目录

[1.概观 2](#_Toc360530851)

[1.1范围 2](#_Toc360530852)

[1.2目的 2](#_Toc360530853)

[1.3条款和附件的说明 2](#_Toc360530854)

[2.规范性引用文件 2](#_Toc360530855)

[3.关键字、定义、缩略字和缩写. 2](#_Toc360530856)

[3.1关键词 2](#_Toc360530857)

[3.2定义 2](#_Toc360530858)

[3.3缩略语 4](#_Toc360530859)

[3.4数学公约 5](#_Toc360530860)

[4.一般概念 5](#_Toc360530861)

[4.1简介 5](#_Toc360530862)

[4.2组件 6](#_Toc360530863)

[4.3 明文记录格式化器 8](#_Toc360530864)

[4.4明文记录逆格式化器 8](#_Toc360530865)

[4.5加密例程 8](#_Toc360530866)

[4.6解密例程 9](#_Toc360530867)

[5.加密模式 11](#_Toc360530868)

[5.1概述本 11](#_Toc360530869)

[5.2带密码分组链接消息认证码的计数器（CCM） 12](#_Toc360530870)

[5.3伽罗瓦/计数器模式（GCM） 13](#_Toc360530871)

[5.4加密哈希消息认证码的密码分组链接（CBC-HMAC） 13](#_Toc360530872)

[5.5 XOR加密XOR可调分组密码加密哈希消息验证码（XTS-HMAC） 14](#_Toc360530873)

[6.密钥管理和初始化向量要求 16](#_Toc360530874)

[6.1随机位发生器 16](#_Toc360530875)

[6.2密钥的入口和出口 17](#_Toc360530876)

[6.3处理密钥 17](#_Toc360530877)

[6.4存储介质上的加密密钥包装 17](#_Toc360530878)

[6.5初始化向量请求 18](#_Toc360530879)

[6.6创建独特的IV一个自足组内 20](#_Toc360530880)

[附件A 21](#_Toc360530881)

[附件B 23](#_Toc360530882)

[附录C 29](#_Toc360530883)

[附录D 29](#_Toc360530884)

## 1.概观

### 1.1范围

本标准指定对加密单元的要求，加密单元提供对包含数据的存储设备的认证和加密功能。全交互需要额外的格式规范（如压缩算法和物理数据格式），这超出这个标准的范围。

### 1.2目的

本标准适用于存储在磁带上的数据进行加密，因为磁带能轻松容纳长度扩大的密文。此外，该标准适用于其他存储设备，如果这些设备支持存储额外的带加密记录的元数据。本标准的算法设计，在一个要求较高的系统上保证数据的保密性和完整性。

### 1.3条款和附件的说明

条款1：概述了这个标准，包括的范围和目的。

条款2：列出了规范性引用文件，这对本标准实施是必不可少的。

条款3：给出了定义，首字母缩写词，在此标准中使用的缩写。

条款4：描述在本标准中发挥重要作用的组件

条款5：说明所使用的加密模式的加密单元。

条款6:介绍了密钥管理和初始化向量的要求。

附录A（资料性附录）列出参考书目，实施这一标准时非​​常有用。

附录B（参考）讨论了一些实施者和使用者应该理解的安全问题。

附件C（信息）提供一个总结文件要求。

附录D（资料性附录）提供了一些有用的测试向量验证的加密单元。

## 2.规范性引用文件

下列参考文档对于本文档的应用是必不可少的（他们必须能够理解和使用，因此每个参考文档在文中被引用，并且解释了和这个文档的关系）。凡是注日期的参考文档，仅引用的版本适用。凡是不注日期的引用文件，其最新版本参考文件（包括任何修订或更正）适用。

## 3.关键字、定义、缩略字和缩写.

### 3.1关键词

就这个标准的目标而言，下列词汇是关键字。

can:指示某种能力的关键词（等价is able to）.要求：见shall.

may:指示一个在这个标准限度内可以允许的行为过程（等价于 is permitted to）

must：仅仅用来描述一个不可避免的情况，并要求符合这个标准。

shall：一个关键字，指示的强制性的、必须遵循的要求，不允许有偏差(shall

等价于 is required to)shall not：指示本标准绝对禁止。should：表明几种可能性中被推荐的特别适合的一个，无须提及或者排除他人，或者说更倾向的一中行为过程，但不强制要求。或者（否定形式）一中不推荐的行为过程，但并不是禁止.

### 3.2定义

对于这个标准的目的，下列术语和定义适用。本节中未定义的术语或者定义可以参考权威词典IEEE标准（第七版）[B6] 5

3.2.1 额外的验证数据（AAD）：信息传递到一个身份验证的加密例程，它被认证，但不加密。

3.2.2 高级加密标准（AES）由NIST FIPS 197定义的分组密码。另请参阅：分组密码。

3.2.3分组密码：加密基元使用加密密钥创建一个固定大小的位串的一个伪随机置换。另请参阅：密钥明文，密文。

3.2.4密码分组链接（CBC）：在加密操作模式中输出密文，从每个密码分组送入以下的密码分组（见NIST SP 800-38A）。

3.2.5密码分组链接初始化向量（CBC-IV）：CBC模式的初始化向量输入，根据NIST SP 800-38A。另请参阅：CBC-HMAC;初始化向量.NOTE-see 5.4

3.2.6带密钥散列消息认证码的密码分组链接（CBC-HMAC）：一组寻求保密性而使用CBC模式的加密模式和寻求完整性使用密钥哈希消息认证码（HMAC）（见NIST FIPS 198）。

3.2.7密钥散列消息认证码的密码分组链接，而且带密钥散列消息认证码使用SHA-1/2算法.（CBC-HMAC-SHA）：拥有密码哈希消息认证码的一组密码分组链接（CBC）模式（见NIST SP 800 - 38A）（见NIST FIPS 198）认证码来自SHA-1/2算法（见NIST FIPS 180-2）。注：见5.4。

#### 3.2.8加密密钥

一个控制加解密伪随机置换的比特串。另请参阅：block cipher。

3.2.9密文记录：加密等长明文记录的结果，明文记录使用加密操作模式。另请参阅：ciphertex;cryptographic mode of operation;plaintext record..

3.2.10 碰撞：两个独立的变量在某一特定情况具有相同的值的一种事件。见：初始化向量明文，密文。

3.2.11 计数器模式（CTR）：由NIST SP 800-38A定义的加密操作模式，密文是带加密计数器的明文的异或

3.2.12 带加密分组链接消息认证码的计数器（CCM）：一种提供保密性和完整性的加密操作模式，由计数器模式保证保密性，由使用加密分组链接的消息认证码保证完整性（见NIST SP 800-38C）。注：见5.2。

3.2.13 声音加密机位发生器（RBG）：设备或算法输出二进制位的序列，看起来是统计独立和客观的。特别是，一个RBG产生不可预测的数字，任何一个从RBG产生的输出不能揭示别的发生器产生的数据。注：见6.1。

3.2.14：密码杂凑函数。

从一个输入总产生一个哈希值的哈希函数，并且具备如下性质：1）在计算上求逆是困难的（从哈希值计算输入）2）在计算上是很难找到两个不同的输入，具有相同的的散列值; 3）在计算上是很难找到一个输入的散列值是一个特定的值3.2.15加密密钥：被用作加密元语的比特.另请参阅：阻止密码加密密钥。

3.2.16加密模式：see cryptogrphic mode of operation。

3.2.17加密操作模式：包括一个分组密码的算法，该算法在特定配置下使用，使用的加密密钥讲明文转换成密文,反之亦然。另请参阅：block cipher;cipher key,plaintext;ciphertext.

3.2.18加密单元：任何一套可以执行加密操作的软件，固件或硬件。

3.2.19解密：从密文产生明文的行为。

3.2.20解密程序：解密操作模式的实例，将密文转换成明文。

3.2.21加密记录：包括一个加密操作或验证操作输出的字段集合（例如，密文，消息认证码），可以包含其他为随后的解密操作所需的信息（例如，额外的认证数据，初始化矢量）。

3.2.22加密：从明文生成密文的行为。

3.2.23加密例程：加密操作模式的实例，将明文转换成密文。

#### 3.2.24加密会话

加密单元在一个间隔生成一套使用首尾一致的变量的加密记录，，如唯一的初始化向量。

3.2.25伽罗瓦/计数器模式（GCM）：通过计数器模式加密提供保密性、通过使用使用伽罗华域算术的消息认证码提供完整性的加密操作模式。 （见麦克格鲁VIEGA的，伽罗瓦/计数器工作模式）。注：见5.3。

3.2.26主机记录：从主机传递给加密单元的明文串。另请参阅：明文记录;加密单元主机。

3.2.27初始化向量（IV）：加密/解密算法的一个输入，这个输入无须保密，但当使用一个特定的密钥时它极有可能是独一无二的。另请参阅：加密，解密加密会话密钥。注：见6.5。

3.2.28密钥加密密钥（KEK）：仅用于加密或者解密其他密钥的密钥。另请参阅：加密密钥加密。

3.2.29加密哈希消息认证码（HMAC）：由NIST FIPS 198定义的包括一个秘密哈希键的消息认证码

3.2.30密钥管理：控制创建、归档、销毁密钥的任何设备或者人。另请参阅：加密密钥。注见4.2.3。

3.2.31消息认证码（MAC）：用于检测在一个加密记录中可以的修改和错误的加密校验和在不知道用于MAC算法的密钥的情况下不能够被伪造。另请参阅：加密记录。

3.2.32随机数：一个字符串，它以一个很低的概率匹配另外一个特定情况下的随机数。另请参阅：初始化向量。

3.2.33明文：没有通过加密转换被掩盖 的信息。

3.2.34明文记录：传递给一个加密程序来产生一个等长密文记录的明文串。参见 plaintext; cryptographic unit; encryption; encrypted record; host record

3.2.35策略：在密码学中,定义密码系统管理各个方面的一组规则（例如，加密，解密，或旁路规则）。

3.2.36随机位发生器（RBG）：See cryptographically-sound random-bit generator.。

3.2.37多重随机数：完全由一个随机位发生器的输出的随机数。见：随机数加密声的随机位发生器。

3.2.38安全散列算法（SHA）：由NIST FIPS定义的一组加密哈希函数180-2。另请参阅加密哈希函数。

3.2.39自足组：以一致的方式产生和使用的IV的一组加密单元。

3.2.40 Xor-encrypt-xor with tweak and ciphertext stealing (XTS)：加密的操作模式,在IEEE标准1619中描述。

### 3.3缩略语

AAD 额外的验证数据AES 高级加密标准CBC 密码分组链接CCM 带密码分组链接消息认证码的计数器CTR 计数器模式FIPS 联邦信息处理标准GCM 伽罗瓦/计数器模式HMAC 加密哈希消息认证码IEEE 电气和电子工程师学会IV 初始化向量KEK 密钥加密键MAC 消息认证码NIST 国家标准机构和技术RBG （加密音）随机位发生器SHA 安全散列算法SP（NIST）特别出版物XOR 异或XTS 异或加密-异或调整和密文盗取

### 3.4数学公约

本标准采用十进制，二进制，十六进制数字。为清楚起见，一般的十进制数代表计数，二进制或十六进制的数字描述位模式或原始二进制数据。二进制数由一个或者更多二进制数字来表示，其下标为2。例如，十进制数26以二进制表示为000110102。

## 4.一般概念

### 4.1简介

这个标准描述的体系结构的元素，这是适合的加密的保密性和存储的数据的完整性。在一个典型的系统之内，此体系结构包括几个部分组成的模型，系统能够安全地存储和检索信息。组件如下：

控制器：控制器控制加密单元的整体操作，并接收加密单元的状态（见4.2.1）

主机：主机以主机目录的方式提供给加密单元明文数据，从加密单元接收明文数据（见4.2.2节）

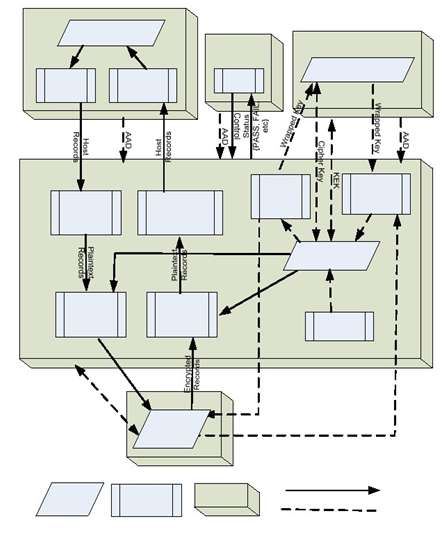
密钥管理器：密钥管理器可能会提供或协商密钥和(或)密钥加密密钥（KEK）给加密单元，并应安全的地保持这些加密密钥的生命周期（请参阅4.2.3）

加密单元：其执行的数据格式，加密和解密，这可能执行密钥管理（见4.2.4）

存储介质：它提供的加密所产生的记录和元数据的非易失性存储加密单元（见4.2.5）

本标准规定加密单元的要求。本标准的实施者应该提供给最终用户关于加密单元的文档，这个文档也可以以任何被最终用户容易得到的形式（如电子，印在纸上）。文档应包括所有该标准指定的所需文本。该文档提供了足够的信息来允许充分利用和详细的安全评估加密单元和它的环境。

一个文档摘要见附录C。图1显示了一个例子，上面列出的组件之间的交互和每个组件包含的子组件。图1中所示的多个组件可能存在在一个单一的实施例，并且相同的组件或子组件的多个实例可以存在在一个单一的系统。



### 4.2组件

#### 4.2.1控制器，

控制加密单元的整体操作的任何实体。控制器发送命令给加密单元，处理加密单元的状态。这是作为控制器定义的策略实现所需要的。可能有多个控制器控制一个特殊的控制单元。一个控制器可能是另外组件的一个组成部分，比如一个主机或者密钥管理者.

#### 4.2.2主机

主机提供主机记录给加密单元来加密，解密之后从加密单元接收主机记录主机记录包含明文数据，可以是加密单元允许的任何大小。

一个典型的主机包括例程将任意主机明文数据转换成主机的记录，反之亦然。这样的主机记录可能是可变长度的，视加密单元的能力而定。在图1这些程序如下：

主机记录格式化：一个例程为了加密单元将明文数据转换成主机记录

主机记录逆格式化：一个例程将主机记录匆匆加密单元转换成明文.

它并不需要一台主机来实现这些功能。主机只需要传递主机记录给加密单元，并接受从一个加密单元的主机记录。

示例：

如果该加密单元包含在一个磁带驱动器，则主机可能运行备份应用程序，备份应用程序获取任意的主机明文数据文件，并将它们整合到备份集，打破这些备份集到可变长度的分组，将分组作为主机记录发送给加密单元.如果加密单元被包含在一个磁盘驱动器，则主机可能是一个操作系统，那么可能会对格式化文件到固定大小的扇区（通常为512字节），在从加密单元发送、接收数据时将扇区作为主机记录.

#### 4.2.3密钥管理器

密钥管理负责被加密单元使用的密钥的生命周期（例如产生、归档和销毁）这样一个加密密钥可以是一个加密密钥或密钥加密密钥（KEK）（见6.3）。密钥管理器可以在密钥存档保持加密密钥。密钥管理和密钥存档的要求是本标准范围之外的，但对整个系统的安全性至关重要（见IEEE P1619.3 [B8]）。

#### 4.2.4加密单元

一个加密单元是软件，固件或硬件的任何组合，能够处理使用5.1指定的至少一个机密模式来处理明文和密文.

加密单元应包含以下子组件：

--明文记录格式化器（见4.3）和（或)明文记录格式化（见4.4）

--加密程序（见4.5）和(或)或解密例程（见4.6）--加密参数（见4.7）

加密单元可能包含以下子组件:

---随机位发生器

---密钥打包程序（见6.4）---密钥解包程序（见6.4）

#### 4.2.5存储介质

存储介质是非易失性存储加密记录和元数据的任何设备或者材料，

存储控制器可以配置加密单元来写特定的密文记录到存储媒介或以加密方式，或以非加密方式。加密单元在存储媒介上可以混合加密记录和明文记录。加密单元可能写额外的非加密数据到存储介质，假设这种信息没有揭露密钥或者试图加密的文明。如果信息对存储介质上的信息的保密性或者完整性构成威胁，加密单元不能写此类信息到

存储介质。

### 4.3 明文记录格式化器

明文记录格式化器是一种将主机记录转换成明文记录的程序，明文记录被传送给加密例程。在最简单的情况下，这个过程可以简单地通过主机记录，就像明文记录。在更复杂的系统，这个程序可以进行压缩，填充，或其他可逆转换。

加密单元从主机接收主机记录，它作为加密的一个基本单元.何时进行加密，加密单元应使用明文记录格式化器将主机记录格式化成明文记录

为了尽量减少缓冲的要求和等待时间，加密单元对明文记录可以定义一个最大值，此值小于加密单元所允许的最大主机记录大小。该明文记录格式化器可能分解主机记录成多个明文记录，以可选的填充或重新格式化方式。

加密单元可能应用填充或进行可逆转换（如压缩）到主机内的记录的数据以形成明文记录。

如果由两个或两个以上的明文记录形成主机记录，加密单元包括足够的信息让明文记录格式化器明确重构每个原始主机记录或者检测恶意篡改。为满足这个要求，加密单元应该使用命令验证或者检测篡改或重新排序的加密记录（见4.6.3）。

文档应说明明文记录格式化器如何从主机记录产生明文记录。

### 4.4明文记录逆格式化器

明文记录逆格式化器是一个程序：将从解密程序收到的解密的明文记录转换成主机记录，加密单元将主机记录传给主机.明文记录逆格式化器应只使用解密程序通过MAC验证的信息。

如果明文记录包含填充或可逆变换，明文记录逆格式化器应该验证这些格式的正确性。如果格式不正确，那么加密单元应当发送特殊信号FAIL给主机和（或）控制器，并且不返回任何主机记录。

文档应说明明文记录逆格式化器如何从明文记录生成主机记录。

### 4.5加密例程

#### 4.5.1概述

加密程序需要格式化明文记录作为输入和产生加密记录作为输出。下面的小节描述所有加密模式中常见的加密程序的特征。

#### 4.5.2输入

加密程序需要以下输入（见5.1限制）：

A）一个秘密密钥

B）一个初始化向量（IV）

C）的IV长度

D）明文记录

E）明文记录的长度

F）额外的验证数据（AAD）

G）ADD的长度

#### 4.5.3输出

加密程序产生一个加密的记录，其中包含以下内容：

a）在密文记录;

b）一个消息认证码（MAC）;

c）任选的IV或重构IV的足够信息;

d）任选的AAD的或重建AAD的足够信息。

密文记录应和明文记录具有相同的长度。一个加密记录可能另外含有IV和AAD。如果加密记录不包含IV和AAD，那么应该有足够的信息在存储介质或者加密单元，以允许重建完整的IV和AAD。 IV和AAD可能包含任何其他信息，这些信息不能威胁加密记录（例如，的保密性和完整性。

加密单元应该将加密记录写入到存储介质。当进行加密时，加密单元不应写明文或加密密钥到未加密的存储介质。加密单元可以写一个加密过的密钥到存储介质（见6.4）。

对于加密，加密密钥应该与一个单一的加密模式相关联，加密单位不应在任何其他加密模式使用家秘密要。密钥管理器应该将每个密码密钥用与一个单一的加密模式相关联。

### 4.6解密例程

#### 4.6.1概述

解密例程使用密钥将存储介质中的加密记录转换成成明文记录，为了明文记录逆格式化器。

下面的小节描述解密的要求，这在标准指定的加密模式中是司空见惯的。

#### 4.6.2解密输入

解密程序需要以下输入：

A）一个秘密密钥

B）的初始化向量（IV）

C）IV的长度

D）密文记录

E）密文记录的长度

F）额外的验证数据（AAD）

G）ADD的长度

h）加密模式判定长度的MAC在解密过程中，加密单元应始终验证的MAC。加密单元应该在发送主机明文之前验证MAC。最推荐的做法是在返回明文之前验证MAC（见B.4）。文档应说明加密单元是否在返回明文之前验证MAC。

如果加密单元返回明文之前验证MAC，如果主机MAC验证失败，那么它不该返回明文。如果MAC验证失败，那么加密单位应当返回特殊信号FAIL给主机和（或）控制器。

如果加密单元验证MAC之前返回给主机明文，那么加密单位应当随后验证的MAC。如果此MAC验证失败，那么加密单元应当返回FAIL信号给主机或控制器。如果该MAC验证通过，那么加密单元应当返回PASS信号给主机或控制器。

如果加密单元能够验证MAC之前返回明文，主机不应操作来自加密单元的任何明文，直到接收到一个完整的主机记录和信号PASS。

文档应定义信号FAIL，并描述了主机和(或)控制器如何接收这个信号。信号FAIL应该识别MAC验证失败的主机记录。

如果加密单元在验证MAC之前能返回明文，那么文档应定义信号FAIL，描述主机和/或控制器如何收到这样一个信号。定义主机记录的最大值和明文的最大比特值，这可能在校对MAC钱返回。

#### 4.6.3命令验证

在解密过程中，加密单元应该进行命令验证，这通过检查每个IV或AAD与先前的IV或AAD前后一致性来实现，他们是基于创建IV或ADD文档机制.（见B.3）。

如果加密单元执行命令验证并检测到不一致的IV或AAD，那么加密单元应返回FAIL给主机和(或)控制器。

如果加密单元支持命令验证，那么文档应指定启用或禁用此功能的方法，应说明机密单元如何通知主机和(或)控制器前后不一致的IV 或ADD，并且说明可能的恢复方法。

#### 4.6.4只验证模式

加密单元可能支持仅验证的模式，它仅验证MAC和给主机和/或控制器返回PASS或者FAIL信号，但不返回任何主机记录。

#### 4.7加密参数

一个加密的参数是一个值，它影响加密信息的机密性或完整性。

当在加密单元存储参数时，加密单位应当保护加密参数免受未经授权的修改，但可能允许信息披露：

---额外的身份验证数据（AAD）

---初始化向量（IV）

---任何不对称公钥[例如（KEK）]

此外，加密的单位应当保护以下加密参数免遭未经授权的修改和信息披露：

---密码钥匙

---随机位发生器的种子密钥

---任何非对称私钥（例如，不对称的私有KEK）

---任何对称KEK

如果这种披露使用加密模式或者使用物理安全连接，加密单元可能会披露加密参数给授权的实体。文档应描述加密单元所使用的所有加密参数。

## 5.加密模式

### 5.1概述本

加密单元在兼容模式下运行时，这节描述标准允许的加密模式。加密单元至少支持一个下表所述的加密模式。

Table 1 —Cryptographic modes

**Family Fully qualified name Description Ref.**

CCM CCM-128-AES-256 Counter with 128-bit cipher block chainingMAC 5.2

GCM GCM-128-AES-256 Galois/Counter Mode with 128-bit MAC 5.3

CBC-HMAC CBC-AES-256-HMAC-SHA-1 Cipher block chaining with 160-bit HMAC 5.4

CBC-AES-256-HMAC-SHA-256 Cipher block chaining with 256-bit HMAC 5.4

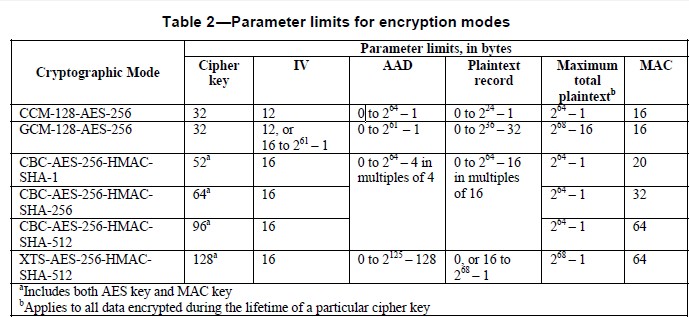
CBC-AES-256-HMAC-SHA-512 Cipher block chaining with 512-bit HMAC 5.4

XTS-HMAC XTS-AES-256-HMAC-SHA-512 Xor-encrypt-xor with tweak and ciphertext 5.5

stealing, with 512-bit HMAC

当独立描述加密和解密例程时，这个标准将使用后缀“ENC”表示加密程序，“DEC”表示解密例程。例如，“CCM-128，AES-256-ENC”是指实施CCM-128，AES-256加密模式的加密例

加密单元应该表格2给出的参数限制之内操作机密模式



所有的长度应是一个整数的字节数。一个加密单元可能施加比表2中更严格的参数。如果不同于表2中，文档应当规定加密单元的参数限制.

### 5.2带密码分组链接消息认证码的计数器（CCM）

一个支持CCM-128，AES-256加密模式的加密单元应该使用由NIST特别出版物800-38C指定的算法（NIST SP 800-38C），说明如下：

A）分组加密算法应该是256位（32字节）的密钥的AES（见NIST FIPS 197）。

B）计数器的生成功能由NIST SP 800-38C的附录A中规定的。

C）格式化功能由附录A中的NIST SP 800-38C指定。

D）在MAC长度（TLEN）应为128位（16字节）。

E）IV的长度应为96位（12字节）。加密程序的输入IV对应CCM要求的随机数N （NIST SP 800-38C）。

F）加密单元可能验证MAC之前返回文明给主机,正如4.6.2所描述的。

G）IV计算应按照6.5的要求。

注意-尽管名称是相似的，用于CBC-MAC计算的IV不对应CBC-HMAC中所用的CBC-IV（见5.4）。CCM的CBC-MAC部分采用了全0的CBC-IV,以此和CBC-HMAC相比较，后者每次调用采用了独特的CBC-IV。

数据长度应使用24位（3字节），参数化设置t=16,q=3。表3示出B0分组的格式。

Z]NZH)00O{A8F8MMBNCRHLN

在表3中，如果的AAD长度是零，变量y应等于二进制'0'; 如果AAD长度为非零,它是一个二进制'1'，。例如，如果没有AAD,B0的第一个字节的二进制值001110102和如果有AAD则是011110102。计数器分组标志字段应包含二进制值000000102（NIST内见A.3SP 800-38C）。所有其他参数应符合附录A中的NIST SP 800-38C。注意--如果MAC验证失败，NIST SP 800-38C不允许返回任何明文，。4.6.2描述的一中情况是标准所允许的。

5.3伽罗瓦/计数器模式（GCM）支持GCM-128，AES-256加密模式的加密单元应该使用McGrew and Viega’s *The Galois/Counter Mode of Operation指定的算法，参数如下:*A）分组加密算法应该是带246位密钥的AES（见NIST FIPS197）。

B）MAC长度应为128位（16字节）。MAC应该被用作GCM算法中定义的标签。

C）IV计算应遵循从6.5的要求。

D）正如4.6.2所述，加密单元在验证MAC之前可能返回给主机明文

E）IV的长度或者12字节或者16字节到2^61-1之间。注1：文档The Galois/Counter Mode of Operation不允许任何明文返回，如果MAC验证失败。4.6.2描述的情况是例外。注2：超过128位（16字节）长度的IV不会增强安全性，因为长的IV在使用前被提取到16字节.

5.4加密哈希消息认证码的密码分组链接（CBC-HMAC）支持CBC-HMAC的加密单元应该使用CBC和HMAC，前者由NIST在800-38A指定后者由NISF FIPS 198指定，说明：

A）分组加密算法是带 256位（32字节）AES密钥的AES（见NIST FIPS 197）。

B）HMAC应使用下列之一的散列函数（见NIST FIPS 180-2）：

1）SHA-1;

2）SHA-256;

3）SHA-512。

C）在MAC长度（即TLEN）应匹配基础散列函数的输出长度[例如，160位（20字节）SHA-1，SHA-256，256位（32字节）或512位（64字节）为SHA-512]。

D）在MAC密钥长度应等于MAC长度（即TLEN）。

E）明文记录长度为16个字节的倍数（见4.3为填充的讨论）。

F）AAD长度应是一个4字节的倍数。

G）加密单元应计算IV，CBC-HMAC中称为CBC-Ivs，根据下列方法之一：

1）设置CBC-IV为随机IV（见6.5.2）;

2）通过使用AES密钥，设置的CBC-IV的结果为加密的一个带有AES分组密码的随机数IV的结果，（见6.5.3），（NIST SP 800-38A，附录C）。

H）的CBC-IV长度应为128位（16字节）。

I）加密单元验证MAC之前可能返回给主机明文（见4.6.2）。

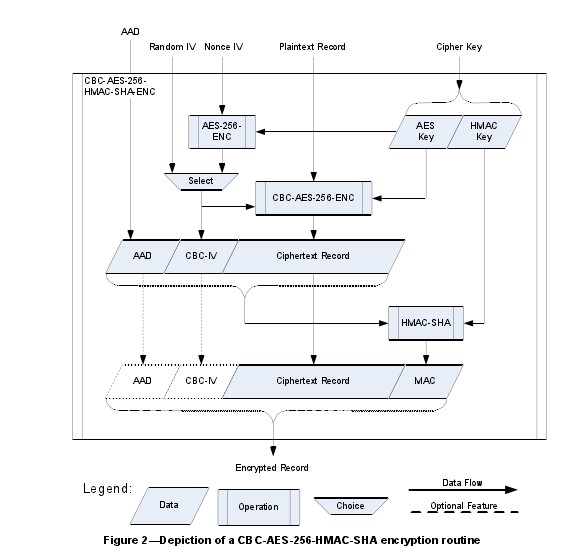
注1：即使明文记录必须是16个字节的整数倍，主机记录可能是任何规模大小，如果加密单元的明文记录格式化器提供填充，以此产生一个16个字节的倍数的明文记录。

对CBC-HMAC来说，加密单元应该通过串接AAD、CBC-IV、密文记录来计算HMAC，如果AAD是可变长度，那么AAD内必有足够的信息使解密例程明确确定AAD结束位置和CBC-IV开始位置

如果加密单元支持CBC-HMAC，那么文档应说明AAD的格式，以及确定AAD的结束和CBC-IV的开始所使用的方法。在解密过程中，加密单元应使用此方法来确定AAD的结束，并发送FAIL信号给主机和(或)控制器,如果AAD不坚持文档格式。

注2：通过在AAD开始的一个定长段中包括一个ADD长度实现先前的要求是可能的。

对于CBC-HMAC，当使用CBC-AES-256-HMAC-SHA-1时密钥长度应为416位，使用CBC-AES-256-HMAC-SHA-256时512位，使用CBC-AES-256-HMAC-SHA-512时768位。在加密和解密例程中加密单元应该使用密钥开始的256比特作为AES密钥。在MAC产生和验证例程中，应该使用剩下的比特做为HMAC密钥。图2显示了CBC-AES-256-HMAC-SHA加密例程。



在图2中，“AAD”和“CBC-IV”框周围的虚线表示，它是可选的目的地加密的记录，如果有足够的信息，这些字段内其他地方重建AAD和CBC-IV的解密例程。

注意，虽然图2中的“选择”框中，显示了两个可能的输入，支持一个特定的实现，该实现CBC-HMAC只需要支持这些选项之一。

### 5.5 XOR加密XOR可调分组密码加密哈希消息验证码（XTS-HMAC）

密码单元，实现了加密模式XTS-HMAC家庭内应使用在IEEE Std1619 XTS-AES-256程序指定的保密性，和HMAC-SHA-512由NIST FIPS198和NIST的FIPS180-2指定生成MAC，具有以下规格：

a）密钥长度为1024位（128字节）组成的串联  
以下几部分组成，依次为：

1）的AES密钥的长度为512位（64字节）到XTS-AES-256作为输入，用于  
程序（见IEEE STD1619）

2）HMAC密钥长度为512位（64字节），作为输入到HMAC-SHA-512  
程序

b）本加密单位应当按6.5计算的IV。 IV是使用指定的调整在IEEE Std1619。

C）的IV长度应为128位（16字节）。

d）在生成的MAC码的长度应为512位（64字节）。

对于XTS-HMAC，加密单元在串联“反倾销协定”，应计算的HMAC调整，密文记录。如果“反倾销协定”具有可变长度，然后将有足够的信息解密例程明确确定反倾销协定的结束和“反倾销协定”允许内调整开始。

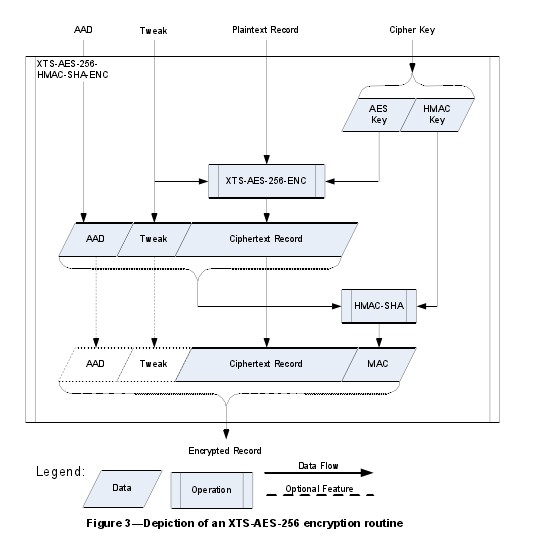
如果加密单元支持XTS-HMAC，那么文件应说明“反倾销协定”的格式，所使用的方法，以确定该的AAD端和调整开始。在解密过程中，加密单元应使用此方法来确定“反倾销协定”结束，并发送特殊信号失败的主机和/或控制器“反倾销协定”，如果不坚持记录格式。

NOTE-1：这是可能的，完成先前的要求包括一个固定长度的长度内的AAD,AAD的开头的字段。

NOTE-2：虽然明文记录是必需的，至少为16字节长，主机记录可能更小，如果明文的加密单元提供格式化填充。

图3显示了XTS-AES-256加密例程

在图3中，“AAD”和“微调”框周围的虚线表示，它是可选的目的地加密的记录，如果有足够的信息，这些字段内其他地方重建AAD和调整解密例程。



## 6.密钥管理和初始化向量要求

### 6.1随机位发生器

如果加密单元，则需要随机数据加密单元应使用加密声随机位发生器（RBG），以产生当前的随机数据。特别是，RBG的设计应它不是知识的基础上计算上可行地预测后续的输出的RBG以前输出的RBG或传递到未加密的信息加密单元。

一个加密单元应该实现一个RBG与NIST FIPS140-2兼容[B15]比较的标准。为实施指导描述适合RBGS的，请参阅ANSI X9.31：1998[B1]ISO/ IEC18031[B11]，凯勒[B13]，NIST SP800-90[B19]。

文档应描述，包括算法和加密单元采用RBG描述信息的来源的随机性。

### 6.2密钥的入口和出口

一个加密单元接收加密密钥的密钥管理使用安全的方法（例如，物理安全接口，加密保护的通信）。

一个加密单元不得作出明文的加密密钥，除了授权的实体使用物理安全端口。一个加密单元进行加密（即，包裹）加密键提供外部。

密钥管理应避免输入相同的密码钥匙插入兼容的加密单元不符合标准的加密单元的加密的目的。在这种情况下，这是有可能的不符合规定的危及安全的数据加密单元（例如，不符合规定的加密单元，可以使用相同的IV的序列作为兼容的加密单元）。

如果加密单元支持加密密钥的入口或出口，然后文件须注明支持的加密密钥的入口和出口的方法。

### 6.3处理密钥

加密单元使用下列方法中的一个或多个来创建加密密钥：

a）生成一个新的加密密钥使用只有一个RBG的输出，并执行以下操作：

1）创建一个包裹密钥使用KEK密钥管理;

2）存档包裹密钥使用一个或多个以下操作：

i）存储在存储介质上的包裹密码密钥和/或

ii）出口包裹密钥的密钥管理。

b）使用的密码密钥的密钥管理器，包括随机信息内的IV，如

6.5.2或6.5.3.3规定。

c）使用加密密钥的密钥管理和使用独特的IV作为一个自足组内

6.6中规定的。

单元可以使用的加密密钥包装，如6.4中指定的，结合的任何项目在前面的列表。

NOTE-控制器需要使用加密密钥加密单元配置时要特别小心密钥管理。在这种情况下，重要的是频繁地生成新的加密密钥的密钥管理器，因为信息泄漏的风险增加，与在相同的密码密钥加密的明文的量的平方（见B.7和B.8）。

### 6.4存储介质上的加密密钥包装

密钥包装是使用一个密钥加密密钥（KEK）的过程中，另一个加密密钥加密使用密钥包装程序（例如，加密密钥）。密钥打开解密的过程中，利用KEK以前包装的加密密钥。如果相同的KEK用于包装，解包，那么它是对称KEK。如果用于包装和展开，那么不对称的公众不同的KEKKEK进行包装和非对称的私人KEK执行解包。

密钥包装例程的例子如下：  
NIST的AES密钥对称KEK裹[B14]  
RSAES-OAEP（见RSA PKCS＃1 v2.1的不对称市民[B20]）与RSA公钥  
KEK进行加密和RSA私钥解密不对称的私人KEK  
ECIES（见IEEE STD1363A-2004[B7]）与椭圆曲线公钥不对称的椭圆曲线作为非对称私人的KEK用于解密的私人密钥进行加密和KEK来

支持密钥包装是可选的。可以使用任何加密单元密钥包装例行在进口或出口，或用于存档内的存储介质或键保护密钥经理。加密单元应该只使用密钥包装程序，在加密的区经过同行评审，例如上面所列出的。

当解开一个包裹被包裹的密钥与非对称的公共KEK，关键经理不应该通过非对称的私人KEK加密单元。相反，密钥管理应检索的包裹的密钥，使用其不对称的私人KEK解开它，然后通过加密密钥的加密单元，使用安全的方法。

如果加密单元支持密钥包装，然后文档应描述所有关键的包装例程加密单元支持。

注意KEK的实力可能会影响整体解决方案的实力。关于安全的讨论见B.2KEK的关注。

### 6.5初始化向量请求

#### 6.5.1概述

加密每个明文记录需要的密码密钥和IV，并使用相同的加密算法的组合加密密钥和IV不止一个明文记录引入安全漏洞（见B.6）。至使用一个以上的加密的加密密钥和IV相同的组合的可能性降至最低明文记录，加密单元生成的IV按照下列方法之一：

1. Random IV对于每一个加密记录，加密单元生成一个新的四，由  
   完全的RBG的输出（见6.5.2）。
2. Nonce IV：使用加密会话，根据6.5.3。

#### 6.5.2使用随机的IV

一个加密单元可能会生成一个随机的IV为每个加密记录的投入。这种随机的四应完全由一个RBG的输出。

#### 6.5.3加密会话

##### 6.5.3.1概述

一个加密会话是一个时间间隔，在其中一个或多个加密单元保持一致的序列的IV加密明文记录。

一个加密单元可以同时维护多个独立的加密会话中，每个独立加密会话使用不同的加密密钥和独立的IV。

##### 6.5.3.2年初加密会话

在加密会话开始之前，控制器应配置的加密单元使用创建或检索到的加密密钥，按6.3的特殊方法。

加密会话之后才开始的下列事件之一：  
a）加密单元将接收的加密密钥的密钥管理器。  
b）在加密单元生成一个新的加密密钥。

##### 6.5.3.3加密部分4的要求

加密单元应加密每个明文记录的加密内唯一的四会话。这项要求可防止明文泄漏加密会话内（见B.6）。

开始加密会话时，加密的单位应当建立四初始值。如果加密单元使用一个密钥的密钥管理器（见6.3），那么初始值四包含至少64位，是来自于一个RBG。这个初始值可能继续从最后四先前的加密会话，如果最后的IV是一致的序列的一部分，从最初初始值，其中包含至少64个随机位。

文件应说明的格式的IV和加密单元的机制来生成每个IV。

##### 6.5.3.4加密会话结束

加密会话终止后，以下事件：

a）加密单元从控制器接收一个命令结束的加密会话。支持这样的命令是可选的。

b）在加密单元失去了加密的会话状态，包括密钥和IV。在这种情况下，加密单位应通知主机。

c）在加密单元是无法创建的IV内的加密会话中是唯一的。在

这种情况下，应该发送的加密单元主机和/或控制器的特殊信号失败。

如果加密单元加密数据加密会话加密会话结束，然后加密的单位不得任何更多的数据进行加密，加密会话，直到另一个开始。

如果加密单元支持加密会话结束的命令，然后文件应描述这个命令。

### 6.6创建独特的IV一个自足组内

本款规定，为创造独特的IV组的加密在一个自足单位。支持这些要求是强制性的加密单元，支持6.3Ç），以及可选的否则。

当创造出独特的IV一个自足组内，适用下列语句：

a）加密单元可能都有一个共同的密钥。

b）遵守本标准仅适用于整个自足组，不是某一个人在组内的加密单元。

c）各加密单元应当被配置来协调的自足组内创造独特的IV。

d）加密单位不得共享密码键，目前还没有加入任何加密单元自足组，除非这样的非自足组成员包括随机信息内的IV中定义的6.3 b）。

e）加密单位应只能用于与外部服务，确保和记录遵守声明为d）。

f）文档应描述系统如何防止任意两点间重复使用相同的IV自足组内如何加密单元的加密单元唯一地识别。这种识别应使用加密方法。

## 附件A

（供参考）

参考书目

**[B1] ANSI X9.31:1998, Digital Signatures Using Reversible Public Key Cryptography for the Financial**

**Services Industry (rDSA), 1998.**

**7**

**[B2] Biham, E., New Types of Cryptanalytic Attacks Using Related Keys, Advances in Cryptology—**

**EUROCRYPT'93, Springer-Verlag, 1994, pp. 398–409.**

**[B3] Canetti, Ran, Shai Halevi, and Michael Steiner. “Mitigating Dictionary Attacks on Password-**

**Protected Local Storage.” Advances in Cryptology—CRYPTO '06. LNCS vol. 4117, pages 160–**

**179. Springer-Verlag, available from the World Wide Web site <http://eprint.iacr.org/2006/276>,**

**2006.**

**[B4] ECRYPT IST-2002-507932 D.SPA.16, ECRYPT Yearly Report on Algorithms and Keysizes**

**(2005), Jan 2006.**

**[B5] Ferguson, Niels, Authentication weaknesses in GCM, available from the World Wide Web site**

**http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/comments/CWC-GCM/Ferguson2.pdf.**

**[B6] IEEE 100, The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition, New York,**

**Institute of Electrical and Electronics Engineers, Inc.**

**[B7] IEEE Std 1363a-2004, IEEE Standard Specifications for Public-Key Cryptography Amendment 1:**

**Additional Techniques.**

**[B8] IEEE P1619.3 (Draft 1, May 2007), Draft Standard for Key Management Infrastructure for**

**Cryptographic Protection of Stored Data.**

**8**

**[B9] IETF RFC 2898, PKCS #5: Password-Based Cryptography Specification Version 2.0., available**

**from the World Wide Web site http://www.ietf.org/rfc/rfc2898.txt, September 2000.**

**[B10] IETF RFC 3766, Determining Strengths For Public Keys Used For Exchanging Symmetric Keys,**

**available from the World Wide Web site http://www.ietf.org/rfc/rfc3766.txt, April 2004.**

**[B11] ISO/IEC 18031, Information Technology—Security techniques—Random bit generation, November**

**2005.**

**[B12] Joux, Antoine, Authentication Failures in NIST version of GCM, available from the World Wide**

**Web site http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/comments/800-38\_Series-**

**Drafts/GCM/Joux\_comments.pdf.**

**[B13] Keller, Sharon S., NIST-Recommended Random Number Generator Based on ANSI X9.31**

**Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms, January 2005.**

**[B14] NIST AES Key Wrap Specification, November 2001.**

**[B15] NIST FIPS 140-2, Federal Information Processing Standard 140-2, Announcing the Standard for**

**Security Requirements for Cryptographic Modules.**

**[B16] NIST Draft Special Publication 800-38D (June 27, 2007), Recommendation for Block Cipher Modes**

**of Operation: Galois/Counter Mode (GCM) and GMAC.**

**[B17] NIST Special Publication 800-57, Recommendation for Key Management—Part 1: General**

**(Revised), May 2006.**

**[B18] NIST Special Publication 800-63, Electronic Authentication Guideline: Recommendations of the**

**National Institute of Standards and Technology, April 2006.**

**[B19] NIST Special Publication 800-90, Recommendation for Random Number Generation Using**

**Deterministic Random Bit Generators.**

**[B20] RSA PKCS #1 v2.1: RSA Cryptography Standard, June 2002.**

## 附件B

（供参考）

安全问题

**B.1威胁模型**

设置这个标准是为了保护存储的数据，攻击者可能拥有完全访问存储介质：据推测，攻击者可能会对能够读取的存储介质中的内容，并也可以写信给它，包括更换一些任意攻击者的数据与存储的数据选择。我们还假定攻击者可能有非常大量的加密数据的访问。这样的威胁模型是适用的情况下，该存储介质还没有紧紧地绑定到加密单元。一个典型的例子是磁带加密，它是加密墨盒分别从磁带驱动器，在那里他们被第一次写入例行访问。

此外，这个标准是为了提供一些保护，即使在高度对抗性的情况下攻击者可以反复直播系统的存储介质的访问，并能监视或修改反复被写入和过书面的存储。然而，仅提供有限的保护重放攻击（见B.3）。

除了看和修改密文，攻击者可能有一些已知的或选择明文能力。这意味着，攻击者可能有一定的先验知识对应的明文密文写在存储介质上，它甚至可能是能够影响到主机写明文记录包含攻击者选择的文本。时，这是一个现实的假设主机是一个多用户的系统，这些用户都没有高度信任，甚至可现实在单用户的情况下（例如，当Web缓存的内容被写入到加密的磁盘）。

另一方面，这种威胁模型不包括传输中的信息的安全性（即，如何加密单元接收和发送的数据安全地存储）。它也不会覆盖最关键方面管理，如能产生，传输，和安全存储的键（没有解决误如使用密码转换到它自己的密钥加密）的密钥管理。许多这些其他标准方面解决。最后，各种物理​​（侧通道）攻击加密单元，如定时，电力，辐射，故障注入，和一个安全的随机位的设计发电机（RBG）超出范围。

**B.2维护加密密钥的安全性**

密码单元的安全性依赖于高质量的加密密钥。在理想的情况下，加密键应该来自一个加密的声音随机位发生器。用户不应使用来源，缺乏随机性，如密码，任何加密密钥。对于攻击者来说，这是比较容易的推出离线字典攻击的密码。 NIST提供指引，估计金额随机性内常用的密码（见NIST SP800-63[B18]）。另请参阅卡内蒂，等。 [B3]和IETF RFC2898[B9]。

有生力量的解决方案是由很多因素，包括实力的密码键，包装键和键，用于确保通信的安全性，以及许多方面的安全授权。 ECRYPT IST-2002-507932[B4]，IETF RFC3766[B10]，NIST SP800-57之间的等效密钥大小对称密钥和各种不同的估计[B17]非对称密钥加密算法。

**B.3重播攻击**

实施者应保持在特定的存储加密记录在适当的逻辑顺序介质上。否则，该加密单元是容易受到重放攻击，攻击者替换与其他一些正确认证的记录（如之前的版本，目前在媒体上记录记录）。

攻击是适用的环境中，攻击者就可以直接读取和写入到存储中，可以重新排列或重复加密记录，假设加密单元不验证订货。这可能是一个强大的攻击力，如果对手拥有丰富知识的明文希望改变更换特定的记录与其他记录备份集的内容，所有的加密具有相同的加密密钥。

许多网络加密标准，如IPsec，使用序列号来处理重放攻击。同样，加密单元应保持的记录顺序。这可能是简单的通过包括AAD或IV领域内的顺序记录数。

维护的顺序记录号码的帮助，但不处理的情况下，追加操作覆盖以前的数据。一个写通计数器的添加有助于确保数据具有正确的记录的数量和正确的顺序写。否则，攻击者可以重放记录不同的磁带，或与当前磁带上写通。经过验证的写通数防止这种攻击。

不幸的是，包括记录号，写通AAD或四场数内，会使其坚硬的，如果不是不可能的，执行原始加密的数据从一个到另一个磁带直接拷贝。传输加密的数据将只可能进行整个磁带副本。

订货验证似乎并不适用的环境中，主机具有随机访问的，如存储在一个硬盘驱动器。防止重放攻击，因此，必须在这些环境中可以通过其他的装置，例如控制访问媒体时，可能或依赖较高的水平的应用。这是超出本标准的范围。

不允许直接读取可能会提供一定程度的操作保护，防止重放攻击和写入（原始）记录在存储介质上的加密。这增加了攻击者的难度剥夺了他们在这些攻击中使用现成的工具，而不是迫使他们实现自己的的工具。它是明确的，然而，业务方面的保护，这是唯一有效的对休闲攻击。总是确定的，资金充足的攻击者可以通过构建自己的工具。

**B.4传递到主机之前明文检查MAC**

这个标准使得拨备对加密单元传递到主机之前检查明文消息认证码（MAC）。其目的是为了允许实现符合，即使他们无法存储整个解密的主机记录，然后将它传递到主机。然而，这样的实现可能有困难，在获得认证，如FIPS140-2[B15]。此外，既NIST SP800-38C和NIST SP800-38D[B16]要求，该装置之前验证的MAC返回任何明文。

由于这项津贴，重要的是，该主机不作用于任何之前的解密后的明文MAC验证完成。加密的数据使用尤其是CTR（计数器）模式（例如，CCM，GCM）可锻铸给攻击者，因为翻转位密文直接翻转中的相应位明文。这也很容易修改明文嵌入在任何CRC，因为CRC残差线性和仅依赖于其他变形的位进行置位的密文。如果攻击者有知识的明文，很容易妥协的数据进行任意修改。 CBC是明显不太容易XTS是不是容易受到这种攻击。

对休闲攻击的加密单元，可以提供业务保护不执行（原始）命令，可以直接读取和写入存储介质上的加密记录（见B.3）。

**B.5加密密钥完整性检查**

使用一个损坏的密钥加密单元，可能会导致资源浪费，甚至可能打开曝光相关密钥攻击（见比哈姆[B2]）。例如，当使用的HMAC之一，本标准模式（见5.4和5.5），其中只有第256位的加密密钥被损坏会导致加密记录，通过MAC检查，但仍然被解密以外的东西原始明文记录。

要避免使用损坏的键，应该采取一些措施，以验证密钥的完整性使用。确保密钥完整性的方法之一是使用“密钥签名。”举个例子，这可能是一个在使用一些更高级别的密钥加密密钥HMAC计算。如需进一步的指导，请参阅NIST SP800 - 57[B17]。

**B.6避免碰撞初始化向量**

本标准中规定的所有的模式，依赖于一个初始化向量（IV），被假定为非重复使用的加密密钥的范围内。在所有这些中，使用相同的组合加密密钥和IV加密两个不同记录（简称为静脉碰撞）的结果在某些曝光攻击和泄漏的明文信息。

这种接触是特别严重的如CCM和GCM模式使用计数器模式加密（即，变成一个流密码块加密）。在这些模式中，重复使用相同的IV在同一加密密钥构成作为再利用的密钥流在流密码，即，具有相同的风险的异或两个密文记录等于XOR两个明文记录，从而允许攻击者学习通过观察密文的明文记录信息。此外，GCM IV碰撞在某些情况下，可能会透露的信息了解的认证密钥（这是产生内部GCM算法）的攻击，从而使攻击者伪造的认证标签（见弗格森[B5]和Joux的[B12]）。

这种接触在其他两个模式（CBC和XTS）则不太严重，但它的存在，甚至有。例如，IV碰撞，攻击者可以看到两个加密的记录，如果是相同的，或者即使一些特定块中这些记录是相同的。要保持在本标准中的加密方式的安全性，因此重要的是要采取积极主动的措施，以避免IV碰撞。

考虑第四碰撞的概率时，重要的是考虑到的可能性相同的加密密钥加载到加密单元，来自不同制造商的，而且，这些IV防撞加密单元可能会使用不同的策略。因此，重要的是每个加密单元IV碰撞安全保证一定程度的行为，无论任何其他的加密单元，可能会得到相同的加密密钥，或有保证键经理和加密单元之间的加密密钥和IV（见6.5）保持一致的状态。

B.7 IV碰撞回避策略的一些例子中描述的，以及分析它们的各种情况下的有效性。

**B.7防止碰撞策略的例子**

**B.7.1例1：使用随机的IV**

在这个例子中，加密单元接收的密码密钥的密钥管理器，可用于某些加密模式，采用n位的IV。随着每一个加密记录，加密单元提取n位RBG用于记录的四。

为了分析这种策略的有效性，人们可以观察到任何两个IVS承担相同的值概率只有（因为所有的IVS是随机的）。当具有相同的加密密钥，加密Ç记录有C（C-1）/2（即c选择2）可能发生的碰撞事件，每一个发生的概率为。运用布尔的不平等（工会界），可以得到上界IV碰撞的概率中给出的方程（1）

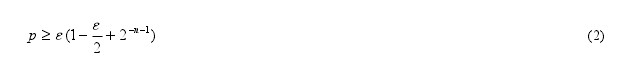


其中

p是概率的任何静脉碰撞易出现  
c是在一个特定的加密密钥加密的记录数  
n是IV的长度，单位是比特

例如，当使用一个128位的IV和加密（约）记录在相同的密钥， IV碰撞的概率为界，P </ +1= （约5.63×）。何时加密相同的记录96位的IV，绑定IV碰撞概率方程（1）：对/ = （1131072）。

从公式（1）的结合是在这种情况下，比较紧。事实上，稍微复杂一些的说法（使用容斥原理的事实，这些碰撞事件是成对的独立）意味着几乎匹配的下界在这种情况下，发生碰撞的概率，给出由公式（2）



其中

C:\Users\Administrator\AppData\Roaming\Tencent\Users\1826117925\QQ\WinTemp\RichOle\`_Z0BVM%FO7$)P4MV1CHXGV.jpg

在上面的例子中使用128位的IV和记录，这对撞概率是下限P>（仍有约），并在96位的IV和的例子中记录下限为（1131072.5）。

**B.7.2例2：增加随机IV**  
在这个例子中，加密单元再次接收来自密钥管理器与一些密码密钥加密模式，采用n位的IV。在收到从密钥管理的关键，加密单元提取n位RBG和存储这些位在状态寄存器。与每个加密记录，加密单元使用状态寄存器的当前内容为IV，然后递增，  
模一个状态寄存器。

这一策略的有效性在很大程度上依赖对密钥管理器。例如，密钥管理从未重新加载相同的两个不同的加密会话密钥有效地保证了IV-碰撞永远不会发生这种策略。另一方面，“最坏情况下的密钥管理器”，始终重新加载相同的密钥加密的每个记录将导致使用相同的IV碰撞概率与随机IV例如B.7.1。

为了得到更多的量化的答案，人们可以观察到任何两个工具变量仍然只承担相同的值，不超过的概率：2的IV要么属于相同的序列，在这种情况下，他们不能  
可能发生碰撞（假设没有序列长于记录），或属于两个不同的序列，在这种情况下，它们是来自于独立的输出的RBG只能以概率相撞。仍然可以使用，因此，方程（1）取得IV的概率的上界碰撞。

在当前示例中，从等式（1）中的表达是一个非常保守的上限持有无论密钥管理器的行为，而实际的碰撞概率，有时可以有太大的更小。例如，假定，只有S加密会话中使用的相同的加密密钥，每个加密会话加密ŗ的不同的记录（所以总记录数为R×S）。请注意，对于任何两个加密会话，任何的IV IV之间的碰撞的概率在这两个加密会话是（2R - 1）/<R /的，因为在这两个加密会话的初始随机数必须在R - 1彼此发生任何碰撞。由于是S加密会话，可以再次使用布尔的不平等的上限值的概率，其中任何两个产生碰撞，如图方程（3）。



其中

p是概率的任何静脉碰撞易出现

S是在一个特定的加密密钥加密的会话的数目

R是，在每一个加密会话进行加密的记录数

n是IV的长度，单位是比特

例如，在S =加密会话，每个R=加密记录（所以总的仍是加密的记录），我们得到了一个上限的p<使用128位的IV和p<当使用96位的IV[比较2公式（1）的边和]。相同的参数B.7.1也可以用在这里表明，这个界限是相当紧张的。

B.7.3例3：只随机的关键

在这个例子中，加密单元选择一个新的为每一个加密的加密密钥，并使用整我作为用于I的IV日加密会话的记录。显然，没有资讯供应商之间的碰撞是可能的相同的加密会话，因此，唯一的风险是关键的碰撞，这恰好可以忽略不计概率（因为加密密钥至少256位长）。

**B.8多少记录一键加密？**

B.7中的示例，可以作为指导的最大数据量，加密的边界与单一的加密密钥。具体而言，给定的最大可接受静脉碰撞的概率，并知识的加密模式，加密单元使用的防撞战略，可以设置一个上限，与单一的加密密钥数据进行加密。

表B.1包含了一个键记录的最大数量的防撞战略在实施例中从B.7.1和B.7.2。对于随机的IV，从方程（2）的表达使用（这是启发式有点更准确的表达公式（1）大ε值），和递增随机的IV，从方程（3）中的表达与R =2  
15被使用（即，假设每个使用加密会话进行加密215记录）。

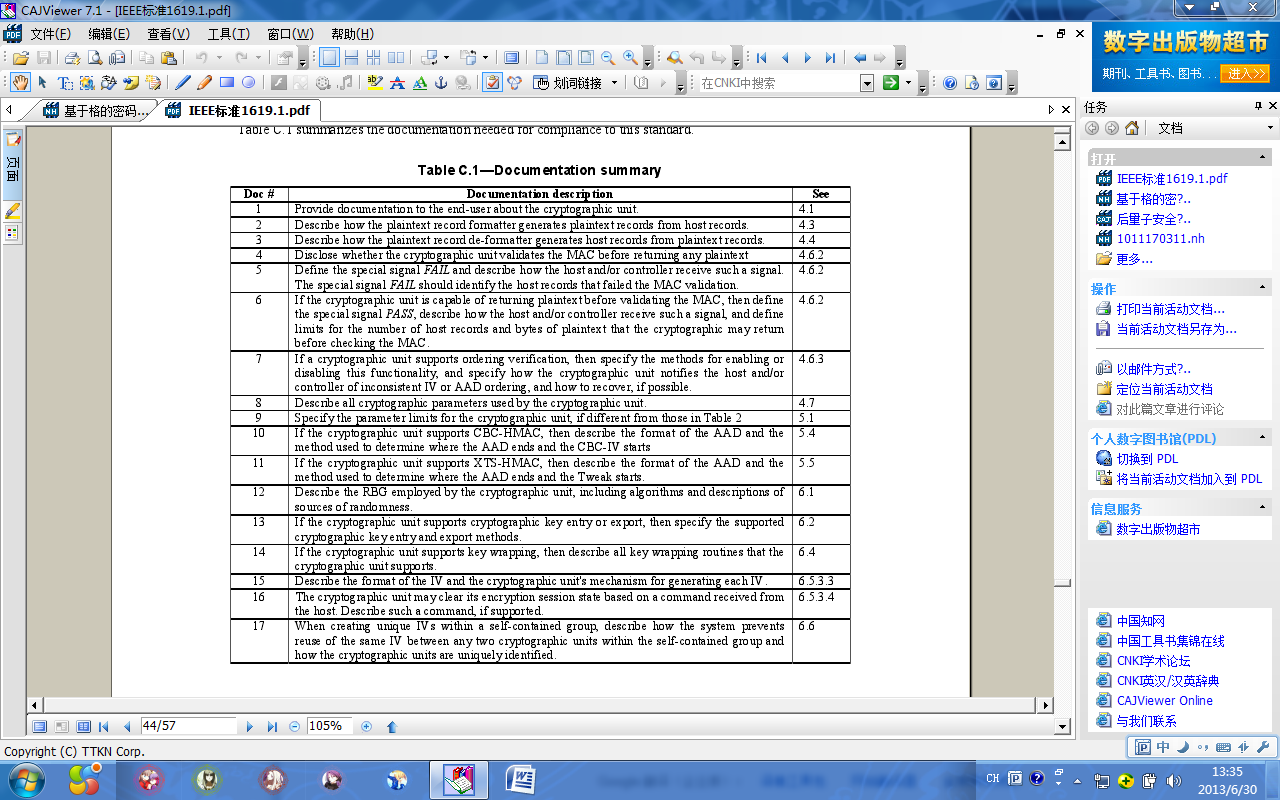
然而，应该强调，在表B.1中的范围只考虑IV-碰撞的概率，在多数设置，还有许多其他的考虑，必须加以考虑。例如，5.1包括一些其他限制，每个加密密钥可以被加密的数据的量，XTS和CBC加密模式需要其自身的局限性，在各自的标准（IEEE STD1619讨论和NIST SP800-38A，分别）。

## 附录C

(供参考)

文档摘要

表C.1总结符合本标准所需要的文档。

****

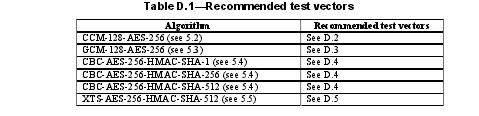
## 附录D

**(供参考)**

**测试向量**

D.1总则

一个加密单元应该测试其加密的功能，包括在本标准的测试向量和/或在参考文件中。表D.1给出了推荐的测试向量，为每个选择加密模式：



对于所有的测试向量，有以下缩写：

另外AAD认证数据

CIV CBC-IV的CBC-HMAC模式（见5.4）

CTX密文记录

DUS数据单元的序列号

HMK160，256或512位的HMAC密钥

IV初始化向量

KEY256位AES密钥

KEY1（XTS）第256位的密钥

KEY2（XTS唯一的）第二个256位的密钥

N / A不适用

非杜撰IV的CBC-HMAC模式（见5.4）

PTX明文记录

RPT一个给定的次数重复前面的AAD

TAG MAC（消息认证代码）

为了提高可读性，例子明确解析XTS-AES-256密钥key1和key2。 HMK是XTS-AES-256 HMAC密钥。

XTS测试向量内所有的数字小尾数位顺序（IEEE标准1619一样）。该基地这些数字是十六进制。

所有其他的测试向量内所有数字大端位顺序，其中最重要的字节是左边。的基地，这些数字是十六进制的，除了“RPT”字段，该字段是十进制。内一个特定的测试向量，如果多行具有相同前缀的开始，这些线路连接在一起。

对于所有的测试向量，每对十六进制数字是一个字节分为左边的数字是最重大和正确的数字是最重要的。

CBC-HMAC测试向量（见D.4），只使用最左边的位HMK，根据密钥大小  
所需的算法。例如，CBC-AES-256-HMAC-SHA-1使用HMK第160位，  
CBC-AES-256-HMAC-SHA-256采用前256位的HMK。

D.2 CCM-128-AES-256 test vectors

D.2.1 CCM-128-AES-256 test vector 1

KEY 0000000000000000000000000000000000000000000000000000000000000000

IV 000000000000000000000000

PTX 00000000000000000000000000000000

CTX c1944044c8e7aa95d2de9513c7f3dd8c

TAG 4b0a3e5e51f151eb0ffae7c43d010fdb

D.2.2 CCM-128-AES-256 test vector 2

KEY 0000000000000000000000000000000000000000000000000000000000000000

IV 000000000000000000000000

AAD 00000000000000000000000000000000

TAG 904704e89fb216443cb9d584911fc3c2

D.2.3 CCM-128-AES-256 test vector 3

KEY 0000000000000000000000000000000000000000000000000000000000000000

IV 000000000000000000000000

AAD 00000000000000000000000000000000

PTX 00000000000000000000000000000000

CTX c1944044c8e7aa95d2de9513c7f3dd8c

TAG 87314e9c1fa01abe6a6415943dc38521

D.2.4 CCM-128-AES-256 test vector 4

KEY fb7615b23d80891dd470980bc79584c8b2fb64ce60978f4d17fce45a49e830b7

IV dbd1a3636024b7b402da7d6f

PTX a845348ec8c5b5f126f50e76fefd1b1e

CTX cc881261c6a7fa72b96a1739176b277f

TAG 3472e1145f2c0cbe146349062cf0e423

D.2.5 CCM-128-AES-256 test vector 5

KEY 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

IV 101112131415161718191a1b

AAD 000102030405060708090a0b0c0d0e0f10111213

PTX 202122232425262728292a2b2c2d2e2f3031323334353637

CTX 04f883aeb3bd0730eaf50bb6de4fa2212034e4e41b0e75e5

TAG 9bba3f3a107f3239bd63902923f80371

D.2.6 CCM-128-AES-256 test vector 6

KEY 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

IV 101112131415161718191a1b

AAD 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

AAD 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

AAD 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

AAD 606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f

AAD 808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9f

AAD a0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebf

AAD c0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedf

AAD e0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff

RPT 0256

PTX 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

CTX 04f883aeb3bd0730eaf50bb6de4fa2212034e4e41b0e75e577f6bf2422c0f6d2

TAG 3376d2cf256ef613c56454cbb5265834

D.2.7 CCM-128-AES-256 test vector 7

KEY 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

IV 101112131415161718191a1b

AAD 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

PTX 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

PTX 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

PTX 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

PTX 606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f

PTX 808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9f

PTX a0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebf

PTX c0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedf

PTX e0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff

CTX 24d8a38e939d2710cad52b96fe6f82010014c4c43b2e55c557d69f0402e0d6f2

CTX 06c53d6cbd3f1c3c6de5dcdcad9fb74f25741dea741149fe4278a0cc24741e86

CTX 58cc0523b8d7838c60fb1de4b7c3941f5b26dea9322aa29656ec37ac18a9b108

CTX a6f38b7917f5a9c398838b22afbd17252e96694a9e6237964a0eae21c0a6e152

CTX 15a0e82022926be97268249599e456e05029c3ebc07d78fc5b4a0862e04e68c2

CTX 9514c7bdafc4b52e04833bf30622e4eb42504a44a9dcbc774752de7bb82891ad

CTX 1eba9dc3281422a8aba8654268d3d9c81705f4c5a531ef856df5609a159af738

CTX eb753423ed2001b8f20c23725f2bef18c409f7e52132341f27cb8f0e79894dd9

TAG ebb1fa9d28ccfe21bdfea7e6d91e0bab

D.2.8 CCM-128-AES-256 test vector 8

KEY fb7615b23d80891dd470980bc79584c8b2fb64ce6097878d17fce45a49e830b7

IV dbd1a3636024b7b402da7d6f

AAD 36

PTX a9

CTX 9d

TAG 3261b1cf931431e99a32806738ecbd2a

D.2.9 CCM-128-AES-256 test vector 9

KEY f8d476cfd646ea6c2384cb1c27d6195dfef1a9f37b9c8d21a79c21f8cb90d289

IV dbd1a3636024b7b402da7d6f

AAD 7bd859a247961a21823b380e9fe8b65082ba61d3

PTX 90ae61cf7baebd4cade494c54a29ae70269aec71

CTX 6c05313e45dc8ec10bea6c670bd94f31569386a6

TAG 8f3829e8e76ee23c04f566189e63c686

D.3 GCM-128-AES-256 test vectors

D.3.1 GCM-128-AES-256 test vector 1

KEY 0000000000000000000000000000000000000000000000000000000000000000

IV 000000000000000000000000

PTX 00000000000000000000000000000000

CTX cea7403d4d606b6e074ec5d3baf39d18

TAG d0d1c8a799996bf0265b98b5d48ab919

D.3.2 GCM-128-AES-256 test vector 2

KEY 0000000000000000000000000000000000000000000000000000000000000000

IV 000000000000000000000000

AAD 00000000000000000000000000000000

TAG 2d45552d8575922b3ca3cc538442fa26

D.3.3 GCM-128-AES-256 test vector 3

KEY 0000000000000000000000000000000000000000000000000000000000000000

IV 000000000000000000000000

AAD 00000000000000000000000000000000

PTX 00000000000000000000000000000000

CTX cea7403d4d606b6e074ec5d3baf39d18

TAG ae9b1771dba9cf62b39be017940330b4

D.3.4 GCM-128-AES-256 test vector 4

KEY fb7615b23d80891dd470980bc79584c8b2fb64ce60978f4d17fce45a49e830b7

IV dbd1a3636024b7b402da7d6f

PTX a845348ec8c5b5f126f50e76fefd1b1e

CTX 5df5d1fabcbbdd051538252444178704

TAG 4c43cce5a574d8a88b43d4353bd60f9f

D.3.5 GCM-128-AES-256 test vector 5

KEY 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

IV 101112131415161718191a1b

AAD 000102030405060708090a0b0c0d0e0f10111213

PTX 202122232425262728292a2b2c2d2e2f3031323334353637

CTX 591b1ff272b43204868ffc7bc7d521993526b6fa32247c3c

TAG 7de12a5670e570d8cae624a16df09c08

D.3.6 GCM-128-AES-256 test vector 6

KEY 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

IV 101112131415161718191a1b

AAD 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

AAD 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

AAD 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

AAD 606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f

AAD 808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9f

AAD a0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebf

AAD c0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedf

AAD e0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff

RPT 0256

PTX 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

CTX 591b1ff272b43204868ffc7bc7d521993526b6fa32247c3c4057f3eae7548cef

TAG a1de5536e97edddccd26eeb1b5ff7b32

D.3.7 GCM-128-AES-256 test vector 7

KEY 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

IV 101112131415161718191a1b

AAD 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

PTX 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

PTX 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

PTX 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

PTX 606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f

PTX 808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9f

PTX a0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebf

PTX c0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedf

PTX e0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff

CTX 793b3fd252941224a6afdc5be7f501b9150696da12045c1c6077d3cac774accf

CTX c3d530d848d665d81a49cbb500b88bbb624ae61d1667229c302dc6ff0bb4d70b

CTX dbbc8566d6f5b158da99a2ff2e01dda629b89c34ad1e5feba70e7aae4328289c

CTX 3629b0588350581ca8b97ccf1258fa3bbe2c5026047ba72648969cff8ba10ae3

CTX 0e05935df0c693741892b76faf67133abd2cf2031121bd8bb38127a4d2eedeea

CTX 13276494f402cd7c107fb3ec3b24784834338e55436287092ac4a26f5ea7ea4a

CTX d68d73151639b05b24e68b9816d1398376d8e4138594758db9ad3b409259b26d

CTX cfc06e722be987b3767f70a7b856b774b1ba2685b368091429fccb8dcdde09e4

TAG 87ec837abf532855b2cea169d6943fcd

D.3.8 GCM-128-AES-256 test vector 8

KEY fb7615b23d80891dd470980bc79584c8b2fb64ce6097878d17fce45a49e830b7

IV dbd1a3636024b7b402da7d6f

AAD 36

PTX a9

CTX 0a

TAG be987d009a4b349aa80cb9c4ebc1e9f4

D.3.9 GCM-128-AES-256 test vector 9

KEY f8d476cfd646ea6c2384cb1c27d6195dfef1a9f37b9c8d21a79c21f8cb90d289

IV dbd1a3636024b7b402da7d6f

AAD 7bd859a247961a21823b380e9fe8b65082ba61d3

PTX 90ae61cf7baebd4cade494c54a29ae70269aec71

CTX ce2027b47a843252013465834d75fd0f0729752e

TAG acd8833837ab0ede84f4748da8899c15

D.3.10 GCM-128-AES-256 test vector 10

KEY dbbc8566d6f5b158da99a2ff2e01dda629b89c34ad1e5feba70e7aae4328289c

IV cfc06e722be987b3767f70a7b856b774

PTX ce2027b47a843252013465834d75fd0f

CTX dc03e524830d30f88e197f3acace66ef

TAG 9984eff6905755d1836f2db04089634c

D.3.11 GCM-128-AES-256 test vector 11

KEY 0e05935df0c693741892b76faf67133abd2cf2031121bd8bb38127a4d2eedeea

IV 74b1ba2685b368091429fccb8dcdde09e4

AAD 7bd859a247961a21823b380e9fe8b65082ba61d3

PTX 90ae61cf7baebd4cade494c54a29ae70269aec71

CTX 6be65e56066c4056738c03fe2320974ba3f65e09

TAG 6108dc417bf32f7fb7554ae52f088f87

D.3.12 GCM-128-AES-256 test vector 12

KEY 0000000000000000000000000000000000000000000000000000000000000000

IV 02cbbc7a03eb4de39d80d1ebc988bfdf

AAD 688e1aa984de926dc7b4c47f44

PTX a2aab3ad8b17acdda288426cd7c429b7ca86b7aca05809c70ce82db25711cb53

PTX 02eb2743b036f3d750d6cf0dc0acb92950d546db308f93b4ff244afa9dc72bcd

PTX 758d2c

CTX ee62552aebc0c3c7daae12bb6c32ca5a005f4a1aaab004ed0f0b30abbf15acf4

CTX c50c59662d4b4468419544e7f981973563ce556ae50859ee09b14d31a053986f

CTX 9ac89b

TAG 9cd0db936e26d44be974ba868285a2e1

D.4 CBC-AES-256-HMAC-SHA test vectors (including HMAC-SHA-1, HMAC-SHA-

256, and HMAC-SHA-512)

D.4.1 CBC-AES-256-HMAC-SHA test vector 1

KEY 0000000000000000000000000000000000000000000000000000000000000000

HMK 0000000000000000000000000000000000000000000000000000000000000000

HMK 0000000000000000000000000000000000000000000000000000000000000000

NON N/A

CIV 00000000000000000000000000000000

PTX 00000000000000000000000000000000

CTX dc95c078a2408989ad48a21492842087

HMAC-SHA-1

TAG 59bb230e817ad3f377d623d2ca97eeffd0fd467c

HMAC-SHA-256

TAG 2cf16e982f18a9009687c8a8bf26cfd31e66bdda7277008d9564dd4779511855

HMAC-SHA-512

TAG bf8b5d45be53465f09ed9a4f53c565f067b3138318195425dfc466856973170d

TAG f8414dceb7d1c8888a622de9ea480840193f8ebd94c34a26bb692a31568e3949

D.4.2 CBC-AES-256-HMAC-SHA test vector 2

KEY 0000000000000000000000000000000000000000000000000000000000000000

HMK 0000000000000000000000000000000000000000000000000000000000000000

HMK 0000000000000000000000000000000000000000000000000000000000000000

AAD 00000000000000000000000000000000

NON N/A

CIV 00000000000000000000000000000000

HMAC-SHA-1

TAG 66040990c7992a2a00d037d0b8631c0db1785897

HMAC-SHA-256

TAG 853c7403937d8b6239569b184eb7993fc5f751aefcea28f2c863858e2d29c50b

HMAC-SHA-512

TAG 65e879d47df1def0af378d32e9f4fe3a824fb51e2143c03322def229361af3b1

TAG 7a724a3d653d05cb9f41f4b90d09e8e2886a78da48537d1cfa62977a82e7374e

D.4.3 CBC-AES-256-HMAC-SHA test vector 3

KEY 0000000000000000000000000000000000000000000000000000000000000000

HMK 0000000000000000000000000000000000000000000000000000000000000000

HMK 0000000000000000000000000000000000000000000000000000000000000000

AAD 00000000000000000000000000000000

NON N/A

CIV 00000000000000000000000000000000

PTX 00000000000000000000000000000000

CTX dc95c078a2408989ad48a21492842087

HMAC-SHA-1

TAG d5adb529213cd69a9a3d69cf2d10b0b469d936fe

HMAC-SHA-256

TAG 16a65111bd8e5a0af5f001f7d9200d44252bcfe5dc34da42315b99213b9cbb4b

HMAC-SHA-512

TAG 6d63ccdc62d5d376cc86eb6a144568d04b8cdf28955509df10a4bbe8c734d5af

TAG 37e8e524d30fed83d324b8dedb06d86636baa67f85caac73cc993f00ecb92dec

D.4.4 CBC-AES-256-HMAC-SHA test vector 4

KEY fb7615b23d80891dd470980bc79584c8b2fb64ce60978f4d17fce45a49e830b7

HMK 1b07a0e93c1f4c3aadff671dd2611ac2fe22d34c6b6d8630c30dd44f41d49fe5

HMK ad0a3dbdd0f13ca27e6523c5e4e2ab12884741a1af9b95f3cf6c0aec3b68ba40

NON N/A

CIV dbd1a3636024b7b402da7d6fe3fb056e

PTX a845348ec8c5b5f126f50e76fefd1b1e

CTX fd057a7f6d17bd747aced7b6fc948567

HMAC-SHA-1

TAG 3bd64954b1b5b0a98ac3a6f95d2e5fe65b5377c0

HMAC-SHA-256

TAG 3e5530fb364c80696b1b2f69e8d0de064a3e07ad1a0b795f00fcdec1649cabcb

HMAC-SHA-512

TAG 444aec157e48e683626bf14d26c9bfd9515d5def34582c034f0c3311dd7d9753

TAG 591f3effe264b8cdfaf755177b8a020a47edb7331fef628523d708aefe09b0da

D.4.5 CBC-AES-256-HMAC-SHA test vector 5

KEY 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

HMK 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

HMK 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

AAD 000102030405060708090a0b0c0d0e0f10111213

NON N/A

CIV 101112131415161718191a1b1c1d1e1f

PTX 202122232425262728292a2b2c2d2e2f

CTX 7b626546c8d79cdeb66edef23b9b7d72

HMAC-SHA-1

TAG c7932ddb8fc2212b56b1207e81019b556f4bb7d9

HMAC-SHA-256

TAG 6b0fe0b40a41e32d2c61726a3d7834014a8ee07873ccfe0c23f3a9073b90b099

HMAC-SHA-512

TAG efac7480579348343d1e9af4fc6896968080439717c3b2c3e63013aa718261f0

TAG e3ee43c6fdb4372f020d64c9fee4bc7743cfd9262d3adf03aec4f8d99fd178e4

D.4.6 CBC-AES-256-HMAC-SHA test vector 6

KEY 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

HMK 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

HMK 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

AAD 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

AAD 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

AAD 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

AAD 606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f

AAD 808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9f

AAD a0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebf

AAD c0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedf

AAD e0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff

NON N/A

CIV 101112131415161718191a1b1c1d1e1f

PTX 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

CTX 7b626546c8d79cdeb66edef23b9b7d723d5f9d5bc2a411f1eb448442250eeca2

HMAC-SHA-1

TAG 49dc3eacadcc028df2bf9a4598e3fec6624c8b38

HMAC-SHA-256

TAG c2ea45b50293d8f62d348ef23aec702268eb66bb3e2248eb9f71a5817709da2f

HMAC-SHA-512

TAG 8b2e672aacc78b6ff58c770fd0d6ed252201ebbae95dceec912c0cf3bf27171b

TAG 3627a6fefc3cc8b9f9e64b542b64c06ebb786f986cdc8296bac15111dbffa82f

D.4.7 CBC-AES-256-HMAC-SHA test vector 7

KEY 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

HMK 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

HMK 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

AAD 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

NON N/A

CIV 101112131415161718191a1b1c1d1e1f

PTX 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

PTX 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

PTX 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

PTX 606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f

PTX 808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9f

PTX a0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebf

PTX c0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedf

PTX e0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff

CTX c9cb7b3859e1a550bcbf11b624022c56c3ad1479e5ce7034d7a03c13d8fb9502

CTX 6f7254c50ce4ebd743486d00e09ddd8e873a7e98984ad43f57088c510e911700

CTX 6acfe2fef69b4010f0f05a93af7d3a93a02085780fd5acb3a4eb870933077752

CTX 2f2c18e310ac0c0c3766bea3e97f71996336e4831f3b411fb2700ddbab565673

CTX 315bf4ab73c7e11abac4d0cfc228f1ac60dd10f85f9c2ade46a9af5eacb6a24a

CTX 43839b942e71ca4ce2080a809a04a849105da07efbbb2f60b9c376e0354e2a27

CTX da1eaa5c7adea77890cc25b6bd48229e17ce518040ceb46a04fc7b62444e77b5

CTX aaf3dbf60a660a2b68ec640622716b07758d99a0f598a73ed8bdae74fa3aae2f

HMAC-SHA-1

TAG 2e08d65f81ff646ad05ab7aaf42903aa760e577a

HMAC-SHA-256

TAG ebfe6f31be473ab22b649602a77f7408508dfa50cad109cbc97f2fe5f8bb8583

HMAC-SHA-512

TAG 7b326204521161942844c0970391344cdac71ce0440325b02203b537dd930799

TAG 0e158541dfc52cfcf69d3e8085658de4c98bc030273bad369fdf28aaad40e63c

D.4.8 CBC-AES-256-HMAC-SHA test vector 8

KEY fb7615b23d80891dd470980bc79584c8b2fb64ce6097878d17fce45a49e830b7

HMK cc84a6cca8f97b8a5624071aec7d09e7cf5bdaff239d467270f9716ba234d109

HMK ac60cf491d5105fc60fc5804c6474bc35cf9ead9123da80f649ca15a98a243d6

AAD 7bd859a2

NON N/A

CIV dbd1a3636024b7b402da7d6f54a67dc8

PTX 90ae61cf7baebd4cade494c54a29ae70

CTX 6cd763ff6144ede649c486f9404a5307

HMAC-SHA-1

TAG efc87d364ccab9d4bdc241185f1d847e2e16d8c4

HMAC-SHA-256

TAG f130415f56372bcd17250339d82118ca347be4cfff9f69181757cf5e98b0a775

HMAC-SHA-512

TAG 1604c3afb72546c2f6a9135df46ae799fdae4d9f5a87fdffd552016c5e4ed98a

TAG 393b62822df55b076e3dc6f9668234919bbdcc99f2b40379754cc6ac30c97250

D.4.9 CBC-AES-256-HMAC-SHA test vector 9

KEY 0000000000000000000000000000000000000000000000000000000000000000

HMK 0000000000000000000000000000000000000000000000000000000000000000

HMK 0000000000000000000000000000000000000000000000000000000000000000

NON 00000000000000000000000000000000

CIV dc95c078a2408989ad48a21492842087

PTX 00000000000000000000000000000000

CTX 08c374848c228233c2b34f332bd2e9d3

HMAC-SHA-1

TAG dedf216e04f467eaad1e5a72b6a7c962c8281f13

HMAC-SHA-256

TAG 1f4dd7b6d7436b5b7d325c0c2411ed4fc02c101949eb8269e8166e8c6325e858

HMAC-SHA-512

TAG d8677480b0466345b3c32baa2c2b502fb3bfba01e759c4d1da04ca7c20dd9e55

TAG 00b3675d0e78e080125b68fd0c584ff3144b1e155a1136785ad723f3c69e23b5

D.5 XTS-AES-256-HMAC-SHA-512 test vectors

D.5.1 XTS-AES-256-HMAC-SHA-512 test vector 1

Key1 2718281828459045235360287471352662497757247093699959574966967627

Key2 3141592653589793238462643383279502884197169399375105820974944592

HMK 0000000000000000000000000000000000000000000000000000000000000000

HMK 0000000000000000000000000000000000000000000000000000000000000000

DUS ff000000000000000000000000000000

PTX 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

PTX 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

PTX 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

PTX 606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f

PTX 808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9f

PTX a0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebf

PTX c0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedf

PTX e0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff

PTX 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

PTX 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

PTX 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

PTX 606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f

PTX 808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9f

PTX a0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebf

PTX c0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedf

PTX e0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff

CTX 1c3b3a102f770386e4836c99e370cf9bea00803f5e482357a4ae12d414a3e63b

CTX 5d31e276f8fe4a8d66b317f9ac683f44680a86ac35adfc3345befecb4bb188fd

CTX 5776926c49a3095eb108fd1098baec70aaa66999a72a82f27d848b21d4a741b0

CTX c5cd4d5fff9dac89aeba122961d03a757123e9870f8acf1000020887891429ca

CTX 2a3e7a7d7df7b10355165c8b9a6d0a7de8b062c4500dc4cd120c0f7418dae3d0

CTX b5781c34803fa75421c790dfe1de1834f280d7667b327f6c8cd7557e12ac3a0f

CTX 93ec05c52e0493ef31a12d3d9260f79a289d6a379bc70c50841473d1a8cc81ec

CTX 583e9645e07b8d9670655ba5bbcfecc6dc3966380ad8fecb17b6ba02469a020a

CTX 84e18e8f84252070c13e9f1f289be54fbc481457778f616015e1327a02b140f1

CTX 505eb309326d68378f8374595c849d84f4c333ec4423885143cb47bd71c5edae

CTX 9be69a2ffeceb1bec9de244fbe15992b11b77c040f12bd8f6a975a44a0f90c29

CTX a9abc3d4d893927284c58754cce294529f8614dcd2aba991925fedc4ae74ffac

CTX 6e333b93eb4aff0479da9a410e4450e0dd7ae4c6e2910900575da401fc07059f

CTX 645e8b7e9bfdef33943054ff84011493c27b3429eaedb4ed5376441a77ed4385

CTX 1ad77f16f541dfd269d50d6a5f14fb0aab1cbb4c1550be97f7ab4066193c4caa

CTX 773dad38014bd2092fa755c824bb5e54c4f36ffda9fcea70b9c6e693e148c151

TAG 1c7105d3c1e8e235ffb013d5e8023729a35cdeacc16af1d7f5f0fec6c036b167

TAG 871649687c5692aaa0ada9773671939bbce2a3d15dcae43671aa6ca5f3a96a6f

D.5.2 XTS-AES-256-HMAC-SHA-512 test vector 2

Key1 2718281828459045235360287471352662497757247093699959574966967627

Key2 3141592653589793238462643383279502884197169399375105820974944592

HMK 1b07a0e93c1f4c3aadff671dd2611ac2fe22d34c6b6d8630c30dd44f41d49fe5

HMK ad0a3dbdd0f13ca27e6523c5e4e2ab12884741a1af9b95f3cf6c0aec3b68ba40

DUS ffff0000000000000000000000000000

PTX 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

PTX 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

PTX 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

PTX 606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f

PTX 808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9f

PTX a0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebf

PTX c0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedf

PTX e0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff

PTX 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

PTX 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

PTX 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

PTX 606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f

PTX 808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9f

PTX a0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebf

PTX c0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedf

PTX e0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff

CTX 77a31251618a15e6b92d1d66dffe7b50b50bad552305ba0217a610688eff7e11

CTX e1d0225438e093242d6db274fde801d4cae06f2092c728b2478559df58e837c2

CTX 469ee4a4fa794e4bbc7f39bc026e3cb72c33b0888f25b4acf56a2a9804f1ce6d

CTX 3d6e1dc6ca181d4b546179d55544aa7760c40d06741539c7e3cd9d2f6650b201

CTX 3fd0eeb8c2b8e3d8d240ccae2d4c98320a7442e1c8d75a42d6e6cfa4c2eca179

CTX 8d158c7aecdf82490f24bb9b38e108bcda12c3faf9a21141c3613b58367f922a

CTX aa26cd22f23d708dae699ad7cb40a8ad0b6e2784973dcb605684c08b8d6998c6

CTX 9aac049921871ebb65301a4619ca80ecb485a31d744223ce8ddc2394828d6a80

CTX 470c092f5ba413c3378fa6054255c6f9df4495862bbb3287681f931b687c888a

CTX bf844dfc8fc28331e579928cd12bd2390ae123cf03818d14dedde5c0c24c8ab0

CTX 18bfca75ca096f2d531f3d1619e785f1ada437cab92e980558b3dce1474afb75

CTX bfedbf8ff54cb2618e0244c9ac0d3c66fb51598cd2db11f9be39791abe447c63

CTX 094f7c453b7ff87cb5bb36b7c79efb0872d17058b83b15ab0866ad8a58656c5a

CTX 7e20dbdf308b2461d97c0ec0024a2715055249cf3b478ddd4740de654f75ca68

CTX 6e0d7345c69ed50cdc2a8b332b1f8824108ac937eb050585608ee734097fc090

CTX 54fbff89eeaeea791f4a7ab1f9868294a4f9e27b42af8100cb9d59cef9645803

TAG ecabc09097f1401bc289548a9b932bf197a1a7002665f36529e5e137395facc9

TAG 7133399c65a05f15cc81abc8067155ccaabd6fa64f744cb1d987d29100c7f523

D.5.3 XTS-AES-256-HMAC-SHA-512 test vector 3

Key1 2718281828459045235360287471352662497757247093699959574966967627

Key2 3141592653589793238462643383279502884197169399375105820974944592

HMK 072126bc492870f666b25023a548a9154b64d06f890ba3542b5198466c60c53d

HMK b4763ddda4de7bbc469113a8cd9196e064ff86b04d1cbbfdfdc305998402756d

AAD 6369757120656854206e776f7262206b706d756a20786f6674207265766f2073

DUS ffffff00000000000000000000000000

PTX 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

PTX 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

PTX 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

PTX 606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f

PTX 808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9f

PTX a0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebf

PTX c0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedf

PTX e0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff

PTX 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

PTX 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

PTX 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

PTX 606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f

PTX 808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9f

PTX a0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebf

PTX c0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedf

PTX e0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff

CTX e387aaa58ba483afa7e8eb469778317ecf4cf573aa9d4eac23f2cdf914e4e200

CTX a8b490e42ee646802dc6ee2b471b278195d60918ececb44bf79966f83faba049

CTX 9298ebc699c0c8634715a320bb4f075d622e74c8c932004f25b41e361025b5a8

CTX 7815391f6108fc4afa6a05d9303c6ba68a128a55705d415985832fdeaae6c8e1

CTX 9110e84d1b1f199a2692119edc96132658f09da7c623efcec712537a3d94c0bf

CTX 5d7e352ec94ae5797fdb377dc1551150721adf15bd26a8efc2fcaad56881fa9e

CTX 62462c28f30ae1ceaca93c345cf243b73f542e2074a705bd2643bb9f7cc79bb6

CTX e7091ea6e232df0f9ad0d6cf502327876d82207abf2115cdacf6d5a48f6c1879

CTX a65b115f0f8b3cb3c59d15dd8c769bc014795a1837f3901b5845eb491adfefe0

CTX 97b1fa30a12fc1f65ba22905031539971a10f2f36c321bb51331cdefb39e3964

CTX c7ef079994f5b69b2edd83a71ef549971ee93f44eac3938fcdd61d01fa71799d

CTX a3a8091c4c48aa9ed263ff0749df95d44fef6a0bb578ec69456aa5408ae32c7a

CTX f08ad7ba8921287e3bbee31b767be06a0e705c864a769137df28292283ea81a2

CTX 480241b44d9921cdbec1bc28dc1fda114bd8e5217ac9d8ebafa720e9da4f9ace

CTX 231cc949e5b96fe76ffc21063fddc83a6b8679c00d35e09576a875305bed5f36

CTX ed242c8900dd1fa965bc950dfce09b132263a1eef52dd6888c309f5a7d712826

TAG a9fe02bbb70c062c93d958bc32936609a25a1ffa2dcd9f33aee88be73d943d4f

TAG dcbd459c0ecb0111c9c74cfcf2d5104f5f8262ae52444d6e744d8046f73ec7f2

D.5.4 XTS-AES-256-HMAC-SHA-512 test vector 4

Key1 2718281828459045235360287471352662497757247093699959574966967627

Key2 3141592653589793238462643383279502884197169399375105820974944592

HMK e19c148c56a3aa6737471aaba4909f06a17705e98bb8ee347e253c26cbf00cc5

HMK 3147ec26beb88413da0268d39bb4a707678277a0c927c10f565496d0fe3349d5

AAD e3e220f1f7f8ef20f9e820ece520e1e9ed20e6ea20e0ea20ecf4faf220f4e2f9

AAD 20e7e1e5f8e420f0e7eee3e420f9f6f6e420ebea2e0d0a

DUS ffffffff000000000000000000000000

PTX 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

PTX 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

PTX 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

PTX 606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f

PTX 808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9f

PTX a0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebf

PTX c0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedf

PTX e0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff

PTX 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

PTX 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

PTX 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

PTX 606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f

PTX 808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9f

PTX a0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebf

PTX c0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedf

PTX e0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff

CTX bf53d2dade78e822a4d949a9bc6766b01b06a8ef70d26748c6a7fc36d80ae4c5

CTX 520f7c4ab0ac8544424fa405162fef5a6b7f229498063618d39f0003cb5fb8d1

CTX c86b643497da1ff945c8d3bedeca4f479702a7a735f043ddb1d6aaade3c4a0ac

CTX 7ca7f3fa5279bef56f82cd7a2f38672e824814e10700300a055e1630b8f1cb0e

CTX 919f5e942010a416e2bf48cb46993d3cb6a51c19bacf864785a00bc2ecff15d3

CTX 50875b246ed53e68be6f55bd7e05cfc2b2ed6432198a6444b6d8c247fab941f5

CTX 69768b5c429366f1d3f00f0345b96123d56204c01c63b22ce78baf116e525ed9

CTX 0fdea39fa469494d3866c31e05f295ff21fea8d4e6e13d67e47ce722e9698a1c

CTX 1048d68ebcde76b86fcf976eab8aa9790268b7068e017a8b9b749409514f1053

CTX 027fd16c3786ea1bac5f15cb79711ee2abe82f5cf8b13ae73030ef5b9e4457e7

CTX 5d1304f988d62dd6fc4b94ed38ba831da4b7634971b6cd8ec325d9c61c00f1df

CTX 73627ed3745a5e8489f3a95c69639c32cd6e1d537a85f75cc844726e8a72fc00

CTX 77ad22000f1d5078f6b866318c668f1ad03d5a5fced5219f2eabbd0aa5c0f460

CTX d183f04404a0d6f469558e81fab24a167905ab4c7878502ad3e38fdbe62a4155

CTX 6cec37325759533ce8f25f367c87bb5578d667ae93f9e2fd99bcbc5f2fbba88c

CTX f6516139420fcff3b7361d86322c4bd84c82f335abb152c4a93411373aaa8220

TAG a6babc0886b9f7c7f16844449dc6fa549d4909969dab34f85287cd5a76bc6c41

TAG d58f3436f0654cad9987e04b95d54900d2a3e09c5264041941b5b56ba26cd7c2

D.5.5 XTS-AES-256-HMAC-SHA-512 test vector 5

Key1 2718281828459045235360287471352662497757247093699959574966967627

Key2 3141592653589793238462643383279502884197169399375105820974944592

HMK 6273d67c8fd3f0b06d1801507f3e42c06dd6d2a831914ad9439790fa9525a349

HMK da015634fee3bc417e41c3012174806c3242ae474e66c9f81f597ebd4b7ca2ca

AAD 4f6e63652075706f6e20612074696d652c207468657265207761732061206e6f

AAD 6e636f6e666f726d696e672073706172726f772077686f206465636964656420

AAD 6e6f7420746f20666c7920736f75746820666f72207468652077696e7465722e

AAD 20486f77657665722c20736f6f6e207468652077656174686572207475726e65

AAD 6420736f20636f6c6420746861742068652072656c756374616e746c79207374

AAD 617274656420736f757468776172642e20496e20612073686f72742074696d65

AAD 2c2069636520626567616e20746f20666f726d206f6e206869732077696e6773

AAD 20616e642068652066656c6c20746f20656172746820696e2061206261726e79

AAD 6172642c20616c6d6f73742066726f7a656e2e204120636f7720706173736564

AAD 20627920616e642063726170706564206f6e20746865206c6974746c65207370

AAD 6172726f772e205468652073706172726f772074686f75676874206974207761

AAD 732074686520656e642e20427574207468656e20746865206d616e7572652077

AAD 61726d65642068696d20616e6420646566726f73746564206869732077696e67

AAD 732e205761726d20616e642068617070792c2061626c6520746f206272656174

AAD 68652c206865207374617274656420746f2073696e672e204a75737420746865

AAD 6e2061206c61726765206361742063616d6520627920616e642068656172696e

AAD 6720746865206368697270696e672c20696e7665737469676174656420746865

AAD 20736f756e64732e205468652063617420636c65617265642061776179207468

AAD 65206d616e7572652c20666f756e6420746865206368697270696e6720737061

AAD 72726f7720616e642070726f6d70746c79206174652068696d2e0a0a54484520

AAD 4d4f52414c204f46205448452053544f52590a0a312e2045766572796f6e6520

AAD 77686f207368697473206f6e20796f75206973206e6f74206e65636573736172

AAD 696c7920796f757220656e656d792e0a322e2045766572796f6e652077686f20

AAD 6765747320796f75206f7574206f662073686974206973206e6f74206e656365

AAD 73736172696c7920796f757220667269656e642e0a332e20416e642c20696620

AAD 796f75277265207761726d20616e6420686170707920696e20612070696c6520

AAD 6f6620736869742c206b65657020796f7572206d6f7574682073687574210a0a

DUS ffffffffff0000000000000000000000

PTX 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

PTX 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

PTX 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

PTX 606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f

PTX 808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9f

PTX a0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebf

PTX c0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedf

PTX e0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff

PTX 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

PTX 202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f

PTX 404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f

PTX 606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f

PTX 808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9f

PTX a0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebf

PTX c0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedf

PTX e0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff

CTX 64497e5a831e4a932c09be3e5393376daa599548b816031d224bbf50a818ed23

CTX 50eae7e96087c8a0db51ad290bd00c1ac1620857635bf246c176ab463be30b80

CTX 8da548081ac847b158e1264be25bb0910bbc92647108089415d45fab1b3d2604

CTX e8a8eff1ae4020cfa39936b66827b23f371b92200be90251e6d73c5f86de5fd4

CTX a950781933d79a28272b782a2ec313efdfcc0628f43d744c2dc2ff3dcb66999b

CTX 50c7ca895b0c64791eeaa5f29499fb1c026f84ce5b5c72ba1083cddb5ce45434

CTX 631665c333b60b11593fb253c5179a2c8db813782a004856a1653011e93fb6d8

CTX 76c18366dd8683f53412c0c180f9c848592d593f8609ca736317d356e13e2bff

CTX 3a9f59cd9aeb19cd482593d8c46128bb32423b37a9adfb482b99453fbe25a41b

CTX f6feb4aa0bef5ed24bf73c762978025482c13115e4015aac992e5613a3b5c2f6

CTX 85b84795cb6e9b2656d8c88157e52c42f978d8634c43d06fea928f2822e465aa

CTX 6576e9bf419384506cc3ce3c54ac1a6f67dc66f3b30191e698380bc999b05abc

CTX e19dc0c6dcc2dd001ec535ba18deb2df1a101023108318c75dc98611a09dc48a

CTX 0acdec676fabdf222f07e026f059b672b56e5cbc8e1d21bbd867dd9272120546

CTX 81d70ea737134cdfce93b6f82ae22423274e58a0821cc5502e2d0ab4585e94de

CTX 6975be5e0b4efce51cd3e70c25a1fbbbd609d273ad5b0d59631c531f6a0a57b9

TAG 0664e417f6740411cc10c55d6a8c6f43b7ad21f95f0f6b4751b6049990d13136

TAG 8fef3f1b42e172fbbec6b8133fdcbb8dccf3fed9c345818dc0ae11ace07e0c43