


Collision-Based Attacks on Block Cipher Modes

Exploiting Collisions and Their Absence

John Preuß Mattsson

Ericsson Research, Stockholm, Sweden
`john.mattsson@ericsson.com`

Keywords: Block Ciphers, Collision Attacks, Near Collisions, Nonce Hiding, Distinguishing Attacks, Entropy Loss, Plaintext Recovery, GCM, CCM

Abstract. Advanced Encryption Standard in Galois/Counter Mode (AES-GCM) is the most widely used Authenticated Encryption with Associated Data (AEAD) algorithm in the world. In this paper, we analyze the use of GCM with all the Initialization Vector (IV) constructions and lengths approved by NIST SP 800-38D when encrypting multiple plaintexts with the same key. We derive attack complexities in both ciphertext-only and known-plaintext models, with or without nonce hiding, for collision attacks compromising integrity and confidentiality. To facilitate the analysis of GCM with random IVs, we derive a new, simplified equation for near birthday collisions. Our analysis shows that GCM with random IVs provides less than 128 bits of security. When 96-bit IVs are used, as recommended by NIST, the security drops to less than 97 bits. Therefore, we strongly recommend NIST to forbid the use of GCM with 96-bit random nonces.

1 Introduction

Galois/Counter Mode (GCM) is an Authenticated Encryption with Associated Data (AEAD) mode of operation, designed by McGrew and Viega [34] and standardized by the National Institute of Standards and Technology (NIST) in SP 800-38D [14]. GCM combines counter mode of encryption with Galois mode of authentication, which is a Wegman-Carter polynomial hash operating in the field $\text{GF}(2^{128})$. Originally designed for block ciphers with a 128-bit block size, such as the Advanced Encryption Standard (AES) [3], but as shown in [9] it can also be adapted for use with any stream cipher like SNOW 5G [15] or Rijndael-256-256 [12] in counter mode.

AES-GCM is the most widely used AEAD algorithm in the world, used in numerous security protocols, including TLS [45], QUIC [52], IPsec [53], MACsec [30], and WiFi WPA3 [56]. It is also supported by many cryptographic Application Programming Interfaces (APIs) such as PKCS #11 [39], Oracle Java SE [25], Microsoft Cryptography API [11], W3C Web Cryptography API [54], the Linux Kernel Crypto API [28], and Apple CryptoKit [7]. Its popularity is well-deserved due to its strong performance and proven security [24, 35]. GCM

is online, fully parallelizable, and can be efficiently pipelined, making it highly effective in both hardware and software, especially on processors with dedicated instructions to accelerate AES and GHASH [18].

Weaknesses in GCM have been discussed by several researchers, including Ferguson [16], Joux [6], Handschuh and Preneel [21], Iwata et al. [24], Saarinen [49], Procter and Cid [43], Mattsson and Westerlund [31], Abdelraheem et al. [1], Forler et al. [17], and Luykx and Preneel [29]. An extensive evaluation of GCM was conducted by Rogaway [47]. It is well-known that reusing a counter value, known as a two-time pad, compromises confidentiality. Furthermore, Joux demonstrated that reusing a single Initialization Vector (IV) in GCM also breaks integrity [6]. NIST has decided to revise NIST SP 800-38D [4]. The proposed changes include removing support for authentication tags shorter than 96 bits, as suggested by [31, 47], and providing clearer guidance on IV constructions such as clarifying that the IV construction used in TLS 1.3 [45] is approved.

In this paper, we analyze the use of GCM with all the IV constructions and lengths approved in NIST SP 800-38D [14] when encrypting multiple plaintexts with the same key. We derive attack complexities in both ciphertext-only and known-plaintext models, considering different nonce hiding transforms [8], for collision attacks compromising integrity and confidentiality. Previous work have mostly focused on advantages in the adaptive chosen-ciphertext model without nonce hiding. The confidentiality attacks enable the attacker to find a large number of colliding keystream blocks and are therefore significantly more severe than distinguishing attacks [24]. Our analysis shows that GCM is severely limited by the narrow 128-bit “block size”. GCM with random IVs or IVs that are not 96 bits provides less than 128 bits of security. Specifically, when 96-bit IVs are used as recommended by NIST, the security for random IVs drops to below 97 bits, and can be as low as 64 bits. Users of AES-GCM expect 128, 192, or 256 bits of security. Therefore, we strongly recommend that NIST revise SP 800-38D to forbid the use of GCM with 96-bit random nonces. Nonce hiding [8] requires collision attacks to be performed in a known-plaintext context rather than a ciphertext-only context. However, our analysis indicates that except for one examined IV construction combined with one examined nonce hiding transform, nonce hiding does not alter the attack complexity. To facilitate our analysis of GCM with random IVs, we derive a new, simplified equation for near birthday collisions. The integrity attacks on GCM also apply to Galois Message Authentication Code (GMAC) [14]. Furthermore, many of the attacks are generic and affect other AEAD algorithms, such as Counter with Cipher Block Chaining-Message Authentication Code (CCM) [13] and ChaCha20-Poly1305 [37], when they are used with random nonces. In the official documentation of many cryptographic libraries, AES-GCM, AES-CCM, and ChaCha20-Poly1305 are commonly used with random nonces. The official documentation of many cryptographic libraries describe the use of AES-GCM, AES-CCM, and ChaCha20-Poly1305 with random nonces.

In Section 4 we derive a formula for the entropy loss of a pseudorandom permutation (PRP) in counter mode and show that the maximum amount of information an attacker can theoretically recover is $\sigma^2/2^b \ln 4$ where σ is the

number of encrypted blocks and b is the block size. This shows that the collision attacks in Section 2 are practically significantly more severe than distinguishing attacks and that ANSSI’s requirement to rekey after a maximum of $2^{b/2-5}$ blocks is a well-thought-out and balanced requirement, effectively limiting the amount of plaintext an attacker could recover to $\approx 2^{-10.47} \approx 0.0007$ bits.

2 Collision Attacks on Galois Counter Mode (GCM)

In this section, we analyze GCM as specified in NIST SP 800-38D [14]. For simplicity, we assume the block cipher is AES [3], the only NIST-approved block cipher. Given an AES algorithm and key K , the authenticated encryption function takes three input strings: plaintext P , additional authenticated data A , and initialization vector IV . The output consists of ciphertext C and authentication tag T .

The AES key length can be 128, 192, or 256 bits, while the block size is always 128 bits, regardless of key size. The plaintext must be shorter than $2^{32} - 2$ 16-byte blocks. The IV length must be between 1 and $2^{61} - 1$ bytes, though NIST recommends that implementations restrict support to 96-bit IVs. NIST SP 800-38D specifies two IV constructions: one deterministic and one based on a Random Bit Generator (RBG). For IVs shorter than 96 bits, the deterministic construction must be used, while for IVs equal to or longer than 96 bits, either construction is permissible:

- In the deterministic construction, the IV is the concatenation of two fields: the fixed field identifying the device and the invocation field. For any given key, no two distinct devices shall share the same fixed field, and no two distinct sets of inputs to any single device shall share the same invocation field. Typically, the invocation field is an integer counter.
- In the RBG-based construction, the IV is the concatenation of two fields: the random field, which must be at least 96 bits long, and the free field, which has no specific requirements. For our analysis, we assume the free field is empty, meaning the length of the random field equals $|IV|$, the length of the initialization vector in bits. The random field must either 1) consist of the output from an approved RBG, or 2) be obtained by incrementing the random field of the previous IV modulo $2^{|IV|}$. The output string from the RBG is called a direct random string, and the random fields that result from applying the incrementing function are called its successors. We will refer to the two different options as the direct RBG-based construction and the successor RBG-based construction.

The deterministic construction guarantees that there are no IV collisions, while the RBG-based construction limits the use of state between invocations of GCM. The use of only direct random strings eliminates state within each device, while using one direct random string and its successors per device eliminates the need to sync fixed fields between devices. Unless an implementation exclusively uses

96-bit IVs generated by the deterministic construction, the number of invocations n of the authenticated encryption function must not exceed 2^{32} for a given key.

The GCM authenticated encryption function is detailed in Section 7.1 of NIST SP 800-38D [14]. The steps relevant to our analysis are:

$$\begin{aligned} H &= \text{AES-ENC}(K, 0^{128}) \\ \text{If } |IV| &= 96, \text{ then } J_0 = IV \parallel 0^{31} \parallel 1 \\ \text{If } |IV| &\neq 96, \text{ then } J_0 = \text{GHASH}(H, IV \parallel \dots) \\ J_0 &= F \parallel I \text{ where } F \text{ is the leftmost 96 bits, and } I \text{ is the rightmost 32 bits} \\ J_1 &= F \parallel (I + 1) \bmod 2^{32} \\ C_i &= \text{AES-ENC}(K, J_{i+1}) \oplus P_i \\ T &= \text{AES-ENC}(K, J_0) \oplus \dots \end{aligned}$$

where “...” indicates data not relevant to our analysis. The steps assume a tag length of 128 bits and a plaintext length that is a multiple of 16 bytes. P_i and C_i denote the i -th block in the plaintext and ciphertext, respectively.

We analyze the security of AES-GCM across all approved IV constructions and lengths specified in NIST SP 800-38D [14], as well as with all basic nonce-hiding transforms specified in [8], when encrypting multiple plaintexts with the same key K . Specifically, we derive concrete complexities of collision attacks finding collisions between initialization vectors IV or between counter values J in different AES-GCM invocations under the same key. Our attacks do not assume any flaws in the random bit generator or GHASH, remaining effective even if their behavior is indistinguishable from a truly random function. A comprehensive analysis of GHASH collision security is provided in the study by Niwa et al. [38]. IV collision attacks on GCM were briefly mentioned in [41, 42], but only in the context of ciphertext-only attacks involving cleartext 96-bit IVs composed of direct random strings.

It is evident that no collisions occur between counter values within a single invocation. In the following, we use the notation IV_k for the initialization vector in invocation k and J_{ik} for the counter value J_i in invocation k . A collision where $IV_k = IV_l$ implies $J_{0k} = J_{0l}$. A collision $J_{0k} = J_{0l}$ (where $k \neq l$) compromises both integrity and confidentiality. A collision $J_{ik} = J_{jl}$ (where $k \neq l$ and i and j are not both being 0) compromises confidentiality but not integrity.

2.1 Probabilities for Collisions and Near Collisions

Collisions. The probability of a collision among m uniformly distributed random integers between 0 and $N - 1$ is approximately given by

$$\approx \frac{(m-1)m}{2N}, \quad (1)$$

where the approximation is valid when $m^2 \ll N$. For $m \gg 1$, this simplifies to

$$\approx \frac{m^2}{2N}. \quad (2)$$

This is a well-known result from the birthday problem.

Near Collisions. A generalization of the birthday problem considers the probability of near collisions [2], specifically, the probability that at least two values are within a distance d of each other. Using the approximation $e^x \approx 1 + x$ on equation (17) from [51], the probability is approximately

$$\approx \frac{(2d+1)(m-1)m}{2N}, \quad (3)$$

where the approximation is valid when $dm^2 \ll N$. For $d \gg 1$ and $m \gg 1$, this simplifies to

$$\approx \frac{dm^2}{N}. \quad (4)$$

The approximations in equations (3) and (4) are also valid when the distance is calculated modulo N , i.e., the distance between a and b is $\min(|a-b|, N-|a-b|)$ instead of $|a-b|$.

2.2 Ciphertext-Only Collision Attacks ($IV_k = IV_1$)

Deterministic Construction. When the deterministic construction is used, collisions between IVs do not occur. That is, $IV_k \neq IV_l$ when $k \neq l$, and $J_{ik} \neq J_{jl}$ when $i \neq j$ or $k \neq l$. Hence, ciphertext-only collision attacks are not feasible.

Direct RBG-Based Construction. Assuming the free field is empty, when using $n \gg 1$ direct truly random cleartext IVs with no successors under the same key, the probability of an IV collision, given by equation (2), is $\approx n^2/2^{|IV|+1}$. An IV collision $IV_k = IV_l$ where $k \neq l$ implies $J_{0k} = J_{0l}$, compromising both confidentiality and integrity. An attacker can detect collisions among n cleartext IVs with approximately n hash function invocations by using a hash table. Thus, the time complexity of a collision attack is $\approx n/(n^2/2^{|IV|+1}) = 2^{|IV|+1}/n$, and the security is only $\approx |IV| + 1 - \log_2 n$. The memory and data complexities are $\mathcal{O}(n)$. For short IVs, the number of hash function invocations is an appropriate complexity measure. For longer IVs, both the computational effort required by the attacker and the probability of a collision increase by a factor $\mathcal{O}(|IV|)$. The complexity of exploiting a collision for plaintext recovery or forgery is negligible compared to $2^{|IV|+1}/n$.

Successor RBG-Based Construction. Assuming the free field is empty, when using $m \gg 1$ direct truly random string are used, each followed by $d \gg 1$ successors obtained by incrementing the random field of the previous IV modulo $2^{|IV|}$. The total number of IVs is $n \approx dm$. The probability that two IVs collide is given by the near-collision probability equation (4) and is $\approx dm^2/2^{|IV|}$. An attacker can, with high probability, detect collisions by hashing prefixes of the m direct random strings. The length s of the prefixes in blocks should be chosen so that $m^2 \ll 2^s \ll 2^{|IV|}/d$ and the work required is $\approx m$. If two prefixes collide, the attacker can check if any of the IVs collide with work $\mathcal{O}(1)$. Thus, the time complexity of a collision attack is $\approx m/(dm^2/2^{|IV|}) \approx 2^{|IV|}/n$, and the security is $\approx |IV| - \log_2 n$. The memory and data complexities are $\mathcal{O}(m)$.

2.3 Known-Plaintext Attacks ($J_{0k} = J_{0l}$) when $|IV| \neq 96$ bits

Since each IV_k is hashed to produce a 128-bit value J_{0k} , there might be collisions $J_{0k} = J_{0l}$ where $k \neq l$ even if $IV_k \neq IV_l$. Such collisions compromise both confidentiality and integrity.

Deterministic Construction. Since there are no collisions between IVs, the probability that $J_{0k} = J_{0l}$ for $k \neq l$, assuming that the output of GHASH behaves ideally with respect to collisions, is $\approx n^2/2^{129}$. An attacker, assuming they have access to known plaintext, can find such a collision with work $\approx n$. If the attacker knows 16 bytes P_{ik} of each plaintext, they can identify collisions by hashing $P_{ik} \oplus C_{ik}$ from all invocations $k = 0 \dots n - 1$. If the first 16 bytes of the plaintexts P_{0k} contain a fixed header (refer to Section 3.4 of [40]), the attacker can find a collision by hashing each C_{0k} . Consequently, the attack complexity is $\approx 2^{129}/n$, and the security against this attack is $\approx 129 - \log_2 n$.

Direct RBG-Based Construction. The probability that $J_{0k} = J_{0l}$ where $k \neq l$ is $\approx n^2/2^{|IV|+1} + n^2/2^{129}$. An attacker can find such a collision with work $\approx n$, assuming a fixed plaintext header. The attack complexity is therefore $\approx n/(n^2/2^{|IV|+1} + n^2/2^{129}) = (1/2^{|IV|+1} + 1/2^{129})^{-1}/n$. If $|IV| = 128$, the attack complexity is $\approx 2^{128}/n$. For $|IV| > 128$, the attack complexity is $\approx 2^{129}/n$. When $|IV| < 128$, the ciphertext-only attack described in Section 2.2 has lower complexity.

Successor RBG-Based Construction. The probability that $J_{0k} = J_{0l}$ where $k \neq l$ is $\approx dm^2/2^{|IV|} + n^2/2^{129}$. An attacker can find such a collision with work $\approx n$, assuming a fixed plaintext header. The attack complexity is therefore $\approx n/(dm^2/2^{|IV|} + n^2/2^{129}) = (1/2^{|IV|} + d/2^{129})^{-1}/m$. If $|IV| > 128$, the attack complexity is $\approx 2^{129}/n$. When $|IV| \leq 128$, the ciphertext-only attack described in Section 2.2 has lower complexity.

2.4 Known-Plaintext Attacks ($J_{ik} = J_{jl}$) when $|IV| \neq 96$ bits

A collision $J_{ik} = J_{jl}$ where $k \neq l$ and i and j are not both 0 does not break integrity but does compromise confidentiality.

Deterministic Construction. For plaintexts of length $\ell \geq 2^{31}$ blocks or larger, the probability of at least two different counter values colliding is $\approx n^2/2^{97}$. If $J_{ik} = J_{jl}$, it is likely that $J_{(i+1)k} = J_{(j+1)l}$, where the addition is modulo 2^{32} . This results in the keystreams $P \oplus C$ in invocations k and l being partially identical. The work for an attacker to find such a collision is $\approx n \cdot 2^{31}$, as they only need to test the first $\approx 2^{31}$ blocks. The attack complexity is $\approx n \cdot 2^{31}/(n^2/2^{97}) = 2^{128}/n$. An assumption in this scenario can be that the plaintexts consist of approximately 2^{32} blocks, with the attacker knowing the first half but not the second half. For plaintexts of length $\ell < 2^{31}$ blocks, the probability of at least two different counter values colliding is $\approx (n^2/2^{97})(2\ell/2^{32}) = \ell n^2/2^{128}$. The work required is $\approx \ell n$, and the attack complexity is $\approx \ell n/(\ell n^2/2^{128}) = 2^{128}/n$. An assumption in this scenario can be that the attacker knows most, but not all, of each plaintext.

RBG-Based Constructions. The analysis is the same as for the deterministic construction. However, for $|IV| < 128$, the ciphertext-only attacks described in Section 2.2 has lower complexity.

2.5 Nonce Hiding Transforms

Cleartext nonces can compromise privacy by enabling tracking and identification of both the sender and receiver. They can also reveal information to an attacker who has compromised the pseudorandom number generator. As discussed in Section 2.2, cleartext nonces can be exploited for ciphertext-only collision attacks, compromising both integrity and confidentiality. Bellare et al. [8] provide a theoretical treatment of nonce-hiding AEADs and propose several nonce-hiding transformations.

HN1 Transform. In the HN1 (Hiding Nonce One) transform [8], employed in DTLS 1.3 [46] and QUIC [52], the encrypted initialization vector transmitted over the network is $IV \oplus \text{AES-ENC}(K_2, C_0)$, where K and K_2 can be derived from the same secret. Assuming a fixed plaintext header, the attacker can detect IV collisions by hashing the encrypted IVs. For the successor RBG-based construction, collisions can still be detected by hashing prefixes of the m direct random strings, see Section 2.2. The work required is $\approx m$, and the prefix collision probability is $\approx m^2/2^{s+1}$, where $m^2 \ll 2^s$. If two prefixes collide, the attacker can then check whether any of the $\approx 2d$ IVs associated with these prefixes also collide by hashing them. The average work remains $\approx m$. Therefore, the HN1 transform does not alter the attack complexities, but it requires the attack to be conducted in a known-plaintext model instead of a ciphertext-only model.

HN2 Transform. In the HN2 transform [8], the encrypted initialization vector transmitted over the network is $\text{AES-ENC}(K_2, IV \parallel x)$, where x is a prefix of C_0 . When the HN2 transform is employed, the most effective collision attack appears to involve hashing all the encrypted IVs, which requires work $\approx n$. For the successor RBG-based construction this increases the attack complexity to $\approx n/(dm^2/2^{|IV|}) \approx 2^{|IV|}/m$. Depending on the parameters, the complexity $\approx 2^{|IV|}/m$ may be lower or higher than the complexities of other attacks in the known-plaintext model described in Sections 2.3 and 2.4.

HN3 Transform. In the HN3 transform [8], the initialization vector used in GCM and transmitted over the network is $\text{PRF}(K_2, IV)$, where PRF is a pseudorandom function family. This effectively converts the deterministic and successor RBG-based constructions into the direct RBG-based construction. Consequently, the attack complexity aligns with that of the direct RBG-based construction described in Sections 2.3 and 2.4.

2.6 Summary

The security of GCM against collision attacks is summarized in Tables 1, 2, and 3. Table 1 summarizes the complexity of collision attacks in the ciphertext-only

model that compromise integrity and confidentiality. Table 2 shows the complexity of collision attacks in the known-plaintext model that compromise integrity and confidentiality. Finally, Table 3 details the complexity of collision attacks in the known-plaintext model that compromise confidentiality. For certain parameters, the attacks in Table 3 are slightly more effective than the attacks in Table 2 for an attacker focused solely on compromising confidentiality. The nonce-hiding HN1 transform does not alter the attack complexities, while the HN3 transform converts the deterministic and successor RBG-based constructions into the direct RBG-based construction.

Table 1. Complexity of ciphertext-only collision attacks ($IV_k = IV_l$) breaking integrity and confidentiality. $1 \ll n \leq 2^{32}$ is the number of cleartext IVs.

	$ IV < 96$	$ IV \geq 96$
Deterministic	∞	∞
RBG Direct	N/A	$2^{ IV +1}/n$
RBG Successor	N/A	$2^{ IV }/n$

Table 2. Complexity against known-plaintext collision attacks ($J_{0k} = J_{0l}$) breaking integrity and confidentiality. $1 \ll n \leq 2^{32}$ is the number of IVs. $m \leq n$ is the number of direct random strings.

	$ IV < 96$	$ IV = 96$	$96 > IV < 128$	$ IV = 128$	$ IV > 128$
Deterministic	$2^{129}/n$	∞	$2^{129}/n$	$2^{129}/n$	$2^{129}/n$
RBG Direct	N/A	$2^{97}/n$	$2^{ IV +1}/n$	$2^{128}/n$	$2^{129}/n$
RBG Successor	N/A	$2^{96}/n$	$2^{ IV }/n$	$2^{128}/n$	$2^{129}/n$
RBG Successor HN2	N/A	$2^{96}/m$	$\min(2^{ IV }/m, 2^{129}/n)$		

3 Analysis of Algorithm and Protocol Specifications

Section 8 of NIST SP 800-38D [14] states the following regarding IV “uniqueness”:

“The probability that the authenticated encryption function ever will be invoked with the same IV and the same key on two (or more) distinct sets of input data shall be no greater than 2^{-32} .”

“The total number of invocations of the authenticated encryption function shall not exceed 2^{32} , including all IV lengths and all instances of the authenticated encryption function with the given key.”

Table 3. Complexity against known-plaintext collision attacks ($J_{ik} = J_{jl}$) breaking confidentiality. $1 \ll n \leq 2^{32}$ is the number of IVs. $m \leq n$ is the number of direct random strings.

	$ IV < 96$	$ IV = 96$	$96 < IV < 128$	$ IV \geq 128$
Deterministic	$2^{128}/n$	∞	$2^{128}/n$	$2^{128}/n$
RBG Direct	N/A	$2^{97}/n$	$2^{ IV +1}/n$	$2^{128}/n$
RBG Successor	N/A	$2^{96}/n$	$2^{ IV }/n$	$2^{128}/n$
RBG Successor HN2	N/A	$2^{96}/m$	$\min(2^{ IV }/m, 2^{128}/n)$	

NIST does not provide a motivation for the probability limit. Expressing requirement as probabilities has several issues. First, it assumes that users know birthday probability formulas (2) and (4) and can calculate that with e.g., direct random strings, a probability of 2^{-32} corresponds to $\approx 2^{(|IV|-31)/2}$ invocations. Additionally, achieving a probability of 2^{-32} is actually impossible with 2^{32} truly random 96-bit IVs. Moreover, probability is not directly related to attack complexity; it only establishes a lower bound on security. This makes it unclear what security level NIST intended the requirement to provide. Our analysis shows that with a collision probability of 2^{-33} , a ciphertext-only attack compromising both integrity and confidentiality requires only complexity 2^{65} . Furthermore, as Rogaway states Section 12.4.10 of [47]:

*“the exposition in the NIST spec seems to kind of “fall apart” in Sections 8 and 9, and in Appendix C. These sections stray from the goal of defining GCM, and make multiple incorrect or inscrutable statements. Here are some examples. Page 18 : **The probability that the authenticated encryption function ever will be invoked with the same IV and the same key on two (or more) sets of input data shall be no greater than 2^{-32}** (here and later in this paragraph, imperatives are preserved in their original bold font). The probabilistic demand excludes use of almost all cryptographic PRGs (including those standardized by NIST), where no such guarantee is known.”*

Theoretically, using a cryptographic pseudorandom generator (PRG) for generating a large number of non-colliding IVs is the wrong approach. Instead, a pseudorandom function family (PRF) should be utilized. While a PRG ensures that a single output appears random, a PRF guarantees that all outputs appear random. The Double-Nonce-Derive-Key-GCM (DNDK-GCM) construction [19] effectively uses a PRF.

3.1 Protocols and Other Algorithms

Many IETF protocols use the NIST-standardized version of GCM [14] with a deterministic construction and an IV length of 96 bits and do therefore not suffer from collision attacks. The exceptions are JOSE [26] and COSE [50], which may

use random IVs, IPsec [53], which uses the pre-standardized version of GCM [34], and CMS [23], which may use all IVs constructions and lengths allowed by NIST.

The collision attacks on GCM compromising integrity also apply to GMAC, which is also standardized in NIST SP 800-38D [14]. The ciphertext-only collision attacks compromising confidentiality listed in Table 1 also apply to CCM [32] and ChaCha20-Poly1305 [33] if used with random nonces. CCM with random nonces would be particularly problematic as it can be used with 7–13 byte nonces. In SP 800-38C NIST states that “The nonce is not required to be random”, suggesting that AES-CCM with random nonces is NIST-approved. Unlike for GCM, NIST does not mandate any specific nonce constructions, maximum collision probabilities, or maximum number of invocations. The security of AES-CCM with random nonces would be $\approx |IV| + 1 - \log_2 n$ where $|IV|$ can be as low as 56 and n can be as large as $\approx 2^{59}$. SP 800-38C only restricts the number of block cipher invocations:

“The total number of invocations of the block cipher algorithm during the lifetime of the key shall be limited to 2^{61} .”

As stated in Section 11.9 of [47], Rogaway and Fergusson suggest that “The nonce is not required to be random” should be interpreted as the nonce need not be unpredictable. It is likely this was NIST’s intention. However, we believe that developers and users are unlikely to interpret the statement in this way. In fact, the official documentation of many cryptographic libraries exemplifies the use of AES-CCM with random nonces and the widely-used Python package PyCryptodome [44] defaults to 11-byte random nonces with AES-CCM, resulting in only $89 - \log_2 n$ bits of security.

4 Entropy Loss in PRP Counter Mode

The collision attacks affecting confidentiality enable the attacker to find a large number of colliding keystream blocks and are therefore significantly more severe than distinguishing attacks. Assuming that the block cipher is a pseudorandom permutation (PRP), the probability of distinguishing attacks making use of the birthday bound is directly related to the entropy loss in the counter mode keystream. The entropy of the first block in the counter mode keystream is $\log_2(N)$. The entropy of the second block is $\log_2(N - 1)$, and the entropy in the σ -th block is $\log_2(N - \sigma + 1)$. The entropy of the first σ blocks is

$$H_{\text{PRP}} = \sum_{i=0}^{\sigma-1} \log_2(N - i) = \log_2(N!) - \log_2((N - \sigma)!) . \quad (5)$$

The entropy of an equally long keystream with perfect secrecy is $H_{\text{PRF}} = \sigma \log_2(N)$ so the entropy loss $\Delta H = H_{\text{PRF}} - H_{\text{PRP}}$ of the counter mode keystream is

$$\Delta H = \sigma \log_2(N) - \log_2(N!) + \log_2((N - \sigma)!) . \quad (6)$$

Sterling’s approximation for the binary logarithm of the factorial is

$$\log_2(x!) = x \log_2(x) - x \log_2(e) + \frac{1}{2} \log_2(2\pi x) + \mathcal{O}\left(\frac{1}{x}\right) . \quad (7)$$

Inserting (7) in (6), assuming that $\sigma \ll N$, and rearranging gives

$$\Delta H = \left(N - \sigma + \frac{1}{2}\right) \log_2\left(\frac{N - \sigma}{N}\right) + \sigma \log_2(e) + \mathcal{O}\left(\frac{1}{N}\right) . \quad (8)$$

Series expansion¹ of (8) at $N = \infty$ gives

$$\Delta H = \sum_{n \geq 1} \frac{(2\sigma - n - 1) \cdot \sigma^n}{n(n+1) \ln 4 \cdot N^n} + \mathcal{O}\left(\frac{1}{N}\right) . \quad (9)$$

Assuming that $\sigma/N \ll 1 \ll \sigma$ gives

$$\lim_{N \rightarrow \infty} \Delta H \approx \frac{\sigma^2}{N \ln 4} . \quad (10)$$

As seen in Fig. 1, (10) is an excellent approximation also for small block sizes. The straight lines with slope -0.50 in the graph indicates that for $\sigma = 2^{b/2-c}$, and a fixed c the relative error shrinks like $\mathcal{O}(2^{-b/2})$. For all practical block ciphers with typical block sizes of 64, 128, or 256 bits, we can approximate the entropy loss with (10) as long as $\sigma/N \ll 1 \ll \sigma \ll N$.

As shown by Shannon, an attacker can theoretically recover up to L bits of information about the plaintext, assuming optimal attack conditions.

When $\sigma = N^{1/2}$ an attacker can theoretically recover up to $1/\ln 4 \approx 0.721$ bits of plaintext. When $N = 2^{128}$ and $\sigma \leq 2^{34.5}$ as required for AES-GCM in TLS 1.3 and DTLS 1.3 an attacker can theoretically recover up to $2^{-59.47}$ bits of the plaintext. In QUIC the limit is $\sigma \leq 2^{35}$ and an attacker can theoretically recover up to $2^{-58.47}$ bits of the plaintext.

The limits in TLS 1.3, DTLS 1.3, and QUIC appear to be far stricter than necessary. There is no universally agreed-upon threshold for how much information an attacker must recover for it to constitute a meaningful attack. However, protecting against the recovery of $2^{-58.47}$ bits, a practically negligible fraction, seems overly cautious and unlikely to be a realistic threat.

We agree with the conclusion in NIST IR 8459 that the key must be changed well before encrypting $N^{1/2}$ blocks of data. We recommend that NIST aligns with ANSSI [5] by requiring that the maximum number of encrypted blocks under the same key must not exceed $N^{1/2}/32$. This appears to be a well-thought-out and balanced requirement, effectively limiting the amount of plaintext an attacker could recover to $\approx 2^{-10.47} \approx 0.0007$ bits.

¹ The series expansion was done with Wolfram|Alpha [55]

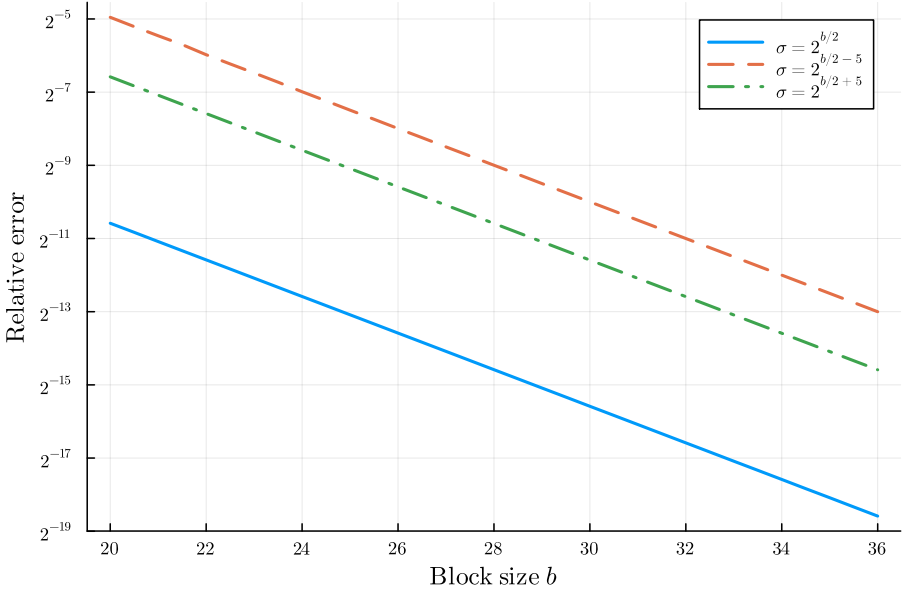


Fig. 1. Relative error $(|\sigma^2/N \ln 4 - \Delta H|/\Delta H)$ of the entropy loss approximation (10) for small block sizes.

5 Conclusions and Recommendations

Our analysis, summarized in Tables 1, 2, and 3, shows that GCM with random IVs or IVs that are not 96 bits provides less than 128 bits of security. Specifically, when 96-bit IVs are used as recommended by NIST, the security for random IVs drops to below 97 bits, and can be as low as 64 bits when 2^{32} directly generated truly random cleartext IVs are employed. Without the assumption that the Random Bit Generator and GHASH functions behave ideally with respect to collisions, the security could be significantly lower.

Without counter value collisions $J_{0k} = J_{0l}$ where $k \neq l$, the security against forgeries in GCM and GMAC is $\approx 2^{129}/\ell$ where ℓ is the plaintext length in blocks. For short plaintexts, the forgery probability is $\approx 2^{128}$, and for maximum length plaintexts the forgery probability is $\approx 2^{97}$. As shown in Table 1 and 2, the RBG-based construction significantly lowers security against forgeries. The attack model is practically serious, as it can be executed by passively observing communications, performing calculations offline, and if successful, allowing any number of forgeries with a success probability of 1.

Without counter value collisions $J_{ik} = J_{jl}$ where $k \neq l$, the best attacks on AES-GCM confidentiality are distinguishing attacks based on the birthday bound. With counter value collisions, collision attacks finding colliding parts of keystream (two-time pad) becomes possible. As shown in Table 1 and 3, the

security of the RGB-based construction significantly lowers security even when the number of IVs, n , is small.

We strongly recommend that NIST disallow the use of the RGB-based construction when $|IV| < 128$, as it significantly lowers the security against forgeries for all plaintext lengths. Additionally, NIST should consider disallowing the RGB-construction when $|IV| \geq 128$ as it significantly lowers the security against forgeries for short plaintext lengths. We also advise NIST to disallow the use of the deterministic construction when $|IV| \neq 96$, as it lowers security and there is no reason to ever use it. We strongly recommend NIST to remove the statement that a collision probability of 2^{-32} is acceptable. NIST should ensure that all remaining options achieve security strength of 128 bits [27] and clearly describe the security strength category [10] of each option.

If NIST intends to continue allowing the RGB-based construction, given the potential use cases for AES-GCM with random IVs, we recommend that NIST mandate that the random field is at least 17 bytes and clearly state the security level against collision attacks. While GCM with non-96-bit IVs has other theoretical weaknesses [1], to our knowledge, none are remotely comparable to ciphertext-only attacks that break integrity and confidentiality with complexity $2^{96}/n$.

If the RGB-based construction is kept, NIST should replace the probability-based IV requirement with an explicit requirement that is easy to understand for developers and users. This requirement should clearly specify the number of authenticated encryption invocations with the same key for different lengths of the random field. NIST should also give examples of PRGs or PRFs that can be used for generating a large number of non-colliding IVs. As the RGB-based construction cannot provide 128-bit security unless the number of invocations is severely limited, a better solution is likely deriving a new key K for each random nonce as suggested in DNDK-GCM [19]. We prefer that NIST disallows all IV constructions except for the deterministic construction with 96-bit IVs and approves a solution like DNDK-GCM for use with random nonces.

We strongly recommend that NIST and IETF explicitly disallow the use of the random nonces in AES-CCM and ChaCha20-Poly1305. We recommend cryptographic libraries to discontinue the use of short random nonces as the default and in examples. Additionally, we suggest that NIST update the terminology in SP 800-38D to use “nonce” instead of “IV”, as “nonce” is now the established term for the AEAD input parameter [32], while “IV” commonly refers to one of the fields used to construct the nonce [45, 53]. Updating SP 800-38D to use the term “nonce” will align it with SP 800-38C. We recommend IETF to update the use of GCM in IPsec [53] to refer to the standardized version of GCM [14].

The security of AES-GCM is severely limited by the narrow 128-bit block size in AES and the 128-bit digest size in GHASH. Future encryption schemes should use 256-bit keys and 256-bit nonces. Shorter nonces could be acceptable for Misuse-Resistant AEs (MRAE) [48] as nonce collisions only lowers the security to Deterministic Authenticated Encryption (DAE). Robust AE (RAE) [22] are especially attractive as they combine misuse-resistance with reforgeability

resilience. However, as interfaces should be designed to minimize user demands and mitigate the consequences of human errors [20], users ideally should not have to handle nonces. Consequently, we believe that future standardized authenticated encryption interfaces should not require nonces as input. One such interface is Authenticated Encryption with Replay protection (AERO) [33, 36], which not only manages nonces but also provides replay protection and nonce hiding.

References

1. Abdelraheem, M.A., Beelen, P., Bogdanov, A., Tischhauser, E.: Twisted Polynomials and Forgery Attacks on GCM. In: Oswald, E., Fischlin, M. (eds.) *Advances in Cryptology – EUROCRYPT 2015*. pp. 762–786. Springer Berlin Heidelberg, Berlin, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_29, <https://ia.cr/2015/1224>
2. Abramson, M., Moser, W.: More Birthday Surprises. *The American Mathematical Monthly* **77**(8), 856–858 (1970). <https://doi.org/10.1080/00029890.1970.11992600>, <https://web.archive.org/web/http://www.vartang.com/wp-content/uploads/2013/03/More-Birthday-Surprises.pdf>
3. Advanced Encryption Standard (AES). National Institute of Standards and Technology, NIST FIPS PUB 197 (May 2023). <https://doi.org/10.6028/NIST.FIPS.197-upd1>
4. NIST to Revise Special Publication 800-38D | Galois/Counter Mode (GCM) and GMAC Block Cipher Modes. National Institute of Standards and Technology (Mar 2024), <https://csrc.nist.gov/News/2024/nist-to-revise-sp-80038d-gcm-and-gmac-modes>
5. Guide des mécanismes cryptographiques. Agence nationale de la sécurité des systèmes d’information, ANSSI-PG-083 (Jan 2020), https://cyber.gouv.fr/sites/default/files/2021/03/anssi-guide-mecanismes_crypto-2.04.pdf
6. Antoine, J.: Authentication Failures in NIST Version of GCM. Comment to NIST (2006), https://csrc.nist.gov/csrc/media/projects/block-cipher-techniques/documents/bcm/comments/800-38-series-drafts/gcm/joux_comments.pdf
7. Apple CryptoKit. Apple, <https://developer.apple.com/documentation/cryptokit/>
8. Bellare, M., Ng, R., Tackmann, B.: Nonces Are Noticed: AEAD Revisited. In: Boldyreva, A., Micciancio, D. (eds.) *Advances in Cryptology – CRYPTO 2019*. pp. 235–265. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-26948-7_9, <https://ia.cr/2019/624>
9. Campagna, M., Maximov, A., Preuß Mattsson, J.: Galois Counter Mode with Strong Secure Tags (GCM-SST). Internet-draft, Internet Engineering Task Force (Feb 2025), <https://datatracker.ietf.org/doc/html/draft-mattsson-cfrg-aes-gcm-sst>
10. Transition to Post-Quantum Cryptography Standards. National Institute of Standards and Technology, NIST IR 8547 (Initial Public Draft) (Nov 2024). <https://doi.org/10.6028/NIST.IR.8547.ipd>
11. Cryptography API: Next Generation. Microsoft (Jul 2021), <https://learn.microsoft.com/en-us/windows/win32/seccng/cng-portal>
12. Daemen, J., Rijmen, V.: The Rijndael Block Cipher. Submission to NIST Advanced Encryption Standard Process (Apr 2003), <https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf>

13. Dworkin, M.: Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality. National Institute of Standards and Technology, NIST SP 800-38D (May 2004). <https://doi.org/10.6028/NIST.SP.800-38C>
14. Dworkin, M.: Recommendations for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. National Institute of Standards and Technology, NIST SP 800-38C (Nov 2007). <https://doi.org/10.6028/NIST.SP.800-38D>
15. Ekdahl, P., Johansson, T., Maximov, A., Yang, J.: SNOW-Vi: An Extreme Performance Variant of SNOW-V for Lower Grade CPUs. Cryptology ePrint Archive, Paper 2021/236 (2021). <https://doi.org/10.1145/3448300.3467829>, <https://ia.cr/2021/236>
16. Ferguson, N.: Authentication Weaknesses in GCM. Comment to NIST (May 2005), <https://csrc.nist.gov/csrc/media/projects/block-cipher-techniques/documents/bcm/comments/cwc-gcm/ferguson2.pdf>
17. Forler, C., List, E., Lucks, S., Wenzel, J.: Reforgeability of Authenticated Encryption Schemes. In: Pieprzyk, J., Suriadi, S. (eds.) Information Security and Privacy. pp. 19–37. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-59870-3_2, <https://ia.cr/2017/332>
18. Gueron, S.: Constructions based on the AES round and polynomial multiplication that are efficient on modern processor architectures. The Third NIST Workshop on Block Cipher Modes of Operation 2023 (Oct 2023), <https://csrc.nist.gov/csrc/media/Presentations/2023/constructions-based-on-the-aes-round/images-media/session-5-gueron-bcm-workshop-2023.pdf>
19. Gueron, S.: Double-Nonce-Derive-Key-GCM (DNDK-GCM) General Design Paradigms and Application. NIST Workshop on the Requirements for an Accordion Cipher Mode 2024 (Jun 2024), <https://csrc.nist.gov/csrc/media/Presentations/2024/double-nonce-derive-key-gcm-dndk-gcm/images-media/session-6-gueron-acm-workshop-2024.pdf>
20. NIST Cryptographic Standards and Guidelines Development Process. National Institute of Standards and Technology, NISTIR 7977 (Mar 2016). <https://doi.org/10.6028/NIST.IR.7977>
21. Handschuh, H., Preneel, B.: Key-Recovery Attacks on Universal Hash Function Based MAC Algorithms. In: Wagner, D.A. (ed.) Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17–21, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5157, pp. 144–161. Springer (2008). https://doi.org/10.1007/978-3-540-85174-5_9
22. Hoang, V.T., Krovetz, T., Rogaway, P.: Robust Authenticated-Encryption AEZ and the Problem That It Solves. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology – EUROCRYPT 2015. pp. 15–44. Springer Berlin Heidelberg, Berlin, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_2, <https://ia.cr/2014/793>
23. Housley, R.: Using AES-CCM and AES-GCM Authenticated Encryption in the Cryptographic Message Syntax (CMS). RFC 5084 (Nov 2007). <https://doi.org/10.17487/RFC5084>
24. Iwata, T., Ohashi, K., Minematsu, K.: Breaking and Repairing GCM Security Proofs. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology – CRYPTO 2012. pp. 31–49. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_3, <https://ia.cr/2012/438>
25. Java Platform, Standard Edition & Java Development Kit Version 23 API Specification. Oracle (2024), <https://docs.oracle.com/en/java/javase/23/docs/api/index.html>

26. Jones, M.B.: JSON Web Algorithms (JWA). RFC 7518 (May 2015). <https://doi.org/10.17487/RFC7518>
27. Recommendation for Key Management: Part 1 – General. National Institute of Standards and Technology, NIST SP 800-57 Part 1 Revision 5 (May 2020). <https://doi.org/10.6028/NIST.SP.800-57pt1r5>
28. Crypto API. The Linux Kernel, <https://www.kernel.org/doc/html/v6.13/crypto/index.html>
29. Luykx, A., Preneel, B.: Optimal Forgeries Against Polynomial-Based MACs and GCM. In: Nielsen, J.B., Rijmen, V. (eds.) *Advances in Cryptology – EUROCRYPT 2018*. pp. 445–467. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-78381-9_17, <https://ia.cr/2018/166>
30. Media Access Control (MAC) Security. Institute of Electrical and Electronics Engineers, IEEE 802.1AE-2018 (Sep 2018). <https://doi.org/10.1109/IEEESTD.2018.8585421>
31. Mattsson, J., Westerlund, M.: Authentication Key Recovery on Galois/Counter Mode (GCM). In: Pointcheval, D., Nitaj, A., Rachidi, T. (eds.) *Progress in Cryptology – AFRICACRYPT 2016*. pp. 127–143. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-31517-1_7, <https://ia.cr/2015/477>
32. McGrew, D.: An Interface and Algorithms for Authenticated Encryption. RFC 5116 (Jan 2008). <https://doi.org/10.17487/RFC5116>
33. McGrew, D., Foley, J.: Authenticated Encryption with Replay protection (AERO). Internet-draft, Internet Engineering Task Force (Feb 2014), <https://datatracker.ietf.org/doc/html/draft-mcgrew-aero>
34. McGrew, D.A., Viega, J.: The Galois/Counter Mode of Operation (GCM). Submission to NIST Modes of Operation Process (May 2005), <https://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/gcm/gcm-revised-spec.pdf>
35. McGrew, D.A., Viega, J.: The Security and Performance of the Galois/Counter Mode (GCM) of Operation. In: Canteaut, A., Viswanathan, K. (eds.) *Progress in Cryptology - INDOCRYPT 2004*. pp. 343–355. Springer Berlin Heidelberg, Berlin, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30556-9_27, <http://ia.cr/2004/193>
36. Minematsu, K.: Authenticated Encryption with Small Stretch or, How to Accelerate AERO. In: *Proceedings, Part II, of the 21st Australasian Conference on Information Security and Privacy - Volume 9723*. p. 347–362. Springer-Verlag, Berlin, Heidelberg (2016). https://doi.org/10.1007/978-3-319-40367-0_22, <https://ia.cr/2015/738>
37. Nir, Y., Langley, A.: ChaCha20 and Poly1305 for IETF Protocols. RFC 8439 (Jun 2018). <https://doi.org/10.17487/RFC8439>
38. Niwa, Y., Ohashi, K., Minematsu, K., Iwata, T.: GCM Security Bounds Reconsidered. In: Leander, G. (ed.) *Fast Software Encryption*. pp. 385–407. Springer Berlin Heidelberg, Berlin, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48116-5_19, <https://ia.cr/2015/214>
39. Cryptographic Token Interface Current Mechanisms Specification Version 3.0. OASIS (Jun 2020), <http://docs.oasis-open.org/pkcs11/pkcs11-curr/v3.0/pkcs11-curr-v3.0.pdf>
40. Preuß Mattsson, J.: Hidden Stream Ciphers and TMTO Attacks on TLS 1.3, DTLS 1.3, QUIC, and Signal. In: Deng, J., Kolesnikov, V., Schwarzmann, A.A. (eds.) *Cryptology and Network Security*. pp. 251–267. Springer Nature Singapore, Singapore (2023). https://doi.org/10.1007/978-981-99-7563-1_12, <https://ia.cr/2023/913>

41. Preuß Mattsson, J., Smeets, B., Thormarker, E.: Proposals for Standardization of Encryption Schemes. The Third NIST Workshop on Block Cipher Modes of Operation 2023 (Oct 2023), <https://csrc.nist.gov/csrc/media/Events/2023/third-workshop-on-block-cipher-modes-of-operation/documents/accepted-papers/Proposals%20for%20Standardization%20of%20Encryption%20Schemes%20Final.pdf>
42. Preuß Mattsson, J., Smeets, B., Thormarker, E.: Comments on NIST's Requirements for an Accordion Cipher. NIST Workshop on the Requirements for an Accordion Cipher Mode 2024 (Jun 2024), <https://csrc.nist.gov/csrc/media/Events/2024/accordion-cipher-mode-workshop-2024/documents/papers/comments-on-NIST-reqs-accordion-cipher.pdf>
43. Procter, G., Cid, C.: On Weak Keys and Forgery Attacks Against Polynomial-Based MAC Schemes. *J. Cryptol.* **28**(4), 769–795 (2015). <https://doi.org/10.1007/S00145-014-9178-9>, <https://ia.cr/2013/144>
44. PyCryptodome's Documentation. PyCryptodome (2024), <https://pycryptodome.readthedocs.io/en/latest/src/cipher/modern.html#ccm-mode>
45. Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446 (Aug 2018). <https://doi.org/10.17487/RFC8446>
46. Rescorla, E., Tschofenig, H., Modadugu, N.: The Datagram Transport Layer Security (DTLS) Protocol Version 1.3. RFC 9147 (Apr 2022). <https://doi.org/10.17487/RFC9147>
47. Rogaway, P.: Evaluation of Some Blockcipher Modes of Operation. Evaluation carried out for the Cryptography Research and Evaluation Committees (CRYPTREC) for the Government of Japan (Feb 2011), <https://web.cs.ucdavis.edu/~rogaway/papers/modes.pdf>
48. Rogaway, P., Shrimpton, T.: A Provable-Security Treatment of the Key-Wrap Problem. In: Vaudenay, S. (ed.) *Advances in Cryptology - EUROCRYPT 2006*. pp. 373–390. Springer Berlin Heidelberg, Berlin, Heidelberg (2006). https://doi.org/10.1007/11761679_23, <https://ia.cr/2006/221>
49. Saarinen, M.J.O.: Cycling Attacks on GCM, GHASH and Other Polynomial MACs and Hashes. In: Canteaut, A. (ed.) *Fast Software Encryption*. pp. 216–225. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34047-5_13, <https://ia.cr/2011/202>
50. Schaad, J.: CBOR Object Signing and Encryption (COSE): Initial Algorithms. RFC 9053 (Aug 2022). <https://doi.org/10.17487/RFC9053>
51. Tausch, H.J.: Simplified Birthday Statistics and Hamming EDAC. *IEEE Transactions on Nuclear Science* **56**(2), 474–478 (2009). <https://doi.org/10.1109/TNS.2009.2012710>
52. Thomson, M., Turner, S.: Using TLS to Secure QUIC. RFC 9001 (May 2021). <https://doi.org/10.17487/RFC9001>
53. Viega, J., McGrew, D.: The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP). RFC 4106 (Jun 2005). <https://doi.org/10.17487/RFC4106>
54. Web Cryptography API. W3C (Jan 2017), <https://www.w3.org/TR/WebCryptoAPI/>
55. Wolfram|Alpha: Wolfram Alpha LLC, https://www.wolframalpha.com/input?i=Series%5B%28N-%CF%83%2B1%2F2%29*lb%28%28N-%CF%83%29%2FN%29%2B%CF%83*lb%28e%29%2C%7BN%2CInfinity%2C2%7D%5D
56. WPA3 Specification Version 3.4. Wi-Fi Alliance (Oct 2024), <https://www.wi-fi.org/system/files/WPA3%20Specification%20v3.4.pdf>