ICS 35.040
L 80

中华人民共和国国家标准

GB/T 36624—2018

信息技术　安全技术　可鉴别的加密机制

Information technology—Security techniques—Authenticated encryption

(ISO/IEC 19772:2009,MOD)

2018-09-17 发布　　2019-04-01 实施

国家市场监督管理总局
中国国家标准化管理委员会　发布

目　　次

前　言

本标准按照 GB/T 1.1—2009 给出的规则起草。

本标准使用重新起草法修改采用 ISO/IEC 19772:2009《信息技术　安全技术　可鉴别的加密机制》。

本标准与 ISO/IEC 19772:2009 的技术性差异及其原因如下：

——关于规范性引用文件，本标准做了具有技术性差异的调整，以适应我国的技术条件，调整的情况集中反映在第 2 章“规范性引用文件”中，具体调整如下：

- 用等同采用国际标准的 GB/T 15852.1—2008 代替了 ISO/IEC 9797-1(见 8.4)；
- 用 GB/T 17964—2008 代替了 ISO/IEC 10116(见 9.3)；
- 用 GB/T 32907—2016 代替了 ISO/IEC 18033-3(见第 5 章)；
- 增加引用了 GB/T 25069—2010(见第 3 章)；

——第 3 章中，直接采用现行国家标准中已定义的术语定义，删除了部分常见的通用定义。

与 ISO/IEC 19772:2009 相比在结构上有较大调整，具体如下：

——对 ISO/IEC 19772:2009 范围一节内容进行修改，其中部分内容移至第 5 章概述；

——考虑到我国国情及技术的实际应用范围，本标准采用了 ISO/IEC 19772:2009 中第 7 章至第 11 章规定的五种可鉴别加密机制，删除 ISO/IEC 19772:2009 中第 6 章规定的可鉴别加密机制；

——将 ISO/IEC 19772:2009 中规范性附录 C 调整为附录 A，相应的，ISO/IEC 19772:2009 中资料性附录 A 和附录 B 分别调整为附录 B 和附录 C；

——附录 C 中给出的数据示例，修改为采用 SM4 算法作为示例。

本标准做了下列编辑性修改：

——纳入了 ISO/IEC 19772:2009 勘误版本的内容。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本标准由全国信息安全标准化技术委员会(SAC/TC 260)提出并归口。

本标准起草单位：中国科学院数据与通信保护研究教育中心、中国科学院软件研究所、北京江南天安科技有限公司。

本标准主要起草人：王琼霄、蔡权伟、张颖君、赵宇航、宋利、吴鹏一、闻楠、林璟锵、荆继武、王明月、宋天林。

信息技术　安全技术　可鉴别的加密机制

1　范围

本标准规定了五种可鉴别的加密机制，通过定义数据串的处理方法来实现以下安全目标：

——数据保密性，保护数据不会向非授权者泄露；

——数据完整性，确保数据接收者能够验证数据是否被修改；

——数据源鉴别，确保数据接收者能够验证数据始发者的身份。

本标准给出了五种可鉴别的加密机制 ASN.1 定义。

本标准适用于需对数据进行保密性、完整性保护及数据源鉴别的应用和系统。

2　规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本(包括所有的修改单)适用于本文件。

GB/T 15852.1—2008　信息技术　安全技术　消息鉴别码　第 1 部分：采用分组密码的机制(ISO/IEC 9797-1：1999，IDT)

GB/T 17964—2008　信息安全技术　分组密码算法的工作模式

GB/T 25069—2010　信息安全技术　术语

GB/T 32907—2016　信息安全技术　SM4 分组密码算法

3　术语和定义

GB/T 15852.1—2008、GB/T 17964—2008、GB/T 25069—2010 界定的以及下列术语和定义适用于本文件。

3.1

可鉴别的加密　authenticated encryption

一种可逆的数据转换，利用密码算法产生数据对应的密文，非授权实体无法在不被发现的情况下对该密文进行修改，同时提供了数据保密性、数据完整性与数据源鉴别。

3.2

可鉴别的加密机制　authenticated encryption mechanism

用于实现数据保密性保护并提供数据完整性和数据源鉴别的密码技术，包括加密和解密两个处理过程。

3.3

数据完整性　data integrity

数据没有遭受以未授权方式所作的更改或破坏的特性。

[GB/T 25069—2010，定义 2.1.36]

3.4

分组密码　block cipher

又称块密码算法，一种对称密码算法，将明文划分成固定长度的分组进行加密。

[GB/T 17964—2008,定义 3.1.2]

3.5

加密系统　encryption system

用于保护数据保密性的密码技术,包括加密算法、解密算法和密钥生成三个处理过程。

3.6

消息鉴别码　message authentication code;MAC

利用对称密码技术和秘密密钥,由消息所导出的数据项。任何持有这一秘密密钥的实体,可利用消息鉴别码检查消息的完整性和始发者。

[GB/T 15852.1—2008,定义 3.2.5]

3.7

分块　partition

将任意长度的字符串划分成为比特数相同的数据分组,除最后一个分组外。若每个等长的分组包含 n 比特,最后一个数据分组包含 r 比特,那么 $0<r\leqslant n$。

3.8

秘密密钥　secret key

由特定的一组实体使用的用于对称加密技术的密钥。

3.9

对称加密系统　symmetric encryption system

基于对称密码技术的加密系统,加密和解密使用相同的秘密密钥。

4　符号

下列符号适用于本文件。

A	额外的可鉴别数据
C	可鉴别的加密的数据
D	可鉴别的加密机制的输入数据
d	分组密码解密算法,$d_K(Y)$表示使用分组密码算法及密钥 K,对分组长度为 n 比特的数据 Y 进行解密得到的结果
e	分组密码加密算法,$e_K(Y)$表示使用分组密码算法及密钥 K,对分组长度为 n 比特的数据 Y 进行加密得到的结果
K	数据始发者和接收者共享的分组密码的秘密密钥,该密钥用于可鉴别的数据加密机制
m	对 D 进行分块后,得到的数据分组的数量
n	一个数据分组的长度(以比特为单位)
t	标记(tag)长度(以比特为单位)
0^i	由长度为 i 比特的 0 构成的分组
1^i	由长度为 i 比特的 1 构成的分组
$\oplus$	逐位异或
$\|$	字符串连接符号
$\#$	将数字转换成二进制表示。若 k 是一个整数 $0\leqslant k\leqslant 2^a$,那么 $\#_a(k)$是一个长度为 a 的比特块,左侧为高位,该二进制表示的值与 k 相等
$\#^{-1}$	将二进制转换为整数表示。若 A 是一个长度为 n 的二进制表示,那么 $\#_n(\#^{-1}(A))=A$

$X\|_s$	截取比特块 X 中的左 s 比特
$X\|^s$	截取比特块 X 中的右 s 比特
$X\ll 1$	将比特块 X 左移一位，$Y=X\ll 1$ 的最右一位始终设置为 0
$X\gg 1$	将比特块 X 右移一位，$Y=X\gg 1$ 的最左一位始终设置为 0
len	取长运算，以比特串 X 作为输入，输出值为 X 的比特数
mod	求模运算

5 概述

本标准规定的五种可鉴别的加密机制均基于分组密码算法，同时要求数据的始发者和接收者共享密钥。其中，可鉴别的加密机制 1 和 4 仅能对加密的数据实现数据鉴别，可鉴别的加密机制 2、3、5 还可支持对未加密的数据进行数据鉴别。在上述三个支持对未加密数据进行数据鉴别的机制中，被保护的数据可划分为两部分：

D：被加密并进行数据完整性保护的数据；

A：额外的可鉴别数据，A 被进行数据完整性保护，但是未被加密；A 可能为空。

本标准中规定的可鉴别的加密机制需满足以下要求，采用可鉴别的加密机制保护数据的始发者和接收者应：

a) 协商使用本标准所规定的某个特定的可鉴别的加密机制；

b) 协商使用某个特定的分组密码算法，分组密码算法应用使用 GB/T 32907—2016 规定的密码算法或国家密码管理部门认可的密码算法；

c) 共享一个秘密密钥 K：除了可鉴别的加密机制 4 以外的所有机制中，此秘密密钥应作为选定的分组密码算法的密钥；在可鉴别的加密机制 4 中，此密钥应作为密钥派生程序的输入。

本标准中规定的可鉴别的加密机制的 ASN.1 模块见附录 A，其他使用指导参见附录 B。

本标准凡涉及密码算法的相关内容，按国家有关法规实施；凡涉及到采用密码技术解决保密性、完整性、真实性、抗抵赖性需求的应遵循密码相关国家标准和行业标准。

6 可鉴别的加密机制 1

6.1 简介

本章定义的可鉴别的加密机制为 Key Wrap。具体示例参见附录 C 的 C.2。

6.2 特定符号与标记

本机制涉及的符号与标记如下：

$C_0, C_1, \cdots, C_m$	可鉴别加密过程产生的输出数据，由 $(m+1)$ 个长度为 64 比特的分组构成的序列
$D_1, D_2, \cdots, D_m$	对数据 D 进行分块后得到的比特分组序列，由 m 个长度为 64 比特的分组构成
$R_1, R_2, \cdots, R_m$	在加密和解密过程中计算的比特分组序列，由 m 个长度为 64 比特的分组构成
Y	在加密和解密过程中使用的长度为 64 比特的分组
Z	在加密和解密过程中计算得到的长度为 128 比特的分组

6.3 具体要求

本机制使用的分组密码算法应是128位的分组密码算法，即$n=128$。本机制要求被保护的数据D应至少包含128比特，且数据长度应是64比特的整数倍[即$\mathrm{len}(D)=64m, m>1$]。

6.4 加密程序

始发者应执行以下步骤来保护数据D：

a) 将D进行分块，得到一个由m个长度为64比特的分组构成的序列$D_1, D_2, \cdots, D_m$。其中，D_1包含D的第一个64比特，D_2包含D的第二个64比特，依此类推。

b) 将64比特块Y的值设为0xA6A6A6A6A6A6A6A6(十六进制表示)，即二进制的(10100110 10100110 … 10100110)。

c) 对于$i=1,2,\cdots,m$，令$R_i=D_i$。

d) 对于$i=1,2,\cdots,6m$，执行以下四个步骤：

1) $Z=e_K(Y \| R_1)$；

2) $Y=Z|_{64} \oplus \#_{64}(i)$；

3) 对于$j=1,2,\cdots,m-1$，令$R_j=R_{j+1}$；

4) $R_m=Z|^{64}$。

e) $C_0=Y$。

f) 对于$i=1,2,\cdots,m$，令$C_i=R_i$。

上述过程的输出，即数据D的可鉴别的加密数据输出为$C=C_0 \| C_1 \| \cdots \| C_m$。加密过程的输出$C$由$(m+1)$个长为64比特的分组构成，比输入$D$多了64比特。

6.5 解密程序

接收者应执行以下步骤来解密和验证可鉴别的加密数据C：

a) 如果$\mathrm{len}(C)$不是64的倍数或者是小于192，计算并输出INVALID。

b) 将C进行分块，得到一个由$m+1$个长度为64比特的数据分组构成的序列$C_0, C_1, \cdots C_m$。其中，C_0包含C的第一个64比特，C_1包含C的第二个64比特，依此类推。

c) 令$Y=C_0$。

d) 对于$i=1,2,\cdots,m$，令$R_i=C_i$。

e) 对于$i=6m, 6m-1, \cdots, 1$执行以下四个步骤：

1) $Z=d_K([Y \oplus \#_{64}(i)] \| R_m)$；

2) $Y=Z|_{64}$；

3) 对于$j=m, m-1, \cdots, 2$，令$R_j=R_{j-1}$；

4) $R_1=Z|^{64}$。

f) 如果$Y=(1010011010100110\cdots10100110)$（即0xA6A6A6A6A6A6A6A6），则输出$D=R_1 \| R_2 \| \cdots \| R_m$，否则输出INVALID。

7 可鉴别的加密机制2

7.1 简介

本章定义的可鉴别的加密机制为CCM。具体示例参见C.3。

7.2 特定符号与标记

本机制涉及的符号与标记如下：

B	用于计算标签(tag)值的比特分组
$B_1, B_2, \cdots, B_v$	用于计算标签(tag)值的比特分组序列，每个分组为 n 比特
$C_1, C_2, \cdots, C_m$	可鉴别加密过程产生的部分输出数据，由 m 个长度为 128 比特的分组构成的序列
$D_1, D_2, \cdots, D_m$	对数据 D 进行分块后得到的比特分组序列，由 m 个长度为 128 比特的分组构成
F	8 比特的标志
L	D 的长度(以字节为单位)，数据长度不包括填充以及长度块 D_0
r	D_m 包含的字节数
S	开始变量，长度为 $(120-8w)$ 比特
T	明文标签值，长度为 t 比特
T'	在解密操作中计算得出的标签值
U	加密后的标签值，长度为 t 比特
v	用于计算标签值的变量
w	消息长度域的长度(以字节为单位)
X	在加密和解密过程中计算的长度为 128 比特的分组
Y	在加密和解密过程中计算的长度为 128 比特的分组

7.3 具体要求

在使用此可鉴别的加密机制前，数据的始发者和接收者首先需协商确定以下事项：

a) t：标签(tag)的长度(以比特为单位)，t 的值应在集合 $\{32,48,64,80,96,112,128\}$ 中选取；

b) w：消息长度域的长度(以字节为单位)，w 的值应在集合 $\{2,3,4,5,6,7,8\}$ 中选取。w 值的选取会影响保护数据的最大长度，最大的信息长度为 2^{8w+3} 位，即 2^{8w} 个字节。

本机制使用的分组密码算法应是 128 位的分组密码算法，即 $n=128$。使用本机制保护的数据 D 及额外的可鉴别数据 A，其长度均应是 8 比特的整数倍。

7.4 加密程序

始发者应执行以下步骤来保护数据 D。令 $L=\mathrm{len}(D)/8$，即 L 是 D 的长度，单位为字节。

a) 选定开始变量 S，S 包含 $(15-w)$ 个字节[即 $(120-8w)$ 比特]。对每个被保护的数据应用选取各不相同的 S，并确保 S 可被消息接收者获知。S 的取值无需保证不可预知的或是秘密的。

b) 对数据 D 进行右填充，填充 $(16-r)$ 个字节的 0(也就是 0 到 120 个比特的 0)，从而使得 D 填充后长度为 128 比特的整数倍。将 D 进行分块，得到一个由 m 个长度为 128 比特的分组构成的序列 $D_1, D_2, \cdots, D_m$。其中，D_1 包含 D 的第一个 128 比特，D_2 包含 D 的第二个 128 比特，依此类推。

c) 如果 $\mathrm{len}(A)=0$，令标志 $F=0^2 \| \sharp_3((t-16)/16) \| \sharp_3(w-1)$。

d) 如果 $\mathrm{len}(A)>0$，令标志 $F=0\|1\| \sharp_3((t-16)/16) \| \sharp_3(w-1)$。

说明：F 最左端的一位是“保留位”，即在当前版本的算法中置为 0，但在将来的算法中可能会用到。之后的一位置为 0 的含义是所有被保护的数据都已经加密。

e) 令 $X=e_K(F \| S \| \sharp_{8w}(L))$。

f) 如果 $\mathrm{len}(A)>0$，执行以下六个步骤：

1) 如果 $0<\mathrm{len}(A)<65\ 280$，令 $B=\#_{16}(\mathrm{len}(A)/8)\ \|\ A$；

2) 如果 $65\ 280\leqslant\mathrm{len}(A)<2^{32}$，令 $B=1^{15}\ \|\ 0\ \|\ \#_{32}(\mathrm{len}(A)/8)\ \|\ A$；

3) 如果 $2^{32}\leqslant\mathrm{len}(A)<2^{64}$，令 $B=1^{16}\ \|\ \#_{64}(\mathrm{len}(A)/8)\ \|\ A$；

4) 将 B 进行分块，得到一个分组序列 $B_1,B_2,\cdots,B_v$。其中，B_1 包含 B 的第一个 n 比特，B_2 包含 B 的第二个 n 比特，依此类推，直至 B_v 包括 B 的最后 k 个比特，其中 $0<k\leqslant n$，由此可得，$\mathrm{len}(B)=(v-1)n+k$；

5) 对 B_v 进行右填充，填充 $(n-k)$ 比特个 0，即 $B_v=B_v\ \|\ 0^{n-k}$；

6) 对于 $i=1,2,\cdots,v$，令 $X=e_K(X\oplus B_i)$。

g) 对于 $i=1,2,\cdots,m$，令 $X=e_K(X\oplus D_i)$。

h) 令 $T=X\mid_t$。

i) $F=(0^5\ \|\ \#_3(w-1))$，且 $Y=(F\ \|\ S\ \|\ 0^{8w})$。

j) 令 $U=T\oplus[e_K(Y)]\mid_t$。

k) 对于 $i=1,2,\cdots,m-1$，执行以下两个步骤：

1) $Y=(F\ \|\ S\ \|\ \#_{8w}(i))$；

2) $C_i=D_i\oplus e_K(Y)$。

l) 令 $Y=(F\ \|\ S\ \|\ \#_{8w}(m))$，且 $C_m=[D_m\oplus e_K(Y)]\mid_{8r}$。

上述过程的输出，即数据 D 的可鉴别的加密数据输出为 $C=C_1\ \|\ C_2\ \|\cdots\|\ C_{m-1}\ \|\ C_m\ \|U$。

7.5 解密程序

接收者应执行以下步骤来解密和验证可鉴别的加密数据 C：

a) 如果 C 的长度不是 8 比特的整数倍，停止计算并输出 INVALID。

b) 如果 C 的长度小于 $(t+8)$ 比特，停止计算并输出 IVALID。

c) 确定整数 m 和 r，使得 C 包含 $(128(m-1)+8r+t)$ 比特，其中 $0<r\leqslant16$。将 C 进行分块，得到一个数据分组的序列：$C_1,C_2,\cdots C_m,U$。其中，C_1 包含 C 的第一个 128 比特，C_2 包含 C 的第二个 128 比特，依此类推，直到 C_m 包含 $8r$ 比特。最后，令 U 为 C 的最后 t 比特。

d) 令标志 $F=(0^5\ \|\ \#_3(w-1))$，且 $Y=(F\ \|\ S\ \|\ 0^{8w})$。

e) 令 $T=U\oplus[e_K(Y)]\mid_t$。

f) 对于 $i=1,2,\cdots,m-1$，执行以下两个步骤：

1) 令 $Y=(F\ \|\ S\ \|\ \#_{8w}(i))$；

2) $D_i=C_i\oplus e_K(Y)$。

g) $Y=(F\ \|\ S\ \|\ \#_{8w}(m))$，且 $D_m=[C_m\oplus e_K(Y)]\mid_{8r}$。

h) 令 $D=D_1\ \|\ D_2\ \|\cdots\|\ D_m$，且令 $L=16m-16+r$。

i) 对 D_m 进行右填充，填充 $(128-8r)$ 比特个 0，即 $D_m=D_m\ \|\ 0^{128-8r}$。

j) 如果 $\mathrm{len}(A)=0$，令标志 $F=0^2\ \|\ \#_3((t-16)/16)\ \|\ \#_3(w-1)$。

k) 如果 $\mathrm{len}(A)>0$，令标志 $F=0\ \|\ 1\ \|\ \#_3((t-16)/16)\ \|\ \#_3(w-1)$。

l) 令 $X=e_K(F\ \|\ S\ \|\ \#_{8w}(L))$。

m) 如果 $\mathrm{len}(A)>0$，执行以下六个步骤：

1) 如果 $0<\mathrm{len}(A)<65\ 280$，令 $B=\#_{16}(\mathrm{len}(A)/8)\ \|\ A$；

2) 如果 $65\ 280\leqslant\mathrm{len}(A)<2^{32}$，令 $B=1^{15}\ \|\ 0\ \|\ \#_{32}(\mathrm{len}(A)/8)\ \|\ A$；

3) 如果 $2^{32}\leqslant\mathrm{len}(A)<2^{64}$，令 $B=1^{16}\ \|\ \#_{64}(\mathrm{len}(A)/8)\ \|\ A$；

4) 将 B 进行分块，得到一个分组序列 $B_1,B_2,\cdots,B_v$。其中，B_1 包含 B 的第一个 n 比特，B_2 包含 B 的第二个 n 比特，依此类推，直至 B_v 包括 B 的最后 k 比特，其中 $0<k\leqslant n$，由

此可得，$\mathrm{len}(B)=(v-1)n+k$；

5） 对 B_v 进行右填充，填充（$n-k$）比特个 0，即 $B_v=B_v \parallel 0^{n-k}$；

6） 对于 $i=1,2,\cdots,v$，令 $X=e_K(X \oplus B_i)$。

n） 对于 $i=1,2,\cdots,m$，$X=e_K(X \oplus D_i)$；

o） $T'=X|_t$。

p） 如果 $T=T'$，那么输出在步骤 h）中计算得出的数据 D 和额外的可鉴别数据 A，否则输出 IVALID。

8 可鉴别的加密机制 3

8.1 简介

本章定义的可鉴别的加密机制为 EAX。具体示例参见附录 C.4。

8.2 特定符号与标记

本机制涉及的符号与标记如下：

$C_1,C_2,\cdots,C_m$	比特分组的序列，是可鉴别加密机制操作得到的部分输出，其中 C_1 到 C_{m-1} 每个分组长为 n 比特，分组 C_m 的长度可能不是 n 比特
$D_1,D_2,\cdots,D_m$	对数据 D 进行分块后得到的比特分组序列，其中 D_1 到 D_{m-1} 每个分组长为 n 比特，分组 D_m 的长度可能不是 n 比特
E_0,E_1,E_2	在加密和解密过程中计算的长度为 n 比特的分组
M	在加密和解密过程中使用的函数
S	开始变量（n 比特）
T	标签（t 比特），与加密后的消息拼接，用于提供数据完整性保护
T'	在解密操作中计算得出的标签值
U	加密后的标签值，长度为 t 比特
W	在加密和解密过程中计算的长度为 n 比特的分组

8.3 具体要求

在使用此可鉴别的加密机制前，数据的始发者和接收者首先需确定标准的长度 t，其中 $0<t\leqslant n$。

8.4 M 函数定义

为定义可鉴别的加密机制中的加密和解密程序需要对函数 M 进行定义。函数 M 以任意长度的比特串以及一个分组密码的密钥作为输入，输出为一个长度为 n 比特的数据分组。此函数的定义如下所示。

假设 X 是一个比特串，K 是选定的分组密码算法的密钥；那么，$M_K(X)$ 应与利用密钥 K 使用 GB/T 15852.1—2008 中的 MAC 算法 5（也称为 OMAC）对数据 X 进行计算得到的结果一致，其中 MAC 算法使用的分组密码算法与可鉴别加密机制采用的分组密码算法相同。

8.5 加密程序

始发者应执行以下步骤来保护数据 D：

a） 选取一个长度为 n 比特的开始变量 S。对每个被保护的数据应用选取各不相同的 S，并确保 S 可被消息接收者获知。S 的取值无需保证不可预知的或是秘密的。

b) 令 $E_0 = M_K(0^n \| S)$。

c) 令 $E_1 = M_K(0^{n-1} \| 1 \| A)$。

d) 令 $W = E_0$。

e) 将 D 进行分块，得到一个数据分组的序列：$D_1, D_2, \cdots D_m$。其中，D_1 包含 D 的第一个 n 比特，D_2 包含 D 的第二个 n 比特，依此类推，直到 D_m 包含 D 的最后 r 比特（$0 < r \leqslant n$）。由此可得，$\mathrm{len}(D) = (m-1)n + r$。

f) 对于 $i = 1, 2, \cdots, m-1$，执行以下两个步骤：

1) $C_i = D_i \oplus e_K(W)$；

2) $W = \#_n(\#^{-1}(W) + 1 \bmod 2^n)$。

g) $C_m = D_m \oplus [e_K(W)]|_r$。

h) $E_2 = M_K(0^{n-2} \| 1 \| 0 \| C_1 \| C_2 \| \cdots \| C_m)$。

i) $T = [E_0 \oplus E_1 \oplus E_2]|_t$。

上述过程的输出，即数据 D 的可鉴别的加密数据输出为 $C = C_1 \| C_2 \| \cdots \| C_m \| T$。

8.6 解密程序

接收者应执行以下步骤来解密和验证可鉴别的加密数据 C：

a) 如果 C 的长度小于 t，则停止计算并输出 INVALID。

b) 确定整数 m 和 r，使得 C 的比特数等于 $(m-1)n + r + t$，其中 $0 < r \leqslant n$。将 C 进行分块，得到一个数据分组的序列：$C_1, C_2, \cdots C_m, T$。其中，C_1 包含 C 的第一个 n 比特，C_2 包含 C 的第二个 n 比特，依此类推，直到 C_m 包含 r 比特（$0 < r \leqslant n$）。最后，令 T 为 C 的最后 t 比特。

c) 令 $E_0 = M_K(0^n \| S)$。

d) 令 $E_1 = M_K(0^{n-1} \| 1 \| A)$。

e) 令 $E_2 = M_K(0^{n-2} \| 1 \| 0 \| C_1 \| C_2 \| \cdots \| C_m)$。

f) $T' = [E_0 \oplus E_1 \oplus E_2]|_t$。

g) 如果 $T \neq T'$，则停止计算并输出 INVALID。

h) 令 $W = E_0$。

i) 对于 $i = 1, 2, \cdots, m-1$，执行以下两个步骤：

1) $D_i = C_i \oplus e_K(W)$；

2) $W = \#_n(\#^{-1}(W) + 1 \bmod 2^n)$。

j) $D_m = C_m \oplus [e_K(W)]|_r$。

k) 输出 D 和 A。

9 可鉴别的加密机制 4

9.1 简介

本章定义的可鉴别的加密机制可以由任意的加密算法和消息鉴别码（MAC）机制组合而成。在本机制中，先对数据进行加密，再对密文计算消息鉴别码（MAC）。

9.2 特定符号与标记

本机制涉及的符号与标记如下：

C'　　对数据 D 加密后得到的比特串

δ　　解密函数，分组密钥 K_1、开始变量 S 及密文 C' 作为函数的输入，使用选定的

	工作模式进行解密，输出解密后的数据，表示为 $\delta_{K_1,S}(C')$
ε	加密函数，分组密钥 K_1 、开始变量 S 及明文 D 作为函数的输入，使用选定的工作模式进行加密，输出加密后的数据，表示为 $\varepsilon_{K_1,S}(D)$
f	MAC 函数，使用密钥 K_2 对数据 X 计算 MAC，表示为 $f_{K_2}(X)$
K_1	分组密码使用的密钥
K_2	MAC 函数使用的密钥
S	开始变量（n 比特）
T	标志（t 比特），与加密后的消息拼接，用于提供数据完整性保护
T'	在解密操作中计算得出的标志值

9.3 具体要求

在使用此可鉴别的加密机制前，数据的始发者和接收者首先需协商确定以下事项：

a) 分组密码算法的工作模式：分组密码算法工作模式可以为 GB/T 17964—2008 中指定的采用除 ECB 模式以外的任意工作模式。

b) MAC 算法：MAC 算法应从 GB/T 15852.1—2008 指定的算法中选择（假设 MAC 算法生成的标签长度为 t 比特）。

c) 密钥获取方法：双方可通过共享密钥 K 获得密钥对 (K_1,K_2) ，其中 K_1 是分组密码算法使用的密钥，K_2 是 MAC 算法使用的密钥。

9.4 加密程序

始发者应执行以下步骤来保护数据 D ：

a) 选取一个适合于所选的分组密码算法工作模式的开始变量 S。在一个密钥的生命周期内，每个被保护的数据应选取各不相同的 S ，并确保 S 可被消息接收者获知。对 S 的进一步要求可参考 GB/T 17964—2008 相关章节内容以及参见附录 B.7 的内容。

b) $C'=\varepsilon_{K_1,S}(D)$ 。

c) $T=f_{K_2}(S\parallel C')$ 。

上述过程的输出，即数据 D 的可鉴别的加密数据输出为 $C=C'\parallel T$ ，以及开始变量 S。

9.5 解密程序

接收者应执行以下步骤来解密和验证可鉴别的加密数据 C ，该加密数据 C 同时伴有开始变量 S ：

a) 如果 C 的长度小于 t ，则停止计算并输出 INVALID。

b) 将 C 最右边的 t 比特作为 T ，将 C 移除最右边 t 比特后的其他部分作为 C' ，即 $C=C'\parallel T$。

c) 令 $T'=f_{K_2}(S\parallel C')$ 。

d) 如果 $T\neq T'$ ，则停止计算并输出 INVALID。

e) $D=\delta_{K_1,S}(C')$ 。

f) 输出 D。

10 可鉴别的加密机制 5

10.1 简介

本章定义的可鉴别的加密机制为 GCM。具体示例参见附录 C.5。

10.2 特定符号与标记

本机制涉及的符号与标记如下：

$C_1, C_2, \cdots, C_m$　比特分组的序列，是可鉴别加密机制操作得到的部分输出，其中 C_1 到 C_{m-1} 每个分组长为 128 比特，分组 C_m 的长度可能不足 128 比特

$D_1, D_2, \cdots, D_m$　对数据 D 进行分块后得到的比特分组，其中 D_1 到 D_{m-1} 每个分组长为 128 比特，分组 D_m 的长度可能不是 128 比特

G　在加密和解密过程中使用的函数

H　在加密和解密过程中使用的长度为 128 比特的分组

inc　输入与输出均为 128 比特分组的函数，$\mathrm{inc}(X) = (X\ |_{96}) \| \ \#_{32}(\#^{-1}(X\ |^{32}) + 1 \bmod 2^{32})$，其中，$X$ 是长度为 128 比特的分组

r　表示待加密消息按每个分组长度为 n 比特分块后，最后一个分组的长度，即消息长度为 $(m-1)n + r$ 比特

R　在 $\mathrm{GF}(2^{128})$ 有限域内乘法计算所使用的长度为 128 比特的分组

S　开始变量（n 比特）

T　标志（t 比特），与加密后的消息拼接，用于提供数据完整性保护

T'　在解密操作中计算得出的标志值

U, V, W, Z　定义 $\mathrm{GF}(2^{128})$ 有限域内乘法计算所使用的长度为 128 比特的分组

$X_0, X_1, \cdots, X_{k+l+1}$　在函数 G 的计算中使用的长度为 128 比特的分组

$Y_0, Y_1, \cdots, Y_m$　在加密和解密过程中使用的长度为 128 比特的分组序列

$\{\}$　长度为 0 的比特串

$\cdot$　在 $\mathrm{GF}(2^{128})$ 有限域内的乘法运算，决定 $\mathrm{GF}(2^{128})$ 的表示形式的多项式为 $1 + \alpha + \alpha^2 + \alpha^7 + \alpha^{128}$

10.3 具体要求

在使用此可鉴别的加密机制前，数据的始发者和接收者首先需协商确定标签长度 t，t 的取值应是 8 的整数倍，且满足 $96 \leqslant t \leqslant 128$（在特定情况下 t 的值也可以为 32 或 64）。

本机制应使用 128 位的分组密码算法，即 $n = 128$。

10.4 乘法运算的定义

假设 U 和 V 都是长度为 128 比特的分组，U 和 V 的乘法运算 $W = U \cdot V$ 定义如下，其中 W 也是长度为 128 比特的分组。

a) 令 $R = 11100001 \| \ 0^{120}$。

b) 令 $W = 0^{128}$。

c) 令 $Z = U$。

d) 对于 $i = 0, 1, \cdots, 127$，执行以下两个步骤：

1) 如果 $v_i = 1$，$W = W \oplus Z$；

2) 如果 $z_{127} = 0$，令 $Z = Z \gg 1$；否则 $Z = (Z \gg 1) \oplus R$。

上述描述中，v_i 表示 V 的第 i 位（$V = v_0 \| \ v_1 \| \cdots \| \ v_{127}$），$v_{127}$ 表示 V 的最右一位。

10.5 函数 G 的定义

函数 G 在加密和解密程序中使用，其输入包括一个长度为 128 比特的分组和两个任意长度的比特

串，输出是一个长度为128比特的分组。使用H代表一个长度为128比特的分组，W和Z分别代表两个任意长度的比特串（W和Z可能为空）。整数k和u满足$\text{len}(W)=128(k-1)+u$且$0<u\leqslant 128$；整数l和v满足$\text{len}(Z)=128(l-1)+v$且$0<v\leqslant 128$。将W分块后，得到分组序列$W_1,W_2,\cdots,W_k$，每个分组长度为128比特，其中最后一个分组W_k的长度为u（$0<u\leqslant 128$）；将Z分块后，得到分组序列$Z_1,Z_2,\cdots,Z_l$，每个分组长度为128比特，其中最后一个分组Z_l的长度为v（$0<v\leqslant 128$）。

$G(H,W,Z)$是128比特的值X_{k+l+1}，当$i=0,1,\cdots,k+l-1$时，X_i的定义如下：

a) $X_0=0^{128}$。

b) 对于$1\leqslant i\leqslant k-1$，$X_i=(X_{i-1}\oplus W_i)\cdot H$；如果$k\leqslant 1$略过此步。

c) $X_k=(X_{k-1}\oplus(W_k\parallel 0^{128-u}))\cdot H$；如果$k=0$略过此步。

d) 对于$k+1\leqslant i\leqslant k+l-1$，$X_i=(X_{i-1}\oplus Z_{i-k})\cdot H$；如果$l\leqslant 1$略过此步。

e) $X_{k+l}=(X_{k+l-1}\oplus(Z_l\parallel 0^{128-v}))\cdot H$；如果$l=0$略过此步。

f) $X_{k+l+1}=(X_{k+l}\oplus[\sharp_{64}(\text{len}(W))\parallel\sharp_{64}(\text{len}(Z))])\cdot H$。

10.6 加密程序

始发者应执行以下步骤来保护数据D，并确保额外的可鉴别数据A的完整性：

a) 选取一个任意长度的开始变量S。对每个被保护的数据应选取各不相同的S，并确保S可被消息接收者获知。无需保证S的取值是不可预知的或是秘密的。

b) 将D进行分块，得到一个数据分组的序列：$D_1,D_2,\cdots D_m$。其中，D_1包含D的第一个128比特，D_2包含D的第二个128比特，依此类推，直到D_m包含D的最后r比特（$0<r\leqslant 128$）。由此可得，$\text{len}(D)=(m-1)n+r$。

c) 令$H=e_K(0^{128})$。

d) 如果$\text{len}(S)=96$，$Y_0=S\parallel 0^{31}\parallel 1$；否则$Y_0=G(H,\{\},S)$。

e) 对于$i=1,2,\cdots,m-1$，执行以下两个上步骤：

   1) $Y_i=\text{inc}(Y_{i-1})$；

   2) $C_i=D_i\oplus e_K(Y_i)$。

f) $Y_m=\text{inc}(Y_{m-1})$。

g) $C_m=D_m\oplus(e_K(Y_m))|_r$。

h) $T=(G(H,A,C_1\parallel C_2\parallel\cdots\parallel C_m)\oplus e_K(Y_0))|_t$。

上述过程的输出，即数据D的可鉴别的加密数据输出为$C=C_1\parallel C_2\parallel\cdots\parallel C_m\parallel T$。

本机制产生的可鉴别的加密数据C比原始数据D增加了t比特，此外，数据始发者还需将变长的开始变量S和额外的可鉴别数据A发送给收者。

10.7 解密程序

接收者应执行以下步骤来解密和验证可鉴别的加密数据C，并验证额外的可鉴别数据A：

a) 如果C的长度小于t，则停止计算并输出INVALID。

b) 确定整数m和r，使得C的比特数等于$(m-1)n+r+t$，其中$0<r\leqslant n$。将C进行分块，得到一个数据分组的序列：$C_1,C_2,\cdots C_m,T$。其中，C_1包含C的第一个n比特，C_2包含C的第二个n比特，依此类推，直到C_m包含r比特（$0<r\leqslant n$）。最后，令T为C的最后t比特。

c) 令$H=e_K(0^{128})$。

d) 如果$\text{len}(S)=96$，$Y_0=S\parallel 0^{31}\parallel 1$；否则$Y_0=G(H,\{\},S)$。

e) $T'=(G(H,A,C_1\parallel C_2\parallel\cdots\parallel C_m)\oplus e_K(Y_0))|_t$。

f) 如果 $T \neq T'$，则停止计算并输出 INVALID。

g) 对于 $i=1,2,\cdots,m-1$，执行以下两个步骤：

1) $Y_i = \mathrm{inc}(Y_{i-1})$；

2) $D_i = C_i \oplus e_K(Y_i)$。

h) $Y_m = \mathrm{inc}(Y_{m-1})$。

i) $D_m = C_m \oplus (e_K(Y_m))|_r$。

j) 输出 D 与额外的可鉴别数据 A。

附 录 A
（规范性附录）
ASN.1 模块

A.1 形式定义

```
AuthenticatedEncryption {
    iso (1)  standard (0)  authenticated-encryption (19772)  asn1-module (0)
        authenticated-encryption-mechanisms (0) }
DEFINITIONS EXPLICIT TAGS ::= BEGIN
-- IMPORTS None; --
OID ::= OBJECT IDENTIFIER
AuthenticatedEncryptionMechanismALGORITHM ::= {
    ae-mechanism1|
    ae-mechanism2|
    ae-mechanism3|
    ae-mechanism4|
    ae-mechanism5|
    ae-mechanism6
}
-- Synonyms --
is19772 OID ::= { iso (1)  standard (0)  authenticated-encryption (19772) }
mechanism OID ::= { is19772 mechanisms (1) }
ae-mechanism1 OID ::= { mechanism 1 }
ae-mechanism2 OID ::= { mechanism 2 }
ae-mechanism3 OID ::= { mechanism 3 }
ae-mechanism4 OID ::= { mechanism 4 }
ae-mechanism5 OID ::= { mechanism 5 }
ae-mechanism6 OID ::= { mechanism 6 }
END -- AuthenticatedEncryption -
```

本标准中规定的五种可鉴别的加密机制，分别对应上述 ASN.1 定义中的 ae-mechanism2、ae-mechanism3、ae-mechanism4、ae-mechanism5、ae-mechanism6。

A.2 后续 OID 应用

本标准规定的 5 种可鉴别的加密机制都使用了分组密码算法，其中机制 4 还使用指定的分组密码运算模式以及 MAC 算法。使用可鉴别的加密机制算法 OID 时，会伴有所使用的分组密码算法及相关参数的 OID。对于机制 4，还会伴有分组密码算法工作模式与 MAC 算法的 OID。

附 录 B
（资料性附录）
可鉴别的加密机制的使用指导

B.1 简介

本附录为本标准中规定的可鉴别的加密机制的使用提供指导。任一机制的使用需要选择与该机制匹配的参数，B.3～ B.7 给出了相关参数选择的建议。本章以下部分对使用本标准所规定的所有机制应满足的需求提供了建议。

所有机制中使用的分组密码算法为 GB/T 32907—2016 指定的密码算法或由国家密码管理机构批准的密码算法。

分组密码算法中的分组长度 n 应至少为 64 比特，推荐使用分组长度为 128 比特（$n=128$ 的分组密码算法。在可鉴别的加密机制 1、2 和 5 中，应使用 $n=128$ 的分组密码算法。

所有机制要求数据始发者和接收者共享秘密密钥 K。该密钥只能由通信双方，或者是为了实现通信双方共享密钥的可信第三方知晓。密钥产生及相关的管理方法不属于本标准范畴，可参考相关标准执行。

所有五种可鉴别加密机制都需要选择标记（tag）的长度。该参数的选择会影响发送给接收者的受保护数据的完整性和来源真实性。

B.2 可鉴别的加密机制的选择

本标准中所有可鉴别的加密机制都具有高级别的安全性。然而，在特定的应用中，某些机制会比其他机制更合适。在进行可鉴别的加密机制选择时，应对表 B.1 中给出的因素以及表后列出的内容进行考虑。

表 B.1 各机制的特点

机制序号	1	2	3	4	5
加密长度为 q 比特的消息所需的分组加密运算次数	12 $[q/n]$	$2q/n$	$2q/n$	取决于选定的加密与 MAC 方式	q/n
是否需要使用授权	否	否	否	取决于选定的加密与 MAC 方式	否
是否针对短消息设计	是	否	否	否	否
加密前是否需知道消息的长度	否	是	否	否	否
是否需要开始变量	否	是	是	是	是

a） 可鉴别的加密机制 2 和 3 是将采用 CTR 模式的分组密码算法与消息鉴别码相结合；

b） 可鉴别的加密机制 4 提供了将标准化的密码算法和 MAC 算法结合的方法。如果这些功能已经可用，那么可鉴别的加密机制 4 在实施上具有一定的优势；

c） 可鉴别的加密机制 5 适于高吞吐量硬件实现，因为该机制的实现不会存在流水线停滞的问题。

B.3 可鉴别的加密机制 1

可鉴别的加密机制 1 要求使用分组长度为 128 比特($n=128$)的分组密码算法。

B.4 可鉴别的加密机制 2

可鉴别的加密机制 2 要求使用分组长度为 128 比特($n=128$)的分组密码算法。

该机制要求标签(tag)长度参数 t 的取值应从集合 {32,48,64,80,96,112,128} 中选取。参数 t 的取值取决于该机制的应用环境,建议使用 $t \geqslant 64$,除非有充分的理由做出不同的选择。

该机制要求消息长度域的长度参数 w 的取值应从集合 {2,3,4,5,6,7,8} 中选取。参数 w 的取值取决于该机制的应用环境,参数 w 值的大小不影响机制的安全性。参数 w 取值越大,可持的消息长度越大;相反的,会使用开始变量 S 的长度减小。即使 w 取最大值 8,开始变量 S 仍可以有 56 比特,56 比特的开始变量可以满足大部分应用对于每个消息使用不同开始变量的要求。对大多数应用而言,建议 w 取值为 4,即 $w=4$,即最大消息长度为 $2^{32} \approx 4 \times 10^9$ 字节。

B.5 可鉴别的加密机制 3

可鉴别的加密机制 3 需要确定标签(tag)长度参数 t($t \leqslant n$)。参数 t 的取值取决于该机制的应用环境,建议使用 $t \geqslant 64$,除非有充分的理由做出不同的选择。

B.6 可鉴别的加密机制 4

可鉴别的加密机制 4 需要选择分组密码算法的工作模式和 MAC 算法,该机制的安全性取决于上述两个因素的安全性。

无论使用何种分组密码的工作模式,强烈建议在可能的开始变量空间中均匀地随机选取开始变量 S。如果不依照此建议执行,将无法确保该机制的安全性,在某些环境下,可能遭受攻击。

选择 MAC 算法时,应考虑结合可鉴别的加密机制的应用环境,并遵循

GB/T 15852.1—2008 给出的意见。如果选用了基于分组密码的 MAC 算法,应注意以下两点:

a) MAC 算法 1 仅可在消息长度确定的情况下使用;

b) 填充方法 1 仅可在消息长度确定的情况下使用。

B.7 可鉴别的加密机制 5

可鉴别的加密机制 5 要求使用分组长度为 128 比特($n=128$)的分组密码算法。

该机制使用变长的开始变量 S,其长度应满足 $1 \leqslant \mathrm{len}(S) \leqslant 2^{64}$。保证开始变量在一个密钥的生存周期内不会被重复使用,这一点对该机制的安全性至关重要。

标签长度参数 t 的取值应是 8 的整数倍,且满足 $96 \leqslant t \leqslant 128$($t=32$ 和 $t=64$ 需谨慎使用,仅在特定应用中可以使用,具体使用说明请见附录 A)。

需要被可鉴别的加密机制进行保护的数据 D 应满足 $\mathrm{len}(D) \leqslant 2^{39}-256$,额外的可鉴别数据 A 应满足 $\mathrm{len}(A) \leqslant 2^{64}$。$D$ 和 A 所包含的数据分组的数量总和不能超过 2^{64}。对于任何给定的密钥 K,加密操作的总数最多为 2^{32},除非在该密钥每次被使用时,对应的开始变量 S 长度为 96 比特,即 $\mathrm{len}(S)=96$。

附 录 C
（资料性附录）
数 据 示 例

C.1 简介

本附录使用 SM4 分组密码算法实现标准中规定的可鉴别加密机制(不包括可鉴别的加密机制 4)，针对每一种机制给出具体的数据示例。本附录给出的数据示例均以 16 制数表示。

C.2 可鉴别的加密机制 1

以下针对相同密钥 K ，分别给出了 4 组明文消息 D_i 、密文 C_i 的数据示例。

K ：000102030405060708090A0B0C0D0E0F

D_1 ：0001020304050607 08090A0B0C0D0E0F

C_1 ：C8965070ACFBE416219080544FEE6453
3D1D7F61FE77B5BF

D_2 ：0001020304050607 08090A0B0C0D0E0F
1011121314151617

C_2 ：49F92F32A6FAC552 C8731D66E4E00B5D
F4EEC22383CA75CC8BBBF9D2F2BB8E4F

D_3 ：000102030405060708090A0B0C0D0E0F
101112131415161718191A1B1C1D1E1F

C_3 ：BE8827031E9585CF183D810683A6794D
38500F9420D6A4D9AC83BC0BD9FE95FA
F25F8067BF34F7B

D_4 ：000102030405060708090A0B0C0D0E0F
101112131415161718191A1B1C1D1E1F
2021222324252627

C_4 ：E2AFC9BA0E146CC684B70B6F1A281E3F
E869E77B4A9D1F0F084ACFD781FAE6C9
139916895539B4F5773D4BE371965BFE

C.3 可鉴别的加密机制 2

以下针对相同密钥 K 和开始变量 S ，分别给出了 6 组明文消息 D_i 、密文 C_i 和加密后标签 U_i 的数据示例，其中 $t=128$，$w=2$(因此，S 应为 104 比特)。

K ：000102030405060708090A0B0C0D0E0F

S：000102030405060708090A0B0C

D_1：空(即 L=0)
C_1：空
U_1：36D53BC3E931A547849F7D044ACE0515

D_2：0001020304050607
C_2：273204E39F4F4F9E
U_2：92D2BF3926B24C4AF2EB8A5945B22F3C

D_3：000102030405060708090A0B0C0D0E0F
C_3：273204E39F4F4F9E602809EC9AA0A411
U_3：143F95B9B1FACDD7FE38C8705FEF8F93

D_4：000102030405060708090A0B0C0D0E0F
1011121314151617
C_4：273204E39F4F4F9E602809EC9AA0A411
C97F81AFF1D6FE96
U_4：0087CD0ED720F051A18DC2FF1BB076DC

D_5：000102030405060708090A0B0C0D0E0F
101112131415161718191A1B1C1D1E1F
C_5：273204E39F4F4F9E602809EC9AA0A411
C97F81AFF1D6FE96BA1EE8304D4EE9F0
U_5：458B0B5A993D40AC57AA1EE01F46D337

D_6：000102030405060708090A0B0C0D0E0F
101112131415161718191A1B1C1D1E1F
2021222324252627
C_6：273204E39F4F4F9E602809EC9AA0A411
C97F81AFF1D6FE96BA1EE8304D4EE9F0
548DFEB8F12C39CC
U_6：CAB0AC757E5DD7A6882BA59AF3D53092

C.4 可鉴别的加密机制 3

以下针对相同密钥 K 和开始变量 S，分别给出了 6 组明文消息 D_i、密文 C_i 和标签 T_i 的数据示例，其中 $t=128$。

K：000102030405060708090A0B0C0D0E0F
S：000102030405060708090A0B0C0D0E0F

D_1：空(即 m=0)
C_1：空

T_1：72A775827AB4258C4332D84CE607D7A1

D_2：0001020304050607
C_2：8D4B0E9BCDF63E0A
T_2：9C864CB61B71647007D3313EABC55A87

D_3：000102030405060708090A0B0C0D0E0F
C_3：8D4B0E9BCDF63E0A1D7566451CE7B43A
T_3：A098F58D2984BB0C06437473D95A2BC2

D_4：000102030405060708090A0B0C0D0E0F
1011121314151617
C_4：8D4B0E9BCDF63E0A1D7566451CE7B43A
4A2FD41A1A01EE4F
T_4：4D86AA50FF95AD05FEB1131D40ED9FF1

D_5：000102030405060708090A0B0C0D0E0F
101112131415161718191A1B1C1D1E1F
C_5：8D4B0E9BCDF63E0A1D7566451CE7B43A
4A2FD41A1A01EE4F02A04D0F52CB5379
T_5：8FB7A88B255A5472A40B45B1FD2F2E2C

D_6：000102030405060708090A0B0C0D0E0F
101112131415161718191A1B1C1D1E1F
2021222324252627
C_6：8D4B0E9BCDF63E0A1D7566451CE7B43A
4A2FD41A1A01EE4F02A04D0F52CB5379
F1B4BC37ED9A5BBF
T_6：9395E68CB9AFDF298D748EEDE0E1C9B6

C.5 可鉴别的加密机制 5

以下针对相同密钥 K 和开始变量 S，给出了 2 组明文消息 D_i、密文 C_i 和标签 T_i 的数据示例，其中 $t=128$，且额外的可鉴别数据为空。

K：00000000000000000000000000000000
S：000000000000000000000000

D_1：空（即 m=0）
C_1：空
T_1：232F0CFE308B49EA6FC88229B5DC858D

D_2：00000000000000000000000000000000
C_2：7DE2AA7F1110188218063BE1BFEB6D89
T_2：B851B5F39493752BE508F1BB4482C557

参 考 文 献

[1] M.Bellare and C.Namprempre,'Authenticated encryption:Relations among notions and analysis of the generic composition paradigm'.In:T.Okamoto (ed.),Advances in Cryptology-ASIACRYPT 2000,6th International Conference on the Theory and Application of Cryptology and Information Security,Kyoto,Japan,December 3-7,2000,Proceedings.Lecture Notes in Computer Science 1976,SpringerVerlag (2000) pp.531-545

[2] M.Bellare,P.Rogaway and D.Wagner,'The EAX mode of operation'.In:B.K.Roy,W.Meier (eds.):Fast Software Encryption,11th International Workshop,FSE 2004,Delhi,India,February 5-7,2004,Revised Papers.Lecture Notes in Computer Science 3017,Springer-Verlag (2004) pp.389-407

[3] ISO/IEC 9797 (all parts),Information technology—Security techniques—Message Authentication Codes (MACs)

[4] ISO/IEC 10118 (all parts),Information technology—Security techniques—Hash-functions

[5] ISO/IEC 11770 (all parts),Information technology—Security techniques—Key management

[6] ISO/IEC 18033-1:2005,Information technology—Security techniques—Encryption algorithms—Part 1:General

[7] T.Krovetz and P.Rogaway,The OCB Authenticated-Encryption Algorithm,IETF draft draft-krovetzocb-00.txt,March 2005

[8] National Institute of Standards and Technology,NIST Special Publication 800-38D:Recommendation for Block Cipher Modes of Operation:Galois/Counter Mode (GCM) and GMAC.November 2007

[9] National Institute of Standards and Technology,AES Key Wrap Specification.NIST,November 2001

[10] National Institute of Standards and Technology,NIST Special Publication 800-38C:Recommendation for Block Cipher Modes of Operation:The CCM Mode For Authentication and Confidentiality.May 2004

[11] J.Schaad and R.Housley,RFC 3394:Advanced Encryption Standard (AES):Key Wrap Algorithm.IETF,September 2002

[12] D.Whiting,R.Housley and N.Ferguson,RFC 3610:Counter with CBC-MAC (CCM).IETF,September 2003

中 华 人 民 共 和 国
国 家 标 准
信息技术 安全技术 可鉴别的加密机制
GB/T 36624—2018
*
中 国 标 准 出 版 社 出 版 发 行
北京市朝阳区和平里西街甲 2 号(100029)
北京市西城区三里河北街 16 号(100045)
网址:www.spc.org.cn
服务热线:400-168-0010
2018 年 9 月第一版
*
书号:155066·1-61205

GB/T 36624-2018