

Refining Galois/Counter Mode for 256-Bit Blocks: Multilinear Authentication and Security Proofs

DOI: <https://doi.org/10.54654/isj.v1i27.6406>

Nguyen Tuan Anh*, Nguyen Bui Cuong, Kieu Viet Dich

Abstract—This paper investigates the feasibility of extending the authenticated encryption mode GCM to a 256-bit block size to increase the amount of authenticated data and enhance security in large-scale information processing systems. Based on an analysis of the limitations of the standard GCM construction, particularly the restriction on the amount of data processed per invocation and the weaknesses arising from the linearity of the GHASH function, we propose a GCM-256 variant employing a 256-bit block cipher together with a multilinear authentication mechanism, replacing the conventional linear GHASH function.

Tóm tắt— Bài báo nghiên cứu khả năng mở rộng chế độ mã hóa có xác thực Galois/Counter Mode (GCM) sang kích cỡ khối 256 bit nhằm tăng dữ liệu được xác thực và nâng cao độ an toàn trong các hệ thống xử lý thông tin dung lượng lớn. Trên cơ sở phân tích những hạn chế của GCM chuẩn, đặc biệt là giới hạn lượng thông tin được xử lý mỗi lần và điểm yếu phát sinh từ tính tuyến tính của hàm GHASH, bài báo đề xuất một biến thể GCM-256 sử dụng mã khối kích thước 256 bit và cơ chế xác thực đa tuyến tính thay cho hàm GHASH tuyến tính truyền thống.

Keywords— GCM; GHASH; authenticated encryption; provable security.

Từ khóa— GCM; GHASH; mã hóa có xác thực; độ an toàn chứng minh được.

I. INTRODUCTION

Authenticated Encryption with Associated Data (AEAD) modes have become a fundamental component of modern secure communication protocols due to their ability to

simultaneously guarantee confidentiality and data integrity. Among them, the Galois/Counter Mode (GCM) [1, 2] is one of the most widely deployed modes in TLS, IPsec, VPN infrastructures, storage systems, and high-speed communication platforms, owing to its efficient parallelizable design and strong compatibility with modern hardware acceleration mechanisms.

In the standard GCM construction, the authentication mechanism is based on the GHASH function defined over the finite field $GF(2^{128})$, with authentication tags of at most 128 bits. Although this security level remains sufficient for most practical applications, the continuous growth in data volume, authentication queries, and long-term storage requirements is introducing new challenges to the security bounds of GCM. In particular, the forgery probability in GCM increases with the number of processed data blocks, while the linearity of GHASH enables certain attacks exploiting the algebraic structure of the underlying finite field.

Furthermore, in the post-quantum setting, the emergence of Grover's algorithm effectively reduces the security level of symmetric cryptographic primitives, thereby motivating the study of larger authentication structures capable of preserving long-term security guarantees. This naturally raises the following question: can GCM be extended to a larger block size in order to strengthen authentication security while preserving the efficiency characteristics that make the mode attractive in practice?

Motivated by these considerations, this paper investigates the construction of a GCM-256 variant operating over the finite field $GF(2^{256})$

This manuscript was received on May 25, 2026. It was reviewed on June 17, 2026, revised on June 20, 2026 and accepted on June 24, 2026.

* Corresponding author

with 256-bit authentication tags. Unlike straightforward extensions of the conventional linear GHASH construction, the proposed approach employs a multilinear authentication function to eliminate weaknesses related to small-subgroup structures in GHASH. In addition, the paper analyzes the theoretical challenges arising from directly extending standard GCM to the 256-bit setting and proposes a construction that admits provable security in the random permutation model.

Our Contributions. In this paper, we analyze several limitations of the standard GCM construction and the linear GHASH function. We then propose a GCM-256 structure based on a 256-bit block cipher. In particular, we construct a multilinear authentication mechanism over $\text{GF}(2^{256})$ in order to mitigate attacks exploiting small-subgroup structures. Finally, we provide preliminary analyses of both the confidentiality and authenticity properties of the proposed GCM-256 scheme. The results indicate that extending GCM to the 256-bit setting not only enhances the level of authentication security, but also opens new directions for the development of AEAD schemes capable of satisfying long-term security requirements in future systems.

Related work. The provable security of GCM has been analyzed in numerous works [3-5]. These results indicate that the security of GCM strongly depends on the number of processed data blocks and the number of adversarial queries. In addition, several weaknesses of GCM have also been identified, including Ferguson's forgery attack against short authentication tags [6], nonce-reuse attacks [7], cycle attacks [8],... Accordingly, several constructions have been proposed to improve and address the limitations of GCM, including: GCM-SIV [9], MGM [10],.... However, these constructions were all designed for the 128-bit block setting. Among the schemes that can potentially be extended directly to the 256-bit setting are MGM [10] and GCM-ACPKM [11]. Nevertheless, such direct extensions may also introduce certain weaknesses, which we analyze in greater detail in Section 3 of this paper.

Outline. The remainder of this paper is organized as follows. Section 2 presents several notations and an overview of the GCM mode. Section 3 introduces the proposed GCM-256 construction. Section 4 analyzes the security of GCM-256 against the cycle attack. Finally, Section 5 evaluates the confidentiality and authenticity of the proposed authenticated encryption scheme in the random permutation model.

II. PRELIMINARIES

A. Notations

Notations: Let $X \parallel Y$ denote the concatenation of two bit strings X and Y . The set $\{0,1\}^n$ denotes the set of all bit strings of length n , while $\{0,1\}^*$ denotes the set of bit strings of arbitrary length. For a bit string X , the notation $|X|$ represents its bit length. Uniform random sampling of an element X from a set $\mathcal{X}$ is denoted by $X \xleftarrow{\$} \mathcal{X}$. The notation $\mathcal{A}^{\mathcal{O}} = b$ indicates that the adversary $\mathcal{A}$ is given oracle access to $\mathcal{O}$ and outputs a bit $b \in \{0,1\}$. For positive integers a and l such that $a \leq 2^l - 1$, let $\text{str}_l(a)$ denote the l -bit binary representation of a . More precisely, if $a = a_{l-1}2^{l-1} + \dots + a_12 + a_0$ with $a_{l-1}, \dots, a_0 \in \{0,1\}$ then $\text{str}_l(a) = a_{l-1} \dots a_1 a_0 \in \{0,1\}^l$. For a bit string $X = X_{l-1} \dots X_1 X_0 \in \{0,1\}^l$, let $\text{int}(X)$ denote the corresponding positive integer $X_{l-1}2^{l-1} + \dots + X_12 + X_0$. For a string $X \in \{0,1\}^{256}$, the function $\text{inc}_{128}(X)$ is defined as follows:

$$\text{inc}_{128}(X) = \text{msb}_{128}X \parallel \text{str}_{128}(\text{int}(\text{lsb}_{128}(X)) + 1 \bmod 2^{128}).$$

B. Overview of GCM

The GCM mode (Galois/Counter Mode), standardized in NIST SP 800-38D and ISO/IEC 19772:2009, employs the CTR mode for data encryption and the universal hash function GHASH for authentication. Given a plaintext P , associated data A , and a nonce N , the ciphertext C is generated using CTR mode, while the authentication tag is defined as follows:

$$T = \text{MSB}_{\tau}(E_K(I_0) \oplus \text{GHASH}_H(A, C)),$$

where $E_K: \{0,1\}^{128} \rightarrow \{0,1\}^{128}$ and $I_0 = N \parallel 0^{31}1$ when $|N| = 96$, or $I_0 = \text{GHASH}_H(\epsilon, N)$ when $|N| \neq 96$. The authentication subkey is

defined by $H = E_K(0^{128})$. The hash function $\text{GHASH}_H(X)$, for an input $X = X_1 \parallel X_2 \parallel \dots \parallel X_m$, where $|X_i| = 128$, is computed as $Y = \sum_{i=1}^m X_i \cdot H^{m-i+1}$, which is a linear operation over the finite field $\text{GF}(2^{128})$ defined by the irreducible polynomial $p(x) = x^{128} + x^7 + x^2 + x + 1$.

Limitations of GCM

- In GCM, the maximum message length processed in a single invocation must not exceed $2^{39} - 256$ bits. This bound corresponds to approximately 64 GiB of data per encryption session. More intuitively, this capacity is equivalent to roughly 10 hours of 4K video or 3 hours of 8K video. Consequently, in large-scale data processing systems or continuous transmission environments, this limit may be reached within a relatively short period of time.

- The GHASH function is linear, which gives rise to several attacks exploiting this property. A representative example is the cycle attack aimed at generating collisions in GHASH. In particular, in 2012, the authors of [8] showed that the multiplicative group structure of $\text{GF}(2^{128})$ possesses a smooth order, yielding 512 subgroups and consequently inducing weak keys in GHASH. An adversary may exploit this property to forge messages without detection. More specifically, for multiplication over $\text{GF}(2^{128})$, or more generally $\text{GF}(2^n)$, suppose that the authentication key $H = E_K(0^n)$ has order l , that is, $H^l = 1 \bmod 2^n$. Then, simply permuting the positions of X_i and X_j such that $i = j \bmod l$ produces two distinct inputs yielding the same GHASH value.

III. PROPOSED GCM-256 WITH A MULTILINEAR AUTHENTICATION MECHANISM

In this section, we propose a GCM-256 variant based on a 256-bit block cipher. Accordingly, multiplication is defined over the finite field $\text{GF}(2^{256})$, and the authentication tag has length 256 bits. We first discuss several possible approaches for constructing a 256-bit-block version of GCM.

The most straightforward approach is to preserve the original GCM structure and

directly extend it to the 256-bit setting. In this case, the initialization vector (IV) length is divided into two cases, namely $|N| = 192$ and $|N| \neq 192$, following the same 3:4 ratio used in the 128-bit-block setting. Multiplication over $\text{GF}(2^{256})$ is performed using the irreducible polynomial $p(x) = x^{256} + x^{10} + x^5 + x^2 + 1$, which minimizes both the number of nonzero terms and the number of bytes required for reduction when $n = 256$. At first glance, the resulting construction can still be proven secure, at least theoretically, following the framework of [4], and the security bounds naturally extend to the case $n = 256$. However, the main difficulty arises in the analysis of the case $|N| \neq 192$. In [4], obtaining the security bound for the 128-bit setting required estimating a quantity denoted by $\alpha_{\max}$ over a space of size 2^{32} . Extending this analysis to the case $n = 256$ would require bounding the same quantity over a space of size 2^{64} , which is computationally infeasible.

A second approach follows the design philosophy of RFC 8645 [11]. To avoid the enormous computational complexity associated with the previous method, the N length is fixed. As a result, the GHASH function is used solely for authentication purposes. The advantage of this approach is that it still admits provable security according to [3], achieves tighter security bounds compared to the variable-length N setting, and avoids the need to estimate $\alpha_{\max}$ over an impractically large space. Nevertheless, this construction, similarly to the previous one, still inherits the fundamental weakness of GHASH, namely its linearity. When operating over $\text{GF}(2^{256})$, the number of smooth-order subgroups increases significantly, specifically to 2048 subgroups.

A third approach is to adopt the MGM design in the 256-bit block setting. However, the use of encrypting the nonces $0 \parallel N$ and $1 \parallel N$ before applying the increment function significantly increases the probability of keystream collisions generated during the encryption and authentication processes.

In this paper, we propose a GCM-256 construction that both admits a clear provable-

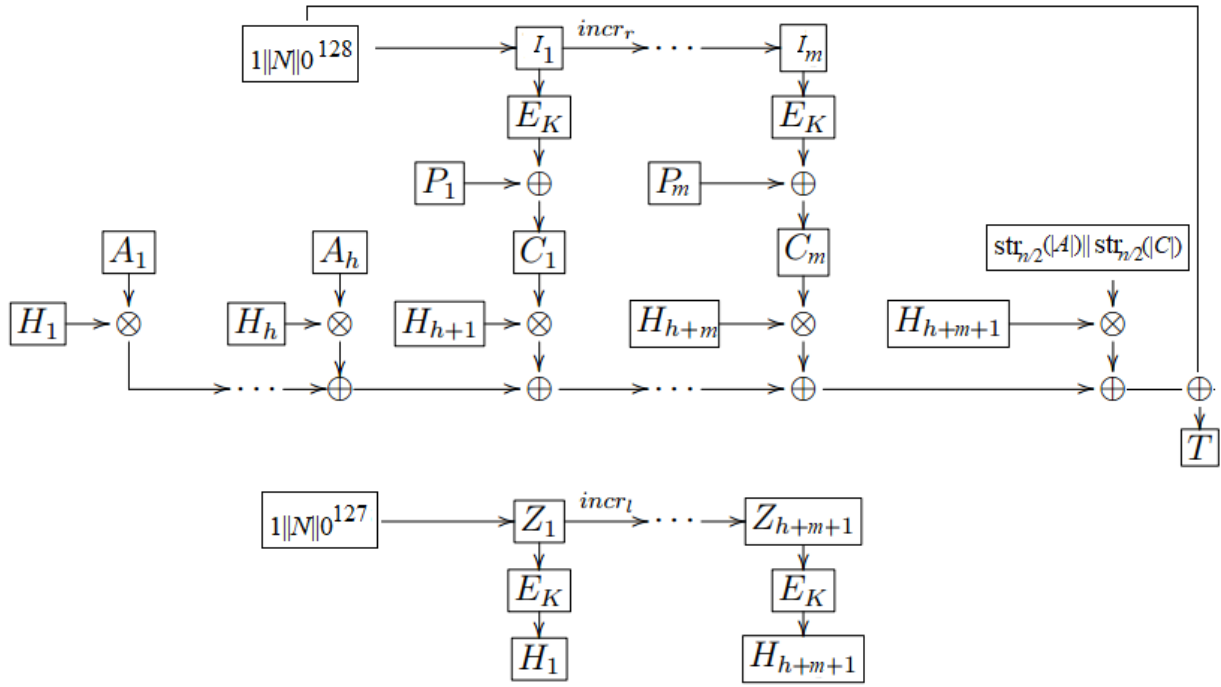


Figure 1. GCM-256 Encryption

security analysis and avoids the linearity weakness of GHASH. The polynomial used in our construction is $p(x) = x^{256} + x^{10} + x^5 + x^2 + 1$, which is an irreducible polynomial. To this end, we first introduce a new hash function, denoted HASH, to replace GHASH, defined as follows.

The algorithm $\text{HASH}_N(A, C)$, based on the block cipher $E: \mathcal{K} \times \{0,1\}^{256} \rightarrow \{0,1\}^{256}$, takes as input a nonce $N \in \{0,1\}^{127}$, and two data strings $A \in \{0,1\}^*$, $C \in \{0,1\}^*$.

1. $a \leftarrow 256 \cdot \lceil |A|/256 \rceil - |A|$,
 $p \leftarrow 256 \lceil |C|/256 \rceil - |C|$, $Z_1 = 0 \| N \| 0^{128}$
2. $X \leftarrow A \| 0^a \| C \| 0^p \| \text{str}_{128}(|A|) \| \text{str}_{128}(|C|)$
3. Partition the string X into 256-bit blocks:
 $X = X_1 \| X_2 \| \dots \| X_x$
4. $Z_i = \text{inc}_{128}(Z_{i-1})$ với $2 \leq i \leq x$
5. $H_i = E_K(Z_i)$ với $1 \leq i \leq x$
6. Return $\sum_{i=1}^x H_i \cdot X_i$

The encryption algorithm GCM-256.Enc , based on the block cipher $E: \mathcal{K} \times \{0,1\}^{256} \rightarrow \{0,1\}^{256}$ takes as input a key $K \in \mathcal{K}$, a nonce $N \in \{0,1\}^{127}$, a plaintext $P \in \{0,1\}^*$, $|P| \leq 2^{136} -$

256, and associated data $A \in \{0,1\}^*$, $|A| + |P| \leq 2^{136}$, and outputs a ciphertext $C \in \{0,1\}^{|P|}$ together with an authentication tag $T \in \{0,1\}^{256}$.

1. $I_0 = 1 \| N \| 0^{128}$
2. $m \leftarrow \lceil |P|/256 \rceil$
3. $I_i = \text{inc}_{128}(I_{i-1})$, with $1 \leq i \leq m$
4. $Y_i = E_K(I_i)$ with $1 \leq i \leq m$
5. $Y = Y_1 \| Y_2 \| \dots \| Y_m$
6. $C = P \oplus \text{Msb}_{|P|}(Y)$
7. $T = E_K(I_0) \oplus \text{HASH}_N(A, C)$
8. Return (C, T)

The decryption algorithm GCM-256.Dec , based on the block cipher $E: \mathcal{K} \times \{0,1\}^{256} \rightarrow \{0,1\}^{256}$, takes as input a key $K \in \mathcal{K}$, a nonce $N \in \{0,1\}^{127}$, associated data $A \in \{0,1\}^*$, a ciphertext $C \in \{0,1\}^*$, $|C| \leq 2^{136} - 256$, $|A| + |C| \leq 2^{136}$, and an authentication tag $T \in \{0,1\}^{256}$, and outputs either a plaintext $P \in \{0,1\}^{|C|}$ or a distinguished failure symbol $\perp$.

1. $I_0 = 1 \| N \| 0^{128}$
2. $T^* = E_K(I_0) \oplus \text{HASH}_N(A, C)$
3. If $T \neq T^*$ then return $\perp$ else go to step 4

4. $m \leftarrow \lceil |C|/256 \rceil$
5. $I_i = \text{inc}_{128}(I_{i-1})$ with $1 \leq i \leq m$
6. $Y_i = E_K(I_i)$ with $1 \leq i \leq m$
7. Return $P = C \oplus \text{Msb}_{|C|}(Y)$

Design rationale

- The GCM-256 construction employs a 256-bit counter block, in which 128 bits are used as the nonce and the remaining 128 bits are reserved for the incrementing function. This equal partitioning, inherited from the nonce construction method of the CTR mode specified in GOST 34.13–2015, allows a larger amount of data to be authenticated per invocation compared to the 3:4 ratio approach. Specifically, the maximum amount of authenticated data that can be processed in a single invocation is $2^{136} - 256$ bit.

- The initial counter values are chosen in the form $1||N||0^{128}$ and $1||N||0^{128}$, and the block cipher E_K is not applied to these values. This design choice is intended to prevent collisions between the counter sequences used during the encryption and authentication processes.

- The use of a multilinear authentication function prevents cycle attacks [8], as will be discussed in greater detail in Section IV.

IV. SECURITY ANALYSIS OF GCM-256 AGAINST CYCLE ATTACKS [8]

In the proposed GCM-256 construction, we do not employ a linear hash function for authentication. Instead, the authentication tag is computed through a multilinear function whose terms are defined by $H_i = E_K(Z_i)$. More precisely, the authentication tag is computed as

$$T = E_K(I_0) \oplus \left(\sum_{i=1}^{h+m+1} H_i \cdot X_i \right)$$

where the values H_i may be regarded as uniformly random values for each distinct nonce N . First, observe that the above authentication expression does not involve any exponentiation of H_i . Consequently, it is impossible to analyze the construction under the assumption that some H_i belongs to a small multiplicative subgroup, as is the case in conventional GHASH-based

attacks. Furthermore, according to the attack strategy described in [8], swapping two blocks X_i and X_j , would require the following equality to hold $H_i \cdot X_i \oplus H_j \cdot X_j = H_i \cdot X_j \oplus H_j \cdot X_i$, or $(H_i \oplus H_j)(X_i \oplus X_j) = 0$. Here, it should be noted that swapping X_i and X_j is meaningful only when $X_i \neq X_j$, since otherwise the resulting data remain unchanged. The values H_i are pairwise distinct, since E_K is a permutation and the inputs Z_i are constructed to be distinct. Moreover, we use the irreducible polynomial to define the finite field $GF(2^{256})$. Therefore, the probability that $(H_i \oplus H_j)(X_i \oplus X_j) = 0$ is zero.

Therefore, the proposed GCM-256 construction is resistant to the cycle attack described in [8], and no weak-key class analogous to that attack exists in the proposed scheme.

V. PROVABLE SECURITY ANALYSIS

A. Security definitions

In this paper, we consider two security notions for GCM-256, namely confidentiality and authenticity, following the framework of [4]. In these models, the block cipher is typically assumed to behave either as a random permutation or as a random function. We denote by $\text{GCM-256}[\text{Perm}]$ and $\text{GCM-256}[\text{Rand}]$ the variants of GCM-256 instantiated with the block cipher E_K modeled as a random permutation and a random function, respectively.

Privacy. The adversary $\mathcal{A}$ is given access either to the encryption oracle GCM-256.Enc or to an oracle, denoted by $\$$, which always returns a uniformly random bit string whose length equals the plaintext length plus 256 bits. The advantage of $\mathcal{A}$ against the privacy of GCM-256 is defined as:

$$\begin{aligned} & \text{Adv}_{\text{GCM-256}}^{\text{priv}}(\mathcal{A}) \\ & \stackrel{\text{def}}{=} \Pr \left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\text{GCM-256.Enc}} \rightarrow 1 \right] \\ & \quad - \Pr[\mathcal{A}^{\$} \rightarrow 1], \end{aligned}$$

where the first probability is taken over the randomness of $\mathcal{A}$ and the random choice of K , while the second probability is taken over the randomness of $\mathcal{A}$ and the oracle $\$$. In this

security notion, the adversary is nonce-respecting, meaning that $\mathcal{A}$ never submits two encryption queries using the same nonce.

Authenticity. The adversary $\mathcal{A}$ is given access to both the encryption and decryption oracles of GCM-256. The forgery advantage of $\mathcal{A}$ against GCM-256 is defined as

$$\text{Adv}_{\text{GCM-256}}^{\text{auth}}(\mathcal{A}) \stackrel{\text{def}}{=} \Pr \left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\text{GCM-256.Enc, GCM-256.Dec}} \text{ forge} \right],$$

where the probability is taken over the randomness of $\mathcal{A}$ and the random choice of K . The adversary is said to successfully forge if the decryption oracle returns a bit string different from $\perp$ for some query (N, A, C, T) , such that the pair (C, T) was not previously returned by the encryption oracle in response to a query (N, A, P) . Accordingly, without loss of generality, we may assume that the decryption algorithm GCM-256.Dec, immediately after checking whether $T = T^*$ (as in Step 3), directly outputs a bit 0 indicating that the adversary has successfully forged, without performing any further computations. Note that the nonce-respecting restriction applies only to encryption queries and does not constrain decryption queries.

B. Privacy

Proposition 1. Consider the GCM-256 mode instantiated with a block cipher modeled as a random permutation, denoted by, GCM-256[Perm]. Let $\mathcal{A}$ be any adversary making at most q queries, with the total plaintext length bounded by σ blocks and the total associated-data length bounded by δ blocks. Then,

$$\text{Adv}_{\text{GCM-256[Perm]}}^{\text{priv}}(\mathcal{A}) \leq \frac{0.5(2\sigma + \delta + 2q)^2}{2^{256}}.$$

Proof. Recall that GCM-256[Perm] denotes the GCM-256 construction instantiated with a random permutation $\pi \xleftarrow{\$} \text{Perm}(256)$ whereas GCM-256[Rand] denotes the GCM-256 construction instantiated with a random function $F \xleftarrow{\$} \text{Rand}(256)$. Then, by the PRP/PRF switching lemma [12], we obtain

$$\text{Adv}_{\text{GCM[Perm]}}^{\text{priv}}(\mathcal{A})$$

$$\leq \text{Adv}_{\text{GCM[Rand]}}^{\text{priv}}(\mathcal{A}) + \frac{0.5(2\sigma + \delta + 2q)^2}{2^{256}}$$

since q encryption queries require at most σ invocations of the block cipher to generate the ciphertext, and at most $\sigma + \delta + 2q$ additional block-cipher calls to compute the authentication tags.

Next, we evaluate the confidentiality of the GCM-256 construction when the underlying block cipher is modeled as a random function. Observe that, since the adversary $\mathcal{A}$ is nonce-respecting, the values I_i and Z_i are always distinct across the q queries. Consequently, the values $E_K(I_i)$ and $E_K(Z_i)$ may be regarded as uniformly and independently sampled from the set $\{0,1\}^{256}$.

On the other hand, consider the i -th encryption query, The ciphertext is computed as $C_i = P_i \oplus \text{msb}_{|P_i|}(Y_{i,1} \parallel \dots \parallel Y_{i,m_i})$, where m_i denotes the block length of P_i and $Y_{i,j} = E_K(I_{i,j})$. Since the values $Y_{i,j}$ are independently and uniformly sampled at random, the keystream $Y_{i,1} \parallel \dots \parallel Y_{i,m_i}$ is also uniformly random. Therefore, for any fixed plaintext P_i the ciphertext C_i may likewise be regarded as a uniformly random value of length $|P_i|$.

The authentication tag is computed as: $T_i = E_K(I_{i,0}) \oplus \text{HASH}_N(A_i, C_i)$. Note that the value $I_{i,0}$ is not used in the computation of either C_i or $\text{HASH}_N(A, C)$. Therefore, $E_K(I_{i,0})$ can be regarded as a uniformly random value that is independent of C_i .

Consequently, the pair (C_i, T_i) is distributed exactly as the output of the random oracle $\$$ in the privacy experiment. Hence, the adversary's view in the GCM-256[Rand] experiment is identical to its view in the random-output experiment. Therefore,

$$\text{Adv}_{\text{GCM[Rand]}}^{\text{priv}}(\mathcal{A}) = 0. \blacksquare$$

C. Authenticity

Consider an adversary $\mathcal{A}$ making q encryption queries $(N_1, A_1, P_1), \dots, (N_q, A_q, P_q)$

and q' decryption queries $(N'_1, A'_1, C'_1, T'_1), \dots, (N'_q, A'_q, C'_q, T'_q)$, where the total plaintext length is at most σ blocks, the total associated-data length is at most δ blocks, and the total ciphertext length is at most γ blocks. We denote by $m_i = \lceil |P_i|/256 \rceil$, $m'_i = \lceil |C'_i|/256 \rceil$, $h_i = \lceil |A_i|/256 \rceil$, $h'_i = \lceil |A'_i|/256 \rceil$, $a_i = 256h_i - |A_i|$, $p_i = 256m_i - |P_i|$, $a'_i = 256h'_i - |A'_i|$, and $p'_i = 256m'_i - |C'_i|$.

Proposition 2. Consider the GCM-256 mode instantiated with a block cipher modeled as a random permutation, denoted by GCM-256[Perm]. Let $\mathcal{A}$ be any adversary making at most q encryption queries and q' decryption queries, where the total plaintext length is bounded by σ blocks, the total associated-data length is bounded by δ blocks, and the total ciphertext length is bounded by γ blocks. Then

$$\begin{aligned} & \text{Adv}_{\text{GCM-256[Perm]}}^{\text{auth}}(\mathcal{A}) \\ & \leq \frac{0.5(2\sigma + \delta + \gamma + 2q + 2q')^2}{2^{256}} + \frac{q'}{2^{256}}. \end{aligned}$$

Proof. Without loss of generality, we assume that the adversary $\mathcal{A}$ makes exactly q encryption queries and q' decryption queries. Then, by the PRP/PRF switching lemma [12], we have:

$$\begin{aligned} & \text{Adv}_{\text{GCM-256[Perm]}}^{\text{auth}}(\mathcal{A}) \\ & \leq \text{Adv}_{\text{GCM-256[Rand]}}^{\text{auth}}(\mathcal{A}) \\ & \quad + \frac{0.5(2\sigma + \delta + \gamma + 2q + 2q')^2}{2^{256}}, \end{aligned}$$

since q encryption queries and q' decryption queries require at most σ block-cipher calls to compute the ciphertext, and at most $\sigma + \delta + \gamma + 2q + 2q'$ block-cipher invocations to compute the authentication tag.

Next, we evaluate $\text{Adv}_{\text{GCM-256[Rand]}}^{\text{auth}}(\mathcal{A})$. We can simulate the adversary $\mathcal{A}$'s attack against the GCM-256[Rand] scheme. Observe that $\mathcal{A}$ remains nonce-respecting for encryption queries. Accordingly, we simulate the block cipher, modeled as a random function, using lazy sampling. Furthermore, we maintain a set $\mathcal{N}$ to store all nonces that have been used. Initially, this set is empty and is updated whenever $\mathcal{A}$ makes a query.

Simulation of the forgery attack process against the GCM-256[Rand] scheme:

If $\mathcal{A}$ makes a query (N_i, A_i, P_i) such that $N_i \notin \mathcal{N}$.

01. $I_{i,0} \leftarrow 1 \parallel N_i \parallel 0^{128}$, $Z_{i,1} = 0 \parallel N_i \parallel 0^{128}$,
 $Y_{i,0} \xleftarrow{\$} \{0,1\}^{256}$
02. for $j \leftarrow 1$ to m_i do
03. $I_{i,j} \leftarrow \text{inc}_{128}(I_{i,j-1})$
04. $Y_{i,j} \xleftarrow{\$} \{0,1\}^{256}$
05. for $j \leftarrow 1$ to $h_i + m_i + 1$ do
06. $Z_{i,j} \leftarrow \text{inc}_{128}(Z_{i,j-1})$
07. $H_{i,j} \xleftarrow{\$} \{0,1\}^{256}$
08. $F(I_{i,j}) \leftarrow S_{i,j}$, $F(Z_{i,j}) \leftarrow H_{i,j}$
09. $C_i \leftarrow P_i \oplus \text{msb}_{|P_i|}(Y_{i,1} \parallel \dots \parallel Y_{i,m_i})$
10. $X_i \leftarrow A_i \parallel 0^{a_i} \parallel C \parallel 0^{p_i} \parallel \text{str}_{128}(|A_i|) \parallel \text{str}_{128}(|C_i|)$
11. $T_i \leftarrow Y_{i,0} \oplus \left(\sum_{j=1}^{h_i+m_i+1} H_{i,j} \cdot X_{i,j} \right)$
12. return (C_i, T_i)

If $\mathcal{A}$ makes a query (N_i, A_i, P_i) such that $N_i \in \mathcal{N}$, i.e., the nonce N_i has already been used in a previous decryption query (N_i, A'_j, C'_j, T'_j) .

13. for $j \leftarrow 1$ to m_i do
14. $I_{i,j} \leftarrow \text{inc}(I_{i,j-1})$
15. $Y_{i,j} \xleftarrow{\$} \{0,1\}^{256}$
16. $F(I_{i,j}) \leftarrow Y_{i,j}$
17. $C_i \leftarrow P_i \oplus \text{msb}_{|P_i|}(Y_{i,1} \parallel \dots \parallel Y_{i,m_i})$
18. $X_i \leftarrow A_i \parallel 0^{a_i} \parallel C \parallel 0^{p_i} \parallel \text{str}_{128}(|A_i|) \parallel \text{str}_{128}(|C_i|)$
19. if $h_i + m_i > h'_j + m'_j$ then
20. for $k \leftarrow h'_j + m'_j + 2$ to $h_i + m_i + 1$
21. $Z_{i,k} \leftarrow \text{inc}_{128}(Z_{i,k-1})$
22. $H_{i,k} \xleftarrow{\$} \{0,1\}^{256}$
23. $F(Z_{i,k}) \leftarrow H_{i,k}$
24. $T_i \leftarrow Y_{i,0} \oplus \left(\sum_{j=1}^{h_i+m_i+1} H_{i,j} \cdot X_{i,j} \right)$
25. Return (C_i, T_i)

If $\mathcal{A}$ makes a query (N'_i, A'_i, C'_i, T'_i) such that $N'_i \notin \mathcal{N}$

26. $I'_{i,0} \leftarrow 1 \parallel N'_i \parallel 0^{128}$, $Z'_{i,1} = 0 \parallel N'_i \parallel 0^{128}$
27. $Y'_{i,0} \xleftarrow{\$} \{0,1\}^{256}$

```

28.  $F(I'_{i,0}) \leftarrow Y'_{i,0}$ 
29. for  $j \leftarrow 1$  to  $h'_i + m'_i + 1$  do
30.    $Z'_{i,j} \leftarrow \text{inc}(Z'_{i,j-1})$ 
31.    $H'_{i,j} \xleftarrow{\$} \{0,1\}^{256}$ 
32.    $F(Z'_{i,j}) \leftarrow H'_{i,j}$ 
33.  $X'_i \leftarrow$ 
 $A'_i \| 0^{a'_i} \| C \| 0^{p'_i} \| \text{str}_{128}(|A'_i|) \| \text{str}_{128}(|C|)$ 
34.  $T'_i \leftarrow Y'_{i,0} \oplus \left( \sum_{j=1}^{h'_i+m'_i+1} H'_{i,j} \cdot X'_{i,j} \right)$ 
35. if  $T'_i = T_i^*$  then bad  $\leftarrow$  true, return 0
36. else return  $\perp$ 

If  $\mathcal{A}$  makes a query  $(N'_i, A'_i, C'_i, T'_i)$  such that  $N'_i \in \mathcal{N}$ , i.e., the nonce  $N'_i$  has already been used in a previous encryption query  $(N'_i, A_j, P_j)$  or decryption query  $(N'_i, A'_j, C'_j, T'_j)$ . Since this nonce has been used before, we assume without loss of generality that the corresponding full tuple is given by  $(N'_i, A_j, P_j, C_j, T_j)$ .

37. if  $h'_i + m'_i > h_j + m_j$  then
38.   for  $k \leftarrow h_j + m_j + 2$  to  $h'_i + m'_i + 1$ 
39.      $Z'_{i,k} \leftarrow \text{inc}_{128}(Z'_{i,k-1})$ 
40.      $H'_{i,k} \xleftarrow{\$} \{0,1\}^{256}$ 
41.      $F(Z'_{i,k}) \leftarrow H'_{i,k}$ 
42.  $T_i^* \leftarrow Y'_{i,0} \oplus \left( \sum_{j=1}^{h'_i+m'_i+1} H'_{i,j} \cdot X'_{i,j} \right)$ 
43. if  $T'_i = T_i^*$  then bad  $\leftarrow$  true, return 0
44. else return  $\perp$ 

```

We observe that the adversary always receives a uniformly random 512-bit string for every encryption query (N_i, A_i, P_i) , or the special symbol $\perp$ for every decryption query (N'_i, A'_i, C'_i, T'_i) as long as the event **bad** does not occur. In this case, the adversary's forgery advantage is equal to zero. Therefore, the probability that the adversary produces a successful forgery is exactly the probability that the event **bad** occurs.

If $N'_i \notin \mathcal{N}$, then the event $T'_i = T_i^*$ is equivalent to $T'_i = Y'_{i,0} \oplus \left(\sum_{j=1}^{h'_i+m'_i+1} H'_{i,j} \cdot X'_{i,j} \right)$. Furthermore, $Y'_{i,0}$ is a uniformly random 256-bit string that is independent of all previously generated values (according to line 27) and has

never been produced before. Therefore $\Pr[T'_i = T_i^*] = 1/2^{256}$.

On the other hand, if $N'_i \in \mathcal{N}$ there exists a previous query with full components $(N'_i, A_{j'}, P_{j'}, C_{j'}, T_{j'})$, where $T_{j'} = Y'_{i,0} \oplus \left(\sum_{j=1}^{h'_i+m'_i+1} H'_{i,j} \cdot X_{i,j} \right)$. Recall that we need to evaluate the probability that $T'_i = Y'_{i,0} \oplus \left(\sum_{j=1}^{h'_i+m'_i+1} H'_{i,j} \cdot X'_{i,j} \right)$, given that $T_{j'} = Y'_{i,0} \oplus \left(\sum_{j=1}^{h'_i+m'_i+1} H'_{i,j} \cdot X_{i,j} \right)$. This is equivalent to:

$$T'_i \oplus T_{j'} = \left(\sum_{j=1}^{h'_i+m'_i+1} H'_{i,j} \cdot X'_{i,j} \right) \oplus \left(\sum_{j=1}^{h'_i+m'_i+1} H'_{i,j} \cdot X_{i,j} \right)$$

or, equivalently,

$$T'_i \oplus T_{j'} = \sum H'_{i,j} \cdot Y_{i,j},$$

where the coefficients $Y_{i,j}$ are determined by the difference between the two inputs.

Note that if $(A'_i, C'_i) = (A_{j'}, C_{j'})$ then, by the definition of a successful forgery, the adversary must satisfy $T'_i \neq T_{j'}$ since replaying a previously returned ciphertext-tag pair is not considered a valid forgery. Therefore, the above equality cannot hold in this case. On the other hand, if $(A'_i, C'_i) \neq (A_{j'}, C_{j'})$, then there exists at least one index j such that $Y'_{i,j} \neq 0$. Furthermore, the values $H'_{i,j}$ are independent and uniformly distributed random variables. Consequently, the linear combination $\sum H'_{i,j} \cdot Y_{i,j}$ is itself uniformly distributed over $\{0,1\}^{256}$. Hence, the probability that $T'_i \oplus T_{j'} = \sum H'_{i,j} \cdot Y_{i,j}$ holds is exactly $1/2^{256}$.

Since there are q' decryption queries, we obtain:

$$\Pr[\mathbf{bad}] = \frac{q'}{2^{256}}.$$

VI. CONCLUSION

This paper has investigated the extension of the authenticated encryption mode GCM from a 128-bit block size to a 256-bit block size, with the aim of enhancing authentication

security for large-scale data processing systems and applications requiring long-term security guarantees. Based on an analysis of the limitations of the standard GCM construction - particularly the bounds on forgery probability and the weaknesses arising from the linear structure of GHASH - we have proposed a GCM-256 construction based on a 256-bit block cipher together with a multilinear authentication mechanism replacing the traditional GHASH function.

The paper has presented in detail the design of the encryption and decryption algorithms, as well as the tag generation mechanism of GCM-256, and has discussed different approaches for extending GCM to a 256-bit block setting. The results indicate that a direct extension of the linear GHASH construction faces significant challenges in both computational complexity and security analysis, whereas the proposed construction avoids cycle-based attacks and achieves provable security in the random permutation model.

Increasing the authentication tag size to 256 bits significantly reduces the forgery probability compared to standard GCM, while also improving the security margin in the post-quantum setting. However, practical deployment of GCM-256 still presents several challenges, including higher computational overhead, the need for efficient hardware support for arithmetic over $\text{GF}(2^{256})$, and compatibility issues with existing standards. Future work may focus on optimizing finite-field multiplication in the 256-bit setting, designing more efficient 256-bit block ciphers, and further analyzing hardware implementations and post-quantum deployment scenarios [13]. In addition, we also consider the specific characteristics of existing protocols in order to propose improved variants (such as achieving robustness against nonce reuse attacks) tailored to particular application scenarios [14].

REFERENCES

- [1]. NIST Special Publication 800-38D, "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC", *National Institute of Standards and Technology*, 2007.
- [2]. ISO/IEC. *Iso/iec 19772:2020 information security — authenticated encryption*. 2020.
- [3]. McGrew, D.A. and J. Viega. "The security and performance of the Galois/Counter Mode (GCM) of operation" in *International Conference on Cryptology in India*, 2004.
- [4]. Iwata, T., K. Ohashi, and K. Minematsu. "Breaking and repairing GCM security proofs" in *Annual Cryptology Conference*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 31-49, 2012.
- [5]. Niwa, Yuichi, et al. "GCM security bounds reconsidered" in *International Workshop on Fast Software Encryption*. Berlin, pp 285-497, 2015.
- [6]. Ferguson, N., "Authentication weaknesses in GCM", May 2005.
- [7]. Joux, A., "Authentication failures in NIST version of GCM", *NIST Comment*, 2006.
- [8]. Saarinen, Markku-Juhani Olavi. "Cycling attacks on GCM, GHASH and other polynomial MACs and hashes" in *International Workshop on Fast Software Encryption*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp 216-225, 2012.
- [9]. RFC, *RFC 8452: AES-GCM-SIV: Nonce misuse-resistant authenticated encryption*, 2019.
- [10]. Akhmetzyanova, Liliya, et al., "Security of multilinear galois mode (MGM)" in *Cryptology ePrint Archive*, 2019.
- [11]. RFC, *RFC 8645: Re-keying Mechanisms for Symmetric Keys*, 2019.
- [12]. Bellare, M. and P. Rogaway. "The security of triple encryption and a framework for code-based game-playing proofs" in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp 409-426, 2005.
- [13]. Long, N. V., Linh, H. D., & Dat, L. Q., "The quantum circuit construction for S-boxes without ancilla qubits: A more detail analysis", *Journal of Science and Technology on Information Security*, vol. 1, no. 21, pp. 75-88, 2024. DOI: 10.54654/isj.v1i21.1031.
- [14]. Anh, N. T., & Phong, T. Q., "On the security of AEAD scheme is recommended for use in Signal protocol", *Journal of Science and Technology on Information Security*, vol. 1, no. 21, pp. 67-74, 2024. DOI: 10.54654/isj.v1i21.1028.

ABOUT THE AUTHOR



Nguyen Tuan Anh

Workplace: Institute of Cryptography Science and Technology, Vietnam Government Information Security Commission

Email:

tuananhnghixuan@gmail.com

Education: The BS in Department of Mathematics, Hanoi University

of Science 2016. He is currently a PHD at the Academy of Cryptography Techniques.

Recent research direction: mode of operation, post-quantum cryptography, provable security, key encapsulation mechanism (KEM).

Tên tác giả: **Nguyễn Tuấn Anh**

Cơ quan công tác: Viện Khoa học - Công nghệ mật mã, Ban Cơ yếu Chính phủ, Việt Nam

Email: tuananhnghixuan@gmail.com

Quá trình đào tạo: Tốt nghiệp cử nhân Toán học tại trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Hà Nội năm 2016. Hiện đang là nghiên cứu sinh tại Học viện Kỹ thuật mật mã.

Hướng nghiên cứu hiện nay: Chế độ hoạt động, mật mã hậu lượng tử, độ an toàn chứng minh được, giao thức bọc khóa.



Nguyen Bui Cuong

Workplace: Institute of Cryptography Science and Technology, Vietnam Government Information Security Commission

Email:

nguyenbuicuong@gmail.com

Education: The BS in Department

of Mathematics, Hanoi National University of Education in 2004. Received a Master's degree in Mathematics from VNU University of Science in 2007. The PhD of Mathematical basis for information technology in Academy of Military Science and Technology in 2018.

Recent research direction: block cipher, mode of operation, post-quantum cryptography, provable security.

Tên tác giả: **Nguyễn Bùi Cường**

Cơ quan công tác: Viện Khoa học-Công nghệ mật mã, Ban Cơ yếu Chính phủ.

Email: nguyenbuicuong@gmail.com

Quá trình đào tạo: Tốt nghiệp chuyên ngành Toán học, Đại học Sư phạm Hà Nội - Đại học Quốc gia Hà Nội năm 2004. Tốt nghiệp Thạc sĩ Toán học, Đại học Khoa học tự nhiên-ĐHQGHN năm 2007. Tiến sĩ năm 2018 tại Viện Khoa học - Công nghệ quân sự, chuyên ngành Cơ sở toán học cho tin học.

Hướng nghiên cứu hiện nay: Mã khối, chế độ hoạt động, mật mã hậu lượng tử, độ an toàn chứng minh được.



Kieu Viet Dich

Workplace: 129Co., LTD, Vietnam Government Information Security Commission

Email: dich.kv@gmail.com

Education: Electronics Engineer in Le Quy Don Technical University in 2012.

Recent research direction: Embedded System, FPGA, IPsec VPN.

Tên tác giả: **Kiều Việt Đức**

Cơ quan công tác: Công ty 129, Ban Cơ yếu Chính phủ, Việt Nam.

Email: dich.kv@gmail.com

Quá trình đào tạo: Tốt nghiệp kỹ sư điện tử, Học Viện Kỹ Thuật Quân Sự năm 2012.

Hướng nghiên cứu hiện nay: Hệ thống nhúng, FPGA, IPsec VPN.