

Nonce Misuse Resilience of GCM with Rijndael-256-256

Mario Marhuenda Beltrán¹ and Bart Mennink²

¹ Radboud University, Nijmegen, The Netherlands

mario.marhuendabeltran@ru.nl

² Maastricht University, Maastricht, The Netherlands

bart.mennink@maastrichtuniversity.nl

Abstract. GCM is the leading authenticated encryption scheme. With its application in TLS alone, it finds millions of uses every day. However, in the last years, various shortcomings of GCM have been observed: mistakes in earlier proofs, non-optimal security degradation in case of shorter tags, the support for a 96-bit nonce only (concatenated with a 32-bit counter) for the most widely implemented variant, but most of all, the brittle resistance against nonce misuse. This latter shortcoming is particularly worrisome in applications where uniqueness of the nonce cannot be guaranteed (such as in distributed systems). In part to salvage this situation, the US National Institute of Standards and Technology (NIST) is contemplating the idea of standardizing Rijndael-256-256. Indeed, sticking to the concept of a 32-bit counter, this would allow the support of 224-bit nonces next to the 32-bit counter. Nonetheless, it may be observed that nonces may still be accidentally reused due to other causes and the larger block size does not solve the fatal effects of nonce misuse, among which the possibility to perform a subkey recovery.

To salvage this, we propose GCM-256. The fix is, as a matter of fact, rather simple and comes almost for free. In a nutshell: whereas a native generalization of GCM to 256-bit blocks would mask the output of GHASH with only half of the output of the initial nonce based block cipher evaluation, we suggest to not discard the other half but use it as key to GHASH. As an effect, we demonstrate that GCM-256 does not only achieves classic confidentiality and authenticity, but as a bonus it achieves nonce misuse resilience, meaning that even if nonces are misused, the scheme still guarantees security for fresh nonces.

Keywords: GCM, Rijndael-256-256, nonce misuse resilience, almost for free

1 Introduction

Authenticated encryption [BN00] is the main concept that secures today’s digital communication. It assures confidentiality of data, meaning that no outsider can learn anything about the content of it, as well as authenticity, meaning that any manipulation gets detected. The most well-known and most widely used authenticated encryption scheme is undoubtedly Galois/Counter Mode, or GCM [MV04]. It has been standardized in NIST’s SP 800 38D [Dwo07] and in ISO/IEC 19772 [ISO09], and it is used as main authenticated encryption scheme in TLS [SMC08], IPsec [VM05], SSH [IS09], Wifi (WPA3) [WPA24], NSA Suite B [NSA10], and IEEE 802.1 [IEE06].

At a very high level, GCM is a fairly clean combination of nonce based counter mode for keystream encryption followed by Wegman-Carter-Shoup [WC81, Bra82, Sho96] for authentication. A bit more concretely, suppose we are given a keyed block cipher $E_K : \{0, 1\}^n \rightarrow \{0, 1\}^n$, the most widely used variant of GCM operates as follows on input of a $(n - 32)$ -bit nonce N , associated data A , and plaintext P :

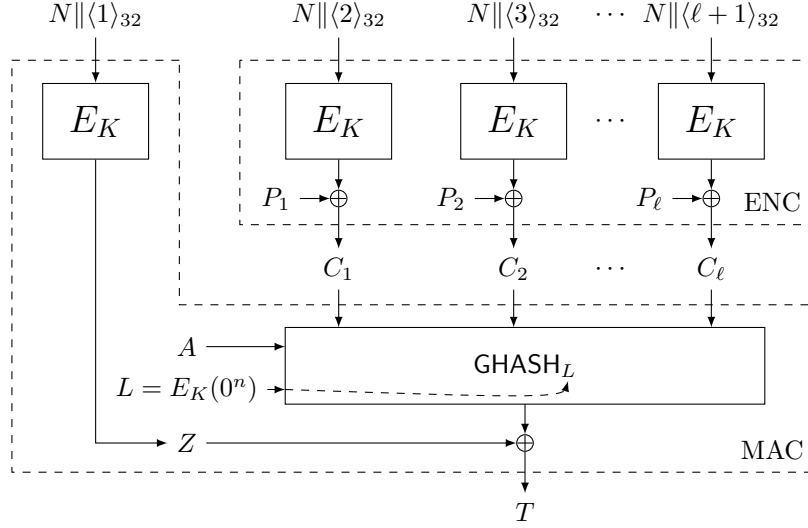


Figure 1: The GCM authenticated encryption scheme for 96-bit nonce. The function GHASH_L is depicted in Figure 4. We refer to Algorithm 1 for the pseudocode. The GCM-ext-1 authenticated encryption scheme is identical, but with 256-bit wires.

- It first computes the keystream

$$S = E_K(N \parallel \langle 2 \rangle_{32}) \parallel E_K(N \parallel \langle 3 \rangle_{32}) \parallel \dots,$$

where $\langle i \rangle_{32}$ denotes the encoding of the integer $i \in \mathbb{N}$ into a binary string of length 32 bits, and then S is added to the plaintext to obtain the ciphertext $C = P \oplus S$;

- It then evaluates GHASH with key $L = E_K(0^n)$ on an injective padding of A and C (cf., (8)), and masks it with $Z = E_K(N \parallel \langle 1 \rangle_{32})$ to get the tag:

$$T = \text{GHASH}_L(A, C) \oplus Z.$$

We refer to Figure 1 for a visualization of the mode. We remark that the formal specification of GCM [MV04] is more general in that it allows for arbitrary-length nonces: if the nonce is not of size $n - 32$ bits, it is first processed with GHASH_L to obtain an n -bit input to the counter mode.

In their original specification, McGrew and Viega [MV04] derived a proof, that was later re-analyzed and fixed by Iwata et al. [IOM12] and Niwa et al. [NOM15]. These results show that, assuming E is a pseudorandom permutation and the nonce is never reused for different encryption evaluations, GCM achieves confidentiality as long as the total complexity does not exceed $2^{n/2}$ and authenticity as long as additionally the total number of forgery attempts does not exceed $2^n/\ell$, where ℓ is the maximum block length. It is worth noting that this result is in fact slightly more general, in that it supports tag truncation to t bits. In this case, the number of forgery attempts should not exceed $2^t/\ell$.

However, the freshness of the nonce is a rather crucial condition! If a nonce N is used twice for two encryptions (A, P) and (A', P') , we necessarily have $C \oplus C' = P \oplus P'$, but more importantly, $L = E_K(0^n)$ can be recovered quite efficiently and this subsequently allows for universal forgeries [Jou06]. The importance of the problem becomes particularly clear if one considers GCM instantiated with AES, the de facto standard instantiation [Dwo07]. In this case, $n = 128$ (the key may be 128, 192, or 256 bits), and the nonce size is $n - 32 = 96$ bits. If one cannot maintain a counter (for whatever reason, e.g., as one is concerned with

a distributed system), random nonces may collide after 2^{48} evaluations, and once this happens, leakage of L implies that one cannot trust encryptions using key K anymore. This is a real-world problem, as also suggested by Kampanakis et al. [KCCP23].

In response to this, the US National Institute of Standards and Technology (NIST) is currently planning to standardize Rijndael-256-256 [NIS24]. This block cipher has a 256-bit key, but more importantly (for our discussion) a 256-bit block size. Intuitively, this seems to solve the issue of short nonces: in above example it would allow nonces of size 224 bits. (Here, we remark that this is also the reason why we restrict our focus to the original version of GCM with 96-bit nonces: the general version is more expensive, and not needed after all if we would instantiate GCM with Rijndael-256-256.)

However, this is only intuition: it only makes sense to reason about the potential of Rijndael-256-256 in GCM once the proper generalization of GCM to 256-bit blocks is considered, and the GCM specification nor the NIST standard consider this. Particularly, it is not clear *how* GCM would be generalized to accommodate a 256-bit block.

In this direction, we can see two logical extensions:

- GCM-ext-1. In the first extension, all wires in Figure 1 are 256 bits. This means that nonces are of size 224 bits (provided the counter is still of size 32 bits), but also GHASH operates with a 256-bit key, over a binary extension field of size 2^{256} ,¹ and with a 256-bit mask. In this case, the bounds of GCM literally carry over, but with $n = 256$ instead of $n = 128$. Note that this is also very similar to the plans that NIST recently outlined [NIS26];
- GCM-ext-2. In the second extension, the encryption wires are 256 bits but the authentication wires are still 128 bits. In this case, the specification of GCM and Figure 1 carry over, except that the subkey for GHASH would be

$$L = \text{left}_{128}(E_K(0^{256})) \quad (1)$$

and the masking would be with

$$Z = \text{left}_{128}(E_K(N \parallel \langle 1 \rangle_{32})). \quad (2)$$

The function is depicted in Figure 2. This description is simpler and is inspired by the observation that, in fact, the security bounds of GCM are quadratic in the block size (the $2^{n/2}$ term) during encryption but only roughly linear in the tag size during authentication (the $2^t/\ell$ term). Stated differently, this observation suggests that having a 256-bit authentication path is overkill [NIS26]. Noting that the bounds on GCM basically already consider a truncated variant, it is quickly observed that the original bounds still carry over, with $n = 256$ and $t = 128$.

1.1 GCM-256 and Its Nonce Misuse Resilience

Actually, judging the existing security bound on GCM, of the two generalizations outlined above, the GCM-ext-2 variant makes the most sense, but still, nonce misuse is devastating as it reveals the subkey (1) and it thus gives room for universal forgeries. We remark that nonce misuse may seem like an unlikely event in case the nonce is of size 224 bits, but nonce misuse can take many different shapes and forms regardless of the nonce size [BZD⁺16].

However, additionally, part of the “randomness” in the block cipher evaluation of (2) gets destroyed. Based on this, we consider the very minor modification on GCM that has a rather high impact: instead of (1) and (2), we set

$$\begin{aligned} Z &= \text{left}_{128}(E_K(N \parallel \langle 1 \rangle_{32})), \\ L &= \text{right}_{128}(E_K(N \parallel \langle 1 \rangle_{32})). \end{aligned} \quad (3)$$

¹This would be over the binary extension field with an irreducible polynomial of degree 256.

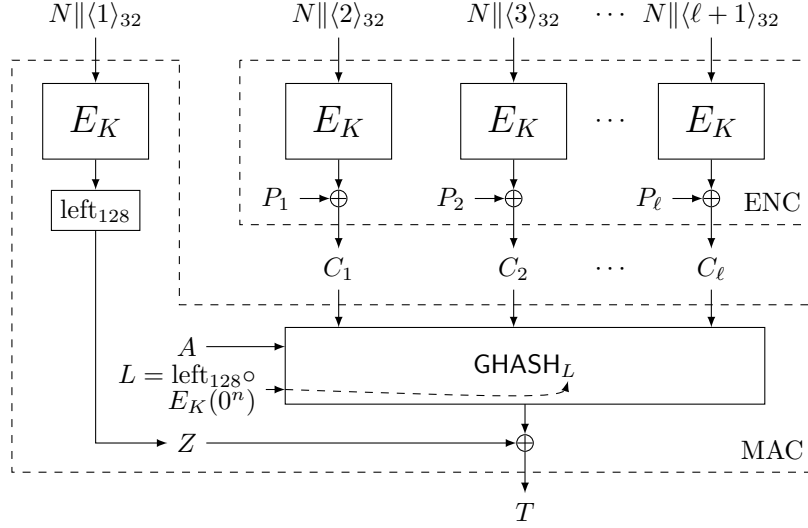


Figure 2: The GCM-ext-2 authenticated encryption scheme. The function GHASH_L is depicted in Figure 4.

Table 1: Security results of GCM, the extensions GCM-ext-1 and GCM-ext-2, and GCM-256.

Mode	Block size	Regular		Nonce misuse resilient		Proof/Bound
		Conf.	Auth.	Conf.	Auth.	
GCM	128	✓	✓	✓	✗	[NOM15, ADL17], Sect. 3.1
GCM-ext-1	256	✓	✓	✓	✗	[NOM15, ADL17], Sect. 3.1
GCM-ext-2	256	✓	✓	✓	✗	[NOM15, ADL17], Sect. 3.1
GCM-256	256	✓	✓	✓	✓	Sect. 4

In words, not only the masking value Z is derived based on the nonce *but also the GHASH subkey*. We dub the function GCM-256, and it is depicted in Figure 3. Note that the idea to also have the GHASH subkey depend on the nonce is not new: it already appears a similar way in ChaCha20-Poly1305 [Ber05, Ber08] (RFC 8439 [NL18]). However, while this idea was impossible to be executed in GCM with AES (unless an extra block cipher call would be made), this potential feature is enabled by the planned standardization of Rijndael-256-256. Because of this, it makes sense, and it is a highly relevant problem, to thoroughly analyze the resulting GCM-256 construction.

We perform that analysis of GCM-256 in this work. Concretely, we prove that GCM-256 achieves the same bounds on confidentiality and authenticity as GCM. However, more interestingly, *as a bonus* we do not prove this in the classical model but under the nonce misuse resilience model of Ashur et al. [ADL17]. This property basically implies that nonce misuse breaks the security of the scheme *for that nonce* only, but evaluations for any new nonce are still confidential and authentic. Note that this was not the case for the original GCM, as nonce misuse revealed the nonce invariant subkey L . Most importantly, GCM-256 achieves confidentiality and authenticity under nonce misuse resilience whereas the earlier version of GCM only achieve confidentiality under nonce misuse resilience [ADL17, Section 4.2].

The security results are summarized in Table 1. Here, it should be noted that GCM is specifically for 128-bit blocks whereas GCM-ext-1, GCM-ext-2, and GCM-256 are for 256-bit blocks. These first three schemes are equally efficient in the number of block cipher evaluations, while GCM-256 saves one block cipher evaluation.

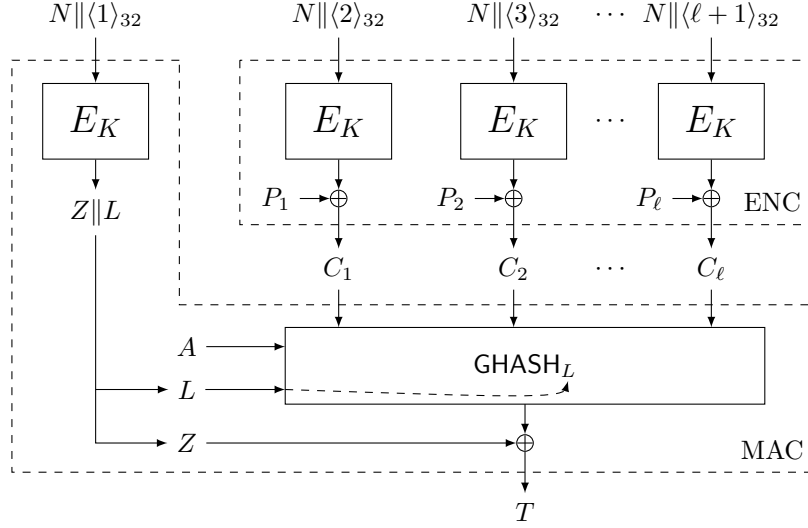


Figure 3: The GCM-256 authenticated encryption scheme. The function GHASH_L is depicted in Figure 4. We refer to Algorithm 2 for the pseudocode.

1.2 Related Constructions

Before proceeding, we briefly highlight two GCM-related constructions. The first one is GCM-RUP [ADL17], a variant of GCM that achieves authenticity under release of unverified plaintext by processing the output of GHASH_L using a tweakable block cipher construction on top of E . The second one is GCM-SST [CMM25], a variant of GCM that solves the suboptimal $2^t/\ell$ security term against forgeries [Fer05, NGR05] by replacing the universal hash function by a cascade of two hash functions with different subkeys, so as to allow for strengthened security under shorter tags. In fact, Inoue et al. [IJMM25] proved that this strengthened security is indeed achieved, and they additionally prove nonce misuse resilience of GCM-SST. In our work, we do not focus on release of unverified plaintext, and neither do we specifically target improved security under specifically short tags.

One may also target nonce misuse resistance: In this model (cf., Section 2.3), evaluations of the scheme for repeated nonces are still secure. However, these schemes are always more expensive, and they are at least two-pass, meaning that one has to go over the entire data twice. An example inspired by GCM is GCM-SIV [GL15]. In our work, we aim for a minimal adjustment over GCM that achieves strengthened security.

Likewise, a different angle of research that salvages the problem of short nonces is to derive a subkey from the master key and the nonce, actually similar to a technique introduced in GCM-SIV [GL15]. Several works have introduced this recently [Val25, BDD25, GR25, KHEC25], however, also this approach consists of a rather structural adjustment to GCM, whereas we aim to minimize the adjustment.

1.3 Outline

We start with preliminaries in Section 2. The original GCM construction, and a discussion of the meaning of its bounds for GCM-ext-1 and GCM-ext-2, is given in Section 3. Then, we define GCM-256 in Section 4, prove its nonce misuse resilient confidentiality in Section 4.1 and its authenticity in Section 4.2. We conclude in Section 5.

2 Preliminaries

Let $\mathbb{N}$ denote the set of natural numbers, $\{0,1\}^*$ the set of binary strings of arbitrary length, and for $n \in \mathbb{N}$ we denote by $\{0,1\}^n$ the set of binary strings of length n . We abbreviate $\{1, \dots, n\}$ as $[n]$. Given a string $A \in \{0,1\}^n$ and $\ell \leq n$, we define $\text{left}_\ell(A)$ to be $A[1 : \ell]$ and $\text{right}_\ell(A)$ to be $A[n - \ell + 1 : n]$. For $n \in \mathbb{N}$ and $i \in \{0, \dots, 2^n - 1\}$, we denote by $\langle i \rangle_n$ the encoding of i into a binary string of length n .

Given $n \in \mathbb{N}$ and a string $A \in \{0,1\}^{2^n}$, we denote by $B, C \stackrel{\text{par}}{\leftarrow} A$ the parsing of A into the bit strings $B, C \in \{0,1\}^n$ such that $A = B \| C$. We alternatively write $B = \text{left}_n(A)$ and $C = \text{right}_n(A)$. Furthermore, for $A \in \{0,1\}^*$, we denote by $(A_1, \dots, A_m) \stackrel{\text{pad}, n}{\leftarrow} A$ the procedure that first pads A with 0s so that it is of length a multiple of n bits, and outputs the unique strings $\{A_i \in \{0,1\}^n\}_{i=1}^m$ satisfying $A = \parallel_{i=1}^m A_i$.

We denote by fun_n (respectively perm_n) the set of functions (respectively permutations) from $\{0,1\}^n$ to $\{0,1\}^n$. If $\mathcal{S}$ is a finite set, we write $x \leftarrow \$ \mathcal{S}$ to denote that x is sampled uniformly from $\mathcal{S}$. Furthermore, we will use $A := B$ to denote that a variable is being defined.

2.1 Block Ciphers

Let $\mathcal{K}$ be a finite set and $n \in \mathbb{N}$. A block cipher with key space $\mathcal{K}$ and message space $\{0,1\}^n$ is a keyed function $E : \mathcal{K} \times \{0,1\}^n \rightarrow \{0,1\}^n$ such that for any $K \in \mathcal{K}$, $E(K, \cdot) = E_K(\cdot)$ is a permutation over $\{0,1\}^n$. In our work, block ciphers are only evaluated in forward direction, and we will require that they achieve “high” PRP security, as defined as follows.

Definition 1 (Pseudorandom Permutation (PRP) Security). Let $E : \mathcal{K} \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a block cipher and let $\mathcal{A}$ be a distinguisher who makes queries to the encryption routine of E_K , i.e., performs a chosen-plaintext attack with a uniformly sampled key $K \leftarrow \$ \mathcal{K}$. The pseudorandom permutation (PRP) advantage of E against $\mathcal{A}$ is defined as

$$\text{Adv}_E^{\text{PRP}}(\mathcal{A}) := |\Pr[\mathcal{A}^{E_K} \rightarrow 1] - \Pr[\mathcal{A}^P \rightarrow 1]|, \quad (4)$$

where $P \leftarrow \$ \text{perm}_n$.

2.2 Universal Hashing

Let $\mathcal{K}$ be a finite set, $\mathcal{X} \subseteq \{0,1\}^*$, and $n \in \mathbb{N}$. We will consider a keyed hash function $H : \mathcal{K} \times \mathcal{X} \rightarrow \{0,1\}^n$, and require it to satisfy almost uniformity and almost XOR universality.

Definition 2 (Almost Uniformity). Let $H : \mathcal{K} \times \mathcal{X} \rightarrow \{0,1\}^n$ be a hash function. It is said to be δ -almost uniform for $\delta \in (0, 1]$ if

$$\Pr_{K \leftarrow \$ \mathcal{K}}[H_K(X) = Y] \leq \delta, \quad \forall X \in \mathcal{X}, \forall Y \in \{0,1\}^n. \quad (5)$$

Definition 3 (Almost (XOR) Universality). Let $H : \mathcal{K} \times \mathcal{X} \rightarrow \{0,1\}^n$ be a hash function, it is said to be ϵ -almost XOR universal for $\epsilon \in [0, 1]$ if

$$\Pr_{K \leftarrow \$ \mathcal{K}}[H_K(X) \oplus H_K(X') = Y] \leq \epsilon, \quad \forall X \neq X' \in \mathcal{X}, \forall Y \in \{0,1\}^n. \quad (6)$$

We say that the set $\mathcal{K}$ is said to be *suffix-free* if for any $X \neq X' \in \mathcal{K}$, X' is not a suffix of X , more precisely, there is no $Y \in \{0,1\}^*$ so that $X = Y \| X'$.

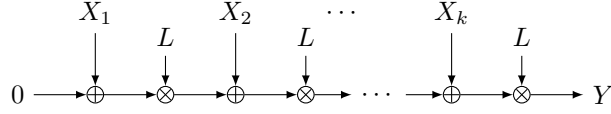


Figure 4: The $\text{GHASH}^{\text{raw}}$ universal hash function, operating with a key L and an *injectively padded* message $X_1\|X_2\|\dots\|X_k$. Note that we denote $X \cdot L$ by $X \otimes L$ for clarity.

2.2.1 GHASH

Let $p(x)$ be an irreducible polynomial over $\mathbb{F}_2$ of degree n . Addition and multiplication of polynomials modulo $p(x)$ defines a finite field $\mathbb{F}_{2^n}$ and we can identify elements of $\{0, 1\}^n$ as elements in this finite field. Henceforth, we describe GHASH over bit strings.

We first describe the raw version on padded bits, denoted $\text{GHASH}^{\text{raw}}$ (noting that the GHASH as used in GCM puts a padding layer on top of it). Fix some $\ell \in \mathbb{N}^+$, let $\mathcal{K} = \{0, 1\}^n \setminus \{0^n\}$, $L \in \{0, 1\}^n$, and $\mathcal{X} \subseteq (\{0, 1\}^n)^{\leq \ell}$. Then, we define

$$\text{GHASH}^{\text{raw}}(L, X) = \text{GHASH}_L^{\text{raw}}(X) = X_1 \cdot L^\ell \oplus \dots \oplus X_\ell \cdot L, \quad (7)$$

where $(X_1, \dots, X_\ell) \xleftarrow{\$} X$. This function is depicted in Figure 4.

It is proven that $\text{GHASH}^{\text{raw}}$ achieves good almost uniformity and almost XOR universality [Tay93, MV04].

Lemma 1 (Security of $\text{GHASH}^{\text{raw}}$ [Tay93, MV04]). *The function $\text{GHASH}^{\text{raw}}$ of (7) is δ -almost uniform for $\delta = \frac{\ell}{2^n}$. If $\mathcal{K}$ is suffix-free, it is ϵ -almost XOR universal for $\epsilon = \frac{\ell}{2^n}$.*

We remark that suffix-freeness of $\mathcal{K}$ is required to avoid attacks of Shen et al. [SSW23].

GCM uses $\text{GHASH}^{\text{raw}}$ in a particular way, and this is the function we dub GHASH. It is defined on three inputs: a key $L \in \mathcal{K}$, and two arbitrary-length bit strings $A, C \in \{0, 1\}^*$ such that $\lceil |A|/n \rceil + \lceil |C|/n \rceil + 1 \leq \ell$. Concretely, we define

$$\text{GHASH}_L(A, C) = \text{GHASH}_L^{\text{raw}}(\text{pad}_n(A) \| \text{pad}_n(C) \| |A|_{64} \| |C|_{64}). \quad (8)$$

where $|\cdot|_{64}$ denotes the length encoded as a 64-bit string. As the padding of A and C into the input to $\text{GHASH}_L^{\text{raw}}$ is injective, the results of Lemma 1 carry over.

2.3 Authenticated Encryption

Let $\mathcal{K}$, $\mathcal{N}$, and $\mathcal{A}$ be sets, and $t \in \mathbb{N}^+$. An authenticated encryption scheme is a pair of functions $\mathbf{E} = (\mathbf{E}.\text{Enc}, \mathbf{E}.\text{Dec})$ such that $\mathbf{E}.\text{Enc} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \{0, 1\}^* \rightarrow \{0, 1\}^* \times \{0, 1\}^t$ is the encryption function that takes secret key $K \in \mathcal{K}$, nonce $N \in \mathcal{N}$, associated data $A \in \mathcal{A}$, and plaintext $P \in \{0, 1\}^*$ and returns a ciphertext $C \in \{0, 1\}^*$ and tag $T \in \{0, 1\}^t$, and $\mathbf{E}.\text{Dec} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \{0, 1\}^* \times \{0, 1\}^t \rightarrow \{0, 1\}^* \cup \{\perp\}$ takes as input the secret key $K \in \mathcal{K}$, nonce $N \in \mathcal{N}$, associated data $A \in \mathcal{A}$, ciphertext $C \in \{0, 1\}^*$, and tag $T \in \{0, 1\}^t$, and returns either a plaintext $P \in \{0, 1\}^*$ or the symbol $\perp$.

The traditional security notions for authenticated encryption schemes are confidentiality (**conf**) and authenticity (**auth**), where the adversary may arbitrarily choose nonces for encryption queries, provided that all nonce values are distinct (see, e.g., [RBBK01]). Such an adversary is called nonce respecting. We summarize these notions below.

Definition 4 (Nonce Respecting Confidentiality (**conf**)). Let $\mathbf{E} = (\mathbf{E}.\text{Enc}, \mathbf{E}.\text{Dec})$ be an authenticated encryption scheme and let $\mathcal{A}$ be a distinguisher against $\mathbf{E}$ with uniformly sampled key $K \leftarrow \$ \mathcal{K}$. The nonce respecting confidentiality (**conf**) advantage of $\mathbf{E}$ against $\mathcal{A}$ is defined as

$$\text{Adv}_{\mathbf{E}}^{\text{conf}}(\mathcal{A}) := \left| \Pr[\mathcal{A}^{\mathbf{E}.\text{Enc}_K} = 1] - \Pr[\mathcal{A}^\$ = 1] \right|,$$

where $\$$ is an oracle that returns a random string of $|P| + t$ bits for any query (N, A, P) . $\mathcal{A}$ may not re-use nonces.

Definition 5 (Nonce Respecting Authenticity (auth)). Let $E = (E.\text{Enc}, E.\text{Dec})$ be an authenticated encryption scheme and let $\mathcal{A}$ be an adversary against E with a uniformly sampled key $K \leftarrow \$ \mathcal{K}$. The nonce respecting authenticity (auth) advantage of E against $\mathcal{A}$ is defined as

$$\text{Adv}_E^{\text{auth}}(\mathcal{A}) := \Pr[\mathcal{A}^{E.\text{Enc}_K, E.\text{Dec}_K} \text{ forges}].$$

We say $\mathcal{A}$ forges if it receives $P \neq \perp$ from the decryption oracle. $\mathcal{A}$ may not re-use nonces in encryption queries. In addition, a trivial forgery is not allowed, namely if $\mathcal{A}$ receives (C, T) from an encryption query (N, A, P) , $\mathcal{A}$ should not make a decryption query (N, A, C, T) after that query.

However, in certain settings, maintaining a nonce is problematic. To salvage this, Rogaway and Shrimpton [RS06] introduced the concept of nonce misuse resistance (nmr) to address scenarios where nonces may be repeated. While their security notions provide the strongest possible guarantees, schemes achieving nmr require heavier computation than nonce respecting authenticated encryption schemes. In particular, the confidentiality notion under nonce misuse resistance necessitates offline encryption: the first ciphertext bit can only be output after the algorithm has processed the entire input (N, A, P) [HRRV15].

Ashur et al. [ADL17] introduced a relaxed concept called Nonce Misuse resiLience (nml). The idea is that, even if nonces are misused, security for *new* nonces is still guaranteed. More technically, under nml, the confidentiality notion splits encryption queries into two categories: challenge queries and non-challenge queries. An adversary $\mathcal{A}$ is allowed to query the real authenticated encryption scheme (real world), or an idealized primitive (ideal world). Non-challenge queries are always answered with the real-world oracle, while the challenge queries are answered with the corresponding real or ideal primitive. In a challenge query, nonce reuse is not allowed, whereas in a non-challenge query, the nonce may be repeated. During authenticity, there are no challenge oracles, instead $\mathcal{A}$ has access to both an encryption and a decryption oracle, and the objective of $\mathcal{A}$ is to forge, that is, fabricate a fresh valid authenticated ciphertext. We remark that the fresh part is necessary to prevent trivial attacks. Nonce misuse is allowed during encryption, however, decryption queries with repeated nonces are not allowed. *A fortiori* forgeries must have fresh nonces.

Formally, we consider the following nonce misuse resilience security notions on E .

Definition 6 (Nonce Misuse Resilience Confidentiality (nml-conf)). Let $E = (E.\text{Enc}, E.\text{Dec})$ be an authenticated encryption scheme and let $\mathcal{A}$ be a distinguisher against E with uniformly sampled key $K \leftarrow \$ \mathcal{K}$. The nonce misuse resilience confidentiality (nml-conf) advantage of E against $\mathcal{A}$ is defined as

$$\text{Adv}_E^{\text{nml-conf}}(\mathcal{A}) := \left| \Pr[\mathcal{A}^{E.\text{Enc}_K, \$} = 1] - \Pr[\mathcal{A}^{\$, E.\text{Enc}_K} = 1] \right|,$$

where $\$$ is an oracle that returns a random string of $|P| + t$ bits for any query (N, A, P) . We denote by $O_1 \in \{E.\text{Enc}_K, \$\}$ the challenge oracle and by $O_2 = E.\text{Enc}_K$ the non-challenge oracle. $\mathcal{A}$ may re-use nonces with its non-challenge oracle O_2 , but it may not re-use nonces with its challenge oracle O_1 . Furthermore, $\mathcal{A}$ may not query one of its oracles with a nonce that it already used for the other one.

Definition 7 (Nonce Misuse Resilience Authenticity (nml-auth)). Let $E = (E.\text{Enc}, E.\text{Dec})$ be an authenticated encryption scheme and let $\mathcal{A}$ be an adversary against E with a uniformly sampled key $K \leftarrow \$ \mathcal{K}$. The nonce misuse resilience authenticity (nml-auth) advantage of E against $\mathcal{A}$ is defined as

$$\text{Adv}_E^{\text{nml-auth}}(\mathcal{A}) := \Pr[\mathcal{A}^{E.\text{Enc}_K, E.\text{Dec}_K} \text{ forges}].$$

We say $\mathcal{A}$ forges if it receives $P \neq \perp$ from the decryption oracle. Here, nonces in encryption queries may repeat, but any nonce value in a decryption query must appear at most once in encryption queries. In addition, a trivial forgery is not allowed, namely if $\mathcal{A}$ receives (C, T) from an encryption query (N, A, P) , $\mathcal{A}$ should not make a decryption query (N, A, C, T) after that query.

3 Original GCM

We will restrict our focus to the case of GCM for 96-bit nonce [Dwo07], but we describe it for the general setting that the tag may be truncated to t bits. The specification is given in Algorithm 1 and depicted (without tag truncation) in Figure 1.

Algorithm 1 GCM for 96-bit nonces

1: function GCM.Enc (K, N, A, P)	1: function GCM.Dec (K, N, A, C, T)
2: $(P_1, \dots, P_m) \xleftarrow{\$} P$	2: $(C_1, \dots, C_m) \xleftarrow{\$} C$
3: for $i = 1, \dots, m$ do	3: for $i = 1, \dots, m$ do
4: $C_i \leftarrow P_i \oplus E_K(N \parallel \langle i+1 \rangle_{32})$	4: $P_i \leftarrow C_i \oplus E_K(N \parallel \langle i+1 \rangle_{32})$
5: $C \leftarrow \text{left}_{ P }(C_1 \parallel \dots \parallel C_m)$	5: $P \leftarrow \text{left}_{ C }(P_1 \parallel \dots \parallel P_m)$
6: $L \leftarrow E_K(0^n)$	6: $L \leftarrow E_K(0^n)$
7: $Z \leftarrow E_K(N \parallel \langle 1 \rangle_{32})$	7: $Z \leftarrow E_K(N \parallel \langle 1 \rangle_{32})$
8: $H \leftarrow \text{GHASH}_L(A, C)$	8: $H \leftarrow \text{GHASH}_L(A, C)$
9: $T \leftarrow \text{left}_t(Z \oplus H)$	9: if $T = \text{left}_t(Z \oplus H)$ then return P
10: return (C, T)	10: else return $\perp$

3.1 Existing Security of GCM

Following McGrew and Viega [MV04] and Iwata et al. [IOM12], Niwa et al. [NOMI15] proved the following bounds on the nonce respecting confidentiality and authenticity of GCM with 96-bit nonces. We adopt the bounds of Inoue et al. [IJMM25], who tailored the ones of Niwa et al. to the specific version of GCM with 96-bit nonces.

Theorem 1 (Confidentiality of GCM [NOMI15, IJMM25]). *Consider an n -bit block cipher E , and consider any conf adversary $\mathcal{A}$ that runs in time θ and makes q queries where each query has input-data of length at most ℓ blocks of at most σ blocks of data across all queries. Then, there exists a PRP adversary $\mathcal{B}$ that runs in time $\mathcal{O}(\sigma\theta)$ and makes $(q + \sigma + 1)$ encryption queries such that*

$$\text{Adv}_{\text{GCM}[E]}^{\text{conf}}(\mathcal{A}) \leq \text{Adv}_E^{\text{prp}}(\mathcal{B}) + \frac{(\sigma + q + 1)^2}{2^{n+1}} + \frac{2(\sigma + q)}{2^n}.$$

Theorem 2 (Authenticity of GCM [NOMI15, IJMM25]). *Consider an n -bit block cipher E , and consider any auth $\mathcal{A}$ that runs in time θ and makes q encryption queries and ν decryption queries where each query has input-data of length at most ℓ blocks of at most σ blocks of data across all queries. Then, there exists a PRP adversary $\mathcal{B}$ that runs in time $\mathcal{O}(\sigma\theta)$ and makes $(q + \nu + \sigma + 1)$ encryption queries such that*

$$\text{Adv}_{\text{GCM}[E]}^{\text{auth}}(\mathcal{A}) \leq \text{Adv}_E^{\text{prp}}(\mathcal{B}) + \frac{(\sigma + q + \nu + 1)^2}{2^{n+1}} + \frac{2(\sigma + q + \nu)}{2^n} + \frac{\nu(\ell + 1)}{2^t}.$$

3.2 Application to GCM-ext-1 and GCM-ext-2

Clearly, the specification of GCM-ext-1 of Section 1 is identical to that of GCM of Algorithm 1 and depicted (without tag truncation) in Figure 1, as the functions are identical though with n twice as large. This also means that the security bounds of Theorems 1 and 2 apply to GCM-ext-1 as well.

The specification of GCM-ext-2 of Section 1 differs from Algorithm 1 only in the following two changes:

- Line 6 for both encryption and decryption get replaced with $L \leftarrow \text{left}_{n/2}(E_K(0^n))$.
- Line 7 for both encryption and decryption get replaced with $Z \leftarrow \text{left}_{n/2}(E_K(N \parallel \langle 1 \rangle_{32}))$.

However, the security proofs of Theorems 1 and 2 first apply a PRP-PRF switch on E , de facto treating E_K as a random function, which results in the observation that truncation to obtain L and Z does not change anything, and these theorems carry over though with a shorter ($t = 128$)-bit tag.

3.3 Insecurity of GCM and Variants Under Nonce Misuse

However, it is well-known that under nonce misuse, GCM is insecure. Indeed, suppose one observes two encryptions (N, A, C, T) and (N, A', C', T') under the same nonce N . By design, we know that $C \oplus C' = P \oplus P'$ for the real world E.Enc , which would not be the case for the ideal world $\$$, thus breaking the confidentiality of GCM in two queries. Additionally, from these two evaluations, one can observe that (in the notation of Algorithm 1 with $'$'s added for the intermediate values of the second queries)

$$T \oplus T' = Z \oplus \text{GHASH}_L(A, C) \oplus Z' \oplus \text{GHASH}_L(A', C').$$

However, as $N = N'$, necessarily $Z = Z'$, and one obtains that

$$T \oplus T' = \text{GHASH}_L^{\text{raw}} \left((\text{pad}_n(A) \parallel \text{pad}_n(C) \parallel |A|_{64} \parallel |C|_{64}) \oplus (\text{pad}_n(A') \parallel \text{pad}_n(C') \parallel |A'|_{64} \parallel |C'|_{64}) \right).$$

From this, L can be easily deduced.

These attacks apply to GCM, GCM-ext-1, and GCM-ext-2, demonstrating that all three fall victim under the nonce misuse attack.

4 GCM-256

Inspired by the attacks of Section 3.3, we present and formally describe GCM-256 in Algorithm 2. We refer to Figure 3 for a depiction of the mode (without final truncation). The change compared to GCM of Algorithm 1 is surprisingly simple, it only consists of combining the generation of Z and L in line 6 for both encryption and decryption, but enables much stronger security guarantees for GCM-256.

We remark that this version of GCM is *specifically to be used with a block cipher* with block size $n = 256$ bits (and probably key size $k = 256$) and with GHASH on 128-bit key and blocks. Concretely, this algorithm has a block size $n = 256$, a nonce size $s = 224$, a counter size $n - s = 32$, and a tag size $t = 128$.

In what remains of this section, we prove nonce misuse resilience confidentiality in Section 4.1 and authenticity in Section 4.2. Note that nonce respecting security follows by implication.

Algorithm 2 GCM-256

<pre>1: function GCM-256.Enc(K, N, A, P) 2: $(P_1, \dots, P_m) \xleftarrow{\\$} P$ 3: for $i = 1, \dots, m$ do 4: $C_i \leftarrow P_i \oplus E_K(N \parallel \langle i+1 \rangle_{32})$ 5: $C \leftarrow \text{left}_{ P }(C_1 \parallel \dots \parallel C_m)$ 6: $Z \parallel L \xleftarrow{\\$} E_K(N \parallel \langle 1 \rangle_{32})$ 7: $H \leftarrow \text{GHASH}_L(A, C)$ 8: $T \leftarrow \text{left}_t(Z \oplus H)$ 9: return (C, T)</pre>	<pre>1: function GCM-256.Dec(K, N, A, C, T) 2: $(C_1, \dots, C_m) \xleftarrow{\\$} C$ 3: for $i = 1, \dots, m$ do 4: $P_i \leftarrow C_i \oplus E_K(N \parallel \langle i+1 \rangle_{32})$ 5: $P \leftarrow \text{left}_{ C }(P_1 \parallel \dots \parallel P_m)$ 6: $Z \parallel L \xleftarrow{\\$} E_K(N \parallel \langle 1 \rangle_{32})$ 7: $H \leftarrow \text{GHASH}_L(A, C)$ 8: if $T = \text{left}_t(Z \oplus H)$ then return P 9: else return $\perp$</pre>
---	---

4.1 Nonce Misuse Resilience of GCM: Confidentiality

We start by proving the nonce misuse resilience confidentiality of GCM-256. The proof resembles a bit the analyses of Niwa et al. [NOMI15] on GCM and Inoue et al. [IJMM25] on GCM-SST.

Theorem 3. *Given any nml-conf $\mathcal{A}$ that runs in time θ and makes q queries consisting of q_1 challenge and $q_2 = q - q_1$ non-challenge encryption queries, where each query has input-data of length at most ℓ blocks of at most σ blocks of data across all queries. Then, there exists a PRP adversary $\mathcal{B}$ that runs in time $\mathcal{O}(\sigma\theta)$ and makes $q + \sigma$ encryption queries such that*

$$\text{Adv}_{\text{GCM-256}}^{\text{nml-conf}}(\mathcal{A}) \leq \text{Adv}_E^{\text{prp}}(\mathcal{B}) + \frac{(\sigma + q)^2}{2^{n+1}}.$$

Proof. Let K be the key sampled uniformly at the beginning of the game, P be sampled uniformly from perm_n . By abusing notation, we denote by $\text{GCM-256}[P]$ the variant of the authenticated encryption scheme GCM-256, but that does not use a key and instead replaces the evaluations of E_K by P . Then, by a hybrid argument:

$$\text{Adv}_{\text{GCM-256}}^{\text{nml-conf}}(\mathcal{A}) \leq \text{Adv}_E^{\text{prp}}(\mathcal{B}) + \text{Adv}_{\text{GCM-256}[P]}^{\text{nml-conf}}(\mathcal{C}), \quad (9)$$

where $\mathcal{B}$ denotes an adversary running in time $\mathcal{O}(\sigma\theta)$ and making at most $q + \sigma$ queries, and $\mathcal{C}$ denotes a computationally-unbounded deterministic adversary, with the same query complexity as $\mathcal{A}$.

Then, we sample F uniformly at random from fun_n , and by the PRP-PRF switch [BR06], we have

$$\text{Adv}_{\text{GCM-256}[P]}^{\text{nml-conf}}(\mathcal{C}) \leq \frac{(\sigma + q)^2}{2^{n+1}} + \text{Adv}_{\text{GCM-256}[F]}^{\text{nml-conf}}(\mathcal{D}), \quad (10)$$

where $\mathcal{D}$ has the same time and query complexity as $\mathcal{C}$.

Let τ and $\bar{\tau}$ be, respectively, the transcripts for the challenge and non-challenge queries containing the keystream, that is:

$$\begin{aligned} \tau &= \left(Z_i^j : j \in [q_1], i \in \{0, \dots, m_j\} \right), \\ \bar{\tau} &= \left(\bar{Z}_i^j : j \in [q_2], i \in \{0, \dots, \bar{m}_j\} \right), \end{aligned}$$

where $Z_i^j = F(N_j \parallel \langle i+1 \rangle_{32})$ and similar for $\bar{Z}_i^j$.

Observe that $F(N\|\cdot)$ is uniformly distributed and independent of $F(N'\|\cdot)$ whenever $N \neq N'$, and since nonce repetition is not allowed for challenge queries, the values in τ are uniformly and independently distributed. Additionally, the non-challenge oracle is never queried for a nonce queried to the challenge oracle, implying that τ is independent of $\bar{\tau}$.

Concluding, the outputs of the calls to the challenge oracle, $\text{GCM-256}[F]$ in the real world and $\$$ in the ideal world, are both uniformly random and independent of the non-challenge oracle, and thus

$$\text{Adv}_{\text{GCM-256}[F]}^{\text{nml-conf}}(\mathcal{D}) = 0.$$

Together with (9) and (10), this completes the proof. $\square$

4.2 Nonce Misuse Resilience of GCM: Authenticity

Next, we prove nonce misuse resilience authenticity of GCM-256. The proof builds further upon the ideas of the proof of Theorem 3, with some difficulties incurring in the coverage of forgery attempts.

Theorem 4. *Consider any nml-auth $\mathcal{A}$ that runs in time θ and makes q encryption queries consisting of q_1 nonce respecting queries and $q_2 = q - q_1$ nonce misusing encryption queries, and ν decryption queries where each query has input-data of length at most ℓ blocks of at most σ blocks of data across all queries. Then, there exists a PRP adversary $\mathcal{B}$ that runs in time $\mathcal{O}(\sigma\theta)$ and makes $q + \nu + \sigma$ encryption queries such that*

$$\text{Adv}_{\text{GCM-256}}^{\text{nml-auth}}(\mathcal{A}) \leq \text{Adv}_E^{\text{prp}}(\mathcal{B}) + \frac{(\sigma + q + \nu)^2}{2^{n+1}} + \frac{\nu\ell}{2^t}.$$

Proof. The first part of this proof is similar to that of the proof of Theorem 3. Concretely, P , F , $\text{GCM-256}[P]$, and $\text{GCM-256}[F]$ be as defined in the proof of Theorem 3. Then, we first replace E_K with the randomly sampled permutation P , incurring a PRP cost, and then we replace P with a randomly sampled function F , and putting that together, we obtain

$$\text{Adv}_{\text{GCM-256}}^{\text{nml-auth}}(\mathcal{A}) \leq \text{Adv}_E^{\text{prp}}(\mathcal{B}) + \frac{(\sigma + q + \nu)^2}{2^{n+1}} + \text{Adv}_{\text{GCM-256}[F]}^{\text{nml-auth}}(\mathcal{D}), \quad (11)$$

$\mathcal{B}$ denotes an adversary running in time $\mathcal{O}(\sigma\theta)$ and making at most $q + \nu + \sigma$ queries, and $\mathcal{D}$ denotes a computationally-unbounded deterministic adversary, with the same query complexity as $\mathcal{A}$. Let τ , $\bar{\tau}$, and $\tilde{\tau}$ be, respectively, the transcripts containing the key streams for the nonce respecting, nonce misusing and decryption queries:

$$\begin{aligned} \tau &= \left(Z_i^j : j \in [q_1], i \in \{0, \dots, m_j\} \right), \\ \bar{\tau} &= \left(\bar{Z}_i^j : j \in [q_2], i \in \{0, \dots, \bar{m}_j\} \right), \\ \tilde{\tau} &= \left(\tilde{Z}_i^j : j \in [\nu], i \in \{0, \dots, \tilde{m}_j\} \right), \end{aligned}$$

where again $Z_i^j = F(N_j \| \langle i+1 \rangle_{32})$ and similar for $\bar{Z}_i^j$ and $\tilde{Z}_i^j$.

Then, we repeat the argument of the proof of Theorem 3 to conclude that $\tau \cup \tilde{\tau}$ is independent of $\bar{\tau}$. This property holds as the nonces in the queries corresponding to elements in $\tau \cup \tilde{\tau}$ are, by assumption, disjoint to those to $\bar{\tau}$. Hence, we may disregard nonce misusing queries.

We now introduce some notation. Let $\mathcal{T}$ denote the set of all transcripts reachable by the interaction of $\mathcal{D}$ with the encryption and decryption oracles, and $\mathcal{T}_{\text{WIN}}$ denote the subset of transcripts in which $\mathcal{D}$ succeeds in making a valid forgery. Since $\mathcal{D}$ stops the

moment it finds a valid forgery, or it runs out of queries, each $\tau \in \mathcal{T}_{\text{WIN}}$ contains at most $q_1 + \nu$ entries, and must end with a valid decryption query.

For convenience, for each tuple in the transcript $\mathcal{T}_{\text{WIN}}$, we append the key streams. In other words, for an encryption tuple (N^j, A^j, C^j, T^j) we append $Z^j = \parallel_{i \geq 0} Z_i^j$, and for a decryption tuple $(\tilde{N}^j, \tilde{A}^j, \tilde{C}^j, \tilde{T}^j)$ we append $\tilde{Z}^j = \parallel_{i \geq 0} \tilde{Z}_i^j$.

Let X denote the transcript random variable generated by the interaction of $\mathcal{D}$ with the encryption and decryption queries. Additionally, we let $\mathcal{T}_{\text{WIN}}^j$ be the subset of $\mathcal{T}_{\text{WIN}}$ denoting the transcripts in which the successful forgery is at the j -th decryption query. By definition, we have

$$\text{Adv}_{\text{GCM-256}[F]}^{\text{nml-auth}}(\mathcal{D}) = \Pr[X \in \mathcal{T}_{\text{WIN}}] = \sum_{j=1}^{\nu} \Pr[X \in \mathcal{T}_{\text{WIN}}^j]. \quad (12)$$

We fix $\tau \in \mathcal{T}_{\text{WIN}}^j$, and by definition, the j -th query of τ is a successful decryption query: $(\tilde{N}, \tilde{A}, \tilde{C}, \tilde{T}, \tilde{Z})$. Let $\mathcal{N}$ be the set of nonces used during encryption up to the point of the j -th decryption query.

We distinguish two cases:

1. $\tilde{N} \notin \mathcal{N}$. *A fortiori*, $\tilde{Z}_0$ is uniform at random and independent of all the keystream blocks appearing in the encryption queries. Therefore, $\tilde{T}$, which equals

$$\tilde{T} = \text{left}_t(\text{left}_{128}(\tilde{Z}_0) \oplus \text{GHASH}_{\text{right}_{128}(\tilde{Z}_0)}(\tilde{A}, \tilde{C})),$$

is also uniformly random (cf., Figure 3), and hence the probability that it is a valid forgery is exactly $1/2^t$;

2. $\tilde{N} \in \mathcal{N}$. Then, there exists a unique encryption query-response tuple (N, A, P, C, T) such that $\tilde{N} = N$. The forgery τ is successful if and only if

$$\text{left}_t(\text{GHASH}_L(\tilde{A}, \tilde{C}) \oplus \text{GHASH}_L(A, C)) = \tilde{T} \oplus T,$$

where $L = \text{right}_{128}(\tilde{Z}_0) = \text{right}_{128}(Z_0)$. By Lemma 1, this happens with probability at most $\ell/2^t$ (noting that the almost XOR universality result of Lemma 1 also holds after truncation [BJKS93]).

In either case, the j -th forgery attempt is successful with probability at most $\ell/2^t$:

$$\Pr[X \in \mathcal{T}_{\text{WIN}}^j] \leq \frac{\ell}{2^t}. \quad (13)$$

The result then follows from (11) to (13). $\square$

5 Conclusion

In this work, we introduced GCM-256, an authenticated encryption scheme based on the popular GCM, designed to exploit the larger block size of Rijndael-256-256, which NIST plans to standardize. In detail, the larger block size of Rijndael-256-256 enables the possibility to cleanly derive the masking value and the GHASH subkey from a single block cipher call. This adjustment clearly comes at essentially no additional cost, but we demonstrate that the improved version GCM-256 not only achieves “normal” confidentiality and authenticity, but also security under nonce misuse resilience. This way, GCM-256 solves a well-known problem that was present in GCM, namely that nonce reuse is devastating and allows an attacker to even retrieve subkey material.

Acknowledgements. Mario Marhuenda Beltrán and Bart Mennink are supported by the Netherlands Organisation for Scientific Research (NWO) under grant VI.Vidi.203.099.

References

- [ADL17] Tomer Ashur, Orr Dunkelman, and Atul Luykx. Boosting Authenticated Encryption Robustness with Minimal Modifications. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, Lecture Notes in Computer Science, pages 3–33. Springer, 2017.
- [BDD25] Ritam Bhaumik, Jean Paul Degabriele, and Chandranan Dhar. #Pencil: A Domain-Extended Committing BBB PRF for Strengthening GCM. Cryptology ePrint Archive, Report 2025/383, 2025. <http://eprint.iacr.org/2025/383>.
- [Ber05] Daniel J. Bernstein. The Poly1305-AES Message-Authentication Code. In Henri Gilbert and Helena Handschuh, editors, *Fast Software Encryption: 12th International Workshop, FSE 2005, Paris, France, February 21-23, 2005, Revised Selected Papers*, Lecture Notes in Computer Science, pages 32–49. Springer, 2005.
- [Ber08] Daniel J. Bernstein. The ChaCha family of stream ciphers, January 2008. <https://cr.yp.to/chacha.html>.
- [BJKS93] Jürgen Bierbrauer, Thomas Johansson, Gregory Kabatianskii, and Ben J. M. Smeets. On Families of Hash Functions via Geometric Codes and Concatenation. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, Lecture Notes in Computer Science, pages 331–342. Springer, 1993.
- [BN00] Mihir Bellare and Chanathip Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In Tatsuaki Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, Lecture Notes in Computer Science, pages 531–545. Springer, 2000.
- [BR06] Mihir Bellare and Phillip Rogaway. The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, Lecture Notes in Computer Science, pages 409–426. Springer, 2006.
- [Bra82] Gilles Brassard. On Computationally Secure Authentication Tags Requiring Short Secret Shared Keys. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology: Proceedings of CRYPTO '82, Santa Barbara, California, USA, August 23-25, 1982*, pages 79–86. Plenum Press, New York, 1982.
- [BZD⁺16] Hanno Böck, Aaron Zauner, Sean Devlin, Juraj Somorovsky, and Philipp Jovanovic. Nonce-Disrespecting Adversaries: Practical Forgery Attacks on GCM in TLS. In Natalie Silvanovich and Patrick Traynor, editors, *10th USENIX Workshop on Offensive Technologies, WOOT 16, Austin, TX, USA, August 8-9, 2016*. USENIX Association, 2016.

- [CMM25] Matt Campagna, Alexander Maximov, and John Preuß Mattsson. Galois Counter Mode with Secure Short Tags (GCM-SST). Internet-Draft draft-mattsson-cfrg-aes-gcm-sst-18, Internet Engineering Task Force, February 2025. <https://datatracker.ietf.org/doc/draft-mattsson-cfrg-aes-gcm-sst/18>.
- [Dwo07] Morris Dworkin. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. NIST Special Publication 800-38D, National Institute of Standards and Technology, U.S. Department of Commerce, 2007. <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf>.
- [Fer05] Niels Ferguson. Authentication Weaknesses in GCM. Public Comment to NIST, 2005. <http://csrc.nist.gov/groups/ST/toolkit/BCM/comments.html>.
- [GL15] Shay Gueron and Yehuda Lindell. GCM-SIV: Full Nonce Misuse-Resistant Authenticated Encryption at Under One Cycle per Byte. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, pages 109–119. ACM, 2015.
- [GR25] Shay Gueron and Thomas Ristenpart. DNDK: Combining nonce and key derivation for fast and scalable AEAD. Cryptology ePrint Archive, Report 2025/785, 2025. <http://eprint.iacr.org/2025/785>.
- [HRRV15] Viet Tung Hoang, Reza Reyhanitabar, Phillip Rogaway, and Damian Vizár. Online Authenticated-Encryption and its Nonce-Reuse Misuse-Resistance. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 493–517. Springer, 2015.
- [IEE06] IEEE Standard for Local and Metropolitan Area Networks Media Access Control (MAC) Security. IEEE Std 802.1AE-2006, 2006.
- [IJMM25] Akiko Inoue, Ashwin Jha, Bart Mennink, and Kazuhiko Minematsu. Generic Security of GCM-SST. In Marc Fischlin and Veelasha Moonsamy, editors, *Applied Cryptography and Network Security - 23rd International Conference, ACNS 2025, Munich, Germany, June 23-26, 2025, Proceedings, Part II*, Lecture Notes in Computer Science, pages 342–369. Springer, 2025.
- [IOM12] Tetsu Iwata, Keisuke Ohashi, and Kazuhiko Minematsu. Breaking and Repairing GCM Security Proofs. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, Lecture Notes in Computer Science, pages 31–49. Springer, 2012.
- [IS09] Kevin Igoe and Jerome Solinas. AES Galois Counter Mode for the Secure Shell Transport Layer Protocol. RFC 5647, August 2009. <https://www.rfc-editor.org/info/rfc5647>.
- [ISO09] Information Technology — Security Techniques — Authenticated Encryption, ISO/IEC 19772:2009. International Standard ISO/IEC 19772, 2009.
- [Jou06] Antoine Joux. Authentication Failures in NIST version of GCM. Public Comment to NIST, 2006. <http://csrc.nist.gov/groups/ST/toolkit/BCM/comments.html>.

- [KCCP23] Panos Kampanakis, Matt Campagna, Eric Crocket, and Adam Petcher. Practical Challenges with AES-GCM and the need for a new cipher. Third NIST Workshop on Block Cipher Modes of Operation 2023, October 2023. <https://csrc.nist.gov/csrc/media/Events/2023/third-workshop-on-block-cipher-modes-of-operation/documents/accepted-papers/Practical%20Challenges%20with%20AES-GCM.pdf>.
- [KHEC25] Panos Kampanakis, Shai Halevi, Nevine Maurice Ebeid, and Matt Campagna. Blockcipher-Based Key Commitment for Nonce-Derived Schemes. In Christina Boura, Atefeh Mashatan, and Ali Miri, editors, *Selected Areas in Cryptography - SAC 2025 - 32nd International Conference, Toronto, ON, Canada, August 13-15, 2025, Revised Selected Papers*, volume 16207 of *Lecture Notes in Computer Science*, pages 209–234. Springer, 2025.
- [MV04] David A. McGrew and John Viega. The Security and Performance of the Galois/Counter Mode (GCM) of Operation. In Anne Canteaut and Kapalee Viswanathan, editors, *Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings*, Lecture Notes in Computer Science, pages 343–355. Springer, 2004.
- [NGR05] Kaisa Nyberg, Henri Gilbert, and Matt Robshaw. Galois MAC with forgery probability close to ideal. Public Comment to NIST, 2005. https://csrc.nist.gov/CSRC/media/Projects/Block-Cipher-Techniques/documents/BCM/Comments/general-comments/papers/Nyberg_Gilbert_and_Robshaw.pdf.
- [NIS24] NIST. NIST Proposes to Standardize a Wider Variant of AES, December 2024. <https://csrc.nist.gov/News/2024/nist-proposes-to-standardize-wider-variant-of-aes>.
- [NIS26] NIST. Second Pre-Draft Call for Comments: GCM and GMAC Block Cipher Modes of Operation, June 2026. <https://csrc.nist.gov/pubs/sp/800/38/d/r1/2prd>.
- [NL18] Yoav Nir and Adam Langley. ChaCha20 and Poly1305 for IETF Protocols. RFC 8439, June 2018. <https://www.rfc-editor.org/info/rfc8439>.
- [NOMI15] Yuichi Niwa, Keisuke Ohashi, Kazuhiko Minematsu, and Tetsu Iwata. GCM Security Bounds Reconsidered. In Gregor Leander, editor, *Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers*, Lecture Notes in Computer Science, pages 385–407. Springer, 2015.
- [NSA10] National Security Agency, Internet Protocol Security (IPsec) Minimum Essential Interoperability Requirements, IPMEIR Version 1.0.0 Core, 2010. http://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml.
- [RBBK01] Phillip Rogaway, Mihir Bellare, John Black, and Ted Krovetz. OCB: a block-cipher mode of operation for efficient authenticated encryption. In Michael K. Reiter and Pierangela Samarati, editors, *CCS 2001, Proceedings of the 8th ACM Conference on Computer and Communications Security, Philadelphia, Pennsylvania, USA, November 6-8, 2001*, pages 196–205. ACM, 2001.
- [RS06] Phillip Rogaway and Thomas Shrimpton. A Provable-Security Treatment of the Key-Wrap Problem. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory*

and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, *Proceedings*, Lecture Notes in Computer Science, pages 373–390. Springer, 2006.

- [Sho96] Victor Shoup. On Fast and Provably Secure Message Authentication Based on Universal Hashing. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, Lecture Notes in Computer Science, pages 313–328. Springer, 1996.
- [SMC08] Joseph A. Salowey, David McGrew, and Abhijit Choudhury. AES Galois Counter Mode (GCM) Cipher Suites for TLS. RFC 5288, August 2008. <https://www.rfc-editor.org/info/rfc5288>.
- [SSW23] Yaobin Shen, François-Xavier Standaert, and Lei Wang. Forgery Attacks on Several Beyond-Birthday-Bound Secure MACs. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part III*, Lecture Notes in Computer Science, pages 169–189. Springer, 2023.
- [Tay93] Richard Taylor. An Integrity Check Value Algorithm for Stream Ciphers. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, Lecture Notes in Computer Science, pages 40–48. Springer, 1993.
- [Val25] Filippo Valsorda. The XAES-256-GCM extended-nonce AEAD, May 2025. <https://c2sp.org/XAES-256-GCM>.
- [VM05] John Viega and David McGrew. The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP). RFC 4106, June 2005. <https://www.rfc-editor.org/info/rfc4106>.
- [WC81] Mark N. Wegman and Larry Carter. New Hash Functions and Their Use in Authentication and Set Equality. *J. Comput. Syst. Sci.*, 22(3):265–279, 1981.
- [WPA24] WPA3 Specification Version 3.3. Wi-Fi Alliance, 2024. <https://www.wi-fi.org/system/files/WPA3%20Specification%20v3.3.pdf>.