ICS 35.030
CCS L 80

中华人民共和国密码行业标准

GM/T 0001.4—2024

祖冲之序列密码算法
第4部分:鉴别式加密机制

ZUC stream cipher algorithm—
Part 4:Authenticated encryption mechanisms

2024-12-27 发布　　2025-07-01 实施

国家密码管理局　　发布

目　　次

前　言

本文件按照 GB/T 1.1—2020《标准化工作导则　第 1 部分：标准化文件的结构和起草规则》的规定起草。

本文件是 GM/T 0001《祖冲之序列密码算法》的第四部分。GM/T 0001 已经发布了以下部分：

——第 1 部分：算法描述；

——第 2 部分：基于祖冲之算法的机密性算法；

——第 3 部分：基于祖冲之算法的完整性算法；

——第 4 部分：鉴别式加密机制。

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专利的责任。

本文件由密码行业标准化技术委员会提出并归口。

本文件起草单位：兴唐通信科技有限公司、中国科学院软件研究所、大唐移动通信设备有限公司、中国信息通信研究院、北京海泰方圆科技股份有限公司。

本文件主要起草人：李鸿利、王勇慧、贾文义、马永彪、张斌、冯程、薛跳跳、白亮、徐晖、王海梅、罗影。

引　言

GM/T 0001《祖冲之序列密码算法》旨在确立祖冲之算法的使用规范，拟由四个部分组成。

——第1部分：算法描述。目的在于指导祖冲之算法相关产品的研制、检测和使用。

——第2部分：基于祖冲之算法的机密性算法。目的在于指导基于祖冲之算法的机密性算法的相关产品的研制、检测和使用。

——第3部分：基于祖冲之算法的完整性算法。目的在于指导基于祖冲之算法的完整性算法的相关产品的研制、检测和使用。

——第4部分：鉴别式加密机制。目的在于指导使用祖冲之算法对数据进行机密性、完整性保护及数据源鉴别。

本部分规定了祖冲之密码算法的鉴别式加密机制，包括ZUC-GXM和ZUC-MUR两种机制。

ZUC-GXM为依赖初始向量不重复使用的鉴别式加密机制，应确保初始向量不重复使用。ZUC-MUR为可支持初始向量重复使用的鉴别式加密机制，初始向量可重复使用，可用于无法保证初始向量不重复应用场景下的信息机密性和完整性保护，如群组通信、并发加密、确定性加密等。

祖冲之序列密码算法
第4部分:鉴别式加密机制

1 范围

本文件规定了祖冲之密码算法的两种鉴别式加密机制:ZUC-GXM 和 ZUC-MUR。

本文件适用于指导使用祖冲之密码算法对数据进行机密性、完整性保护及数据源鉴别。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中,注日期的引用文件,仅该日期对应的版本适用于本文件;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 25069 信息安全技术 术语

GB/T 33133.1 信息安全技术 祖冲之序列密码算法 第1部分:算法描述

3 术语和定义

GB/T 25069 界定的以及下列术语和定义适用于本文件。

3.1

认证性 authenticity

一个实体是其所声称实体的性质。

注:又称真实性。

[GB/T 25069—2022,3.769,有修改]

3.2

机密性 confidentiality

采用密码技术保证信息不泄露的性质。

注:又称保密性。

[GB/T 25069—2022,3.259,有修改]

3.3

鉴别式加密 authenticated encryption

一种可逆的数据转换,这种数据转换利用加密算法产生数据的对应密文,未经授权实体无法在不被发现的情况下对其修改,即提供了数据机密性、数据完整性与数据源鉴别。

[GB/T 25069—2022,3.298]

3.4

泛杂凑函数 universal hash function

由密钥确立的映射,将一定范围内任意长比特串映射到定长比特串,满足:对于所有不同的输入,其输出在密钥均匀随机的前提下发生碰撞的概率极小。

3.5

密钥派生算法 key derivation function;KDF

通过作用于某一秘密值和其他参数,产生一个或多个密钥的算法。

4 符号

下列符号和缩略语适用于本文件。

A	附属数据，只进行完整性保护而不进行机密性保护的数据
C	密文
Conv	转换函数
Encode	编码函数
GHASH	基于有限域多项式运算的泛杂凑函数
H	泛杂凑函数的密钥，长度为 128 比特
IV	初始向量，长度与所用 ZUC 算法初始向量相同
K	ZUC-GXM 的工作密钥，长度与所用 ZUC 算法工作密钥相同
K_1	ZUC-MUR 的第一个工作密钥，长度与所用 ZUC 算法工作密钥相同
K_2	ZUC-MUR 的第二个工作密钥，长度与所用 ZUC 算法工作密钥相同
Klen	工作密钥的比特长度
L_X	|X|的 2 进制表示，长度为 64 比特，如|X|＝22 时，$L_X=0^{59}\|\|10110$
$LSB_n(X)$	比特串 X 的最右边 n 比特
$MSB_n(X)$	比特串 X 的最左边 n 比特
P	明文
Partition(X)	X 的 128 比特分组表示，若最后一个分组小于 128 比特，使用比特 0 填充右边至 128 比特，若 X 的长度为 128 比特的整数倍时(X 可为空串)，则无需填充。即 Partition(X)＝X||0^s，s＝128×⌈|X|/128⌉－|X|
Tag	消息鉴别码
v	初始向量的比特长度
|X|	比特串 X 的比特长度
⌈X⌉	不小于 X 的最小整数
ZUC	祖冲之序列密码算法或祖冲之算法
ZUC-GXM	ZUC 算法的一种鉴别式加密机制，要求 IV 不重复使用
$ZUC_L(IV,K)$	以 IV 为初始向量，以 K 为祖冲之算法的工作密钥，L 为输出密钥流的比特长度，按照文件 GB/T 33133.1 给出的方法运行 ZUC 算法产生⌈L/32⌉个 32 比特字的密钥流，并从中截取最左边 L 比特作为密钥流输出
ZUC-MUR	ZUC 算法的一种鉴别式加密机制，IV 可重复使用
τ	消息鉴别码的比特长度
≪n	左移 n 位
⊕	异或运算
⊗	有限域上的乘法
⊥	表示解密失败的返回值
||	字符串连接符
0^n	n 个“0”组成的比特串
$\{0,1\}^n$	所有 n 比特串组成的集合

5 基本运算及辅助函数

5.1 概述

在本文件定义的鉴别式加密机制 ZUC-GXM 和 ZUC-MUR 中，涉及的辅助函数包括：编码函数

Encode、泛杂凑函数 GHASH，以及转换函数 $Conv_v$，其中 GHASH 中涉及有限域 $GF(2^{128})$中的运算。

5.2 有限域运算

泛杂凑函数 GHASH 使用的运算为有限域 $GF(2^{128})=GF(2)[x]/f(x)$中的运算，其中 $f(x)=x^{128}+x^7+x^2+x+1$。将 $GF(2^{128})$中的元素表示为 GF(2)上次数小于 128 的多项式，则$\{0,1\}^{128}$中的比特串 $a=a_0a_1\cdots a_{126}a_{127}$和 $GF(2^{128})$中的多项式 $a(x)=a_{127}x^{127}+a_{126}x^{126}+\cdots+a_1x+a_0$ 对应。

$GF(2^{128})$中的两个多项式“⊗”为 GF(2)上多项式相乘后模 f(x)。具体地，令 X 和 Y 是两个 128 位的比特串，则 128 位的比特串 $Z=X\otimes Y$ 可以如下计算：

a) 令 X 为比特串 $x_0x_1\cdots x_{127}$；

b) 令 $Z=0^{128}$，U=Y；

c) 对于 i=0，1，…，127，执行以下两个步骤：

1) 若 $x_i=1$，则令 $Z=Z\oplus U$；

2) 若 $LSB_1(U)=0$，则令 $U=U\gg1$；否则令 $U=(U\gg1)\oplus(11100001||0^{120})$。

5.3 编码函数 Encode

Encode 的输入为长度小于 2^{64}比特的附属数据 A 和长度小于 2^{64}比特的数据 X，Encode(A，X)的计算步骤为：

a) 计算 $A'=Partition(A)$，$X'=Partition(X)$；

b) 计算 $Encode(A,X)=A'||X'||L_A||L_X$。

5.4 泛杂凑函数 GHASH

GHASH 的输入为 128 比特的密钥 H 和比特长度为 128 整数倍的数据 X，输出为数据 Y，$Y=GHASH_H(X)$的计算步骤如下：

a) 将 X 分为 128 比特的块 $X=X_1||X_2||\cdots||X_t$；

b) 令 $Y=0^{128}$；

c) 对 j=1，2，…，t，计算 $Y=H\otimes(Y\oplus X_j)$。

5.5 转换函数 Conv

函数 $Y=Conv_v(X)$为将数据 X 转化为 v 比特长的数据 Y 的运算，即：

$$Y=\begin{cases}X||0^{v-|X|}，若|X|\leqslant v\\MSB_v(X)，若|X|>v\end{cases}$$

6 ZUC-GXM

6.1 通用要求

ZUC-GXM 包括鉴别式加密算法 ZUC-GXM_E 和鉴别式解密算法 ZUC-GXM_D，初始向量不应重复。

ZUC-GXM_E 的输入为 IV、H、K、A、P、τ，输出为密文 C 和 τ 比特鉴别码 Tag。

ZUC-GXM_D 的输入为 IV、H、K、A、C、Tag、τ，输出为明文 P 或⊥。

注 1：如需减少密钥存储量，H、K 能通过密钥派生算法计算得到，密钥派生算法的相关示例见附录 A 的 A.2。

注 2：ZUC-GXM 的性质和使用指导见附录 B 的 B.2，ZUC-GXM 的示例见附录 C 的 C.2。

6.2 鉴别式加密算法

ZUC-GXM_E(IV,H,K,A,P,τ)的计算步骤如下：

a) 产生密钥流 Z：运行 ZUC_L(IV,K)产生密钥流 Z，$Z=Z_0||Z_1$，其中 $L=|P|+\tau'$，$\tau'=32\times\lceil\tau/32\rceil$，$Z_0$ 为 τ'比特，Z_1 为|P|比特；

b) 计算密文 C：$C=P\oplus Z_1$；

c) 计算 X=Encode(A,C)；

d) 计算 $Y=GHASH_H(X)$；

e) 计算鉴别码 Tag：$Tag=MSB_\tau(Z_0)\oplus MSB_\tau(Y)$；

f) 输出(C,Tag)。

6.3 鉴别式加密算法流程

鉴别式加密算法流程见图 1。

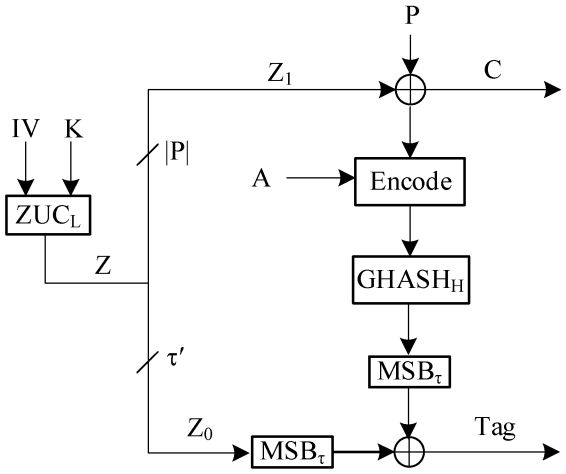

图 1 ZUC-GXM 鉴别式加密算法流程

6.4 鉴别式解密算法

ZUC-GXM_D(IV,H,K,A,C,Tag,τ)的计算步骤如下：

a) 产生密钥流 Z：运行 ZUC_L(IV,K)产生密钥流 Z，$Z=Z_0||Z_1$，其中 $\tau'=32\times\lceil\tau/32\rceil$，$L=|P|+\tau'$，$Z_0$ 为 τ'比特，Z_1 为|C|比特；

b) 计算 X=Encode(A,C)；

c) 计算 $Y=GHASH_H(X)$；

d) 计算鉴别码 Tag′：$Tag'=MSB_\tau(Z_0)\oplus MSB_\tau(Y)$；

e) 若 Tag′≠Tag，报错并退出，返回⊥；否则输出明文 P：$P=C\oplus Z_1$。

6.5 鉴别式解密算法流程

鉴别式解密算法流程见图 2。

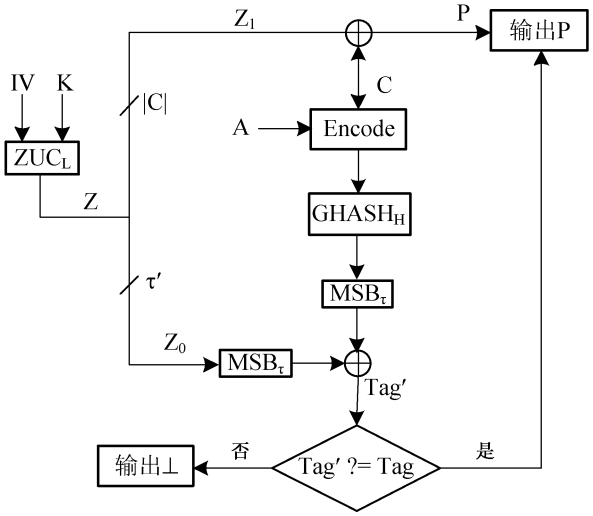

图 2　ZUC-GXM 鉴别式解密算法流程

7　ZUC-MUR

7.1　通用要求

ZUC-MUR 包括鉴别式加密算法 ZUC-MUR_E 和鉴别式解密算法 ZUC-MUR_D。

ZUC-MUR_E 的输入为 IV、H、K_1、K_2、A、P、τ，输出为密文 C 和 τ 比特鉴别码 Tag。

ZUC-MUR_D 的输入为 IV、H、K_1、K_2、A、C、Tag、τ，输出为明文 P 或⊥。

注 1：如需减少密钥存储量，密钥 H、K_1、K_2 能通过密钥派生算法计算得到，密钥派生算法的相关示例见 A.3。

注 2：ZUC-MUR 的性质和使用指导见 B.3，ZUC-MUR 的示例见 C.3。

7.2　鉴别式加密算法

ZUC-MUR_E(IV，H，K_1，K_2，A，P，τ) 的计算步骤如下：

a）　计算 X＝Encode(A，P)；

b）　计算 Y＝$GHASH_H$(X)；

c）　计算鉴别码 Tag：Tag＝$ZUC_\tau(Conv_v(Y)\oplus IV, K_2)$；

d）　产生密钥流 Z：Z＝$ZUC_{|P|}(Conv_v(Tag)\oplus IV, K_1)$；

e）　计算密文 C：C＝P ⊕Z；

f）　输出(C，Tag)。

7.3　鉴别式加密算法流程

鉴别式加密算法流程见图 3。

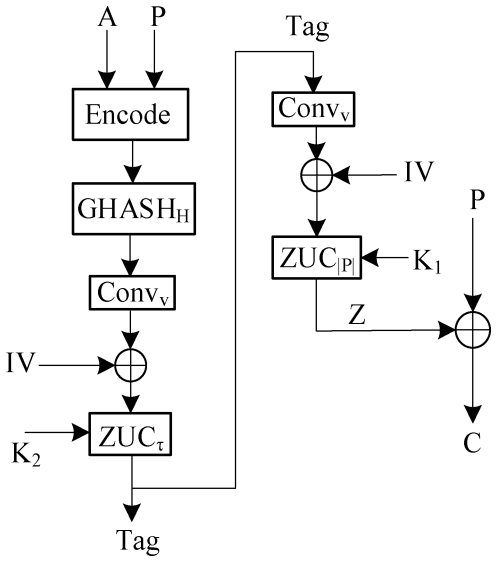

图 3　ZUC-MUR 鉴别式加密算法流程

7.4　鉴别式解密算法

ZUC-MUR_D(IV,H,K_1,K_2,A,C,Tag,τ)的计算步骤如下：

a)　产生密钥流 Z：$Z=ZUC_{|C|}(Conv_v(Tag)\oplus IV,K_1)$；

b)　计算明文 P：$P=C\oplus Z$；

c)　计算 X＝Encode(A,P)；

d)　计算 $Y=GHASH_H(X)$；

e)　计算鉴别码 Tag′：$Tag'=ZUC_\tau(Conv_v(Y)\oplus IV,K_2)$；

f)　若 Tag′≠Tag，则报错并退出，返回⊥；否则输出明文 P。

7.5　鉴别式解密流算法流程

鉴别式解密算法流程见图 4。

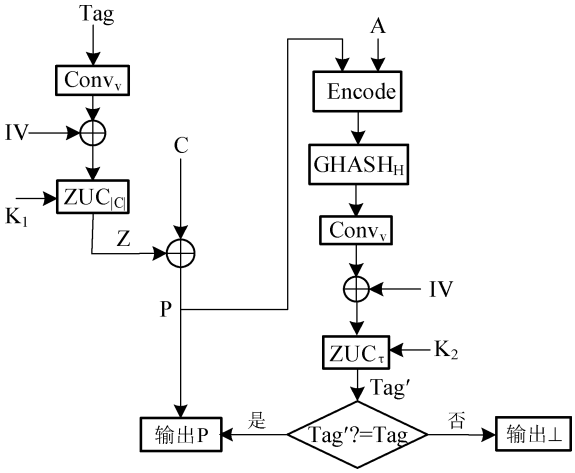

图 4　ZUC-MUR 鉴别式解密算法流程

附 录 A
（资料性）
密钥派生算法

A.1 概述

ZUC-GXM 需要两个相互独立的随机密钥，ZUC-MUR 需要三个相互独立的随机密钥，为了减少密钥存储量，本附录给出 ZUC-GXM 的密钥派生算法 KDF1，以及 ZUC-MUR 的密钥派生算法 KDF2 的示例作为参考。

A.2 密钥派生算法 1

密钥派生算法 KDF1(IV_0，K_0)，以 v 比特初始向量 IV_0（IV_0 可取任意 v 比特数据，未特别约定时取值为 $IV_0=0^v$）和 klen 比特初始密钥 K_0 为输入，运行 ZUC 算法产生 128 比特密钥 H 和 klen 比特密钥 K，即 $H||K=KDF1(IV_0,K_0)=ZUC_{128+klen}(IV_0,K_0)$。密钥派生算法 1 流程见图 A.1。

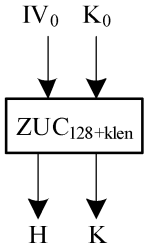

图 A.1 密钥派生算法 1 流程

A.3 密钥派生算法 2

密钥派生算法 KDF2(IV_0，K_0)，以 v 比特初始向量 IV_0（IV_0 可取任意 v 比特数据，未特别约定时取 $IV_0=0^v$）和 klen 比特初始密钥 K_0 为输入，运行 ZUC 算法产生 128 比特密钥 H 和两个 klen 比特密钥 K_1、K_2，即 $H||K_1||K_2=KDF2(IV_0,K_0)=ZUC_{128+2klen}(IV_0,K_0)$。密钥派生算法 2 流程见图 A.2。

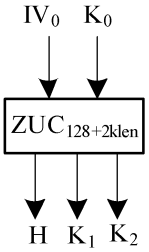

图 A.2 密钥派生算法 2 流程

附 录 B
（资料性）
性质和使用指导

B.1 概述

本附录给出两种鉴别式加密机制 ZUC-GXM 和 ZUC-MUR 的性质和使用指导。

B.2 ZUC-GXM

B.2.1 性质：

a) 在 ZUC 算法的理想安全假设下，使用不同初始向量 IV 的 ZUC-GXM 密文具有独立随机性，同时满足存在性不可伪造。
b) 支持任意长度的明文和附属数据，不需要填充。
c) 运算过程只调用 ZUC 算法一次，即只执行一次算法初始化操作。
d) 解密时先验证鉴别码的正确性，如果验证失败，无需执行解密操作。

B.2.2 使用指导：

a) 对于同一密钥，要求加密方对每条消息产生新的初始向量 IV。
b) ZUC-GXM 机制可用于数据的传输保护，可对数据同时提供机密性和完整性保护功能。
c) 解密时如果鉴别码验证失败，不应输出明文。
d) 建议鉴别码输出长度不小于 64 比特。

B.3 ZUC-MUR

B.3.1 性质：

a) 在 ZUC 算法的理想安全假设下，不同的三元组(IV，A，P)对应的 ZUC-MUR 密文具有独立随机性，同时满足存在性不可伪造；相同的三元组(IV，A，P)对应的 ZUC-MUR 密文和鉴别码相同。
b) 支持任意长度的明文和附属数据，不需要填充。
c) 运算过程需要调用 ZUC 算法两次，即执行两次算法初始化操作。
d) 解密时需要先解密出明文，然后才能验证鉴别码的正确性。

B.3.2 使用指导：

a) ZUC-MUR 机制可用于数据的传输保护，可对数据同时提供机密性和完整性保护功能。
b) ZUC-MUR 支持初始向量的重用，可用于不便于频繁更换初始向量的应用场景。
c) 解密时如果鉴别码验证失败，不应输出明文。
d) 建议鉴别码输出长度不小于 64 比特。

附 录 C
（资料性）
数据示例

C.1 概述

本附录使用 GB/T 33133.1 规定的 ZUC 算法，Tag 长度取 64 比特或 128 比特，给出 ZUC-GXM 和 ZUC-MUR 的具体运算数据示例，数据示例以 16 进制数表示。

C.2 ZUC-GXM

C.2.1 示例 1

|Tag|=128
|A|=152
|P|=0

IV： b3a6db3c 870c3e99 245e0d1c 06b747de
H： 6db45e4f 9572f4e6 fe0d91ac da6801d5
K： edbe06af ed807557 6aad04af dec91d32
A： 9de18b1f dab0ca99 02b9729d 492c807e
c599d5
P： 空

Z： 4d2715d4 22f5bd20 fe836e0e ef837d57
X： 9de18b1f dab0ca99 02b9729d 492c807e
c599d500 00000000 00000000 00000000
00000000 00000098 00000000 00000000
Y： 6733ba7a 941051e7 7a79bc43 31372a85

C： 空
Tag：2a14afae b6e5ecc7 84fad24d deb457d2

C.2.2 示例 2

|Tag|=128
|A|=0
|P|=0

IV： 2923be84 e16cd6ae 529049f1 f1bbe9eb
H： 27bede74 018082da 87d4e5b6 9f18bf66

K： 32070e0f 39b7b692 b4673edc 3184a48e
A： 空
P： 空

C： 空
Tag：5d8a045a c89a681a 4bc91038 0bbadccf

C.2.3 示例 3

|Tag|＝128
|A|＝0
|P|＝120

IV： 2d208683 2cc2fe3f d18cb51d 6c5e99a5
H： 9d6cb516 23fd847f 2e45d7f5 2f900db8
K： 56131c03 e457f622 6b547763 3b873984
A： 空
P： ffffffff ffffffff ffffffff ffffff

C： b78e2f30 cf70252d 58767997 f1b086
Tag：efb30feb bfe0c88a 1e77b1dd e9d45525

C.2.4 示例 4

|Tag|＝128
|A|＝256
|P|＝376

IV： bb8b76cf e5f0d933 5029008b 2a3b2b21
H： ee767d50 3bb3d5d1 b585f57a 0418c673
K： e4b5c1f8 578034ce 6424f58c 675597ac
A： fcdd4cb9 7995da30 efd95719 4eac4d2a
8610470f 99c88657 f462f68d ff7561a5
P： 5fee5517 627f17b2 2a96caf9 7b77ec7f
667cc47d 13c34923 be244130 0066a6c1
50b24d66 c947ca7b 2e708eb6 2bb352

C： b56da5c9 9238b04a 45e3d9d9 6f12f3dc
052e428f a5a58172 92ee23db dad9782c
f66f55c8 46e55dc6 8f47eaf8 378e70
Tag：51c7aedd 9e1c7d74 c38059f5 e7e3a742

C.2.5 示例 5

```
------------------------------------------------------------
|Tag|=64
|A|=384
|P|=256
------------------------------------------------------------
IV:  3615df81  0cc677f1  5080faa1  dd44aad3
H:   fdfaddc4  76785c25  906fe42b  a63a93b7
K:   f405d652  b6362e70  f8362bd3  83b7298b
A:   5fee5517  627f17b2  2a96caf9  7b77ec7f
     667cc47d  13c34923  be244130  0066a6c1
     50b24d66  c947ca7b  2e708eb6  2bb352fc
P:   dd4cb979  95da30ef  d957194e  ac4d2a86
     10470f99  c88657f4  62f68dff  7561a5f3

C:   1134ffc1  19ad163e  91498947  4be6c072
     fd5867f3  989d8b15  899ebd10  a4a248c9
Tag: 8829aaa4  f9891822
```

C.3 ZUC-MUR

C.3.1 示例 1

```
------------------------------------------------------------
|Tag|=128
|A|=256
|P|=376
------------------------------------------------------------
IV:  bb8b76cf  e5f0d933  5029008b  2a3b2b21
H:   ee767d50  3bb3d5d1  b585f57a  0418c673
K1:  e4b5c1f8  578034ce  6424f58c  675597ac
K2:  608053f6  af9efda5  62d95dc0  13bea6b5
A:   fcdd4cb9  7995da30  efd95719  4eac4d2a
     8610470f  99c88657  f462f68d  ff7561a5
P:   5fee5517  627f17b2  2a96caf9  7b77ec7f
     667cc47d  13c34923  be244130  0066a6c1
     50b24d66  c947ca7b  2e708eb6  2bb352

X:   fcdd4cb9  7995da30  efd95719  4eac4d2a
     8610470f  99c88657  f462f68d  ff7561a5
     5fee5517  627f17b2  2a96caf9  7b77ec7f
     667cc47d  13c34923  be244130  0066a6c1
     50b24d66  c947ca7b  2e708eb6  2bb35200
```

```
        00000000  00000100 00000000  00000178
Y:      e42ce8e1  542db606 2d097d4f  134d6131
Z:      90bbc1aa  52bfcdbd 9e896cfc  3524a17b
        f2b512b9  e2f1b5a6 c93e0604  58d63342
        e8978b04  769fe803 39fa0ae8  03ad0669
C:      cf5594bd  30c0da0f b41fa605  4e534d04
        94c9d6c4  f132fc85 771a4734  58b09583
        b825c662  bfd82278 178a845e  281e54
Tag:    15c5d1a7  8a42c4dc d67db05f  a1a640a0
```

C.3.2 示例 2

```
------------------------------------------------------------
|Tag|=128
|A|=0
|P|=0
------------------------------------------------------------
IV:     2923be84  e16cd6ae 529049f1  f1bbe9eb
H:      27bede74  018082da 87d4e5b6  9f18bf66
K1:     32070e0f  39b7b692 b4673edc  3184a48e
K2:     27636f44  14510d62 cc15cfe1  94ec4f6d
A:      空
P:      空

C:      空
Tag:    c0016e07  72c9983d 0fd9fd8c  1b012845
```

C.3.3 示例 3

```
------------------------------------------------------------
|Tag|=128
|A|=0
|P|=120
------------------------------------------------------------
IV:     2d208683  2cc2fe3f  d18cb51d  6c5e99a5
H:      9d6cb516  23fd847f  2e45d7f5  2f900db8
K1:     56131c03  e457f622  6b547763  3b873984
K2:     a8898153  4db331a3 86de3e52  fb46029b
A:      空
P:      ffffffff  ffffffff  ffffffff  ffffff

C:      234c2d51  eaa582da  9be3cc38  28aa67
Tag:    0a7afb7d  817efa07  77826f1e  33a53cf3
```

C.3.4 示例 4

```
------------------------------------------------------------
|Tag|=128
|A|=152
|P|=0
------------------------------------------------------------
IV:  b3a6db3c 870c3e99 245e0d1c 06b747de
H:   6db45e4f 9572f4e6 fe0d91ac da6801d5
K1:  edbe06af ed807557 6aad04af dec91d32
K2:  61d4fca6 b2c2bb48 b4b11725 31333620
A:   9de18b1f dab0ca99 02b9729d 492c807e
     c599d5
P:   空

C:   空
Tag: 8213c296 06d02bba 10f13ffa d1d26a42
```

C.3.5 示例 5

```
------------------------------------------------------------
|Tag|=64
|A|=384
|P|=256
------------------------------------------------------------
IV:  b3a6db3c 870c3e99 245e0d1c 06b747de
H:   6db45e4f 9572f4e6 fe0d91ac da6801d5
K1:  edbe06af ed807557 6aad04af dec91d32
K2:  61d4fca6 b2c2bb48 b4b11725 31333620
A:   9de18b1f dab0ca99 02b9729d 492c807e
     c599d5e9 80b2eac9 cc53bf67 d6bf14d6
     7e2ddc8e 6683ef57 4961ff69 8f61cdd1
P:   b3124dc8 43bb8ba6 1f035a7d 0938251f
     5dd4cbfc 96f5453b 130d890a 1cdbae32

C:   dabbbe23 d8f0ea42 e31a9bdd 9706a427
     5d8aacd2 cf27c4a4 c0d0ba6f b8f31da7
Tag: a276827b 74509357
```

中华人民共和国密码
行业标准
祖冲之序列密码算法
第4部分:鉴别式加密机制
GM/T 0001.4—2024
*
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)
网址 www.spc.net.cn
总编室:(010)68533533 发行中心:(010)51780238
读者服务部:(010)68523946
中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销
*
开本 880×1230 1/16 印张 1.25 字数 26 千字
2025年6月第1版 2025年6月第1次印刷
*
书号:155066·2-39085 定价 38.00 元

GM/T 0001.4-2024