AES-GCM v1[1]

Designers: David A. McGrew and John Viega[2]

Submitter: Daniel J. Bernstein[3]
djb@cr.yp.to

2014.01.28

[1] "AES-GCM v1" is the same as AES-GCM. All CAESAR submissions are required to include version numbers in cipher names to avoid confusion in case of subsequent updates. Perhaps there will be an "AES-GCM v2" that, e.g., fixes the handling of nonce lengths different from 12 bytes; see Section 3.

[2] Of course, AES-GCM is actually an evolution of a long line of previous designs from many authors. For example, a large part of the AES-GCM authenticated cipher is precisely the AES block cipher, which was designed by Joan Daemen and Vincent Rijmen. All readers of CAESAR submissions are expected to be familiar with AES in particular; CAESAR submissions are not expected to give citations or credits for AES. In any event, designer lists are not the same as scientific bibliographies.

[3] Usually the designers of a CAESAR submission would be the same as the submitters. AES-GCM is not actually a CAESAR submission. AES-GCM is the CAESAR baseline: every CAESAR submission must explain why users would prefer that cipher over AES-GCM. This document, minus the footnotes, is merely an illustration of what a CAESAR submission might look like. Submitters are expected to read the complete list of requirements in the call for submissions; the CAESAR secretary may eliminate submissions that do not meet those requirements. In case of discrepancies between this document and the call for submissions, the call for submissions is authoritative. Submitters are also expected to read the authoritative AES-GCM documentation from McGrew, Viega, and NIST. This document is written by the CAESAR secretary, not by the AES-GCM designers.

Chapter 1

Specification

1.1 Parameters

AES-GCM has three parameters: key length, nonce length, and tag length.

Parameter space: Each parameter is an integer number of bytes. The key length is either 16 bytes (128 bits), 24 bytes (192 bits), or 32 bytes (256 bits).[1] The nonce length is between 1 byte and $2^{61}-1$ bytes.[2] The tag length is between 8 bytes and 16 bytes.[3] Not all combinations of lengths are permissible: if the tag length is below 16 bytes then the nonce length must be 12 bytes.[4]

1.2 Recommended parameter sets

Primary recommended parameter set `aes128gcmv1`: 16-byte (128-bit) key,[5] 12-byte (96-bit) nonce,[6] 16-byte (128-bit) tag.[7]

[1] McGrew and Viega, "The Galois/Counter Mode of Operation (GCM)", May 31, 2005: key length "is appropriate for the underlying block cipher".

[2] McGrew and Viega: "any number of bits between 1 and 2^{64}". But 2^{64} is an error: the number of bits later needs to be encoded as an 8-byte big-endian integer, prohibiting 2^{64}. NIST correctly sets a limit of $2^{64}-1$.

[3] McGrew and Viega: "length can be any value between 64 and 128". NIST SP 800-38D is different: it allows 128, 120, 112, 104, and 96, plus 64 or 32 for "certain applications".

[4] McGrew and Viega: "If an IV with a length other than 96 bits is used with a particular key, then that key must be used with a tag length of 128 bits."

[5] McGrew and Viega do not prioritize any particular key length. The CAESAR secretary's assessment is that AES-GCM is most commonly used with 128-bit keys.

[6] McGrew and Viega: "96-bit IV values can be processed more efficiently, so that length is recommended for situations in which efficiency is critical." NIST SP 800-38D recommends that implementations support only "the length of 96 bits, to promote interoperability, efficiency, and simplicity of design".

[7] McGrew and Viega: "A tag length of 128 bits should be used whenever possible, because this value provides the best security." Accompanying footnote: "For some message authentication codes, a slight reduction in the size of the tag improves resistance against certain attacks. This is not true for GCM."

Secondary recommended parameter set `aes256gcmv1`: 32-byte (256-bit) key, 12-byte (96-bit) nonce, 16-byte (128-bit) tag.

1.3 Authenticated encryption

The inputs to authenticated encryption are a plaintext P, associated data A, a public message number N, and a key K. The number of bytes in P is[8] at most 68719476704, i.e., $2^{36} - 32$. The number of bytes in A is[9] at most $2^{61} - 1$. The number of bytes in N is the nonce length. The number of bytes in K is the key length. There is no secret message number; i.e., the secret message number is empty.

The output of authenticated encryption is a ciphertext (C, T) obtained by concatenating[10] an unauthenticated ciphertext C and a tag T. The unauthenticated ciphertext C is obtained by counter-mode encryption of P, so the number of bytes in C equals the number of bytes in P. The tag T is an authenticator of C; the number of bytes in T is the tag length. The total ciphertext length is the number of bytes in P plus the tag length.

Here are the details. Define $H = \mathrm{AES}_K(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, where each 0 means a zero byte.

Define n as the number of bytes in N. If $n = 12$, define J as the 16-byte string $(N, 0, 0, 0, 1)$. Otherwise define

$$J = \mathrm{GHASH}_H(\mathrm{pad}\,N, 0, 0, 0, 0, 0, 0, 0, 0, \mathrm{bits}\,N),$$

where $\mathrm{pad}\,N$ means N zero-padded to the nearest multiple of 16 bytes, and $\mathrm{bits}\,N$ is the 8-byte big-endian encoding of $8n$. (Zero-padding appends zeros, so N is a prefix of $\mathrm{pad}\,N$.) The function GHASH is defined below.

Write J as (J_0, J_1, J_2, J_3) where each J_i is a 4-byte string.

Define p as the number of bytes in P. Define P_i as the ith 16-byte block of P for $1 \le i \le \lceil p/16 \rceil$; if $p \bmod 16 \ne 0$ then P_i has only $p \bmod 16$ bytes for $i = \lceil p/16 \rceil$, but aside from this each P_i has 16 bytes. Define

$$C_i = P_i \oplus \mathrm{AES}_K(J_0, J_1, J_2, J_3 + i).$$

Here $J_3 + i$ means the sum of J_3 and i modulo 2^{32}, where strings are interpreted as integers in big-endian form; and $\oplus$ is xor truncated to the minimum length of the two inputs. (Truncation removes suffix bytes, so a truncated string is a prefix of the original string.) Note that C_i for $i = \lceil p/16 \rceil$ uses only the first $p \bmod 16$ bytes of the AES output if p is not a multiple of 16.

[8] McGrew and Viega: “any number of bits between 0 and $2^{39} - 256$”.

[9] McGrew and Viega: “any number of bits between 0 and 2^{64}”. NIST: “$\le 2^{64} - 1$”. As above, the number of bits needs to be encoded as an 8-byte big-endian integer, prohibiting 2^{64}.

[10] McGrew and Viega specify C and T as separate outputs with no particular order. AES-GCM applications generally use the concatenation (C, T).

Define C as the concatenation of C_i for $1 \le i \le \lceil p/16 \rceil$. Finally, define T as the truncation of

$$\mathrm{GHASH}_H(\mathrm{pad}\, A, \mathrm{pad}\, C, \mathrm{bits}\, A, \mathrm{bits}\, P) \oplus \mathrm{AES}_K(J_0, J_1, J_2, J_3)$$

to the tag length. Here $\mathrm{pad}\, A$ means A zero-padded to a multiple of 16 bytes, $\mathrm{pad}\, C$ means C zero-padded to a multiple of 16 bytes, $\mathrm{bits}\, A$ is the 8-byte big-endian encoding of $8a$ where a is the number of bytes in A, and $\mathrm{bits}\, P$ is the 8-byte big-endian encoding of $8p$.

1.4 The GHASH function

The GHASH function takes two inputs: a 16-byte string H, and a $16n$-byte string $(X_1, X_2, \ldots, X_n)$ for any positive integer n. It produces as output a 16-byte string $\mathrm{GHASH}_H(X_1, X_2, \ldots, X_n)$ defined as $X_1 H^n + X_2 H^{n-1} + \cdots + X_n H$.

Here addition, multiplication, and powering refer to operations in the finite field $\mathbf{F}_2[z]/(z^{128} + z^7 + z^2 + z + 1)$. The 16-byte string

$$\begin{aligned}(&2^7 c_0 + 2^6 c_1 + 2^5 c_2 + 2^4 c_3 + 2^3 c_4 + 2^2 c_5 + 2c_6 + c_7,\\ &2^7 c_8 + 2^6 c_9 + 2^5 c_{10} + 2^4 c_{11} + 2^3 c_{12} + 2^2 c_{13} + 2c_{14} + c_{15},\\ &\vdots,\\ &2^7 c_{120} + 2^6 c_{121} + 2^5 c_{122} + 2^4 c_{123} + 2^3 c_{124} + 2^2 c_{125} + 2c_{126} + c_{127}),\end{aligned}$$

where each $c_i \in \{0, 1\}$, is interpreted as the field element $c_0 + c_1 z + \cdots + c_{127} z^{127}$.

Chapter 2

Security goals

goal	aes128gcmv1 bits of security	aes256gcmv1 bits of security
confidentiality for the plaintext	128	256
integrity for the plaintext	96	96
integrity for the associated data	96	96
integrity for the public message number	96	96

There is no secret message number. The public message number is a nonce. The cipher does not promise any integrity or confidentiality if the legitimate key holder uses the same nonce to encrypt two different (plaintext, associated data) pairs under the same key.

The numbers in the table are actually on different scales: 2^{96} is the expected number of online forgery attempts for a successful forgery, while 2^{128} (or 2^{256}) is the expected number of key guesses to find the secret key. Any successful forgery or successful key guess should be assumed to completely compromise confidentiality and integrity of all messages.

The table above assumes that the legitimate key holder limits the concatenation of P and A to 68719476704 bytes. More generally, for a limit of B 128-bit blocks (i.e., $16B$ bytes), the integrity goal is that a forgery does not succeed with probability larger than $(B+1)/2^{128}$. The table also assumes that the legitimate key holder does not approach 2^{64} blocks encrypted under a single key.

Additional security goal beyond the recommended 12-byte nonces: It is safe to use a single key with nonces of different lengths.[1]

[1] McGrew and Viega: "For a fixed value of the key, each IV value must be distinct, but need not have equal lengths."

Chapter 3

Security analysis

AES outputs for distinct inputs are indistinguishable from independent uniform random strings[1] when the number of inputs does not approach 2^{64}. For the recommended 12-byte nonces, the AES inputs $(J_0, J_1, J_2, J_3 + i) = (N, 1 + i)$ used by AES-GCM cannot collide with each other or with the input used to compute H, since plaintexts have at most $2^{32} - 2$ blocks. The outputs are thus indistinguishable from independent uniform random strings. The encryption of P into C is thus indistinguishable from a one-time pad, and T is indistinguishable from an independent one-time polynomial authenticator of C.

The best attacks known against AES-GCM do not violate the security goals.[2] One can prove security of AES-GCM with explicit bounds showing that these attacks are close to best possible.[3]

For nonce lengths different from 12 bytes[4] there is a mistake in the AES-GCM design, specifically in the hashing that produces J. This mistake compromises the original "proof" of security. Iwata, Ohashi, and Minematsu at Crypto 2012 pointed out this mistake and gave a replacement proof with somewhat worse, but still usable, security bounds. Anyway, the only recommended nonce length here is 12 bytes.

[1] NIST's call for AES submissions identified "the extent to which the algorithm output is indistinguishable from [the output of] a [uniform] random permutation" as one of the "most important" factors in evaluating candidates. This factor has been amply studied for AES. CAESAR submissions that rely solely on this property of AES can simply assert this property; they are not expected to repeat or summarize the studies. AES is a special case: submissions that rely on the security of other previously published components are expected to discuss the security of those components.

[2] A more convincing security-analysis section would explain what these attacks do and would say quantitatively how effective the attacks are. This is a rather minimalist security-analysis section; security analyses typically continue for many pages.

[3] A more convincing security-analysis section would say what these bounds are and would include the security proof.

[4] The security analysis is not required to cover anything beyond the recommended parameter sets. However, broader security analyses provide a foundation for the possibility of recommending other parameter sets later; designers should consider this possibility in advance.

Chapter 4

Features

This cipher has many useful features, especially for encrypting and authenticating network packets.[1]

In software, this cipher provides high speed.[2] In hardware, this cipher provides high throughput (over 10 gigabits per second) and low latency in low area.[3] This cipher is pipelineable (the multiplications can take place in parallel with the cipher calls) and highly parallelizable across blocks. It is also online, allowing encryption to produce ciphertext blocks before subsequent plaintext blocks (or the plaintext length) are known, and decryption to produce plaintext blocks before subsequent ciphertext blocks (or the ciphertext length) are known; note, however, that applications must not use the resulting plaintext until the authenticator has been verified. This cipher also uses only AES block encryption, not AES block decryption.[4]

This cipher is also incremental: modifying one block of a plaintext modifies only the corresponding ciphertext block and the authenticator, and computing the updated authenticator is very fast even for long messages. Incrementality is useful in, e.g., reducing encryption latency for applications that see the first block of a plaintext after the rest of the plaintext. This cipher can efficiently preprocess A without seeing P or the message number,[5] can efficiently prepro-

[1] This is a rather minimalist "Features" section, written as a brief summary of the features advertised by the AES-GCM designers. This is not meant to indicate that the "Features" sections in CAESAR submissions are expected to be so short. This is also not meant to indicate (1) that this assessment of AES-GCM features is clear or accurate; (2) that the features listed here are desirable; or (3) that features not listed here are less desirable.

[2] McGrew and Viega: "excellent performance by using table-driven field operations". McGrew and Viega actually comment on performance in considerably more detail; a better-written "Features" section would include those comments. On the other hand, text predicting performance is never as convincing as publicly verifiable measurements of optimized software and hardware.

[3] McGrew and Viega: "can be implemented in hardware to achieve high speeds with low cost and low latency".

[4] NIST: "The GCM functions require only the forward direction of the underlying block cipher (i.e., the inverse direction is not required)."

[5] NIST: "If some or all of the additional, non-confidential data is fixed, then the corre-

cess P without seeing A, can efficiently preprocess portions of A and P without seeing the other portions, can precompute various cipher calls,[6] etc. Beware that it is important for security to disclose only one authenticator per nonce, but incrementality applies across nonces.

This cipher combines standard, well-understood techniques:[7] encrypt-then-MAC; counter-mode encryption; and polynomial authentication. This cipher is provably secure assuming that AES is indistinguishable from uniform. AES has been extensively studied.

Recommended parameters: The 96-bit nonce is simplest and fastest, and avoids the mistake mentioned in Section 3; it was already recommended before the mistake was discovered. The 128-bit authenticator provides adequate protection against forgeries, taking into account the linear increase of forgery probability with message length. The 128-bit key option provides adequate protection against key searches, and the 256-bit key option provides ample protection.

Length limits: The length limit on plaintexts is above the 2^{32}-byte limit that many environments (e.g., FAT filesystems) already impose upon files. Larger files can be split into smaller files, with each of the smaller files separately encrypted and authenticated under an appropriate message number; this also has the advantage of reducing the amount of data to be retransmitted in case some data is corrupted.

Unfortunately, this cipher does not have any advantages over AES-GCM.[8] Fortunately, this cipher also does not have any disadvantages compared to AES-GCM.

sponding elements of the GCM authentication mechanism can be pre-computed."

[6] NIST: "If the unique initialization string is predictable, and the length of the confidential data is known, then the block cipher invocations within the GCM encryption mechanism can be pre-computed."

[7] McGrew and Viega: "well-understood theoretical foundation".

[8] Every cipher submitted to CAESAR is required to explain in the "Features" section why users should prefer that cipher over AES-GCM. If this were a real submission then obviously it would fail this requirement.

Chapter 5

Design rationale

The main goal of this cipher is extremely high throughput.[1] This requires high parallelizability and hardware-friendly operations.[2]

Encrypt-then-MAC is simple and secure if the encryption and authentication methods are secure. Parallelizability rules out most encryption methods but allows counter mode. It also rules out most authentication methods but allows polynomial authentication. A binary field is the obvious choice for high-speed hardware.

The designers have not hidden any weaknesses in this cipher.[3] All choices inside AES were amply explained during the AES competition.[4] The security proof for AES-GCM rules out weaknesses outside AES.

[1] McGrew and Viega: "our primary motivation is to achieve high data rates".

[2] This is a minimalist design-rationale section, written as a brief summary of the four pages of rationale provided by McGrew and Viega. This is not meant to indicate that the design-rationale sections in CAESAR submissions are expected to be so short, or that this assessment of AES-GCM advantages is accurate, or that the advantages listed here are important, or that advantages not listed here are less important.

[3] Actually, this is overstating the confidence level of the author of this document. Have there been any statements from the AES-GCM designers about this? Submitters are expected to confirm this statement with designers if the submitters are not the same as the designers.

[4] AES is a special case because it has been so thoroughly scrutinized. CAESAR submissions that rely on existing components other than AES are expected to analyze the possibility of hidden weaknesses inside those components.

Chapter 6

Intellectual property

Known patents, patent applications, planned patent applications, and other intellectual-property constraints relevant to use of the cipher: Patent 7840003 covers pipelined hardware for GCM. Patent 7991152 covers implementing GCM using PCLMULQDQ. Patent application 20090080646 covers 2-way parallelization of the polynomial evaluation in GCM. No other constraints known.

If any of this information changes, the submitter will promptly (and within at most one month) announce these changes on the `crypto-competitions` mailing list.

Chapter 7

Consent

The submitter hereby consents to all decisions of the CAESAR selection committee regarding the selection or non-selection of this submission as a second-round candidate, a third-round candidate, a finalist, a member of the final portfolio, or any other designation provided by the committee. The submitter understands that the committee will not comment on the algorithms, except that for each selected algorithm the committee will simply cite the previously published analyses that led to the selection of the algorithm. The submitter understands that the selection of some algorithms is not a negative comment regarding other algorithms, and that an excellent algorithm might fail to be selected simply because not enough analysis was available at the time of the committee decision. The submitter acknowledges that the committee decisions reflect the collective expert judgments of the committee members and are not subject to appeal. The submitter understands that if he disagrees with published analyses then he is expected to promptly and publicly respond to those analyses, not to wait for subsequent committee decisions. The submitter understands that this statement is required as a condition of consideration of this submission by the CAESAR selection committee.