---------------------------------------------------------------------

# 文档声明

## 文档性质说明

本文档是NIST Special Publication 800-38D - Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC标准的部分内容的中文译本。翻译内容并非该标准的完整官方版本，仅覆盖原文作者选择或认为相关的特定章节或主题。本译文旨在方便中文读者理解标准的核心概念，不可替代原始英文标准。

## 合规性提醒

本译文是非正式的、由个人提供的翻译，不具任何官方效力、法律效力或认证意义。仅用于信息参考与学习目的。任何涉及密码实现、产品认证、合规性评估或法律事务的场景，必须查阅、引用并遵循NIST官方网站发布的原始、最新版本的英文标准。

## 版权与使用说明

原始标准版权：NIST Special Publication 800-38D 的版权由美国国家标准与技术研究院(NIST)所有。NIST 出版物通常被视为美国政府的作品，通常属于 公有领域(Public Domain)。用户可自由查阅、下载和使用原始标准。

​ 衍生译文版权：本译文的编排、呈现及可能的注释（如有）由译者（作者）完成。使用者可基于知识共享原则（如 CC BY 或 CC BY-SA）免费阅读、参考、分享本译文，但需注明来源（即包含本声明或链接）。

​ 禁止商业性直接利用：不得将本译文（包括格式排版）用于任何形式的直接商业性出售或牟利。若需商业使用，应基于原始NIST文档。

## 内容免责

​​“作者个人见解”强调：翻译本身涉及对技术术语和复杂概念的理解与转换。译者在力求准确反映原文意图的基础上，包含其个人的理解与诠释。因此，译文可能与原始标准在措辞、细节侧重或细微含义上存在差异。​​

​​非权威性：译者与发布者不声明或保证本译文的精确性、完整性、可靠性或适用性。

​​**无责任承担**：使用者理解并同意，译者、发布者及相关方对因使用、误用或无法使用本译文所载信息而导致的任何损失、损害或后果（包括技术、财务或法律后果）概不负责。

​​最终依据：所有技术实现、决策和争议解决应且仅应以NIST官方发布的原始英文标准为准。

内容完善说明​​

本译文基于特定版本的NIST SP 800-38D进行翻译。原文标准可能被修订和更新。译者不承诺本译文会持续追踪或反映NIST标准的最新修订。译文内容可能存在疏漏或错误。欢迎读者提供建设性反馈以便改进，但译者保留是否采纳修改的最终决定权。

---------------------------------------------------------------------

**美国国家标准与技术研究院特别出版物800-38D 2007年11月**

# 1 目的

本出版物是关于对称密钥分组密码操作模式的一系列建议中的第四部分。

# 2 摘要

本建议规定了GCM（伽罗瓦域的计数器模式）——一种用于带关联数据的认证加密算法。以及GMAC（伽罗瓦域的消息认证码）——用于为数据生成消息认证码。

# 3 引言

本建议规定了GCM算法，用于带关联数据的认证加密。GCM由对称加密分组密码构建，分组大小为16字节，是AES算法的一种操作模式。GCM使用计数器模式的变体来提供数据的机密性保证，在GF(2)上定义通用的哈希函数提供数据的完整性校验（对于加密数据，每次调用最多约64GB；对于未加密的附加数据，长度不受限制）。

如果GCM的输入仅仅限于不加密的数据，则称作GMAC。

GCM提供更强大的加密认证，特别是GCM可以检测到数据的意外修改和故意的、未授权的修改。

GCM的两个函数称为**认证加密函数**和**认证解密函数**。这两个函数都是相对高效且可并行化，能够在软件和硬件中实现高吞吐量。

GCM函数不需要预先知道机密数据和附加的非机密数据的长度。数据长度在数据处理时进行计算。

GCM函数仅需要基础分组密码的正向方向。

可以独立于从加密形式恢复数据来验证受保护的数据是否完整。

如果唯一的初始化字符串是可预测的，并且机密数据的长度已知，则GCM加密机制中的分组密码调用可以预先计算。

如果部分或者全部附加的非机密数据是固定的，则GCM认证机制的元素可以预先计算。

使用GCM的一个重要注意事项是初始化字符串唯一性的要求可能会损害安全保证。除非确保符合唯一性要求，否则不应部署此模式。

# 4 约定

## 4.1 定义和缩写

1. AAD（Additional Authenticated Data）

附加认证数据，是认证加密函数的输入数据，仅被认证（验证完整性和来源），但不被加密。

1. AES（Advanced Encryption Standard）

高级加密标准，是一种广泛使用的对称加密算法。

1. Approved（认可的）

指FIPS（联邦信息处理标准）或NIST（美国国家标准与技术研究院）推荐的算法或技术，具体包括两种情况：一是在FIPS或NIST推荐标准中指定的；二是被FIPS或NIST推荐标准采纳的。

1. Authenticated Decryption（认证解密）

GCM（伽罗瓦 / 计数器模式）中的一种功能，将密文解密为明文，同时验证密文和AAD的真实性。

1. Authenticated Encryption（认证加密）

GCM中的一种功能，将明文加密为密文，同时基于AAD和密文生成一个认证标签。

1. Authentication Tag（Tag，认证标签）

数据的加密校验和，用于检测数据的意外错误和人为篡改。

1. Authenticity（真实性）

指数据确实来自其声称的来源这一属性。

1. Bit（比特）

二进制数字，即0或1。

1. Bit String（比特串）

有限的、有序的比特序列。

1. Block（块）

对于特定的分组密码，指长度等于该分组密码块大小的比特串。

1. Block Cipher（分组密码）

一种参数化的置换族，作用于固定长度的比特串；决定置换方式的参数是一个称为“密钥”的比特串。

1. Block Size（块大小）

对于特定的分组密码和密钥，指输入（或输出）比特串的固定长度。

1. Byte（字节）

由8个比特组成的序列。

1. Byte String（字节串）

有限的、有序的字节序列。

1. Ciphertext（密文）

明文经过加密后的形式。

1. Direct Random String（直接随机串）

在基于RBG（随机数生成器）的IV（初始化向量）构造中，指RBG的输出串，用作IV的随机字段。

1. Exclusive-OR（异或，XOR）

对两个等长比特串进行的按位加法（模2运算），结果为“1”当且仅当两个对应比特不同，否则为“0”。

1. FIPS（Federal Information Processing Standard）

联邦信息处理标准，是美国政府发布的一系列关于信息处理的标准，常用于规范密码算法、数据安全等领域。

1. Fixed Field（固定字段）

在确定性IV构造中，指用于标识认证加密函数实例所属设备或上下文的字段（内容固定，不随每次调用变化）。

1. Forward Cipher Function（正向密码函数）

对于给定分组密码，由密钥选择决定的一种块置换函数（即加密过程中使用的核心函数）。

1. Free Field（自由字段）

在基于RBG的IV构造中，指内容不受限制的字段（可灵活设置，无需遵循固定规则）。

1. Fresh（全新的）

用于描述新生成的密钥，指该密钥与之前使用过的任何密钥都不相同（确保密钥唯一性，避免重复使用导致安全风险）。

1. GCM（Galois/Counter Mode）

伽罗瓦 / 计数器模式，一种认证加密模式，结合了计数器模式（CTR）的加密功能和伽罗瓦域上的乘法运算实现认证，能同时提供加密和数据完整性校验。

1. ICB（Initial Counter Block）

初始计数器块，是GCM等模式中用于生成加密计数器序列的初始块。

1. Initialization Vector（初始化向量，IV）

一种非重复值（nonce），与特定明文和AAD的认证加密调用相关联，用于确保相同密钥下加密相同明文时产生不同密文，增强安全性。

1. Inverse Cipher Function（反向密码函数）

对于给定密钥，与正向密码函数互为逆运算的函数（即解密过程中使用的核心函数）。

1. Invocation Field（调用字段）

在确定性IV构造中，指用于标识特定设备或上下文中认证加密函数输入集合的字段（区分同一设备 / 上下文中的不同调用）。

1. IT（Information Technology）

信息技术，涵盖与信息处理、存储、传输相关的技术领域。

1. IV（Initialization Vector）

初始化向量（同25条）。

1. Key（密钥）

分组密码的参数，用于从置换族中选择特定的正向密码函数（决定加密 / 解密的具体规则，是密码算法安全性的核心）。

1. Least Significant Bit (s)（最低有效位）

比特串中最右侧的比特（类似于十进制数中最右边的数字，权重最小）。

1. Mode of Operation（Mode，工作模式）

基于分组密码的一种数据加密转换算法，定义了如何将固定长度的分组密码应用于任意长度的数据（如GCM、CBC、CTR等均为常见工作模式）。

1. Most Significant Bit (s)（最高有效位）

比特串中最左侧的比特（类似于十进制数中最左边的数字，权重最大）。例如，在比特串1011中，最左侧的“1”即为最高有效位。

1. NIST（National Institute of Standards and Technology）

美国国家标准与技术研究院，是负责制定和推广技术标准（包括密码学标准）的联邦机构，其发布的标准（如AES、GCM等）在全球信息安全领域具有广泛影响力。

1. Nonce（临时值）

在特定上下文中仅使用一次的值，用于避免重复使用相同参数导致的安全漏洞（如IV本质上就是一种nonce）。例如，在加密通信中，每次会话使用的nonce必须唯一，防止攻击者通过重复模式破解加密。

1. Permutation（置换）

一种可逆函数，即对输入数据进行重新排列后，可通过逆操作恢复原始数据。在分组密码中，置换是核心操作之一，用于混淆数据以增强加密强度。

1. Plaintext（明文）

认证加密函数的输入数据，既会被认证（验证完整性和来源），也会被加密（转换为密文以保证机密性）。例如，用户发送的原始消息在加密前即为明文。

1. Random Field（随机字段）

在基于RBG（随机比特生成器）的IV构造中，指要么是直接随机串，要么是其后续生成的串（即通过对直接随机串应用递增函数得到的结果）。

1. RBG（Random Bit Generator）

随机比特生成器，用于产生不可预测的比特序列，是生成密钥、nonce、随机字段等安全参数的基础。

1. Successor（后续串）

在基于RBG的IV构造中，指对直接随机串应用一次或多次适当的递增函数后得到的结果。例如，若直接随机串为0010，递增一次后可能得到0011，该结果即为后续串。

1. XOR（Exclusive-OR）

异或（同之前的“Exclusive-OR”），按位对两个等长比特串进行模2加法运算，是密码学中常用的基础操作（如用于数据加密、校验和计算等）。

## 4.2 符号

### 4.2.1 变量

1. A：附加认证数据（Additional Authenticated Data）
2. C：密文（Ciphertext）
3. H：哈希子密钥（Hash Subkey）
4. ICB：初始计数器块（Initial Counter Block）
5. IV：初始化向量（Initialization Vector）
6. K：分组密码密钥（Block Cipher Key）
7. P：明文（Plaintext）
8. R：块乘法运算算法中的常数（Constant within the algorithm for the block multiplication operation）
9. T：认证标签（Authentication Tag）
10. t：认证标签的位长度（Bit length of the authentication tag）

### 4.2.2 运算和函数

1. 0s：由s个'0'位组成的位串（The bit string that consists of s ‘0’ bits）。
2. CIPHₖ(X)：分组密码在密钥K下对块X应用正向密码函数的输出（The output of the forward cipher function of the block cipher under the key K applied to the block X）。
3. GCTRₖ(ICB, X)：给定分组密码在密钥K下，对具有初始计数器块ICB的位串X应用GCTR函数的输出（The output of the GCTR function for a given block cipher with key K applied to the bit string X with an initial counter block ICB）。
4. GHASHₕ(X)：在哈希子密钥H下对X应用GHASH函数的输出（The output of the GHASH function under the hash subkey H applied to the bit string X）。
5. incₛ(X)：将位串X的最右s位视为整数的二进制表示，按1模2ˢ递增后的结果（The output of incrementing the right-most s bits of the bit string X, regarded as the binary representation of an integer, by 1 modulo 2ˢ）。
6. int(X)：位串X为其二进制表示的整数（The integer for which the bit string X is a binary representation）。
7. len(X)：位串X的位长度（The bit length of the bit string X）。
8. LSBₛ(X)：由位串X的最右s位组成的位串（The bit string consisting of the s right-most bits of the bit string X）。
9. MSBₛ(X)：由位串X的最左s位组成的位串（The bit string consisting of the s left-most bits of the bit string X）。
10. ⎡x⎤：不小于实数x的最小整数（The least integer that is not less than the real number x）。
11. [x]ₛ：非负整数x的s位二进制表示，其中x<2ˢ（The binary representation of the non-negative integer x as a string of s bits, where x<2ˢ）。
12. X >> 1：丢弃位串X的最右位并在左侧前置一个'0'位得到的位串（The bit string that results from discarding the rightmost bit of the bit string X and prepending a ‘0’ bit on the left）。
13. X || Y：两个位串X和Y的连接（The concatenation of two bit strings X and Y）。
14. X ⊕ Y：两个等长位串X和Y的按位异或（The bitwise exclusive-OR of two bit strings X and Y of the same length）。
15. X • Y：两个块X和Y的乘积，视为某个二元伽罗瓦域的元素（The product of two blocks, X and Y, regarded as elements of a certain binary Galois field）。
16. Xⁱ：在乘积'・'下块X的第i次幂（For a positive integer i, the ith power of X under the product ‘・’）。
17. x·y：两个整数x和y的乘积（The product of two integers, x and y）。

# 5 GCM的要素

以下三节介绍GCM的要素以及相关的符号和要求。第5.1节讨论基础分组密码和密钥。第5.2节讨论GCM的认证加密和认证解密函数的数据要素。第5.3节讨论这两个函数中用于机密性和认证的密码原语。

## 5.1 分组密码

GCM的操作取决于基础对称密钥分组密码的选择，因此可以被视为分组密码的一种操作模式（简称模式）。GCM密钥就是分组密码密钥（简称密钥）。

对于任何给定的密钥，该模式的基础分组密码由两个互为逆函数的函数组成。分组密码的选择包括指定分组密码的两个函数之一作为正向密码函数，如参考文献[2]中AES算法的规范所述。GCM不使用反向密码函数。

正向密码函数是固定长度位串上的置换；这些串称为块。块的长度称为块大小。密钥表示为K，由此产生的分组密码的正向密码函数表示为CIPH。

基础分组密码应是批准的，块大小应为128位，密钥大小至少为128位。密钥应随机生成，或接近随机生成，即每个可能的密钥被生成的可能性（几乎）相等。因此，密钥将很可能是全新的，即与任何先前的密钥不等。密钥应保密，且仅用于所选分组密码的GCM模式。第8.1节讨论了关于密钥建立和管理的其他要求。

## 5.2 GCM的两个函数

构成GCM的两个函数称为认证加密和认证解密。认证加密函数对机密数据进行加密，并根据机密数据和任何附加的非机密数据计算认证标签。认证解密函数对机密数据进行解密，但要以标签的验证为前提。

实现可以将输入限制为非机密数据，即没有任何机密数据。GCM的这种变体称为伽罗瓦消息认证码（GMAC）。对于GMAC，认证加密和解密函数成为对非机密数据生成和验证认证标签的函数。

这些函数的输入和输出数据的要求和符号在第5.2.1节和第5.2.2节中讨论。计算这些函数的算法在第7节中给出。

### 5.2.1 认证加密函数

#### 5.2.1.1 输入数据

在选择批准的分组密码和密钥后，认证加密函数有三个输入串：

明文，表示为P；

附加认证数据（AAD），表示为A；

初始化向量（IV），表示为IV。

明文和附加认证数据是GCM保护的两类数据。GCM保护明文和附加认证数据的真实性；GCM还保护明文的机密性，而附加认证数据则保持明文状态。例如，在网络协议中，附加认证数据可能包括地址、端口、序列号、协议版本号和其他指示如何处理明文的字段。

初始化向量本质上是一个随机数，即在指定上下文中唯一的值，它确定了对要保护的输入数据调用认证加密函数。第8节精确规定了初始化向量（和密钥）的唯一性要求，第8.2节给出了两种构造初始化向量的框架。第9.1节和第9.2节讨论了确保该要求的实际考虑。附录A总结了参考文献[5]中详细说明的初始化向量唯一性的关键重要性。

认证加密函数的输入串的位长度应满足以下要求：

len(P) ≤ 2³⁹ - 256；

len(A) ≤ 2⁶⁴ - 1；

1 ≤ len(IV) ≤ 2⁶⁴ - 1。

尽管GCM是在位串上定义的，但明文、附加认证数据和初始化向量的位长度都应为8的倍数，以便这些值是字节串。

实现可以进一步限制这些输入的位长度，只要符合上述要求；例如，实现可以设定更小的最大值。实现允许的位长度称为支持的位长度。应为整个实现确定三个输入各自的一组支持位长度，与密钥无关。

对于初始化向量，建议实现将支持的长度限制为96位，以促进互操作性、效率和设计的简单性。

#### 5.2.1.2 输出数据

认证加密函数的输出数据由以下两个位串组成：

密文，表示为C，其位长度与明文相同；

认证标签，简称标签，表示为T。

标签的位长度，表示为t，是一个安全参数，如附录B中所述。通常，t可以是以下五个值之一：128、120、112、104或96。对于某些应用，t可以是64或32；附录C给出了使用这两种标签长度的指南，包括在这些情况下对输入数据长度和密钥生命周期的要求。

实现不应支持除上一段中七个选择之外的t值。实现可以将其支持限制为这些值中的一个。每个密钥应关联一个从支持的选择中确定的固定t值。

### 5.2.2 认证解密函数

在选择批准的分组密码、密钥和相关标签长度后，认证解密函数的输入是如第5.2.1节所述的IV、A、C和T的值。输出是以下之一：

与密文C对应的明文P；

本文档中表示为FAIL的特殊错误代码。

输出P表明T是IV、A和C的正确认证标签；否则，输出为FAIL。附录B讨论了在每种情况下可以推断出的认证保证。

实现对认证解密函数支持的len (C)、len (A) 和len (IV) 值应与其对认证加密函数支持的len (P)、len (A)和len (IV) 值相同。

## 5.3 用于机密性和认证的原语

GCM中用于明文机密性的机制是计数器模式[10]的一种变体，具有特定的增量函数，表示为inc₃₂，用于生成必要的计数器块序列。用于明文加密的第一个计数器块是通过对由初始化向量生成的块进行递增生成的。

GCM中的认证机制基于一个称为GHASH¹的哈希函数，其特点是在二元伽罗瓦域内与一个称为哈希子密钥的固定参数相乘。

GCM 的设计者对GHASH 的定义有所不同，他们附加了其两个参数长度的编码。在本建议中，这些编码被纳入认证加密 / 解密函数的定义中。认证加密 / 解密函数的规范最终是等效的，但本建议中简化的GHASH版本并未获得设计者在参考文献[7]中为其GHASH定义所展示的所有属性。

哈希子密钥，表示为H，是通过将分组密码应用于“零”块生成的。由此产生的该哈希函数实例，表示为GHASHₕ，用于将附加认证数据和密文的编码压缩为单个块，然后对该块进行加密以生成认证标签。

GHASH是一种密钥哈希函数，但就其本身而言，不是一种密码哈希函数。本建议仅批准GHASH在GCM的上下文中使用。

GCM函数执行过程中的中间值应保密。特别是，这一要求排除了在系统中GCM使用哈希子密钥公开用于其他目的的情况，例如作为不可预测的值或作为密钥的完整性检查值。

# 6 GCM 的数学组件

本节介绍第7节中认证加密和认证解密函数规范中出现的数学组件。第6.1节给出了位串上基本运算和函数的示例。第6.2节定义了增量函数。第6.3节定义了“相乘”块的算法1。第6.4节定义了由此乘法构造的GHASH函数的算法2。第6.5节定义了GCTR函数的算法3。

算法1-3的规范包括输入、输出、算法步骤、图表和摘要。允许产生任何输入的正确输出的等效步骤集。在函数的多次调用中通常固定的输入称为前提条件，尽管它们也可以被视为（变化的）输入。

## 6.1 位串上基本运算和函数的示例

在本文档中，'0'位和'1'位用新的等宽字体表示，以帮助将它们与整数0和1区分开。

给定实数x，上舍入函数，表示为⎡x⎤，是不小于x的最小整数。例如，⎡2.1⎤=3，⎡4⎤=4。

给定正整数s，0ˢ表示由s个'0'位组成的串。例如，0⁸=00000000。

位串上的连接运算表示为 ||；例如，001||10111=00110111。

对于等长的位串，异或（XOR）运算，表示为⊕，指定每个位位置的位的模2加法，即不进位。例如，10011⊕10101=00110。

给定位串X，X的位长度表示为len (X)。例如，len (00010)=5。

给定位串X和非负整数s，且len (X)≥s，函数LSBₛ(X)和MSBₛ(X)分别返回X的s个最低有效位（即最右位）和s个最高有效位（即最左位）。例如，LSB₃(111011010)=010，MSB₄(111011010)=1110。

给定位串X，（单次）右移函数，表示为X>>1，是 MSBₗₑₙ(X)(0||X)。例如，0110111>>1=0011011。

给定正整数s和小于2ˢ的非负整数x，整数到字符串函数，表示为[x]ₛ，是x的二进制表示，为位长度s的字符串，最低有效位在右侧。例如，对于（十进制）整数39，二进制表示（基数2）是100111，因此[39]₈=00100111。

给定（非空）位串X，字符串到整数函数，表示为int (X)，是满足[x]ₗₑₙ(X)=X的整数x。换句话说，int (X)是小于2ˡᵉⁿ⁽ˣ⁾的非负整数，其二进制表示是X。例如，int (00011010)=26。

## 6.2 增量函数

对于正整数s和位串X，且len (X)≥s，s位增量函数，表示为incₛ(X)，定义如下：incₛ(X)=MSB(ₗₑₙ(X)-ₛ(X))||[int (LSBₛ(X))+1 mod 2ˢ]ₛ

换句话说，该函数将字符串的最右s位视为整数的二进制表示，按模2ˢ递增；其余的最左len (X)-s位保持不变。

## 6.3 块上的乘法运算

令R为位串11100001||0¹²⁰。给定两个块X和Y，下面的算法1计算“乘积”块，表示为X・Y。

**算法 1：X・Y**

输入：块X，Y

输出：块X・Y

步骤：

1.令x₀x₁...x₁₂₇表示X中的位序列。

2.令Z₀=0¹²⁸，V₀=Y。

3.对于i=0到127，按如下方式计算块Zᵢ₊₁和Vᵢ₊₁：

Zᵢ₊₁=

Zᵢ，如果xᵢ=0；

Zᵢ⊕Vᵢ，如果xᵢ=1。

Vᵢ₊₁=

Vᵢ>>1，如果LSB₁(Vᵢ)=0；

(Vᵢ>>1)⊕R，如果LSB₁(Vᵢ)=1。

4.返回 Z₁₂₈。

在2¹²⁸种可能的分组（对）上定义的・运算，对应于含有2¹²⁸个元素的二进制伽罗瓦（有限）域的乘法运算。固定分组R决定了该域的一种表示形式，即通过次数小于128的二元多项式的模乘法来表示。将字符串解读为多项式的约定是“小端序”：也就是说，如果u是多项式的变量，那么分组 x₀x₁…x₁₂₇对应的多项式为x₀ + x₁u + x₂u² + … + x₁₂₇u¹²⁷。乘法过程中，“同类项”的系数相加采用异或（XOR）运算。约化模数是对应于R || 1的128次多项式。参考文献[6]详细讨论了该域。

对于正整数i，分组X在这种乘法运算下的i次幂记为Xⁱ。例如，H²=H・H，H³=H・H・H，以此类推。

## 6.4 GHASH函数

下面的算法2规定了 GHASH 函数：

**算法 2：GHASHₕ(X)**

前提条件：块H，哈希子密钥。

输入：位串X，其中len (X)=128\*m，m为正整数。

输出：块GHASHₕ(X)

步骤：

1.令X₁，X₂，...，Xₘ₋₁，Xₘ表示唯一的块序列，使得X=X₁||X₂||...||Xₘ₋₁||Xₘ。

2.令Y₀为“零块”，0¹²⁸。

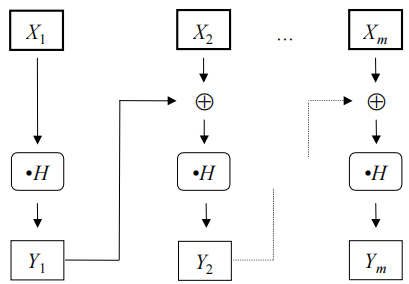
3.对于i=1，...，m，令Yᵢ=(Yᵢ₋₁⊕Xᵢ)・H。

4.返回Yₘ。

实际上，GHASH函数计算X₁・Hᵐ⊕X₂・Hᵐ⁻¹⊕...⊕Xₘ₋₁・H²⊕Xₘ・H。参考文献[6]描述了在硬件和软件中优化GHASH实现的方法。

下图1说明了GHASH 函数，没有零块Y₀，因为它与X₁的异或不会改变X₁。

图 1：GHASHₕ(X₁||X₂||...||Xₘ)=Yₘ



## 6.5 GCTR函数

下面的算法3规定了GCTR函数。建议的符号未指明基础分组密码的选择。

**算法 3：GCTRₖ(ICB, X)**

前提条件：批准的分组密码CIPH，块大小为128位；密钥K。

输入：初始计数器块ICB；任意长度的位串X。

输出：位长度为len (X) 的位串Y。

步骤：

1.如果X是空串，则返回空串作为Y。

2.令n=⎡len (X)/128⎤。

3.令X₁，X₂，...，Xₙ₋₁，Xₙ\*表示唯一的位串序列，使得X=X₁||X₂||...||Xₙ₋₁||Xₙ\*；X₁，X₂，...，Xₙ₋₁是完整的块。

4.令CB₁=ICB。

5.对于i=2 到n，令CBᵢ=inc₃₂(CBᵢ₋₁)。

6.对于i=1 到n-1，令Yᵢ=Xᵢ⊕CIPHₖ(CBᵢ)。

7.令Yₙ\*=Xₙ\*⊕MSBₗₑₙ(Xₙ\*)(CIPHₖ(CBₙ))。

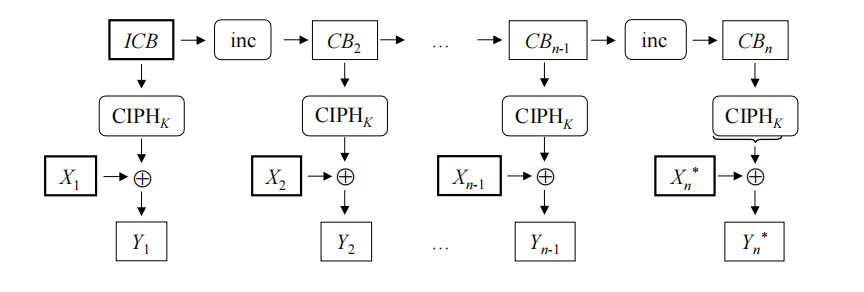
8.令Y=Y₁||Y₂||...||Yₙ\*。

9.返回Y。

在步骤1和2中，将任意长度的输入串尽可能多地划分为块序列，使得序列中最右边的串可能是“部分”块。在步骤3和4中，对初始计数器块输入应用32位增量函数以生成计数器块序列；输入块是该序列的第一个块。在步骤5和6中，将分组密码应用于计数器块，并将结果与输入串分区的相应块（或部分块）进行异或。在步骤7中，将结果序列连接起来形成输出。

下图2说明了GCTR 函数。

图 2：GCTRₖ(ICB, X₁||X₂||...||Xₙ\*)=Y₁||Y₂||...||Yₙ\*



# 7 GCM规范

第7.1节和第7.2节规定了GCM的认证加密和认证解密函数的算法4和算法5。规范包括输入、输出、算法步骤、图表和摘要。建议的符号未指明基础分组密码的选择。在函数的多次调用中通常固定的输入称为前提条件；然而，一些前提条件也可以被视为（变化的）输入。前提条件和其他输入应满足第5节中的要求。

对于这两种算法，允许产生正确输出的等效步骤集。例如，在算法5中，标签的验证可以在明文计算之前进行。

## 7.1 认证加密函数的算法

下文将详细说明认证加密函数的算法4：

**算法 4：GCM-AEₖ(IV, P, A)**

前提条件：经过批准的、块大小为128位的分组密码CIPH；密钥K；支持的输入输出长度定义；与密钥相关联的支持的标签长度t。

输入：初始化向量IV（其长度受支持）；明文P（其长度受支持）；附加认证数据A（其长度受支持）。

输出：密文C；认证标签T。

步骤：

1.令H = CIPHₖ(0¹²⁸)。

2.定义块J₀如下：若len (IV) = 96，则令J₀ = IV || 0³¹ || 1；若len (IV) ≠ 96，则令s = 128×⌈len (IV)/128⌉ - len (IV)，且令J₀ = GHASHₕ(IV || 0ˢ⁺⁶⁴ || [len (IV)]₆₄)。

3.令C = GCTRₖ(inc₃₂(J₀), P)。

4.令u = 128×⌈len (C)/128⌉ - len (C)，且令v = 128×⌈len (A)/128⌉ - len (A)。

5.定义块S 如下：S = GHASHₕ(A || 0ᵛ || C || 0ᵘ || [len (A)]₆₄ || [len (C)]₆₄)。

6.令T = MSBₜ(GCTRₖ(J₀, S))。

7.返回 (C, T)。

步骤1中，通过对“零”块应用分组密码，生成GHASH函数的哈希子密钥H。步骤2中，由IV生成预计数器块J₀。具体来说，当IV的长度为96位时，将填充串0³¹||1附加到IV后，形成J₀；否则，先对IV填充最少数量的“0”位（可能为零），使得到的串的长度为块大小（128位）的整数倍，再在该串后附加64个“0”位，接着附加IV长度的64位表示，然后对得到的串应用GHASH函数，形成J₀。步骤3中，对J₀应用32位递增函数，生成用于对明文P调用GCTR函数的初始计数器块，该调用的输出即为密文C。

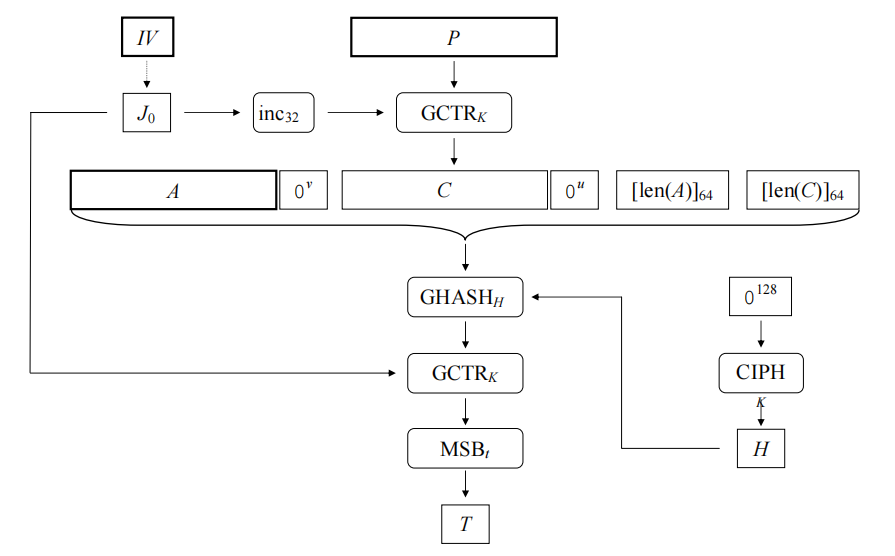
步骤4和步骤5中，分别对附加认证数据A和密文C填充最少数量的“0”位（可能为零），使得到的串的位长为块大小的整数倍。将这些串的拼接结果后附加A和C长度的64位表示，然后对结果应用GHASH函数，生成单个输出块S。步骤6中，使用步骤2中生成的J₀，通过GCTR函数对S进行加密，并将结果截断到指定的标签长度t，形成认证标签T。步骤7中，返回密文C和标签T作为输出。

上述认证加密函数如图3所示（图中未展示由IV确定J₀的过程，即步骤2）。

P→[GCTRₖ]→C，其中J₀经 inc₃₂处理后作为GCTR的输入之一

A||0ᵛ||C||0ᵘ||[len (A)]₆₄||[len (C)]₆₄→[GHASHₕ]→S→[GCTRₖ]→[MSBₜ]→T，其中 0¹²⁸→[CIPH]→H 为GHASH的输入之一

图 3：GCM-AEₖ(IV, P, A) = (C, T)



## 7.2 认证解密函数的算法

下面将详细说明认证解密函数的算法 5：

**算法 5：GCM-ADₖ(IV, C, A, T)**

前提条件：经过批准的、块大小为128位的分组密码CIPH；密钥K；支持的输入输出长度定义；与密钥相关联的支持的标签长度t。

输入：初始化向量IV；密文C；附加认证数据A；认证标签T。

输出：明文P或表示非真实性的标识FAIL。

步骤：

1.若IV、A 或 C 的位长不受支持，或者 len (T)≠t，则返回 FAIL。

2.令H = CIPHₖ(0¹²⁸)。

3.定义块 J₀如下：若 len (IV)=96，则 J₀=IV||0³¹||1；若 len (IV)≠96，则令 s=128×⌈len (IV)/128⌉-len (IV)，且 J₀=GHASHₕ(IV||0ˢ⁺⁶⁴||[len (IV)]₆₄)。

4.令P = GCTRₖ(inc₃₂(J₀), C)。

5.令u=128×⌈len (C)/128⌉-len (C)，且令 v=128×⌈len (A)/128⌉-len (A)。

6.定义块 S 如下：S=GHASHₕ(A||0ᵛ||C||0ᵘ||[len (A)]₆₄||[len (C)]₆₄)。

7.令T'=MSBₜ(GCTRₖ(J₀, S))。

8.若T=T'，则返回 P；否则返回FAIL。

步骤1中，验证实现对IV、密文C、附加认证数据A和认证标签T的长度的支持情况。步骤2中，通过对“零”块应用分组密码，生成GHASH函数的哈希子密钥H。步骤3中，按照认证加密函数（7.1节步骤2）的方式形成预计数器块J₀。步骤4中，对J₀应用32位递增函数，生成用于对密文C调用GCTR函数的初始计数器块，该调用的输出即为与给定IV对应的明文P。

步骤5和步骤6中，分别对附加认证数据A和密文C填充最少数量的“0”位（可能为零），使得到的串的位长为块大小的整数倍。将这些串的拼接结果后附加A和C长度的64位表示，然后对结果应用GHASH函数，生成单个输出块S。步骤7中，使用步骤3中生成的J₀，通过GCTR函数对S进行加密，并将结果截断到指定的标签长度t，形成标签T'。步骤8中，将步骤7的结果与作为输入接收的标签T进行比较，若两者相同，则返回明文P；否则，返回FAIL。

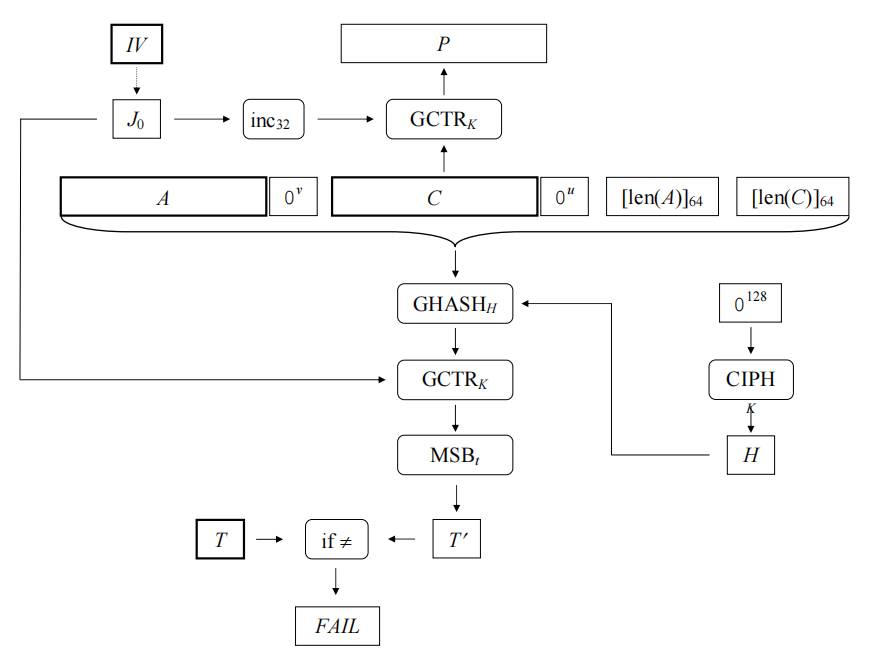
允许采用能产生正确输出的等效步骤集，例如可以先验证标签，再计算明文。

上述认证解密函数如图4所示（图中未展示由IV确定J₀的过程，即步骤3）。

C→[GCTRₖ]→P，其中J₀经 inc₃₂处理后作为GCTR的输入之一

A||0ᵛ||C||0ᵘ||[len (A)]₆₄||[len (C)]₆₄→[GHASHₕ]→S→[GCTRₖ]→[MSBₜ]→T'，其中0¹²⁸→[CIPH]→H为 GHASH 的输入之一，T'与T比较，若不等则输出FAIL

图 4：GCM-ADₖ(IV, C, A, T) = P 或 FAIL



# 8 关于IV和密钥的唯一性要求

GCM中的IV必须满足以下“唯一性”要求：

在两组（或多组）不同的输入数据上，使用相同的IV和相同的密钥调用认证加密函数的概率不得大于2⁻³²。

遵守这一要求对GCM的安全性至关重要。对于使用给定密钥的所有认证加密函数实例，若有任何一个IV被重复使用，那么该实现可能容易遭受文献[5]中描述且附录A中总结的伪造攻击。在实际应用中，这一要求几乎与密钥的保密性同等重要。

8.1节将讨论密钥建立在支持这一要求方面所起的作用。8.2节将给出两种被允许的、可满足这一要求的IV构造方法。8.3节将给出对认证加密函数调用次数的限制。

## 8.1 密钥建立

对于GCM而言，以下要求（通常是一般秘密密钥密码算法的规范）对于支持第8节中的唯一性要求具有明确的重要性：在其预期用户之间建立的任何GCM密钥都应以极高的概率保证是全新的。

实际上，如果密钥生成机制能够抵抗篡改，那么5.1节中的要求应能确保密钥在生成时是全新的。要实现这种抵抗能力，通常会对密钥生成机制的管理提出要求。

具体而言，如果密钥生成机制是确定性的，那么对该机制的管理应能提供强有力的保证，确保任何外部实体都无法诱导该机制重复使用先前的输入集，或者以其他方式导致重复输出先前的结果。例如，可以使用文献[9]中允许的以下协议的密钥派生函数来建立GCM密钥：传输层安全协议、互联网密钥交换协议v1和v2以及安全外壳协议。

同样，如果必须将新密钥传输给其预期接收者，那么传输 / 分发方法应能提供强有力的防“重放”保证，确保任何一方都无法诱导用先前的密钥替代预期的密钥。

GCM密钥应在经过批准的密钥管理结构框架内建立，以确保其全新性、保密性和真实性；此类结构的细节不在本建议的范围内。

## 8.2 IV构造

本建议提供了两种构造IV的框架。第一种构造（8.2.1节描述）依靠确定性元素来满足第8节中的唯一性要求；第二种构造（8.2.2节描述）依靠经过批准的随机位生成器（RBG）输出的足够长的字符串，且该RBG需具备足够的安全强度。

对于任何严格小于96位的受支持IV长度，在使用给定密钥的所有认证加密函数实例中，都必须使用8.2.1节中的构造方法。

对于任何96位或更长的受支持IV长度，在使用给定密钥的所有认证加密函数实例中，必须恰好使用其中一种构造方法，而不能同时使用两种。

例如，假设某个实现支持64位、96位、128位和160位的IV长度。对于64位IV，唯一的选择是8.2.1节中的构造方法；对于其他三种IV长度，一种可能的选择组合是：96位IV使用8.2.1节中的构造方法，128位和160位IV使用8.2.2节中的构造方法。

### 8.2.1 确定性构造

在确定性构造中，IV是两个字段的拼接，这两个字段分别称为固定字段和调用字段。固定字段应标识设备，或更一般地说，标识认证加密函数实例的上下文；调用字段应标识特定设备中认证加密函数的输入集。

对于任何给定的密钥，任何两个不同的设备不得共享相同的固定字段，且任何单个设备的任何两个不同的输入集不得共享相同的调用字段。遵守这两项要求意味着遵守第8节中关于IV的唯一性要求。

若有需要，固定字段本身可由两个或多个更小的字段构造而成。此外，只要其余位能确保对于使用相同密钥的其他设备，固定字段不会被完整重复，这些更小的字段中可能有一个字段包含任意位（即不一定是确定性的，也不一定是设备唯一的）。

同样，当只有一个上下文需要标识时（例如，当全新密钥仅限于通信协议的单个会话时），整个固定字段可由任意位组成。在这种情况下，如果会话中的不同参与者共享相同的固定字段，那么该协议应确保不同的数据输入具有不同的调用字段。

调用字段通常要么是整数计数器，要么是由本原多项式驱动的线性反馈移位寄存器，以确保最大周期长度。在这两种情况下，每次调用认证加密函数时，调用字段都会递增。

对于每个受支持的IV长度，固定字段和调用字段的长度和位置在密钥的整个生命周期内都应保持固定。为了促进默认96位IV长度的互操作性，本建议建议（但不要求）IV的前32位（即最左侧）为固定字段，后64位（即最右侧）为调用字段。

### 8.2.2 基于RBG的构造

在基于RBG的构造中，IV是两个字段的拼接，这两个字段分别称为随机字段和自由字段。对于实现所支持的、且与基于RBG的构造一起使用的每个IV长度，这些字段的长度在密钥的整个生命周期内都应保持固定。此外，随机字段的长度至少应为96位；自由字段可以为空。

如果i是与基于RBG的构造相关联的受支持IV长度，那么令r (i)表示随机字段的位长。随机字段要么由1）经过批准的、具有足够安全强度的RBG输出的r (i)位字符串组成，要么由2）对给定密钥的前一个IV的随机字段应用r (i)位递增函数得到的结果组成。RBG输出的r (i)位字符串称为直接随机字符串，通过应用r (i)位递增函数得到的随机字段称为其后续字符串。

对自由字段中的位没有要求。例如，它们可以像确定性构造中的固定字段那样标识设备，但在基于RBG的构造中，这些标识符不需要为每个设备都唯一。对于与基于RBG的构造相关联的任何IV长度，建议自由字段为空，以便随机字段就是整个IV。

任何两个不同设备中的RBG实例化都应是独立的，这样所有RBG实例化的直接随机字符串的分布有望是均匀的。例如，如果RBG实例化的初始化仅依赖于秘密种子，那么每个实例化都应使用不同的种子进行初始化。

## 8.3 对调用次数的限制

以下要求适用于所有使用1）非96位长度IV的确定性构造，或2）任何长度IV的基于RBG的构造的实现。换句话说，除非某个实现仅使用通过确定性构造生成的96位IV：

认证加密函数的总调用次数不得超过2³²次，包括使用给定密钥的所有IV长度和所有认证加密函数实例。

这是一项“全局”要求，可通过对使用给定密钥的每个认证加密函数实例设置适当的“局部”限制来实现。例如，假设某个实现由2¹⁰个设备组成，这些设备仅支持64位、96位和128位的IV。满足上述要求的一种方法是将每个设备的调用次数限制为：64位IV调用2²⁰次，96位IV调用2²¹次，128位IV调用2²⁰次。

对于基于RBG的IV构造，结合r (i)≥96的要求，上述要求足以确保满足第8节中的唯一性要求，这一点可从文献[4]中的讨论得出。

对于确定性构造，两个字段的长度意味着两项额外的操作限制。这些限制适用于任何受支持的IV长度，包括96位：

调用字段的位长限制了在任何给定的固定字段和密钥下认证加密函数的调用次数。具体而言，如果s表示调用字段中的位数，那么在不违反唯一性要求的情况下，认证加密函数在不同输入集上的调用次数不能超过2ˢ次。

同样，s位的固定字段意味着，对于给定的密钥和给定长度的IV，能够实现认证加密函数的不同设备 / 上下文的数量限制为2ˢ个。

# 9 验证实现的实际考虑因素

GCM 实现的设计者以及在特定系统中部署和维护该实现的信息技术（IT）专业人员，在满足第8节中的唯一性要求方面都扮演着重要角色，下面9.1节和9.2节将对此进行讨论。

这两节中的附加要求旨在在验证程序中证明对第8节中唯一性要求的符合性。具体而言，与文献[9]中的要求类似，这些两节中的要求适用于根据FIPS PUB 140-2（文献 [3]）或任何后续版本的FIPS 140的要求进行验证的实现。

未根据FIPS PUB 140-2的要求进行验证的实现可将9.1节和9.2节中的要求视为建议。

## 9.1 设计考虑因素

为了防止未授权方控制或影响IV的生成，GCM只能在满足FIPS PUB 140-2要求的密码模块中实现。具体而言，该模块的密码边界应包含一个“生成单元”，该单元根据8.2节中的一种构造方法生成IV。

用于根据FIPS 140-2要求进行验证的模块文档应描述该模块如何符合IV的唯一性要求。文档至少应涉及本节中的考虑因素，并清晰记录配置、部署和维护更大系统中的GCM实现的IT专业人员的职责。

GCM 模块的三个重要设计考虑因素如下：

如8.1节所述，应确保密钥的新鲜性。

在使用IV调用认证加密函数之前，IV应作为FIPS PUB 140-2中定义的关键安全参数。在此调用之前，IV应获得与根据FIPS PUB 140-2要求验证的模块中的其他关键安全参数相同的保护。

模块断电不得导致IV重复。如果生成单元无法从断电中恢复，那么在建立新鲜密钥之前，认证加密函数应进入故障状态。

从8.2节中实现的IV构造会影响从断电中恢复的选项。对于确定性构造，构造IV所需的所有确定性元素在恢复供电时都必须可用。例如，这些元素可以存储在非易失性存储器中。

恢复供电时，不得立即对该密钥重复使用先前的IV或任何其他先前的IV。避免这种重复的一种方法是确保定期存储在非易失性存储器中的调用字段值始终比序列中的操作值超前一个或多个值。

基于RBG的构造的一个潜在优势是，RBG可以设计为以简单的方式从断电中恢复，即无需维护系统的IT专业人员采取行动。例如，RBG可以包含一个非确定性的位源，该位源在恢复供电时会自动供RBG使用。

或者，RBG的整个状态可以定期存储在非易失性存储器中。在这种情况下，与确定性构造类似，恢复供电时，设计应确保在RBG的状态推进到生成该密钥的最后一个IV的状态之后，RBG才能输出用于新IV的字符串。换句话说，第一个新IV的直接随机字符串应以极高的概率不同于断电前生成的任何IV中的直接随机字符串。

即使该过程不是自动的，维护系统的IT专业人员也可以在恢复供电时重新初始化RBG（例如，使用新的种子）。在这种情况下，要么模块的设计，要么维护系统的IT专业人员，应确保符合8.2.2节中关于RBG的独立性条件。

## 9.2 操作考虑因素

是否符合IV的唯一性要求，进而GCM的安全性，最终取决于在特定系统中配置、部署和维护GCM模块的IT专业人员。GCM模块的文档应向IT专业人员提供针对该模块特定设计的详细说明。

关于唯一性要求的一些典型操作考虑因素如下：

任何GCM模块的配置或任何操作值是否容易受到任何未授权方的控制或影响？

对于任何给定的密钥，如何在所有曾经实现认证加密函数的模块中强制执行配置选择？

如8.1节所述，如何确保密钥的新鲜性？

如果某个实现不专门使用通过确定性构造生成的96位IV，那么如何确保 8.3 节中关于调用次数的要求得到满足？

如9.1节所述，模块内的IV生成如何从断电中恢复而不违反IV的唯一性要求？

以下考虑因素特定于确定性构造：

如何将设备标识符安装到固定字段中，以确保符合8.2.1节中关于固定字段的要求，无论是对于最初部署的模块还是任何后续部署的模块？

如8.3节中的第一个要点所述，调用字段的长度是否足以支持任何模块在任何密钥的生命周期内可能发生的所有调用？

如8.3节中的第二个要点所述，对于任何给定的密钥和IV长度，固定字段的长度是否足以支持将实现认证加密的模块数量？

如果可以更改固定字段和调用字段长度的默认设置，那么如何在使用任何给定密钥的所有模块中强制执行该选择？

以下考虑因素特定于基于RBG的构造：

如何初始化用于IV生成的 RBG，以确保符合 8.2.2 节中的独立性要求，无论是对于最初部署的模块还是任何后续部署的模块？

# 10 一致性

可以声称符合本建议的实现分为两类：GCM和GMAC。

如5.2.1.1 节和5.2.1.2 节所述，实现可以限制其支持的明文、附加认证数据、IV和认证标签的位长。尽管此类限制可能会影响互操作性，但只有完全取消明文（即 GMAC）才被视为不同类别的实现。

对于本建议中规定的每种算法，一致性实现可以用任何数学上等效的步骤集替换给定的步骤集。换句话说，允许采用能为任何输入产生正确输出的不同程序。

# 11 其他

## 附录 A：IV 唯一性要求的重要性

第8节包含了在使用任何给定密钥的所有认证加密函数实现中，GCM的IV的唯一性要求。本附录总结了为什么这一要求对GCM的安全性至关重要，这一点最初在文献[5]中有所描述。

密码算法的安全运行不仅取决于算法步骤的正确实现，还取决于对相关要求的遵守。任何对称密钥算法的一个明显例子是密钥需要保密。对于GCM，5.2.1节中关于IV唯一性的要求几乎同样重要，尽管并不那么明显。

首先考虑文献[10]中计数器模式对计数器块唯一性的要求。如果违反这一要求，那么计数器模式加密函数可能无法为与重复计数器块对应的数据流提供预期的机密性，尽管其他数据流不会受到影响。

可以说，计数器模式解密函数存在更严重的漏洞。计数器模式密文具有 “可延展性”，即翻转其任何位都会导致解密时明文的相应位翻转。因此，如果攻击者知道某个明文 - 密文对，并且能够诱导解密函数使用来自已知对的计数器块，那么，通过选择在已知密文中翻转哪些位，攻击者可以控制解密结果，其影响范围可达已知明文的长度。

这种漏洞促使将计数器模式加密机制与认证机制相结合，就像在GCM中所做的那样。其思想是防止攻击者篡改后的密文被接受为有效。然而，如文献[5]中详细描述的，GCM中的认证保证关键取决于IV的唯一性。

具体而言，如果在使用给定密钥的GCM认证加密函数中，IV被重复使用，那么攻击者很可能能够从生成的密文中确定哈希子密钥。然后，攻击者可以轻松构造伪造的密文。具体来说，给定认证加密函数调用的密文和标签输出，以及相关的IV和附加认证数据输入，攻击者可以替换任何密文和附加认证数据字符串，并使用该子密钥生成有效的替代标签。尽管这种伪造会使用相同的IV，但在大多数情况下，认证解密函数无法检测到这种重复。因此，认证保证基本上会丢失。

更糟糕的是，认证的丢失意味着GCM继承了其计数器模式密文的问题性可延展性。在上述场景中，知道与给定密文对应的原始明文后，攻击者可以构造替代密文，使其对应于任何相同长度的期望明文。换句话说，攻击者实际上可以控制认证解密函数的明文输出。

## 附录 B：认证保证

认证加密函数生成的认证标签提供了一种机制，通过执行认证解密函数，可以获得对明文、附加认证数据（和IV）的真实性的保证。这种保证的性质取决于认证解密函数的输出：

如果输出是明文，那么该模式的设计提供了强有力但并非绝对的保证，即数据的声称来源创建了该标签，即明文、附加认证数据（以及IV和标签）是真实的。因此，该模式还提供了强有力的保证，即此信息随后未被有意或无意地篡改。

如果输出是FAIL，那么可以确定至少有一个给定的输入（即密文、附加认证数据、IV或标签）是不真实的。

在第一种情况下，这种保证并非绝对的，因为原则上存在伪造的可能性。换句话说，攻击者（即无权访问密钥或认证加密函数的一方）可能能够为某些密文、附加认证数据和IV三元组生成正确的标签。

与任何基于标签的认证机制一样，如果攻击者随机选择一个t位标签，那么对于给定数据，该标签正确的概率预计为1/2ᵗ。然而，对于GCM，攻击者可以选择标签来增加这一概率，其增加幅度与密文和附加认证数据的总长度成正比。因此，GCM不太适合用于短标签长度或极长消息。

具体而言，如果n表示密文和附加认证数据的编码（即7.1节和7.2节中S的定义中GHASH函数的输入）中的总块数，那么存在一种构造 “定向” 密文伪造的方法，其预期成功率约为n/2ᵗ。此外，此攻击中的每次成功伪造1）会增加后续定向伪造成功的概率，2）会泄露有关哈希子密钥H的信息。最终，H可能会被完全泄露，其后果如附录A末尾所述：认证保证完全丢失。

文献[1]描述了这种攻击，文献[7]详细阐述了GCM的安全属性。附录C给出了关于使用两个最短标签长度（32位和64位）的指南和要求。

除了这种攻击之外，攻击者可能会尝试对认证解密的给定输入系统地猜测许多不同的标签，从而增加其中一个（或多个）标签最终被接受为有效的概率。因此，实现GCM的系统或协议应监控并在必要时限制每个密钥的未成功验证尝试次数。

此外，与大多数分组密码模式一样，GCM的安全保证会随着使用单个密钥处理的数据量增加而降低。因此，在密钥的生命周期内，通过调用认证加密函数保护的明文和附加认证数据的总块数应受到限制。对于大多数应用，合理的限制是2⁶⁴，这与8.3节中关于调用次数的要求一致。

## 附录 C：使用短标签的要求和指南

对于某些语音或视频应用，短认证标签可能是合适的。伪造大型流中的部分单个已认证“数据包”可能是可容忍的，因为大型流中的每个数据包可能只包含整体信息的很小一部分。

然而，即使对于语音和视频应用，由于附录B中总结并在文献[1]中详细描述的定向伪造攻击，短标签对于GCM可能存在问题。如果没有本附录中的要求和指南，攻击者可能会实际获取哈希子密钥H，之后认证保证将完全丢失。尽管如此，本建议并未完全禁止短标签，因为知识渊博的安全专业人员应该能够管理与这种攻击及其潜在改进相关的风险。

以下指南隐含地描述了可能适合使用短标签的特殊情况。在本文中，数据包指的是认证解密函数的一组输入。

在认证解密函数中未通过完整性检查的数据包应被静默丢弃。换句话说，接收数据包所依据的控制协议 / 系统不应提供关于单个数据包完整性的ACK/NACK响应。然而，接收方应在内部记录认证错误（以无法从侧信道信息检测到的方式），如果错误百分比超过正常范围，则终止连接或通知用户。这是任何协议 / 系统和任何算法选择的标准安全实践。

数据包中的附加认证数据应仅限于必要的头部信息，不应包含与加密数据一起需要认证的消息。

由大量数据包组成的整体消息的实质内容或含义不应因单个任意数据包的伪造而丢失或受损。例如，数据包可以携带一系列语音或视频数据片段，但单个数据包不应携带.txt或.doc文件。理想情况下，加密所依据的明文数据不应具有可猜测的固定模式。

控制协议 / 系统应频繁建立新的GCM密钥（从而建立新的哈希子密钥H），具体频率取决于单个数据包中可能出现的密文和附加认证数据的最大组合长度。此外，对于32位标签，此组合长度应非常小（约为数十或数百字节）；64位标签则将最大组合长度扩展到数百万字节。

符合这些指南的协议示例是承载互联网协议语音的安全实时传输协议，该协议运行在用户数据报协议之上。

下表1和表2分别量化了上述第4点中针对32位和64位标签的建议。每行有两个条目：1）单个数据包中密文和附加认证数据的最大组合长度（以字节为单位），2）在使用给定密钥的所有GCM实例中，认证解密函数的相应最大调用次数。

表 1：32位标签的限制

|  |  |
| --- | --- |
| 单个数据包中密文和附加认证数据的最大组合长度（字节） | 认证解密函数的最大调用次数 |
| 2⁵ | 2²² |
| 2⁶ | 2²⁰ |
| 2⁷ | 2¹⁸ |
| 2⁸ | 2¹⁵ |
| 2⁹ | 2¹³ |
| 2¹⁰ | 2¹¹ |

对于任何支持32位或64位标签的实现，必须执行表1或表2中的某一行。具体而言，支持的明文 / 密文和附加认证数据的长度应确保每个有效数据包都满足该行中的长度限制，并且控制协议 / 系统应确保在认证解密函数的调用次数超过该行中给出的最大值之前更换密钥。也可以执行更小的最大值。

表 2：64 位标签的限制

|  |  |
| --- | --- |
| 单个数据包中密文和附加认证数据的最大组合长度（字节） | 认证解密函数的最大调用次数 |
| 2¹⁵ | 2³² |
| 2¹⁷ | 2²⁹ |
| 2¹⁹ | 2²⁶ |
| 2²¹ | 2²³ |
| 2²³ | 2²⁰ |
| 2²⁵ | 2¹⁷ |

## 附录 D：防止消息重放

如附录B所述，在认证解密函数中成功验证标签可保证数据的真实性；然而，向认证解密函数提供输入数据的一方可能不是明文和附加认证数据的原始来源。换句话说，与许多其他认证机制一样，GCM本身并不能防止攻击者拦截认证加密调用的输出，并在稍后将其 “重放” 给认证解密函数，例如，试图冒充有权访问密钥的一方。在某些协议中，攻击者甚至可能能够使用验证者自己在协议早期生成的数据。

控制该过程的系统或协议可以通过监控为认证解密提供的IV的任何重复来防止此类事件。或者，可以将某些标识信息纳入附加认证数据中。此类信息的示例包括顺序消息编号或时间戳。在成功验证真实性后，此信息可提供检测重放消息、乱序消息或丢失消息的方法。

## 附录 E：参考文献

1. Ferguson, N., Authentication Weaknesses in GCM, Natl. Inst. Stand. Technol. [网页], http://www.csrc.nist.gov/groups/ST/toolkit/BCM/documents/comments/ CWC-GCM/Ferguson2.pdf, 2005 年 5 月 20 日。
2. FIPS Publication 197, The Advanced Encryption Standard (AES), U.S. DoC/NIST, 2001 年 11 月 26 日。
3. FIPS Publication 140-2, Security Requirements for Cryptographic Modules, U.S. DoC/NIST, 2001 年 5 月 25 日。
4. IEEE P1619.1™/D23, Draft Standard for Authenticated Encryption with Length Expansion for Storage Devices.
5. Joux, Authentication Failures in NIST version of GCM, Natl. Inst. Stand. Technol. [网页], http://www.csrc.nist.gov/groups/ST/toolkit/BCM/documents/comments/80038\_Series-Drafts/GCM/Joux\_comments.pdf.
6. D. McGrew, J. Viega, The Galois/Counter Mode of Operation (GCM), Natl. Inst. Stand. Technol. [网页], http://www.csrc.nist.gov/groups/ST/toolkit/BCM/documents/ proposedmodes/gcm/gcm-revised-spec.pdf, 2005 年 5 月 31 日。
7. D. McGrew and J. Viega. The Security and Performance of the Galois/Counter Mode (GCM) of Operation. Proceedings of INDOCRYPT ’04, Springer-Verlag, 2004. 全文可从 IACR Cryptology ePrint Archive 获取：Report 2004/193, [网页], http://eprint.iacr.org/2004/193/, 2004 年 10 月 7 日。
8. National Institute of Standards and Technology and Communications Security Establishment, Implementation Guidance for FIPS Pub. 140-2 and the Cryptographic Module Validation Program, Natl. Inst. Stand. Technol. [网页], http://csrc.nist.gov/ groups/STM/cmvp/documents/fips140-2/FIPS1402IG.pdf.
9. NIST Special Publication 800-38A, 2001 ED, Version 1, Recommendation for Block Cipher Modes of Operation—Methods and Techniques, December 2001, Natl. Inst. Stand. Technol. [网页], http://www.csrc.nist.gov/publications/nistpubs/800-38a/ sp800-38a.pdf.

**本译文归属 github 用户 WhatTheFuck-cyber**