by $\mathbf{A}^{f(\cdot)}(X) \Rightarrow 1$ we refer to the event that \mathbf{A} on input X and access to an oracle function $f(\cdot)$ returns 1.

SYNTAX AND SECURITY OF KEYED COMPRESSION FUNCTIONS. We denote a keyed compression function by $F : \mathcal{K} \times (\{0, 1\}^n \times \{0, 1\}^m) \rightarrow \{0, 1\}^n$, where m and n are two positive integers, and the key space \mathcal{K} is a non-empty set of strings. The notations $F_K(H, M) = F(K; H, M)$ are equivalent. We can alternatively think of F_K as a single argument function whose domain is $\{0, 1\}^{n+m}$ and write $F_K(H||M) = F_K(H, M)$. Given a keyless compression function $F' : \{0, 1\}^n \times \{0, 1\}^b \rightarrow \{0, 1\}^n$ (e.g. **sha-256** : $\{0, 1\}^{256} \times \{0, 1\}^{512} \rightarrow \{0, 1\}^{256}$) we convert it to a keyed compression function F by dedicating k bits of its b -bit input block to the secret key; i.e. we define $F_K(H, M) = F'(H, K||M)$. For example in the case of **sha-256** we have $n = 256$ and we will set $k = 256$ which will give us $m = 512 - k = 256$. We assess the security of compression functions in the sense of pseudorandom function security described below.

PSEUDORANDOM FUNCTIONS (PRFs) AND TWEAKABLE PRFs. We denote by $\text{Func}(m, n) = \{f : \{0, 1\}^m \rightarrow \{0, 1\}^n\}$ the set of all functions from m -bit strings to n -bit strings and by $\text{Func}(\mathcal{M}, n) = \{f : \mathcal{M} \rightarrow \{0, 1\}^n\}$ the set of all functions from a set \mathcal{M} to n -bit strings. A random function $R \xleftarrow{\$} \text{Func}(m, n)$ is a function selected uniformly at random from $\text{Func}(m, n)$. We define a random function R' with input from set \mathcal{M} and n -bit output in a similar manner.

Let $\text{Func}^{\mathcal{T}}(m, n)$ be the set of all functions $\{\tilde{f} : \mathcal{T} \times \{0, 1\}^m \rightarrow \{0, 1\}^n\}$, where \mathcal{T} is a set of tweaks. A tweakable random function (RF) with the tweak space \mathcal{T} , m -bit input and n -bit output is a map $\tilde{R} : \mathcal{T} \times \{0, 1\}^m \rightarrow \{0, 1\}^n$ selected uniformly at random from $\text{Func}^{\mathcal{T}}(m, n)$; i.e. $\tilde{R} \xleftarrow{\$} \text{Func}^{\mathcal{T}}(m, n)$. We use $\tilde{R}^{(T)}(\cdot)$ and $\tilde{R}(T, \cdot)$ interchangeably, for every $T \in \mathcal{T}$. Notice that each tweak T names a random function $\tilde{R}^{(T)} : \{0, 1\}^m \rightarrow \{0, 1\}^n$ and distinct tweaks name distinct (independent) random functions.

Let $F : \mathcal{K} \times \{0, 1\}^m \rightarrow \{0, 1\}^n$ be a keyed function and let $\tilde{F} : \mathcal{K} \times \mathcal{T} \times \{0, 1\}^m \rightarrow \{0, 1\}^n$ be a keyed and tweakable function, where the key space \mathcal{K} is some nonempty set. Let $F_K(\cdot) = F(K, \cdot)$ and $\tilde{F}_K^{(T)}(\cdot) = \tilde{F}(K, T, \cdot)$. Let \mathbf{A} be an adversary. Then:

$$\begin{aligned} \mathbf{Adv}_F^{\text{prf}}(\mathbf{A}) &= \Pr \left[K \xleftarrow{\$} \mathcal{K} : \mathbf{A}^{F_K(\cdot)} \Rightarrow 1 \right] - \Pr \left[R \xleftarrow{\$} \text{Func}(m, n) : \mathbf{A}^{R(\cdot)} \Rightarrow 1 \right] \\ \mathbf{Adv}_{\tilde{F}}^{\text{prf}}(\mathbf{A}) &= \Pr \left[K \xleftarrow{\$} \mathcal{K} : \mathbf{A}^{\tilde{F}_K^{(\cdot)}(\cdot)} \Rightarrow 1 \right] - \Pr \left[\tilde{R} \xleftarrow{\$} \text{Func}^{\mathcal{T}}(m, n) : \mathbf{A}^{\tilde{R}^{(\cdot)}(\cdot)} \Rightarrow 1 \right] \end{aligned}$$

The resource parametrized advantage functions are defined accordingly, considering that the adversarial resources of interest here are the time complexity (t) of the adversary and the total number of queries (q) asked by the adversary (note that we just consider fixed-input-length functions, so the lengths of queries are fixed and known). We say that F is $(t, q; \epsilon)$ -PRF if $\mathbf{Adv}_F^{\text{prf}}(t, q) \leq \epsilon$. We say that \tilde{F} is $(t, q; \epsilon)$ -tweakable PRF if $\mathbf{Adv}_{\tilde{F}}^{\text{prf}}(t, q) \leq \epsilon$.

Extending these definitions for variable-input-length functions is straightforward; namely, for a VIL function $G : \mathcal{K} \times \mathcal{D} \rightarrow \{0, 1\}^n$, with a non-empty key

space \mathcal{K} and message space $\mathcal{D} = \{0, 1\}^*$, the ideal primitive to which a randomly selected function G_K is compared will be $R \stackrel{\$}{\leftarrow} \text{Func}(\mathcal{D}, n)$. The resource of interest in this case is the total length of all processed queries in n -bit blocks σ for some positive n .

IV-BASED ENCRYPTION SCHEMES. An IV-based encryption scheme is a privacy-only scheme, with a rather weak security notion described below, as for example the CBC mode. We say that an encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is an IV-based encryption scheme if the encryption function \mathcal{E} takes a tuple (K, IV, M) as input, such that $K \in \mathcal{K}$, $\text{IV} \in \{0, 1\}^\tau$ for some fixed positive τ and $M \in \{0, 1\}^*$. We call IV the initialization vector. The notations $\mathcal{E}(K, \text{IV}, M)$, $\mathcal{E}_K(\text{IV}, M)$ and $\mathcal{E}_K^{\text{IV}}(M)$ are used interchangeably. We also assume that if $\mathbb{C} = \mathcal{E}_K^{\text{IV}}(M)$, then we have $|\mathbb{C}| = |M| + \tau$ and $\mathbb{C} = \text{IV}||C$; i.e. the ciphertext reveals IV . We define the advantage of an adversary \mathbf{A} in breaking the $\$$ -privacy of Π as

$$\text{Adv}_\Pi^{\text{priv}\$}(\mathbf{A}) = \Pr \left[K \stackrel{\$}{\leftarrow} \mathcal{K} : \mathbf{A}^{\mathcal{E}_K^{\$}(\cdot)} \Rightarrow 1 \right] - \Pr \left[\mathbf{A}^{\$(\cdot)} \Rightarrow 1 \right]$$

with $\$(\cdot)$ being a random string oracle that on input M returns a random string of length $|M| + \tau$ and $\mathcal{E}_K^{\$}$ returning $\mathcal{E}_K^{\text{IV}}$ with $\text{IV} \stackrel{\$}{\leftarrow} \{0, 1\}^\tau$. It is assumed, that the adversary never asks a query outside the proper message space of Π . Note that in the $\text{priv}\$$ security game, the IV is chosen by the challenger. We remark that we make use of an IV-based scheme as a building block for our misuse-resistant scheme.

SYNTAX OF AN AEAD SCHEME. A nonce-based authenticated encryption with associated data, AEAD for short, is a symmetric key scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. The key space \mathcal{K} is some non-empty finite set. The encryption algorithm $\mathcal{E} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M} \rightarrow \mathcal{C} \cup \{\perp\}$ takes four arguments, a secret key $K \in \mathcal{K}$, a nonce $N \in \mathcal{N}$, an associated data (a.k.a. header data) $A \in \mathcal{A}$ and a message $M \in \mathcal{M}$, and returns either a ciphertext $\mathbb{C} \in \mathcal{C}$ or a special symbol \perp indicating an error. The decryption algorithm $\mathcal{D} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \rightarrow \mathcal{M} \cup \{\perp\}$ takes four arguments (K, N, A, \mathbb{C}) and either outputs a message $M \in \mathcal{M}$ or an error indicator \perp .

For correctness of the scheme, it is required that $\mathcal{D}(K, N, A, \mathbb{C}) = M$ for any \mathbb{C} such that $\mathbb{C} = \mathcal{E}(K, N, A, M)$. It is also assumed that if algorithms \mathcal{E} and \mathcal{D} receive parameter not belonging to their specified domain of arguments they will output \perp . We write $\mathcal{E}_K(N, A, M) = \mathcal{E}(K, N, A, M)$ and similarly $\mathcal{D}_K(N, A, \mathbb{C}) = \mathcal{D}(K, N, A, \mathbb{C})$.

We assume that the message and associated data can be any binary string of arbitrary but finite length; i.e. $\mathcal{M} = \{0, 1\}^*$ and $\mathcal{A} = \{0, 1\}^*$, but the key and nonce are some fixed-length binary strings, i.e. $\mathcal{N} = \{0, 1\}^{|N|}$ and $\mathcal{K} = \{0, 1\}^k$, where the positive integers $|N|$ and k are respectively the nonce length and the key length of the scheme in bits. We assume that $|\mathcal{E}_K(N, A, M)| = |M| + \tau$ for some positive fixed constant τ ; that is, we will have $\text{IV}||C = \mathbb{C}$ where $|C| = |M|$ and $|\text{IV}| = \tau$. We call C the core ciphertext and IV the initialization vector (or IV for short). The IV is not to be confused with the nonce. The nonce is an input to the encryption algorithm of the AEAD scheme and it is used

to randomize the encryption, while the role of what we call IV in this paper is to authenticate a message with associated data and randomize *one part* of the encryption algorithm in such a way, that will ensure the misuse-resistant property. The IV here is generated by the encryption algorithm itself. It can be viewed as a variant of an authentication tag, with the difference that the IV is prepended rather than appended to the core ciphertext.

NONCE RESPECTING AND NONCE MISUSING ADVERSARIES. We say that an adversary \mathbf{A} is nonce-respecting if it never repeats a nonce in its *encryption* queries. That is, if \mathbf{A} queries the encryption oracle $\mathcal{E}_K(\cdot, \cdot, \cdot)$ with the queries $(N^1, A^1, M^1) \cdots (N^q, A^q, M^q)$ then N^1, \dots, N^q must be distinct. If there are at least two queries (N^i, A^i, M^i) and (N^j, A^j, M^j) that share the same nonce, i.e. $N^i = N^j$, then we say that \mathbf{A} is a nonce-misusing (or a nonce-reusing) adversary. Note that adversaries of both types may repeat nonces in their decryption queries.

AE SECURITY. To establish the security of MR-OMD scheme, we use the all-in-one MRAE security notion introduced in [25]. As shown in [25], the all-in-one security notion is equivalent to the conventional two-requirement security notion (that combines IND-CPA for privacy and INT-CTXT for integrity), as put forth in [2, 3, 16].

Definition 1. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a nonce based AEAD scheme. The MRAE-advantage of an adversary \mathbf{A} in attacking the scheme Π is defined as:

$$\mathbf{Adv}_{\Pi}^{\text{mrae}}(\mathbf{A}) = \Pr \left[K \stackrel{\$}{\leftarrow} \mathcal{K} : \mathbf{A}^{\mathcal{E}_K(\cdot, \cdot, \cdot), \mathcal{D}_K(\cdot, \cdot, \cdot)} \Rightarrow 1 \right] - \Pr \left[\mathbf{A}^{\mathcal{S}(\cdot, \cdot, \cdot), \perp(\cdot, \cdot, \cdot)} \Rightarrow 1 \right].$$

To prevent trivial wins, we forbid \mathbf{A} to ask a query (N, A, \mathbb{C}) of the decryption oracle, after obtaining result \mathbb{C} upon query (N, A, M) from the encryption oracle; we also assume that \mathbf{A} never repeats an encryption query (N, A, M) . On query (N, A, M) , the random-bit oracle $\mathcal{S}(\cdot, \cdot, \cdot)$ returns a random string of length $|M| + \tau$ if the inputs N, A and M belong to the respective input domains and \perp otherwise. The $\perp(\cdot, \cdot, \cdot)$ oracle returns \perp on every query.

The resource-based advantage function $\mathbf{Adv}_{\Pi}^{\text{mrae}}(\mathbf{r})$ is parametrized by adversarial resource vector $\mathbf{r} = (t, \sigma_A, \sigma_M, q_e, q_d)$ where t denotes the time complexity, $\sigma_A = (\sum_{i=1}^{q_e} |A^i| + \sum_{j=1}^{q_d} |A^j|)$ is the total length of associated data in all queries, $\sigma_M = (\sum_{i=1}^{q_e} |M^i| + \sum_{j=1}^{q_d} |\mathbb{C}^j - \tau|)$ is the total length of plaintexts in all queries, q_e denotes the maximal number of encryption queries and q_d the maximal number of decryption queries made by the adversary.

Clearly, the MRAE security notion implies the nonce-respecting security; the latter being a special case of the former, where adversary cannot repeat the nonce and hence no query to the encryption oracle is repeated. We denote the conventional nonce-respecting notion by “nr-ae” and let the corresponding resource-parametrized advantage function be $\mathbf{Adv}_{\Pi}^{\text{nr-ae}}(\mathbf{r})$, measuring the maximal insecurity over all “nonce-respecting” adversaries \mathbf{A} having resources bounded by \mathbf{r} .

If an adversary repeats a whole query to the encryption oracle (i.e. queries (N, A, M) twice) then it is impossible to have any IND-CPA privacy, as the ciphertext will reveal this repetition. This is the minimal amount of additional information leakage by a MRAE scheme compared to a semantically secure scheme and a scheme fulfilling Definition 1 is considered as maximally robust to nonce-reuse. Note that such trivial winning strategy is disallowed in Definition 1.

We sometimes use simplified notation for adversary's oracles and the choice of the key in a security game. For a scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, the notations $K \xleftarrow{\$} \mathcal{K} : \mathbf{A}^{\mathcal{E}_{\mathcal{K}}(\cdot, \cdot), \mathcal{D}_{\mathcal{K}}(\cdot, \cdot)}$ and $\mathbf{A}^{\Pi_{\mathcal{K}}(\cdot, \cdot), \Pi_{\mathcal{K}}^{-1}(\cdot, \cdot)}$ are equivalent.

4 Specification of MR-OMD

MR-OMD is a compression function mode of operation for nonce-misuse resistant AEAD. It has the following parameters.

- keyed compression function $F : \mathcal{K} \times (\{0, 1\}^n \times \{0, 1\}^m) \rightarrow \{0, 1\}^n$
- IV length $\tau < n$

where the key space $\mathcal{K} = \{0, 1\}^k$ and $m \leq n$.

Let MR-OMD- F denote the MR-OMD mode of operation using a keyed compression function $F_K : \{0, 1\}^n \times \{0, 1\}^m \rightarrow \{0, 1\}^n$ with $m \leq n$ and an unspecified tag length. We let MR-OMD[F, τ] denote the MR-OMD mode of operation using the keyed compression function F_K and the IV of length τ . The encryption algorithm of MR-OMD[F, τ] takes four input arguments (secret key $K \in \{0, 1\}^k$, nonce $N \in \{0, 1\}^{|N|}$, associated data $A \in \{0, 1\}^*$, message $M \in \{0, 1\}^*$) and outputs $\mathbb{C} = \text{IV} || C \in \{0, 1\}^{|M|+\tau}$. The decryption algorithm of MR-OMD[F, τ] inputs four arguments (secret key $K \in \{0, 1\}^k$, nonce $N \in \{0, 1\}^{|N|}$, associated data $A \in \{0, 1\}^*$, ciphertext $\text{IV} || C \in \{0, 1\}^*$) and either outputs the whole $M \in \{0, 1\}^{|C|}$ at once or an error message \perp .

A schematic representation of the encryption algorithm of MR-OMD[F, τ] is shown in figure 1. The construction of the decryption algorithm is similar to the encryption except that the ciphertext is first decrypted using IV from the input and then the IV from input is compared to IV' computed over the associated data and decrypted message. Figure 2 shows the algorithmic description of the encryption and decryption algorithms of MR-OMD[F, τ]

COMPUTING THE MASKING VALUES. As seen from the description of MR-OMD in Figure 1, before each call to the underlying keyed compression function, we xor a masking value Δ . Seven different sets of masking values are used:

- masks $\Delta_{N,i,j}$ for $j \in \{0, \dots, 5\}$ are used in the IV generation process,
- masks $\Delta_{V,i}$ are used in the encrypt/decryption process.

In the following, all multiplications are in $GF(2^n)$, $\text{ntz}(i)$ denotes the number of trailing zeros (i.e. the number of rightmost bits that are zero) in the binary representation of a positive integer i .

Initialization. Let $L_* = F_K(0^n, 0^m)$; $L(0) = 8 \cdot L_*$, and $L(i) = 2 \cdot L(i-1)$ for $i \geq 1$. The rule to compute $L(i)$ is described as a part of the initialization, because these values can be precomputed and stored in a table. Further on let $\Delta_{N,0,0} = F_K(N||10^{n-1-|N|}, 0^m)$; $\Delta_{N,0,1} = F_K(N||10^{n-1-|N|}, 0^m) \oplus L_*$.

Masking sequence for IV generation. For $i \geq 1$ let $\Delta_{N,i,0} = \Delta_{N,i-1,0} \oplus L(\text{ntz}(i))$; and $\Delta_{N,i,1} = \Delta_{N,i-1,1} \oplus L(\text{ntz}(i))$. For $i \geq 1$ and $j, j' \in \{0, \dots, 5\}$: $\Delta_{N,i,j} = \Delta_{N,i,j'} \oplus (\langle j \rangle_n \oplus \langle j' \rangle_n) \cdot L_*$.

Masking sequence for encryption. Let $\bar{\Delta}_{IV,0} = F_K(IV||10^{n-1-\tau}, 0^m) \oplus 6 \cdot L_*$. We have $\bar{\Delta}_{IV,i} = \bar{\Delta}_{IV,i-1} \oplus L(\text{ntz}(i))$ for $i \geq 1$.

5 Security Analysis

We analyse the security of MR-OMD in two cases: (1) as a MRAE, considering adversaries that are nonce-reusing; (2) in the case that adversaries are nonce-respecting. As MR-OMD is designed as a nonce-misuse resistant scheme, we first focus on analysing the security bounds in the nonce-misuse scenario. Clearly, an upper-bound for the MRAE insecurity (i.e. MRAE advantage) also upper-bounds the insecurity in the nonce-respecting case. Intuitively, the latter can be lower than the former. This is made explicit by Theorem 1 and Theorem 2.

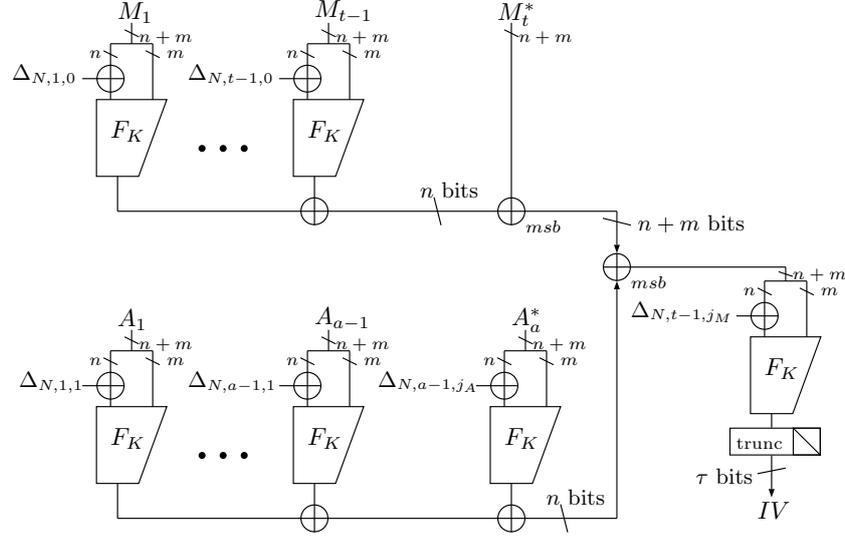
5.1 Security in the Case of Nonce Misuse

Theorem 1 gives the MRAE security of MR-OMD. The high-level structure of the proof is similar to those of previous MRAE schemes following the synthetic-IV (SIV) design paradigm [24], such as HBS [15] and BTM [14], but the details differ. We first prove the security in the information-theoretic setting using tweakable random functions. To obtain the information-theoretic security, we prove security of MR-OMD.HASH as a PRF and that of MR-OMD. \mathcal{E} as a secure IV-based encryption scheme. Consequently, we prove security of MR-OMD in the MR-AE sense using the previous two results. A complexity-theoretic security bound is then determined by instantiating the tweakable random functions using the XE construction from [22].

Theorem 1. *Fix $n \geq 1$ and $\tau \in \{0, 1, \dots, n\}$. Let $F : \mathcal{K} \times (\{0, 1\}^n \times \{0, 1\}^m) \rightarrow \{0, 1\}^n$ be a PRF, where the key space $\mathcal{K} = \{0, 1\}^k$ for $k \geq 1$ and $1 \leq m \leq n$. Then*

$$\mathbf{Adv}_{\text{MR-OMD}[F,\tau]}^{\text{mrae}}(t, \sigma, q_e, q_d) \leq \mathbf{Adv}_F^{\text{prf}}(t', 2\sigma) + \frac{3.5\sigma^2}{2^n} + \frac{0.5q_e^2}{2^\tau} + \frac{q_d}{2^\tau}$$

where q_e and q_d are, respectively, the number of encryption and decryption queries, $t' = t + cn\sigma$ for some constant c and σ is the total number of calls made to the underlying compression function F .



If $|M_t| < n + m$ set $M_t^* = M_t || 10^{n+m-1-|M_t|}$ and $j_M = 4$. Otherwise $M_t^* = M_t$, $j_M = 2$.
 If $|A_a| < n + m$ set $A_a^* = A_a || 10^{n+m-1-|A_a|}$ and $j_A = 5$. Otherwise $A_a^* = A_a$, $j_A = 3$.

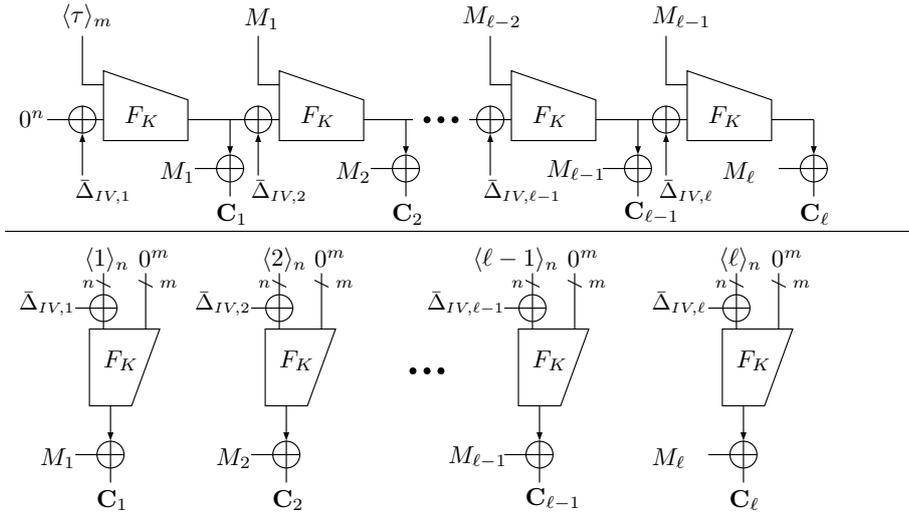


Fig. 1: The encryption process of MR-OMD $[F, \tau]$ and PMR-OMD $[F, \tau]$ using a keyed compression function $F_K : (\{0, 1\}^n \times \{0, 1\}^m) \rightarrow \{0, 1\}^n$ with $m \leq n$. **(Top)** The process of generating the IV. Both associated data and message are parsed into $n + m$ bit blocks and padded if needed as shown. **(Bottom)** The encryption process (upper part for MR-OMD and lower for PMR-OMD). The output ciphertext is $IV || C$. For operations \oplus and \oplus_{msb} see our convention in Section 2.

<pre> 1: Algorithm INITIALIZE(K) 2: $L_* \leftarrow F_K(0^n, 0^m)$ 3: $L_*^{(2)} \leftarrow 2 \cdot L_*$ 4: $L_*^{(4)} \leftarrow 2 \cdot L_*^{(2)}$ 5: $L_*^{(6)} \leftarrow L_*^{(4)} \oplus L_*^{(2)}$ 6: $L(0) \leftarrow 2 \cdot L_*^{(4)}$ 7: for $i \leftarrow 1, 2, \dots$ do 8: $L(i) = 2 \cdot L(i-1)$ 9: return 1: Algorithm HASH$_K(N, A, M)$ 2: $b \leftarrow n + m$ 3: $A_1 A_2 \dots A_{a-1} A_a \stackrel{b}{\leftarrow} A$ 4: $M_1 M_2 \dots M_{\ell-1} M_\ell \stackrel{b}{\leftarrow} M$ 5: $\Sigma_A \leftarrow 0^n; \Sigma_M \leftarrow 0^n$ 6: $\Delta_M \leftarrow F_K(N 10^{n-1- N }, 0^m)$ 7: $\Delta_A \leftarrow \Delta_M \oplus L_*$ 8: for $i \leftarrow 1$ to $a-1$ do 9: $\Delta_A \leftarrow \Delta_A \oplus L(\text{ntz}(i))$ 10: $\text{Left} \leftarrow A_i[b-1 \dots m]$ 11: $\text{Right} \leftarrow A_i[m-1 \dots 0]$ 12: $\Sigma_A \leftarrow \Sigma_A \oplus F_K(\text{Left} \oplus \Delta_A, \text{Right})$ 13: if $A_a = b$ then 14: $\Delta_A \leftarrow \Delta_A \oplus L_*^{(2)}$ 15: $\text{Left} \leftarrow A_a[b-1 \dots m]$ 16: $\text{Right} \leftarrow A_a[m-1 \dots 0]$ 17: $\Sigma_A \leftarrow \Sigma_A \oplus F_K(\text{Left} \oplus \Delta, \text{Right})$ 18: else if $A > 0$ then 19: $\Delta_A \leftarrow \Delta_A \oplus L_*^{(4)}$ 20: $A_a^* \leftarrow A_a 10^{b- A_a -1}$ 21: $\text{Left} A_a^*[b-1 \dots m]$ 22: $\text{Right} \leftarrow A_a^*[m-1 \dots 0]$ 23: $\Sigma_A \leftarrow \Sigma_A \oplus F_K(\text{Left} \oplus \Delta_A, \text{Right})$ 24: for $i \leftarrow 1$ to $t-1$ do 25: $\Delta_M \leftarrow \Delta_M \oplus L(\text{ntz}(i))$ 26: $\text{Left} \leftarrow M_i[b-1 \dots m]$ 27: $\text{Right} \leftarrow M_i[m-1 \dots 0]$ 28: $\Sigma_M \leftarrow \Sigma_M \oplus F_K(\text{Left} \oplus \Delta_M, \text{Right})$ 29: if $M_t = b$ then 30: $\Delta_M \leftarrow \Delta_M \oplus L_*^{(2)}$ 31: $\text{Left} \leftarrow M_t[b-1 \dots m] \oplus \Sigma_A$ 32: $\text{Right} \leftarrow M_t[m-1 \dots 0]$ 33: $\text{IV} \leftarrow F_K(\text{Left} \oplus \Delta_M, \text{Right})$ </pre>	<pre> 34: else 35: $\Delta_M \leftarrow \Delta_M \oplus L_*^{(4)}$ 36: $M_t^* \leftarrow M_t 10^{b-1-M_t -1}$ 37: $\text{Left} \leftarrow M_t^*[b-1 \dots m] \oplus \Sigma_A$ 38: $\text{Right} \leftarrow M_t^*[m-1 \dots 0]$ 39: $\text{IV} \leftarrow F_K(\text{Left} \oplus \Delta_M, \text{Right})$ 40: return $\text{IV}[n-1 \dots n-\tau]$ 1: Algorithm $\mathcal{E}_K(N, A, M)$ 2: if $N > n-1$ then 3: return \perp 4: $M_1 M_2 \dots M_{\ell-1} M_\ell \stackrel{m}{\leftarrow} M$ 5: $\text{IV} \leftarrow \text{HASH}_K(N, A, M)$ 6: $\Delta \leftarrow F_K(\text{IV} 10^{n-1-\tau}, 0^m)$ 7: $\Delta \leftarrow \Delta \oplus L(0) \oplus L_*^{(6)}$ 8: $H \leftarrow 0^n$ 9: $H \leftarrow F_K(H \oplus \Delta, \langle \tau \rangle_m)$ 10: for $i \leftarrow 1$ to $\ell-1$ do 11: $C_i \leftarrow H \oplus M_i$ 12: $\Delta \leftarrow \Delta \oplus L(\text{ntz}(i+1))$ 13: $H \leftarrow F_K(H \oplus \Delta, M_i)$ 14: $C_\ell \leftarrow H \oplus M_\ell$ 15: $\mathbb{C} \leftarrow \text{IV} C_1 C_2 \dots C_\ell$ 16: return \mathbb{C} 1: Algorithm $\mathcal{D}_K(N, A, \mathbb{C})$ 2: if $N > n-1$ or $\mathbb{C} < \tau$ then 3: return \perp 4: $\text{IV} C_1 C_2 \dots C_{\ell-1} C_\ell \stackrel{m}{\leftarrow} \mathbb{C}$ 5: $H \leftarrow 0^n$ 6: $\Delta \leftarrow F_K(\text{IV} 10^{n-1-\tau}, 0^m)$ 7: $\Delta \leftarrow \Delta \oplus L(0) \oplus L_*^{(6)}$ 8: $H \leftarrow F_K(H \oplus \Delta, \langle \tau \rangle_m)$ 9: for $i \leftarrow 1$ to $\ell-1$ do 10: $M_i \leftarrow H \oplus C_i$ 11: $\Delta \leftarrow \Delta \oplus L(\text{ntz}(i+1))$ 12: $H \leftarrow F_K(H \oplus \Delta, M_i)$ 13: $M_\ell \leftarrow H \oplus C_\ell$ 14: $\text{IV}' \leftarrow \text{HASH}_K(N, A, M)$ 15: if $\text{IV}' = \text{IV}$ then 16: return $M \leftarrow M_1 M_2 \dots M_\ell$ 17: else 18: return \perp </pre>
--	--

Fig. 2: Definition of MR-OMD $[F, \tau]$. The function $F : \mathcal{K} \times (\{0, 1\}^n \times \{0, 1\}^m) \rightarrow \{0, 1\}^n$ is a keyed compression function with $\mathcal{K} = \{0, 1\}^k$ and $m \leq n$. The IV length is $\tau \in \{0, 1, \dots, n\}$. Algorithms \mathcal{E} and \mathcal{D} can be called with arguments $K \in \mathcal{K}$, $N \in \{0, 1\}^{\leq n-1}$, and $A, M, \mathbb{C} \in \{0, 1\}^*$.

Remark 1. We can verify that $\sigma = \lceil \sigma_A / (m+n) \rceil + \lceil \sigma_M / (m+n) \rceil + \lceil \sigma_M / (m) \rceil + \sum_{i=1}^{q_e} 1_{|M^i|=0} + \sum_{j=1}^{q_d} 1_{|C^j|=\tau} + q_e + q_d$.

Proof. The proof is obtained by combining Lemma 3, Lemma 1 and Lemma 2 in subsection 5.1.1 with Lemma 4 and Lemma 5 in subsection 5.1.2. \square

5.1.1 Generalization of MR-OMD based on Tweakable Random Functions We define the scheme $\text{MR-OMD}[\tilde{R}, \tau]$, a generalization of $\text{MR-OMD}[F, \tau]$ that uses a tweakable random function $\tilde{R} : \mathcal{T} \times (\{0, 1\}^n \times \{0, 1\}^m) \rightarrow \{0, 1\}^n$, as depicted in figure 3. The tweak space \mathcal{T} consists of seven mutually exclusive sets of tweaks; namely, $\mathcal{T} = \mathcal{N} \times \mathbb{N} \times \{0\} \cup \mathcal{N} \times \mathbb{N} \times \{1\} \cup \mathcal{N} \times \mathbb{N} \times \{2\} \cup \mathcal{N} \times \mathbb{N} \times \{3\} \cup \mathcal{N} \times \mathbb{N} \times \{4\} \cup \mathcal{N} \times \mathbb{N} \times \{5\} \cup \mathcal{IV} \times \mathbb{N}$, where $\mathcal{N} = \{0, 1\}^{|\mathcal{N}|}$ is the set of nonces, $\mathcal{IV} = \{0, 1\}^\tau$ is the set of IV-s and \mathbb{N} is the set of positive integers.

Lemma 1. *Let $\text{MR-OMD}[\tilde{R}, \tau]$ be the MR-OMD scheme that uses tweakable RF \tilde{R} . Then*

$$\text{Adv}_{\text{MR-OMD}[\tilde{R}, \tau].\text{HASH}}^{\text{prf}}(\sigma) \leq \frac{0.5\sigma^2}{2^n}$$

where $\sigma = \sum_{i=1}^q (|A^i|_{m+n} + |M^i|_{m+n})$ is the total number of calls to the underlying tweakable RF \tilde{R} in all q queries asked by a nonce-misusing adversary.

The proof of the lemma is provided in Appendix A.

Before we proceed, we have to introduce a new notation. The purpose of this notation is to make the security analysis better structured. Consider the encryption algorithm $\text{MR-OMD}[\tilde{R}, \tau].\mathcal{E}_K(N, A, M)$. The algorithm can be split into two parts. First, it computes $\text{IV} = \text{MR-OMD}[\tilde{R}, \tau].\text{HASH}_K(N, A, M)$. The second part comprises all the steps after computing the IV. We can formalize the second step as $\text{MR-OMD}[\tilde{R}, \tau].\bar{\mathcal{E}}_K(\text{IV}, M)$, so that, if we simplify the notation, we have

$$\mathcal{E}_K(N, A, M) = \bar{\mathcal{E}}_K(\text{HASH}_K(N, A, M), M).$$

We define $\text{MR-OMD}[\tilde{R}, \tau].\bar{\mathcal{D}}_K(\text{IV}, M)$ in a similar manner.

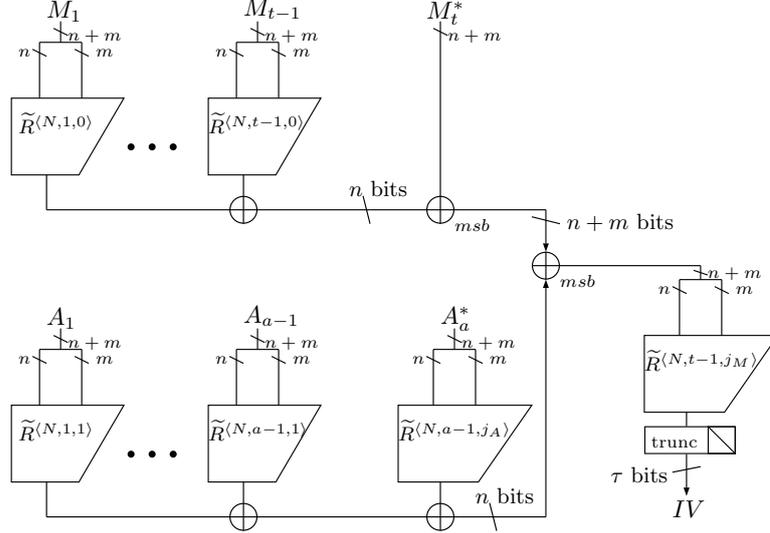
Lemma 2. *Let $\text{MR-OMD}[\tilde{R}, \tau]$ be the MR-OMD scheme that uses tweakable RF \tilde{R} . Then*

$$\text{Adv}_{\text{MR-OMD}[\tilde{R}, \tau].\bar{\mathcal{E}}}^{\text{priv}\$}(q_e) \leq \frac{0.5q_e^2}{2^\tau}$$

where q_e is the number of all encryption queries asked by the adversary.

The proof is provided in Appendix B.

Lemma 3. *Let $\text{MR-OMD}[\tilde{R}, \tau]$ be the MR-OMD scheme that uses tweakable RF \tilde{R} . Let \mathbf{A} be an MR-AE adversary attacking $\text{MR-OMD}[\tilde{R}, \tau]$. Let q_e be the number of encryption queries and q_d the number of decryption queries made by*



If $|M_t| < n + m$ set $M_t^* = M_t || 10^{n+m-1-|M_t|}$ and $j_M = 4$. Otherwise $M_t^* = M_t$, $j_M = 2$.
 If $|A_a| < n + m$ set $A_a^* = A_a || 10^{n+m-1-|A_a|}$ and $j_A = 5$. Otherwise $A_a^* = A_a$, $j_A = 3$.

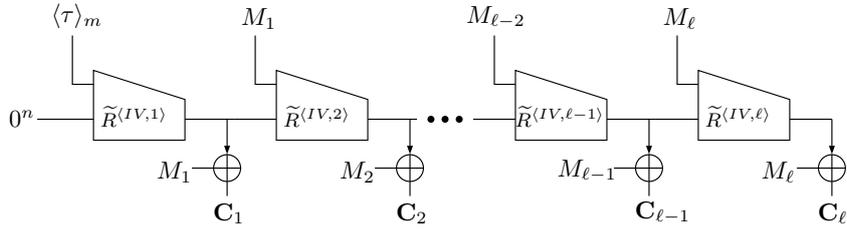


Fig. 3: The scheme $\text{MIR-OMD}[\tilde{R}, \tau]$ using a tweakable random function $\tilde{R} : \mathcal{T} \times (\{0, 1\}^n \times \{0, 1\}^m) \rightarrow \{0, 1\}^n$. **(Top)** The process of generating the IV. Both associated data and message are parsed into $n + m$ bit blocks. If the last block of the message M_t does not have length $n + m$ bits, it is padded as shown. Similar applies for associated data. **(Bottom)** The encryption process. The output ciphertext is $IV || C$. For operations \oplus and \oplus_{msb} see our convention in Section 2.

\mathbf{A} and let σ be the total number of calls to the underlying tweakable RF \tilde{R} in all \mathbf{A} 's queries. Then there exist adversaries \mathbf{E} and \mathbf{R} , such that

$$\mathbf{Adv}_{\text{MR-OMD}[\tilde{R},\tau].\text{HASH}}^{\text{prf}}(\mathbf{R}) + \mathbf{Adv}_{\text{MR-OMD}[\tilde{R},\tau].\mathcal{E}}^{\text{priv}\$}(\mathbf{E}) \geq \mathbf{Adv}_{\text{MR-OMD}[\tilde{R},\tau]}^{\text{mrae}}(\mathbf{A}) - \frac{q_d}{2\tau}$$

where \mathbf{E} asks at most q_e queries and \mathbf{R} asks at most $q = q_e + q_d$ queries in total. Both \mathbf{E} and \mathbf{R} are limited to a total number σ of calls to underlying tweakable RF \tilde{R} in all their queries.

Proof. For the sake of readability, we shall refer to $\text{MR-OMD}[\tilde{R},\tau]$ by Π throughout this proof. The proof proceeds in two steps, similarly as in [25].

In the first step, we start with the scheme $\bar{\Pi}$, which is the same as Π , except that we replace the algorithm $\Pi.\text{HASH}$ by $\text{Func}(\{0,1\}^{|N|} \times \{0,1\}^* \times \{0,1\}^*, \tau)$. To instantiate $\bar{\Pi}$, a tweakable RF \tilde{R} is picked for $\Pi.\bar{\mathcal{E}}$ and a RF $\rho \xleftarrow{\$} \text{Func}(\{0,1\}^{|N|} \times \{0,1\}^* \times \{0,1\}^*, \tau)$ is picked to instantiate the "HASH", so the key is now formed by \tilde{R}, ρ . Let $\bar{p} = \mathbf{Adv}_{\bar{\Pi}}^{\text{mr-ae}}(\mathbf{A})$ with unchanged limits on resources q_e, q_d, σ . We have

$$\begin{aligned} \bar{p} &= \Pr \left[\mathbf{A}_{\tilde{R},\rho}^{\bar{\Pi}_{\tilde{R},\rho}(\cdot,\cdot,\cdot), \bar{\Pi}_{\tilde{R},\rho}^{-1}(\cdot,\cdot,\cdot)} \Rightarrow 1 \right] - \Pr \left[\mathbf{A}_{\tilde{R},\rho}^{\mathcal{S}(\cdot,\cdot,\cdot), \perp(\cdot,\cdot,\cdot)} \Rightarrow 1 \right] \\ &= \bar{p}_1 + \bar{p}_2 \end{aligned}$$

with

$$\begin{aligned} \bar{p}_1 &= \Pr \left[\mathbf{A}_{\tilde{R},\rho}^{\bar{\Pi}_{\tilde{R},\rho}(\cdot,\cdot,\cdot), \bar{\Pi}_{\tilde{R},\rho}^{-1}(\cdot,\cdot,\cdot)} \Rightarrow 1 \right] - \Pr \left[\mathbf{A}_{\tilde{R},\rho}^{\bar{\Pi}_{\tilde{R},\rho}(\cdot,\cdot,\cdot), \perp(\cdot,\cdot,\cdot)} \Rightarrow 1 \right] \\ \bar{p}_2 &= \Pr \left[\mathbf{A}_{\tilde{R},\rho}^{\bar{\Pi}_{\tilde{R},\rho}(\cdot,\cdot,\cdot), \perp(\cdot,\cdot,\cdot)} \Rightarrow 1 \right] - \Pr \left[\mathbf{A}_{\tilde{R},\rho}^{\mathcal{S}(\cdot,\cdot,\cdot), \perp(\cdot,\cdot,\cdot)} \Rightarrow 1 \right] \end{aligned}$$

We proceed by bounding the terms \bar{p}_1 and \bar{p}_2 . To bound \bar{p}_2 , we construct an adversary \mathbf{E} for attacking the $\text{priv}\$$ security of $\bar{\Pi}.\bar{\mathcal{E}}$ from \mathbf{A} . \mathbf{E} is equipped with its own oracle $e(\cdot)$, which implements either $\bar{\Pi}.\bar{\mathcal{E}}$ or the random bits oracle. We let \mathbf{E} run \mathbf{A} . On \mathbf{A} 's query (N, A, M) to the encryption oracle, \mathbf{E} queries its own oracle $e(\cdot)$ with M and returns the result to \mathbf{A} . On any query from \mathbf{A} to decryption oracle, \mathbf{E} returns \perp . When \mathbf{A} halts and outputs bit b , \mathbf{E} stops and outputs b as well. If $e(\cdot) = \bar{\Pi}.\bar{\mathcal{E}}_{\tilde{R}}^{\mathcal{S}}(\cdot)$, then \mathbf{E} simulates $\bar{\Pi}_{\tilde{R},\rho}(\cdot,\cdot,\cdot), \perp(\cdot,\cdot,\cdot)$ correctly (the assumption, that \mathbf{A} does not repeat queries, is needed here). If $e(\cdot) = \mathcal{S}(\cdot)$, then \mathbf{E} correctly simulates $\mathcal{S}(\cdot,\cdot,\cdot), \perp(\cdot,\cdot,\cdot)$. We deduce $\bar{p}_2 \leq \mathbf{Adv}_{\bar{\Pi}.\bar{\mathcal{E}}}^{\text{priv}\$}(\mathbf{E})$.

To give a bound on \bar{p}_1 , we shall reveal the tweakable RF \tilde{R} to \mathbf{A} . Clearly, an upper bound of the advantage in this case will also be valid if \mathbf{A} does not have \tilde{R} , since having \tilde{R} only makes the attack easier:

$$\begin{aligned} \bar{p}_1 &= \Pr \left[\mathbf{A}_{\tilde{R},\rho}^{\bar{\Pi}_{\tilde{R},\rho}(\cdot,\cdot,\cdot), \bar{\Pi}_{\tilde{R},\rho}^{-1}(\cdot,\cdot,\cdot)} \Rightarrow 1 \right] - \Pr \left[\mathbf{A}_{\tilde{R},\rho}^{\bar{\Pi}_{\tilde{R},\rho}(\cdot,\cdot,\cdot), \perp(\cdot,\cdot,\cdot)} \Rightarrow 1 \right] \\ &\leq \Pr \left[\mathbf{A}(\tilde{R})_{\tilde{R},\rho}^{\bar{\Pi}_{\tilde{R},\rho}(\cdot,\cdot,\cdot), \bar{\Pi}_{\tilde{R},\rho}^{-1}(\cdot,\cdot,\cdot)} \Rightarrow 1 \right] - \Pr \left[\mathbf{A}(\tilde{R})_{\tilde{R},\rho}^{\bar{\Pi}_{\tilde{R},\rho}(\cdot,\cdot,\cdot), \perp(\cdot,\cdot,\cdot)} \Rightarrow 1 \right] \end{aligned}$$

In this setting, \mathbf{A} can only tell the difference between the two games, if the decryption query returns something other than \perp (then \mathbf{A} stops and outputs 1). This happens, if \mathbf{A} builds a query (A, \mathbb{C}) to $\bar{\Pi}_{R,\rho}^{-1}$, that successfully verifies and decrypts, and that happens if $\mathbf{IV} = \rho(N, A, M)$ and $M = \bar{\Pi} \cdot \bar{\mathcal{D}}_{\tilde{R}}(\mathbf{IV}, \mathbb{C})$. Recall, that the adversary is assumed not to query its decryption oracle with (N, A, \mathbb{C}) if it had previously obtained \mathbb{C} from an encryption query (N, A, M) . Having \tilde{R} , \mathbf{A} can compute $M = \bar{\Pi} \cdot \bar{\mathcal{D}}_{\tilde{R}}(\mathbf{IV}, \mathbb{C})$ for any pair \mathbf{IV}, \mathbb{C} , but it never knows a pair $\mathbf{IV}, (N, A, M)$, s.t. $\mathbf{IV} = \rho(N, A, M)$ and (N, A, M) has not been queried to the encryption oracle before. \mathbf{A} is thus left to guess the correct \mathbf{IV} and the probability of producing a decryption query, that does not result in \perp is at most $1/2^\tau$. If we consider all queries made by \mathbf{A} , we have $\bar{p}_1 \leq q_d/2^\tau$. We then have $\bar{p} \leq \mathbf{Adv}_{\bar{\Pi}, \tilde{\mathcal{E}}}^{\text{priv}^\S}(\mathbf{E}) + q_d/2^\tau$.

At the beginning of the second step of the proof, we point out that

$$\mathbf{Adv}_{\bar{\Pi}}^{\text{mrae}}(\mathbf{A}) = \bar{p} + \Pr \left[\mathbf{A}^{\bar{\Pi}_{\tilde{R}}(\cdot, \cdot, \cdot), \bar{\Pi}_{\tilde{R}}^{-1}(\cdot, \cdot, \cdot)} \Rightarrow 1 \right] - \Pr \left[\mathbf{A}^{\bar{\Pi}_{R,\rho}(\cdot, \cdot, \cdot), \bar{\Pi}_{R,\rho}^{-1}(\cdot, \cdot, \cdot)} \Rightarrow 1 \right].$$

We construct an adversary \mathbf{R} for attacking $\bar{\Pi} \cdot \text{HASH}$ as PRF, that uses \mathbf{A} as a subroutine. \mathbf{R} is equipped with its own oracle $r(\cdot, \cdot, \cdot)$, which implements either $\bar{\Pi} \cdot \text{HASH}$ or a corresponding RF ρ . The adversary \mathbf{R} picks $\tilde{R} \xleftarrow{\S} \text{Func}^\mathcal{T}(m+n, n)$ and runs \mathbf{A} . On \mathbf{A} 's query (N, A, M) to encryption query, \mathbf{R} sets $\mathbf{IV} \leftarrow r(N, A, M)$, computes $\mathbb{C} \leftarrow \bar{\Pi} \cdot \tilde{\mathcal{E}}_{\tilde{R}}(\mathbf{IV}, M)$ and returns $\mathbf{IV} \parallel \mathbb{C}$ to \mathbf{A} . When \mathbf{A} asks a decryption query $(N, A, \mathbf{IV} \parallel \mathbb{C})$, \mathbf{R} first computes $M \leftarrow \bar{\Pi} \cdot \bar{\mathcal{D}}_{\tilde{R}}(\mathbf{IV}, \mathbb{C})$, then it returns M to \mathbf{A} only if $\mathbf{IV} = r(N, A, M)$, otherwise it returns \perp . When \mathbf{A} stops and outputs bit b , let \mathbf{E} stop and output b . It is easy to see, that if $r = \rho$, then \mathbf{R} correctly simulates $\bar{\Pi}_{\tilde{R},\rho}(\cdot, \cdot, \cdot), \bar{\Pi}_{\tilde{R},\rho}^{-1}(\cdot, \cdot, \cdot)$. It remains to show, that if $r = \bar{\Pi} \cdot \text{HASH}_{\tilde{R}}$ the adversary \mathbf{R} simulates $\bar{\Pi}_{\tilde{R}^*}(\cdot, \cdot, \cdot), \bar{\Pi}_{\tilde{R}^*}^{-1}(\cdot, \cdot, \cdot)$ correctly for some $\tilde{R}^* \in \text{Func}^\mathcal{T}(m+n, n)$. To do so, we give following argument. If we indeed have that $r = \bar{\Pi} \cdot \text{HASH}_{\tilde{R}}$, then the challenger for \mathbf{R} has picked the tweakable RF $\tilde{R}' \xleftarrow{\S} \text{Func}^\mathcal{T}(m+n, n)$, while \mathbf{R} has picked the tweakable RF $\tilde{R} \xleftarrow{\S} \text{Func}^\mathcal{T}(m+n, n)$ independently. The construction of $\bar{\Pi}$ is such, that the set of tweaks $\mathcal{T}_e = \mathcal{IV} \times \mathbb{N}$ used in $\bar{\Pi} \cdot \tilde{\mathcal{E}}$ is disjoint with the set of tweaks $\mathcal{T}_h = \mathcal{T} \setminus \mathcal{T}_e$ used in $\bar{\Pi} \cdot \text{HASH}$. For every $R, R' \in \text{Func}^\mathcal{T}(m+n, n)$ there is some $R^* \in \text{Func}^\mathcal{T}(m+n, n)$ such that:

$$R^{*\langle t_e \rangle}(\cdot) = R^{\langle t_e \rangle}(\cdot) \text{ for all } t_e \in \mathcal{T}_e, \quad R^{*\langle t_h \rangle}(\cdot) = R'^{\langle t_h \rangle}(\cdot) \text{ for all } t_h \in \mathcal{T}_h,$$

so the oracles simulated by \mathbf{R} are equivalent with oracles $\bar{\Pi}_{\tilde{R}^*}(\cdot, \cdot, \cdot), \bar{\Pi}_{\tilde{R}^*}^{-1}(\cdot, \cdot, \cdot)$. We will denote, that three tweakable functions R, R', R^* have the property just described by $R, R' \leftrightarrow R^*$. It remains to show, that the distribution of R^* is

uniform. We have

$$\begin{aligned}
 \Pr \left[\widetilde{R}^* = R^* \right] &= \sum_{R, R': R, R' \leftrightarrow R^*} \Pr \left[\widetilde{R} = R, \widetilde{R}' = R' \right] \\
 &= \sum_{R, R': R, R' \leftrightarrow R^*} \left(\frac{1}{2^{n \cdot |\mathcal{T}| \cdot 2^{m+n}}} \right)^2 \\
 &= |\{R, R' | R, R' \leftrightarrow R^*\}| \cdot \left(\frac{1}{2^{n \cdot |\mathcal{T}| \cdot 2^{m+n}}} \right)^2 \\
 &= 2^{n \cdot |\mathcal{T}_h| \cdot 2^{m+n}} \cdot 2^{n \cdot |\mathcal{T}_e| \cdot 2^{m+n}} \cdot \left(\frac{1}{2^{n \cdot |\mathcal{T}| \cdot 2^{m+n}}} \right)^2 \\
 &= \frac{1}{2^{n \cdot |\mathcal{T}| \cdot 2^{m+n}}}
 \end{aligned}$$

so the distribution of \widetilde{R}^* is indeed uniform, and simulation of \mathbf{A} 's oracles is correct. We deduce $\mathbf{Adv}_H^{\text{mr-ae}}(\mathbf{A}) \leq \bar{p} + \mathbf{Adv}_{H.HASH}^{\text{prf}}(\mathbf{R})$. This concludes the proof. \square

5.1.2 Instantiating Tweakable RFs with PRFs The proof of Theorem 1 is completed in the same way as in [8]. First, the tweakable RF \widetilde{R} is replaced by a tweakable PRF $\widetilde{F} : \mathcal{K} \times \mathcal{T} \times (\{0, 1\}^n \times \{0, 1\}^m) \rightarrow \{0, 1\}^n$, where $\mathcal{K} = \{0, 1\}^k$. This will increase the security bound as shown in Lemma 4.

Lemma 4. *Let $\widetilde{R} : \mathcal{T} \times (\{0, 1\}^n \times \{0, 1\}^m) \rightarrow \{0, 1\}^n$ be a tweakable RF and $\widetilde{F} : \mathcal{K} \times \mathcal{T} \times (\{0, 1\}^n \times \{0, 1\}^m) \rightarrow \{0, 1\}^n$ be a tweakable PRF. Then*

$$\mathbf{Adv}_{\text{MR-OMD}[\widetilde{F}, \tau]}^{\text{mrae}}(t, q_e, q_d, \sigma) \leq \mathbf{Adv}_{\text{MR-OMD}[\widetilde{R}, \tau]}^{\text{mrae}}(q_e, q_d, \sigma) + \mathbf{Adv}_{\widetilde{F}}^{\text{prf}}(t', \sigma)$$

where q_e and q_d are, respectively, the number of encryption and decryption queries, $t' = t + cn\sigma$ for some constant c and σ is the total number of calls to the underlying tweakable PRF \widetilde{F} in all queries asked by the MR-AE adversary.

Consequently, we instantiate the tweakable PRF from an ordinary PRF by the means of xoring a mask to (a part of) the input of the PRF, exactly as in [8]. The tweaks in MR-OMD are either of the form $T = (\alpha, i, j)$ where $\alpha \in \mathcal{N}$, $1 \leq i \leq 2^{n-5}$ and $j \in \{0, \dots, 5\}$ or of the form $T' = (\mathcal{IV}, i)$ with $\alpha \in \mathcal{TV}$, $1 \leq i \leq 2^{n-5}$. We can have a unified notation for all the tweaks as $T = (\alpha, i, j)$ where $\alpha \in \mathcal{N} \cup \mathcal{TV}$, $1 \leq i \leq 2^{n-5}$ and $j \in \{0, \dots, 5\}$ if $\alpha \in \mathcal{N}$ and $j = 6$ if $\alpha \in \mathcal{TV}$. The masking function $\Delta_K(T) = \Delta_K(\alpha, i, j)$ outputs an n -bit mask such that the following two properties hold for any fixed string $H \in \{0, 1\}^n$:

1. $\Pr[\Delta_K(\alpha, i, j) = H] \leq 2^{-n}$ for any (α, i, j)
2. $\Pr[\Delta_K(\alpha, i, j) \oplus \Delta_K(\alpha', i', j') = H] \leq 2^{-n}$ for $(\alpha, i, j) \neq (\alpha', i', j')$

where the probabilities are taken over random selection of the secret key K .

It is easy to verify that these two properties are satisfied by the specific masking scheme of MR-OMD as described in Section 4.

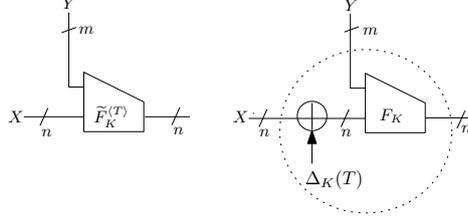


Fig. 4: Building a tweakable PRF $\tilde{F}_K^{(T)} : \{0, 1\}^n \times \{0, 1\}^m \rightarrow \{0, 1\}^n$ using a PRF $F_K : \{0, 1\}^n \times \{0, 1\}^m \rightarrow \{0, 1\}^n$ by applying the method of [17].

The transition from tweakable PRFs to PRFs with xor-masks being exactly the same, we use the result on security bound from [8].

Lemma 5. *Let $\tilde{F} : \mathcal{K} \times (\{0, 1\}^n \times \{0, 1\}^m) \rightarrow \{0, 1\}^n$ be a function family with key space \mathcal{K} . Let $\tilde{F} : \mathcal{K} \times \mathcal{T} \times (\{0, 1\}^n \times \{0, 1\}^m) \rightarrow \{0, 1\}^n$ be defined by $\tilde{F}_K^{(T)}(X||Y) = F_K((X \oplus \Delta(T))||Y)$ for every $T \in \mathcal{T}, K \in \mathcal{K}, X \in \{0, 1\}^n, Y \in \{0, 1\}^m$ and $\Delta_K(T)$ is the masking function of MR-OMD as defined in Section 4. If F is PRF then \tilde{F} is tweakable PRF; more precisely*

$$\mathbf{Adv}_{\tilde{F}}^{\text{prf}}(t, q) \leq \mathbf{Adv}_F^{\text{prf}}(t', 2q) + \frac{3q^2}{2^n}$$

For the proofs for both Lemma 4 and Lemma 5, the reader can refer to [8] and [17].

5.2 Security in the Nonce-Respecting Case

Intuitively, one would expect that the security bound in the nonce-respecting setting should be somewhat better than the one in the nonce-reuse case. Theorem 2 gives a bound on the AE security of MR-OMD in the nonce-respecting scenario, confirming this intuition.

Theorem 2. *Fix $n \geq 1$ and $\tau \in \{0, 1, \dots, n\}$. Let $F : \mathcal{K} \times (\{0, 1\}^n \times \{0, 1\}^m) \rightarrow \{0, 1\}^n$ be a PRF, where the key space $\mathcal{K} = \{0, 1\}^k$ for $k \geq 1$ and $1 \leq m \leq n$. Then*

$$\mathbf{Adv}_{\text{MR-OMD}[F, \tau]}^{\text{nr-ae}}(t, \sigma, q_e, q_d) \leq \mathbf{Adv}_F^{\text{prf}}(t', 2\sigma) + \frac{3\sigma^2}{2^n} + \frac{0.5q_e^2}{2^\tau} + \frac{q_d}{2^\tau}$$

where q_e and q_d are, respectively, the number of encryption and decryption queries, $t' = t + cn\sigma$ for some constant c and σ is the total number of calls made to the underlying compression function F .

Remark 2. We can verify that $\sigma = \lceil \sigma_A / (m+n) \rceil + \lceil \sigma_M / (m+n) \rceil + \lceil \sigma_M / (m) \rceil + \sum_{i=1}^{q_e} 1_{|M^i|=0} + \sum_{j=1}^{q_d} 1_{|C^j|=\tau} + q_e + q_d$.

Proof. The steps to prove this theorem are in fact almost the same as for Theorem 1. The only difference is in the proof for the security of the HASH algorithm as a PRF. This is easy to see, as HASH is the component of MR-OMD where the nonce is used. Lemma 6 gives the PRF security bound for HASH in the nonce-respecting setting. The bound stated in Theorem 2 is obtained combining Lemma 6 with Lemma 3 and Lemma 2 in subsection 5.1.1 and Lemma 4 and Lemma 5 in subsection 5.1.2. \square

Lemma 6. *Assume that adversaries are nonce-respecting. Let $\text{MR-OMD}[\tilde{R}, \tau]$ be the MR-OMD scheme that uses tweakable RF \tilde{R} . Then*

$$\text{Adv}_{\text{MR-OMD}[\tilde{R}, \tau].\text{HASH}}^{\text{prf}}(q, \sigma) = 0$$

where $\sigma = \sum_{i=1}^q (|A^i|_{m+n} + \max\{|M^i|_{m+n}, 1\})$ is the total number of calls to the underlying tweakable RF \tilde{R} in all q queries asked by a nonce-respecting adversary.

Proof. Recall the proof of Lemma 1: the $\text{MR-OMD}[\tilde{R}, \tau].\text{HASH}$ behaves as a RF unless there is a collision in the input to the final RF. This is because the final random function, determined by the final tweak, may be the same for several messages.

Now, considering that adversaries are nonce-respecting, we have that for every query (N^i, A^i, M^i) , $1 \leq i \leq q$ made by the adversary the nonce is distinct, i.e. $N^i \neq N^j$ if $i \neq j$. Each query is processed using a subset \mathcal{T} , that is disjoint with tweak sets used to process all the other queries. Therefore, when a query is processed, the final tweak is always fresh (never used before) and the random function is independent from all others so far. The distribution of $\text{MR-OMD}[\tilde{R}, \tau].\text{HASH}$ is then identical with that of $\text{Func}^{\mathcal{T}}(m+n, n)$. \square

6 Parallelizable MR-OMD

The MR-OMD scheme described in section 4 is designed to be substantially similar to OMD; hence, being able to share a lot of common code/hardware, while achieving different (stronger) security goals than OMD itself. This similarity also implies that the encryption/decryption process in MR-OMD is kept serial as it is in OMD. However, we notice that the two-pass construction (in contrast to OMD which is one-pass) also opens up the possibility of having a parallelizable encryption/decryption process.

The IV in MR-OMD is computed from both associated data and message using a PRF. Thus, the encryption is always dependent on the whole query (via IV) and we no longer need to apply the serial, chaining encryption of OMD. So, in specific applications where there are possibilities for parallel computation,

we might want to modify MR-OMD to exploit this fact. For this purpose, we propose PMR-OMD. PMR-OMD uses the same algorithms Initialize and HASH as MR-OMD, while the encryption/decryption algorithm uses counter mode. Schematic visualisation can be found in Figure 1. This replacement will of course get us further from the original OMD, which may be inconvenient in hardware implementations; however, in software implementations, the parallel execution might be exactly what we want, especially in general purpose CPUs with multiple cores. The PMR-OMD is almost fully parallelizable, with a single bottleneck in processing the final message block in its HASH algorithm.

6.1 Security of PMR-OMD

The security bound of PMR-OMD is exactly the same as those of MR-OMD, both in the nonce-misusing and nonce-respecting settings. This is because the proof of $\$$ -privacy of the counter mode is essentially the same as the one of $\$$ -privacy of the original OMD encryption. The remaining components of MR-OMD (and thus also the proofs) remain unchanged. We therefore omit the proofs of the following theorems.

Theorem 3. *Fix $n \geq 1$ and $\tau \in \{0, 1, \dots, n\}$. Let $F : \mathcal{K} \times (\{0, 1\}^n \times \{0, 1\}^m) \rightarrow \{0, 1\}^n$ be a PRF, where the key space $\mathcal{K} = \{0, 1\}^k$ for $k \geq 1$ and $1 \leq m \leq n$. Then*

$$\mathbf{Adv}_{\text{PMR-OMD}[F, \tau]}^{\text{mrae}}(t, \sigma, q_e, q_d) \leq \mathbf{Adv}_F^{\text{prf}}(t', 2\sigma) + \frac{3.5\sigma^2}{2^n} + \frac{0.5q_e^2}{2^\tau} + \frac{q_d}{2^\tau}$$

where q_e and q_d are, respectively, the number of encryption and decryption queries, $t' = t + cn\sigma$ for some constant c and σ is the total number of calls made to the underlying compression function F .

Remark 3. We can verify that $\sigma = \lceil \sigma_A / (m+n) \rceil + \lceil \sigma_M / (m+n) \rceil + \lceil \sigma_M / m \rceil + \sum_{i=1}^{q_e} 1_{|M^i|=0} + \sum_{j=1}^{q_d} 1_{|C^j|=\tau} + q_e + q_d$.

Theorem 4. *Fix $n \geq 1$ and $\tau \in \{0, 1, \dots, n\}$. Let $F : \mathcal{K} \times (\{0, 1\}^n \times \{0, 1\}^m) \rightarrow \{0, 1\}^n$ be a PRF, where the key space $\mathcal{K} = \{0, 1\}^k$ for $k \geq 1$ and $1 \leq m \leq n$. Then*

$$\mathbf{Adv}_{\text{PMR-OMD}[F, \tau]}^{\text{nr-ae}}(t, \sigma, q_e, q_d) \leq \mathbf{Adv}_F^{\text{prf}}(t', 2\sigma) + \frac{3\sigma^2}{2^n} + \frac{0.5q_e^2}{2^\tau} + \frac{q_d}{2^\tau}$$

where q_e and q_d are, respectively, the number of encryption and decryption queries, $t' = t + cn\sigma$ for some constant c and σ is the total number of calls made to the underlying compression function F .

Remark 4. We can verify that $\sigma = \lceil \sigma_A / (m+n) \rceil + \lceil \sigma_M / (m+n) \rceil + \lceil \sigma_M / m \rceil + \sum_{i=1}^{q_e} 1_{|M^i|=0} + \sum_{j=1}^{q_d} 1_{|C^j|=\tau} + q_e + q_d$.

References

1. Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Tischhauser, E., Yasuda, K.: Parallelizable and authenticated online ciphers. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT (1). Lecture Notes in Computer Science, vol. 8269, pp. 424–443. Springer (2013), <http://dblp.uni-trier.de/db/conf/asiacrypt/asiacrypt2013-1.html#AndreevaBLMTY13>
2. Bellare, M., Namprempre, C.: Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 531–545. Springer (2000)
3. Bellare, M., Rogaway, P.: Encode-Then-Encipher Encryption: How to Exploit Nonces or Redundancy in Plaintexts for Efficient Cryptography. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 317–330. Springer (2000)
4. Bellare, M., Rogaway, P., Wagner, D.: The EAX mode of operation. In: FSE. pp. 389–407 (2004)
5. Bernstein, D.J.: Cryptographic competitions: CAESAR. <http://competitions.cr.yp.to>
6. Black, J., Rogaway, P.: Cbc macs for arbitrary-length messages: The three-key constructions. *J. Cryptology* 18(2), 111–131 (2005)
7. Canvel, B., Hiltgen, A.P., Vaudenay, S., Vuagnoux, M.: Password Interception in a SSL/TLS Channel. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 583–599. Springer (2003)
8. Cogliani, S., Ștefania Maimuț, D., Naccache, D., do Canto, R.P., Reyhanitabar, R., Vaudenay, S., Vizár, D.: Offset merkle-damgård (omd) version 1.0 a caesar proposal. Proposal in CAESAR competition (mar 2014)
9. Datta, N., Nandi, M.: Elmd. CAESAR submission (2013), <http://competitions.cr.yp.to/caesar.html>
10. Dworkin, M.: Recommendation for block cipher modes of operation: the CCM mode for authentication and confidentiality. NIST Special Publication 800-38C (May 2004)
11. Dworkin, M.: Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) and GMAC. NIST Special Publication 800-38D (Nov 2007)
12. Fleischmann, E., Forler, C., Lucks, S.: McOE: A family of almost foolproof on-line authenticated encryption schemes. In: FSE. pp. 196–215 (2012)
13. Fleischmann, E., Forler, C., Lucks, S., Wenzel, J.: McOE: A foolproof on-line authenticated encryption scheme. *IACR Cryptology ePrint Archive* 2011, 644 (2011)
14. Iwata, T., Yasuda, K.: BTM: A single-key, inverse-cipher-free mode for deterministic authenticated encryption. In: *Selected Areas in Cryptography*. pp. 313–330 (2009)
15. Iwata, T., Yasuda, K.: HBS: A single-key mode of operation for deterministic authenticated encryption. In: FSE. pp. 394–415 (2009)
16. Katz, J., Yung, M.: Unforgeable Encryption and Chosen Ciphertext Secure Modes of Operation. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 284–299. Springer (2001)
17. Krovetz, T., Rogaway, P.: The Software Performance of Authenticated-Encryption Modes. In: Joux, A. (ed.) FSE 2011. LNCS, vol. 6733, pp. 306–327. Springer (2011)
18. McGrew, D.A., Viega, J.: The security and performance of the galois/counter mode (GCM) of operation. In: INDOCRYPT. pp. 343–355 (2004)

19. Namprempe, C., Rogaway, P., Shrimpton, T.: Reconsidering Generic Composition. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT. Lecture Notes in Computer Science, vol. 8441, pp. 257–274. Springer (2014)
20. Procter, G., Cid, C.: On Weak Keys and Forgery Attacks against Polynomial-based MAC Schemes. IACR Cryptology ePrint Archive (full version to appear in the Journal of Cryptology. A short version of this paper was presented at Fast Software Encryption 2013) p. 144 (2013)
21. Rogaway, P.: Authenticated-Encryption with Associated-Data. In: ACM Conference on Computer and Communications Security. pp. 98–107 (2002)
22. Rogaway, P.: Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In: ASIACRYPT. pp. 16–31 (2004)
23. Rogaway, P., Bellare, M., Black, J., Krovetz, T.: OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption. In: ACM Conference on Computer and Communications Security. pp. 196–205 (2001)
24. Rogaway, P., Shrimpton, T.: A Provable-Security Treatment of the Key-Wrap Problem. In: EUROCRYPT. pp. 373–390 (2006)
25. Rogaway, P., Shrimpton, T.: Deterministic authenticated-encryption: A provable-security treatment of the key-wrap problem. IACR Cryptology ePrint Archive 2006, 221 (2006)
26. Saarinen, M.J.O.: Cycling Attacks on GCM, GHASH and Other Polynomial MACs and Hashes. In: Canteaut, A. (ed.) FSE. Lecture Notes in Computer Science, vol. 7549, pp. 216–225. Springer (2012)
27. Vaudenay, S.: Security Flaws Induced by CBC Padding - Applications to SSL, IPSEC, WTLS ... In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 534–546. Springer (2002)
28. Whiting, D., Housley, R., Ferguson, N.: Intel®SHA Extensions New Instructions Supporting the Secure Hash Algorithm on Intel®Architecture Processors. Intel White Paper (Jul 2013), <https://software.intel.com/en-us/articles/intel-sha-extensions>

A Proof of Lemma 1

We assume w.l.o.g. that the adversary does not repeat a query. Let q denote the number of queries asked by the adversary and let r denote the number of distinct nonces among all the nonces in the q queries. We partition the queries into sets $\mathcal{Q}^1, \dots, \mathcal{Q}^r$, so that for any two queries $N, A, M \in \mathcal{Q}^i$ and $N', A', M' \in \mathcal{Q}^j$ we have $N = N'$ if $i = j$ and $N \neq N'$ otherwise. Let $q_i = |\mathcal{Q}^i|$ for $i = 1, \dots, r$, then we have $q = \sum_{i=1}^r q_i$. For any $1 \leq i \leq r$, we will denote the queries from \mathcal{Q}^i as $\mathcal{Q}^i = \{(N^i, A^{i,1}, M^{i,1}), \dots, (N^i, A^{i,q_i}, M^{i,q_i})\}$. Let $b = n + m$. We will use notation $H_{\tilde{R}}^\tau$ instead of $\text{MIR} - \text{OMD}[\tilde{R}, \tau].\text{HASH}$ throughout the proof. First, we claim that

$$\mathbf{Adv}_{H_{\tilde{R}}^\tau}^{\text{prf}}(\sigma) \leq \max \left\{ \sum_{h=1}^r \sum_{1 \leq i < j \leq q_h} \Pr [(N^h, A^{h,i}, M^{h,i}), (N^h, A^{h,j}, M^{h,j}) \text{ collide}] \right\}$$

where the maximum is taken over the choice of r, q_1, q_2, \dots, q_r and the queries $(N^{1,1}, A^{1,1}, M^{1,1}), \dots, (N^{r,q_r}, A^{r,q_r}, M^{r,q_r})$ so that we have $\sum_{i=1}^r \sum_{j=1}^{q_i} |A^{i,j}|_b + ||M^{i,j}||_b \leq \sigma$ and where for two queries (N, A, M) and (N, A', M') , with $a = |A|_b$,

$t = \|M\|_b$, $a' = |A'|_b$, $t' = \|M'\|_b$, the event ‘ $(N, A, M), (N, A', M')$ collide’ means that (N, A, M) and (N, A', M') are distinct (as tuples) and we have $X = X'$, where

$$\begin{aligned} X &= \left(\tilde{R}^{\langle N,1,1 \rangle} (A_1) \oplus \dots \oplus \tilde{R}^{\langle N,a-1,1 \rangle} (A_{a-1}) \oplus \tilde{R}^{\langle N,a-1,j_A \rangle} (A_a^*) \oplus \right. \\ &\quad \left. \tilde{R}^{\langle N,1,0 \rangle} (M_1) \oplus \dots \oplus \tilde{R}^{\langle N,t-1,0 \rangle} (M_{t-1}) \right) \oplus_{msb} M_t^* \\ X' &= \left(\tilde{R}^{\langle N,1,1 \rangle} (A'_1) \oplus \dots \oplus \tilde{R}^{\langle N,a'-1,1 \rangle} (A'_{a'-1}) \oplus \tilde{R}^{\langle N,a'-1,j_{A'} \rangle} (A'_{a'}^*) \oplus \right. \\ &\quad \left. \tilde{R}^{\langle N,1,0 \rangle} (M'_1) \oplus \dots \oplus \tilde{R}^{\langle N,t'-1,0 \rangle} (M'_{t'-1}) \right) \oplus_{msb} M_{t'}^* \end{aligned}$$

That is, we claim that the advantage $\mathbf{Adv}_{H_R^\tau}^{\text{prf}}$ is bounded by the probability of collision on the input to the final tweakable RF among queries with the same nonce. To prove this claim, note that the HASH algorithm can be viewed as

$$H_R^\tau(N, A, M) = \tilde{R}^{\langle T_{\text{final}} \rangle} (h_{\tilde{R}}(N, A, M))$$

where the function $h_{\tilde{R}} \in \text{Func}(\{0, 1\}^{|N|} \times \{0, 1\}^* \times \{0, 1\}^*, n)$ is defined as

$$\begin{aligned} h_{\tilde{R}}(N, A, M) \mapsto &\left(\tilde{R}^{\langle N,1,1 \rangle} (A_1) \oplus \dots \oplus \tilde{R}^{\langle N,a-1,1 \rangle} (A_{a-1}) \oplus \tilde{R}^{\langle N,a-1,j_A \rangle} (A_a^*) \oplus \right. \\ &\left. \tilde{R}^{\langle N,1,0 \rangle} (M_1) \oplus \dots \oplus \tilde{R}^{\langle N,t-1,0 \rangle} (M_{t-1}) \right) \oplus_{msb} M_t^*. \end{aligned}$$

The final tweakable RF $\tilde{R}^{\langle \text{final} \rangle}$ is independent from $h_{\tilde{R}}$, because its tweak T_{final} is not used anywhere in $h_{\tilde{R}}$. Moreover, the final tweakable RFs of queries with different nonces are completely independent as well. Therefore, unless there is a collision on the output of $h_{\tilde{R}}$ among queries that share the same value of the nonce, the construction $\tilde{R}^{\langle T_{\text{final}} \rangle} (h_{\tilde{R}}(N, A, M))$ behaves as a truly RF and cannot be distinguished from such. This completes the proof of the claim. We note that this claimed bound can be also proven using a straightforward extension of the game-playing argument from [6].

Now, we proceed to prove the bound in Lemma 1 by bounding the collision probabilities. We note that the set of final tweaks used to process queries (N, A, M) with $|M|$ a multiple of b and the set of final tweaks used to process queries (N, A', M') with M' whose final block is incomplete are mutually exclusive, so we only need to consider collisions among inputs of the same type. To bound the probability of collision $\Pr[(N, A, M), (N, A', M') \text{ collide}]$, an exhaustive case-analysis needs to be done. Here the queries (N, A, M) and (N, A', M') are fixed and the probability is over the choice of \tilde{R} . We let $a = |A|_b$, $t = \|M\|_b$, $a' = |A'|_b$, $t' = \|M'\|_b$ throughout all cases.

Case 1: $|M|$ and $|M'|$ are multiples of b and (w.l.o.g) $|A|$ is a multiple of b and $|A'|$ is not. The collision on $h_{\tilde{R}}$ occurs, if $h_{\tilde{R}}(N, A, M) = h_{\tilde{R}}(N, A', M')$.

Let $S^{(1)}(N, A, M) = h_{\tilde{R}}(N, A, M) \oplus \tilde{R}^{\langle N, a-1, j_A \rangle}(A_a)$ be a partial result of evaluating $h_{\tilde{R}}(N, A, M)$. Given any two queries $(N, A, M), (N, A', M')$, a collision on $h_{\tilde{R}}$ is then equivalent to

$$\begin{aligned} S^{(1)}(N, A, M) \oplus \tilde{R}^{\langle N, a-1, 3 \rangle}(A_a) &= S^{(1)}(N, A', M') \oplus \tilde{R}^{\langle N, a'-1, 5 \rangle}(A'_{a'}) \\ \tilde{R}^{\langle N, a'-1, 5 \rangle}(A'_{a'}) \oplus \tilde{R}^{\langle N, a-1, 3 \rangle}(A_a) &= S^{(1)}(N, A', M') \oplus S^{(1)}(N, A, M) \end{aligned}$$

As can be seen in the first equation, the two tweaks used to process the last blocks of A and A' come from mutually exclusive sets, so the two RFs used to process these blocks will always be chosen independently at random. The collision occurs, if the xor of the outputs of these independent RFs is equal to a distinct value. Probability of collision is then $1/2^n$.

Case 2: $|M|, |M'|$ are both multiples of b and $|A|, |A'|$ are either both multiples of b , or they both are not. In case that both A and A' have an incomplete final block, we can assume, that $A \leftarrow A \parallel 10^{b-1-|A| \bmod b}$ and $A' \leftarrow A' \parallel 10^{b-1-|A'| \bmod b}$. This does not affect the probability of collision, because this mapping is injective for associated data with incomplete last block, and because the set of tweaks used to process final blocks of AD with full-length final block ($j_A = 3$) is mutually exclusive with the set of tweaks used to process final block of messages with incomplete final block ($j_A = 5$). Thus in the following sub-cases, we can w.l.o.g. assume that $|A|, |A'|$ are multiples of b .

Case 2a: $a \neq a'$. W.l.o.g. assume that $a > a'$. Similarly as in **Case 1**, we can express partial evaluation of $h_{\tilde{R}}(A, M) - S^{(2a)}(A, M)$ - as follows:

$$\begin{aligned} S^{(2a)}((N, A, M)) \mapsto & \left(\tilde{R}^{\langle N, 1, 1 \rangle}(A_1) \oplus \dots \oplus \tilde{R}^{\langle N, a', 1 \rangle}(A_{a'}) \oplus \right. \\ & \left. \tilde{R}^{\langle N, 1, 0 \rangle}(M_1) \oplus \dots \oplus \tilde{R}^{\langle N, t-1, 0 \rangle}(M_{t-1}) \right) \oplus_{msb} M_t^* \end{aligned}$$

The collision occurs, if $\tilde{R}^{\langle N, a'+1, 1 \rangle}(A_{a'+1}) \oplus \dots \oplus \tilde{R}^{\langle N, a-1, j_A \rangle}(A_a) = h_{\tilde{R}}(N, A', M') \oplus S^{(2a)}(N, A, M)$. Again, this happens if a xor of outputs of multiple independent RFs equals to a distinct value. The probability of finding a tuple of RFs' inputs producing this equality is $1/2^n$.

Case 2b: $a = a'$ and $A \neq A'$. Because $A \neq A'$, there must be an i , s.t. $1 \leq i \leq a$ and $A_i \neq A'_i$. Again, we construct a partial evaluation $S^{(2b)}(N, A, M) = h_{\tilde{R}}(N, A, M) \oplus \tilde{R}^{\langle N, i, j_i \rangle}(A_i)$. The collision occurs if $\tilde{R}^{\langle N, i, j_i \rangle}(A_i) \oplus \tilde{R}^{\langle N, i, j_i \rangle}(A'_i) = S^{(2b)}(N, A, M) \oplus S^{(2b)}(N, A', M')$. In other words, we have a collision if the result of $\tilde{R}^{\langle N, i, 1 \rangle}(A_i) \oplus \tilde{R}^{\langle N, i, 1 \rangle}(A'_i)$ equals to a distinct value. Because $A_i \neq A'_i$, the probability of this event is $1/2^n$.

Case 2c: $A = A'$ and $t \neq t'$. W.l.o.g. assume that $t > t'$. Similarly as in **Case 2a**, we let $S^{(2c)}(N, A, M)$ be the partial result of $h_{\tilde{R}}(N, A, M)$:

$$\begin{aligned} S^{(2c)}(N, A, M) \mapsto & \left(\tilde{R}^{\langle N, 1, 1 \rangle}(A_1) \oplus \dots \oplus \tilde{R}^{\langle N, a-1, j_A \rangle}(A_a) \oplus \right. \\ & \left. \tilde{R}^{\langle N, 1, 0 \rangle}(M_1) \oplus \dots \oplus \tilde{R}^{\langle N, t', 0 \rangle}(M_{t'}) \right) \oplus_{msb} M_t^* \end{aligned}$$

The collision occurs if $\tilde{R}^{(N,t'+1,0)}(M_{t'+1}) \oplus \dots \oplus \tilde{R}^{(N,t-1,0)}(M_{t-1}) = S^{(2e)}(N, A, M) \oplus h_{\tilde{R}}(N, A', M')$. The probability of this event is $1/2^n$.

Case 2d: $A = A'$, $t = t'$ and M, M' differ in blocks with index i , $i < t$. We let $S^{(2d)}(N, A, M) = h_{\tilde{R}}(N, A, M) \oplus \tilde{R}^{(N,i,0)}(M_i)$. The collision occurs if $\tilde{R}^{(N,i,0)}(M_i) \oplus \tilde{R}^{(N,i,0)}(M'_i) = S^{(2b)}(N, A, M) \oplus S^{(2b)}(N, A', M')$. By similar argument as in **Case 2b**, the probability of collision is thus $1/2^n$.

Case 2e: $A = A'$, $t = t'$ and M, M' differ only in the last block. Then we have $h_{\tilde{R}}(N, A, M) \neq h_{\tilde{R}}(N, A', M')$ so the probability of collision is 0.

Case 3: $|M|$ and $|M'|$ are not multiples of b and (w.l.o.g) $|A|$ is a multiple of b and $|A'|$ is not. This case is analogous to **Case 1**, the only difference is that both M_t and M'_t will be padded before processing (which is of no consequence). By similar argument as in **Case 1**, we conclude that probability of collision is $1/2^n$.

Case 4: $|M|, |M'|$ are both multiples of b and $|A|, |A'|$ are either both multiples of b , or they both are not. In case that both A and A' have an incomplete final block, we can assume, that $A \leftarrow A \parallel 10^{b-1-|A| \bmod b}$ and $A' \leftarrow A' \parallel 10^{b-1-|A'| \bmod b}$. This does not affect the probability of collision (by the argument in the **Case 2**). Thus in the following sub-cases, we can assume that $|A|, |A'|$ are multiples of b . As in previous case, we can also let $M \leftarrow M \parallel 10^{b-1-|M| \bmod b}$ and $M' \leftarrow M' \parallel 10^{b-1-|M'| \bmod b}$ without changing the probabilities of collisions - again, we make use of the fact, that this mapping is injective for M, M' with incomplete final blocks and that the tweak set used to process queries where M and has incomplete final block is disjoint with the tweak set used to process queries where M has full-length final block. This effectively transforms this case into **Case 2**. We therefore list all the sub-cases only briefly.

Case 4a: $a \neq a'$. The probability of collision is $1/2^n$, similarly as in the **Case 2a**.

Case 4b: $a = a'$ and $A \neq A'$. The probability of collision is $1/2^n$, similarly as in the **Case 2b**.

Case 4c: $A = A'$ and $t \neq t'$. The probability of collision is $1/2^n$, similarly as in the **Case 2c**.

Case 4d: $A = A'$, $t = t'$ and M, M' differ in blocks with index i , $i < t$. The probability of collision is $1/2^n$, similarly as in the **Case 2d**.

Case 4e: $A = A'$, $t = t'$ and M, M' differ only in last block. The probability of collision is 0, similarly as in the **Case 2e**.

We deduce that for any two queries (N, A, M) and (N, A', M') we have that $\Pr[(N, A, M), (N, A', M') \text{ collide}] \leq \frac{1}{2^n}$. Using this result, we can bound the

advantage $\mathbf{Adv}_{\widetilde{H}_R}^{\text{prf}}(q_e, \sigma, \ell_{\max})$:

$$\begin{aligned} \mathbf{Adv}_{\widetilde{H}_R}^{\text{prf}}(\sigma) &\leq \max \left\{ \sum_{h=1}^r \sum_{1 \leq i < j \leq q_h} \Pr \left[(N^{h,i}, A^{h,i}, M^{h,i}), (N^{h,j}, A^{h,j}, M^{h,j}) \text{ collide} \right] \right\} \\ &\leq \max \left\{ \sum_{h=1}^r \sum_{1 \leq i < j \leq q_h} \frac{1}{2^n} \right\} \leq \max \left\{ \sum_{h=1}^r q_h^2 \frac{1}{2^n} \right\} \leq \frac{0.5\sigma^2}{2^n} \end{aligned}$$

This completes the proof. \square

B Proof of Lemma 2

For the sake of readability, we will use Π to refer to $\text{MIR} - \text{OMD}[\widetilde{R}, \tau].\bar{\mathcal{E}}$ throughout this proof. We observe the advantage of the adversary \mathbf{A} in two mutually exclusive cases:

$$\mathbf{Adv}_{\Pi}^{\text{priv}\$}(\mathbf{A}) = \mathbf{Adv}_{\Pi}^{\text{priv}\$|\text{IVcoll}}(\mathbf{A}) \Pr[\text{IVcoll}] + \mathbf{Adv}_{\Pi}^{\text{priv}\$|\neg\text{IVcoll}}(\mathbf{A}) \Pr[\neg\text{IVcoll}]$$

where IVcoll denotes the event, that there is a collision among IVs and:

$$\begin{aligned} \mathbf{Adv}_{\Pi}^{\text{priv}\$|\text{IVcoll}}(\mathbf{A}) &= \left(\Pr \left[K \xleftarrow{\$} \mathcal{K} : \mathbf{A}^{\Pi_K^{\$(\cdot)}} \Rightarrow 1 \mid \text{IVcoll} \right] - \Pr \left[\mathbf{A}^{\$(\cdot)} \Rightarrow 1 \mid \text{IVcoll} \right] \right) \\ \mathbf{Adv}_{\Pi}^{\text{priv}\$|\neg\text{IVcoll}}(\mathbf{A}) &= \left(\Pr \left[K \xleftarrow{\$} \mathcal{K} : \mathbf{A}^{\Pi_K^{\$(\cdot)}} \Rightarrow 1 \mid \neg\text{IVcoll} \right] - \Pr \left[\mathbf{A}^{\$(\cdot)} \Rightarrow 1 \mid \neg\text{IVcoll} \right] \right) \end{aligned}$$

First, consider the case that there is no collision on IVs. This implies, that all tweaks $\langle \text{IV}_j, i \rangle$, $1 \leq j \leq q_e$ used to encrypt the queried messages M^1, \dots, M^{q_e} are distinct and all the RFs $\widetilde{R}^{\langle \text{IV}_j, i \rangle}$, used in the encryption queries, are independent. Thus, all the ciphertexts $\mathbb{C}^1, \dots, \mathbb{C}^{q_e}$ the adversary sees appear to be independent random strings. We deduce $\mathbf{Adv}_{\Pi_K}^{\text{priv}\$|\neg\text{IVcoll}}(\mathbf{A}) = 0$.

We bound $\mathbf{Adv}_{\Pi}^{\text{priv}\$|\text{IVcoll}}(\mathbf{A})$ by the probability $\Pr[\text{IVcoll}]$. We have

$$\Pr[\text{IVcoll}] \leq \sum_{1 \leq i < j \leq q_e} \Pr[\text{IV}^i = \text{IV}^j] \leq \sum_{1 \leq i < j \leq q_e} \frac{1}{2^\tau} \leq \frac{0.5q_e^2}{2^\tau}$$

We deduce $\mathbf{Adv}_{\Pi}^{\text{priv}\$|\text{IVcoll}}(\mathbf{A}) \leq \frac{0.5q_e^2}{2^\tau}$. This concludes the proof. \square