SM9 标识密码算法
第 4 部分：密钥封装机制和公钥加密算法

目　　次

SM9 标识密码算法

第 4 部分：密钥封装机制和公钥加密算法

本部分规定了用椭圆曲线对实现的基于标识的密钥封装机制和公钥加密与解密算法，并提供相应的流程。利用密钥封装机制可以封装密钥给特定的实体。公钥加密与解密算法即基于标识的非对称密码算法，该算法使消息发送者可以利用接收者的标识对消息进行加密，唯有接收者可以用相应的私钥对该密文进行解密，从而获取消息。

1 术语

1.1

秘密密钥 secret key
在密码体制中收发双方共同拥有的、而第三方不知道的一种密钥。

1.2

消息 message
任意有限长度的比特串。

1.3

明文 plaintext
未加密的信息。

1.4

密文 ciphertext
经过变换、信息内容被隐藏起来的数据。

1.5

加密 encipherment
为了产生密文，即隐藏数据的信息内容，由密码算法对数据进行(可逆)变换。

1.6

解密 decipherment
加密对应的逆过程。

1.7

密钥派生函数 key derivation function
通过作用于共享秘密和双方都知道的其它参数，产生一个或多个共享秘密密钥的函数。

1.8

消息认证码 message authentication code；MAC

一种认证算法作用于特定的密钥和消息比特串所得出的一段码字，可用来鉴别数据的来源和检验数据的完整性。用于求取消息认证码的函数称作消息认证码函数。

1.9

加密主密钥 encryption master key

处于标识密码密钥分层结构最顶层的密钥，包括加密主私钥和加密主公钥，其中加密主公钥公开，加密主私钥由 KGC 秘密保存。KGC 用加密主私钥和用户的标识生成用户的加密私钥。在标识密码中，加密主私钥一般由 KGC 通过随机数发生器产生，加密主公钥由加密主私钥结合系统参数产生。

1.10

标识 identity

可唯一确定一个实体身份的信息。标识应由实体无法否认的信息组成，如实体的可识别名称、电子邮箱、身份证号、电话号码、街道地址等。

1.11

密钥生成中心 key generation center；KGC

在本部分中，负责选择系统参数、生成加密主密钥并产生用户加密私钥的可信机构。

2 符号

下列符号适用于本部分。

A, B：使用公钥密码系统的两个用户。

cf：椭圆曲线阶相对于 N 的余因子。

cid：用一个字节表示的曲线的识别符，其中 0x10 表示 F_p(素数 $p>2^{191}$) 上常曲线(即非超奇异曲线)，0x11 表示 F_p 上超奇异曲线，0x12 表示 F_p 上常曲线及其扭曲线。

Dec( )：分组解密算法。

de_B：用户B的私钥。

Enc( )：分组加密算法。

e：从 $\mathbb{G}_1\times\mathbb{G}_2$ 到 $\mathbb{G}_T$ 的双线性对。

eid：用一个字节表示的双线性对 *e* 的识别符，其中 0x01 表示 Tate 对，0x02 表示 Weil 对，0x03 表示 Ate 对，0x04 表示 R-ate 对。

$\mathbb{G}_T$：阶为素数N的乘法循环群。

$\mathbb{G}_1$：阶为素数N的加法循环群。

$\mathbb{G}_2$：阶为素数N的加法循环群。

g^u：乘法群 $\mathbb{G}_T$ 中元素 g 的 u 次幂，即 $g^u=\underbrace{g\cdot g\cdot\ldots\cdot g}_{u个}$，$u$ 是正整数。

H_v( )：密码杂凑函数。

H_1( )：由密码杂凑函数派生的密码函数。

hid：在本部分中，用一个字节表示的私钥生成函数识别符，由 KGC 选择并公开。

ID_B：用户B的标识，可以唯一确定用户B的公钥。

$KDF(\ )$：密钥派生函数。

M：待加密的消息。

M'：解密得到的消息。

$MAC(\ )$：消息认证码函数。

N：循环群$\mathbb{G}_1$、$\mathbb{G}_2$和$\mathbb{G}_T$的阶，为大于2^{191}的素数。

$P_{pub\text{-}e}$：加密主公钥。

P_1：群 $\mathbb{G}_1$ 的生成元。

P_2：群 $\mathbb{G}_2$ 的生成元。

ke：加密主私钥。

$<P>$：由元素 P 生成的循环群。

$[u]P$：加法群 $\mathbb{G}_1$、$\mathbb{G}_2$ 中元素 P 的 u 倍。

$x\|y$：x 与 y 的拼接，x 和 y 是比特串或字节串。

$[x, y]$：不小于 x 且不大于 y 的整数的集合。

$\oplus$：长度相等的两个比特串按比特的模2加运算。

β：扭曲线参数。

3 算法参数与辅助函数

3.1 总则

在现代密码系统中，密钥是控制密码变换的重要参数，而且密码的安全性极大地依赖于对密钥的安全保护。密钥封装机制使得封装者可以产生和加密一个秘密密钥给目标用户，而唯有目标用户可以解封装该秘密密钥，并把它作为进一步的会话密钥。

本部分规定了一个用椭圆曲线对实现的基于标识的密钥封装机制。解封装用户持有一个标识和一个相应的加密私钥，该加密私钥由密钥生成中心通过加密主私钥和解封装用户的标识结合产生。封装者利用解封装用户的标识产生并加密一个秘密密钥给对方，解封装用户则用相应的加密私钥解封装该秘密密钥。

本部分还规定了一个用椭圆曲线对实现的基于标识的公钥加密算法。该公钥加密算法是上述密钥封装机制和消息封装机制的结合，消息封装机制包括基于密钥派生函数的序列密码以及结合密钥派生函数的分组密码算法两种类型，该算法可提供消息的机密性。在基于标识的加密算法中，解密用户持有一个标识和一个相应的加密私钥，该加密私钥由密钥生成中心通过加密主私钥和解密用户的标识结合产生。加密用户用解密用户的标识加密数据，解密用户用自身加密私钥解密数据。

3.2 系统参数组

系统参数组包括曲线识别符 cid；椭圆曲线基域 F_q 的参数；椭圆曲线方程参数 a 和 b；扭曲线参数 β(若 cid 的低 4 位为 2)；曲线阶的素因子 N 和相对于 N 的余因子 cf；曲线 $E(F_q)$相对于 N 的嵌入次数 k；$E(F_{q^{d_1}})$ (d_1 整除 k)的 N 阶循环子群 $\mathbb{G}_1$ 的生成元 P_1；$E(F_{q^{d_2}})$ (d_2 整除 k)的 N 阶循环子群 $\mathbb{G}_2$ 的生成元 P_2；双线性对 e 的识别符 eid；（选项）$\mathbb{G}_2$ 到 $\mathbb{G}_1$ 的同态映射 ψ。

双线性对 e 的值域为 N 阶乘法循环群 $\mathbb{G}_T$。

3.3 系统加密主密钥和用户加密密钥的产生

KGC产生随机数$ke \in [1, N-1]$作为加密主私钥，计算$\mathbb{G}_1$中的元素$P_{pub\text{-}e}=[ke]P_1$作为加密主公钥，则加密主密钥对为$(ke, P_{pub\text{-}e})$。KGC秘密保存ke，公开$P_{pub\text{-}e}$。

KGC选择并公开用一个字节表示的加密私钥生成函数识别符hid。

用户B的标识为ID_B，为产生用户B的加密私钥de_B，KGC首先在有限域F_N上计算$t_1=H_1(ID_B||hid,N)+ke$，若$t_1=0$则需重新产生加密主私钥，计算和公开加密主公钥，并更新已有用户的加密私钥；否则计算$t_2 = ke \cdot t_1^{-1}$，然后计算$de_B=[t_2]P_2$。

3.4 辅助函数

3.4.1 概述

在本部分规定的基于标识的密钥封装机制和公钥加密算法中，涉及到五类辅助函数：密码杂凑函数、密钥派生函数、消息认证码函数、随机数发生器和分组密码算法。这五类辅助函数的强弱直接影响密钥封装机制和公钥加密算法的安全性。

3.4.2 密码杂凑函数

3.4.2.1 密码杂凑函数 $H_v(\ )$

密码杂凑函数$H_v(\ )$的输出是长度恰为v比特的杂凑值。本部分规定使用国家密码管理主管部门批准的密码杂凑函数，如SM3密码杂凑算法。

3.4.2.2 密码函数 $H_1(\ )$

密码函数$H_1(Z, n)$的输入为比特串Z和整数n，输出为一个整数$h_1\in[1, n-1]$。 $H_1(Z, n)$需要调用密码杂凑函数$H_v(\ )$。关于密码杂凑函数$H_v(\ )$，应符合本部分5.4.2.1的规定。

密码函数 $H_1(Z, n)$：

输入：比特串 Z，整数 n。

输出：整数$h_1\in[1, n-1]$。

步骤 1：初始化一个 32 比特构成的计数器 ct=0x00000001；

步骤 2：计算 $hlen=8\times\lceil(5\times(\log_2 n))/32\rceil$；

步骤 3：对 i 从 1 到$\lceil hlen/v\rceil$执行：

步骤 3.1：计算 $Ha_i=H_v(0x01||Z||ct)$；

步骤 3.2：ct++；

步骤 4：若 $hlen/v$ 是整数，令 $Ha!_{\lceil hlen/v\rceil}=Ha_{\lceil hlen/v\rceil}$，

否则令 $Ha!_{\lceil hlen/v\rceil}$为 $Ha_{\lceil hlen/v\rceil}$最左边的$(hlen-(v\times\lfloor hlen/v\rfloor))$比特；

步骤 5：令 $Ha = Ha_1 \| Ha_2 \| \cdots \| Ha_{\lceil hlen/v\rceil-1} \| Ha!_{\lceil hlen/v\rceil}$，将 Ha 的数据类型转换为整数；

步骤6：计算$h_1=(Ha \bmod (n\text{-}1))+1$。

3.4.2.3 密码函数 $H_2(\ )$

密码函数$H_2(Z, n)$的输入为比特串Z和整数n，输出为一个整数$h_2\in[1, n-1]$。$H_2(Z, n)$需要调用密码杂凑函数$H_v(\ )$。关于密码杂凑函数$H_v(\ )$，应符合本部分5.4.2.1的规定。

密码函数 $H_2(Z, n)$：

输入：比特串 Z，整数 n。

输出：整数$h_2\in[1, n-1]$。

步骤 1：初始化一个 32 比特构成的计数器 ct=0x00000001；

步骤 2：计算 $hlen=8\times\lceil(5\times(\log_2 n))/32\rceil$；

步骤 3：对 i 从 1 到$\lceil hlen/v\rceil$执行：

步骤 3.1：计算 $Ha_i=H_v(0x02||Z||ct)$；

步骤 3.2：ct++；

步骤 4：若 $hlen/v$ 是整数，令 $Ha!_{\lceil hlen/v\rceil}=Ha_{\lceil hlen/v\rceil}$，

否则令 $Ha!_{\lceil hlen/v\rceil}$为 $Ha_{\lceil hlen/v\rceil}$最左边的($hlen-(v\times\lfloor hlen/v\rfloor)$)比特；

步骤 5：令 $Ha = Ha_1 \| Ha_2 \| \cdots \| Ha_{\lceil hlen/v\rceil-1} \| Ha!_{\lceil hlen/v\rceil}$，将 Ha 的数据类型转换为整数；

步骤6：计算$h_2=(Ha \bmod (n-1))+1$。

3.4.3 密钥派生函数

本部分采用的密钥派生函数$KDF(\ )$。

3.4.4 分组密码算法

分组密码算法包括加密算法$Enc(K_1, m)$和解密算法$Dec(K_1, c)$。$Enc(K_1, m)$表示用密钥K_1对明文m进行加密，其输出为密文比特串c；$Dec(K_1, c)$表示用密钥K_1对密文c进行解密，其输出为明文比特串m或“错误”。密钥K_1的比特长度记为K_1_len。

本部分规定使用国家密码管理主管部门批准的分组密码算法，如SM4分组密码算法。

3.4.5 消息认证码函数

消息认证码函数 $MAC(K_2, Z)$的作用是防止消息数据 Z 被非法篡改，它在密钥 K_2 的控制下，产生消息数据比特串 Z 的认证码，密钥 K_2 的比特长度记为 K_2_len。在本部分的基于标识的加密算法中，消息认证码函数使用密钥派生函数生成的密钥对密文比特串求取消息认证码，从而使解密者可以鉴别消息的来源和检验数据的完整性。

消息认证码函数需要调用密码杂凑函数。

设密码杂凑函数为 $H_v(\ )$，其输出是长度恰为 v 比特的杂凑值。

消息认证码函数 $MAC(K_2, Z)$：

输入：比特串 K_2(比特长度为 K_2_len 的密钥)，比特串 Z(待求取消息认证码的消息)。

输出：长度为 v 的消息认证码数据比特串 K。

步骤1：$K=H_v(Z\|K_2)$。

3.4.6 随机数发生器

本部分规定使用国家密码管理主管部门批准的随机数发生器。

4 密钥封装机制及流程

4.1 密钥封装算法及流程

4.1.1 密钥封装算法

为了封装比特长度为$klen$的密钥给用户B，作为封装者的用户A需要执行以下运算步骤：

A1： 计算群 $\mathbb{G}_1$ 中的元素 $Q_B=[H_1(ID_B\|hid, N)]P_1+P_{pub\text{-}e}$；

A2： 产生随机数 $r\in[1, N-1]$；

A3： 计算群 $\mathbb{G}_1$ 中的元素 $C=[r]Q_B$，将 C 的数据类型转换为比特串；

A4： 计算群 $\mathbb{G}_T$ 中的元素 $g=e(P_{pub\text{-}e}, P_2)$；

A5： 计算群 $\mathbb{G}_T$ 中的元素 $w=g^r$，将 w 的数据类型转换为比特串；

A6： 计算 $K=KDF(C\|w\|ID_B, klen)$，若 K 为全 0 比特串，则返回 A2。

A7： 输出(K, C)，其中 K 是被封装的密钥，C 是封装密文。

4.1.2 密钥封装算法流程

密钥封装算法流程如图1。

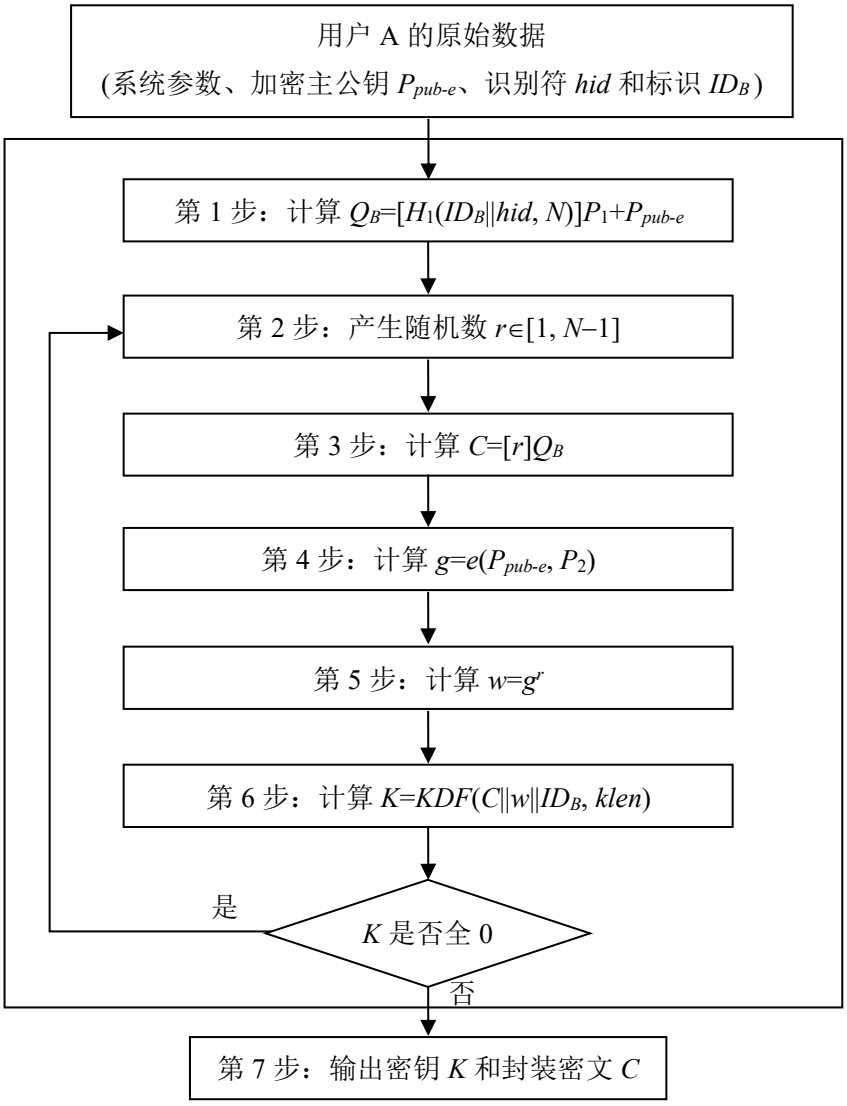

图 1　密钥封装算法流程

4.2　解封装算法及流程

4.2.1　解封装算法

用户B收到封装密文C后，为了对比特长度为$klen$的密钥解封装，需要执行以下运算步骤：

B1：　验证 $C\in\mathbb{G}_1$ 是否成立，若不成立则报错并退出；

B2：　计算群 $\mathbb{G}_T$ 中的元素 $w'=e(C, de_B)$，将 w'的数据类型转换为比特串；

B3：　将 C 的数据类型转换为比特串，计算封装的密钥 $K'=KDF(C||w'||ID_B, klen)$，若 K'为全 0 比特串，则报错并退出；

B4：　输出密钥 K'。

4.2.2　解封装算法流程

解封装算法流程如图2。

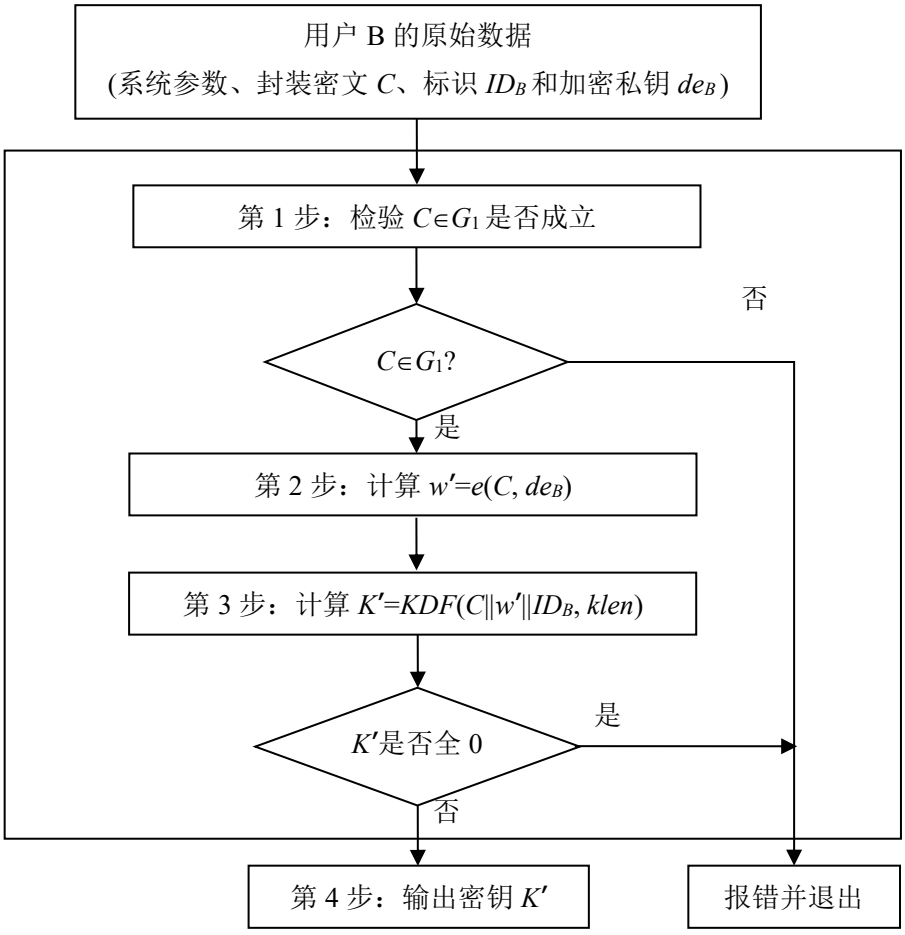

图 2　解封装算法流程

5　公钥加密算法及流程

5.1　加密算法及流程

5.1.1　加密算法

设需要发送的消息为比特串 M，$mlen$ 为 M 的比特长度，K_1_len 为分组密码算法中密钥 K_1 的比特长度，K_2_len 为函数 $MAC(K_2, Z)$中密钥 K_2 的比特长度。

为了加密明文 M 给用户 B，作为加密者的用户 A 应实现以下运算步骤：

A1：　计算群 $\mathbb{G}_1$ 中的元素 $Q_B=[H_1(ID_B\|hid, N)]P_1+P_{pub\text{-}e}$；

A2：　产生随机数 $r \in [1, N-1]$；

A3：　计算群 $\mathbb{G}_1$ 中的元素 $C_1=[r]Q_B$，将 C_1 的数据类型转换为比特串；

A4：　计算群 $\mathbb{G}_T$ 中的元素 $g=e(P_{pub\text{-}e}, P_2)$；

A5：　计算群 $\mathbb{G}_T$ 中的元素 $w=g^r$，按将 w 的数据类型转换为比特串；

A6：　按加密明文的方法分类进行计算：

a)　如果加密明文的方法是基于密钥派生函数的序列密码算法，则

1)　计算整数 $klen=mlen+K_2\_len$，然后计算 $K=KDF(C_1\|w\|ID_B, klen)$。令 K_1 为 K 最左边的 $mlen$ 比特，K_2 为剩下的 K_2_len 比特，若 K_1 为全 0 比特串，则返回 A2；

2) 计算 $C_2=M\oplus K_1$。

b) 如果加密明文的方法是结合密钥派生函数的分组密码算法，则

1) 计算整数 $klen=K_1\_len+K_2\_len$，然后计算 $K=KDF(C_1\|w\|ID_B, klen)$。令 K_1 为 K 最左边的 K_1_len 比特，K_2 为剩下的 K_2_len 比特，若 K_1 为全 0 比特串，则返回 A2；

2) 计算 $C_2=Enc(K_1, M)$。

A7： 计算 $C_3=MAC(K_2, C_2)$；

A8： 输出密文 $C=C_1\|C_3\|C_2$。

5.1.2 加密算法流程

加密算法流程如图3。

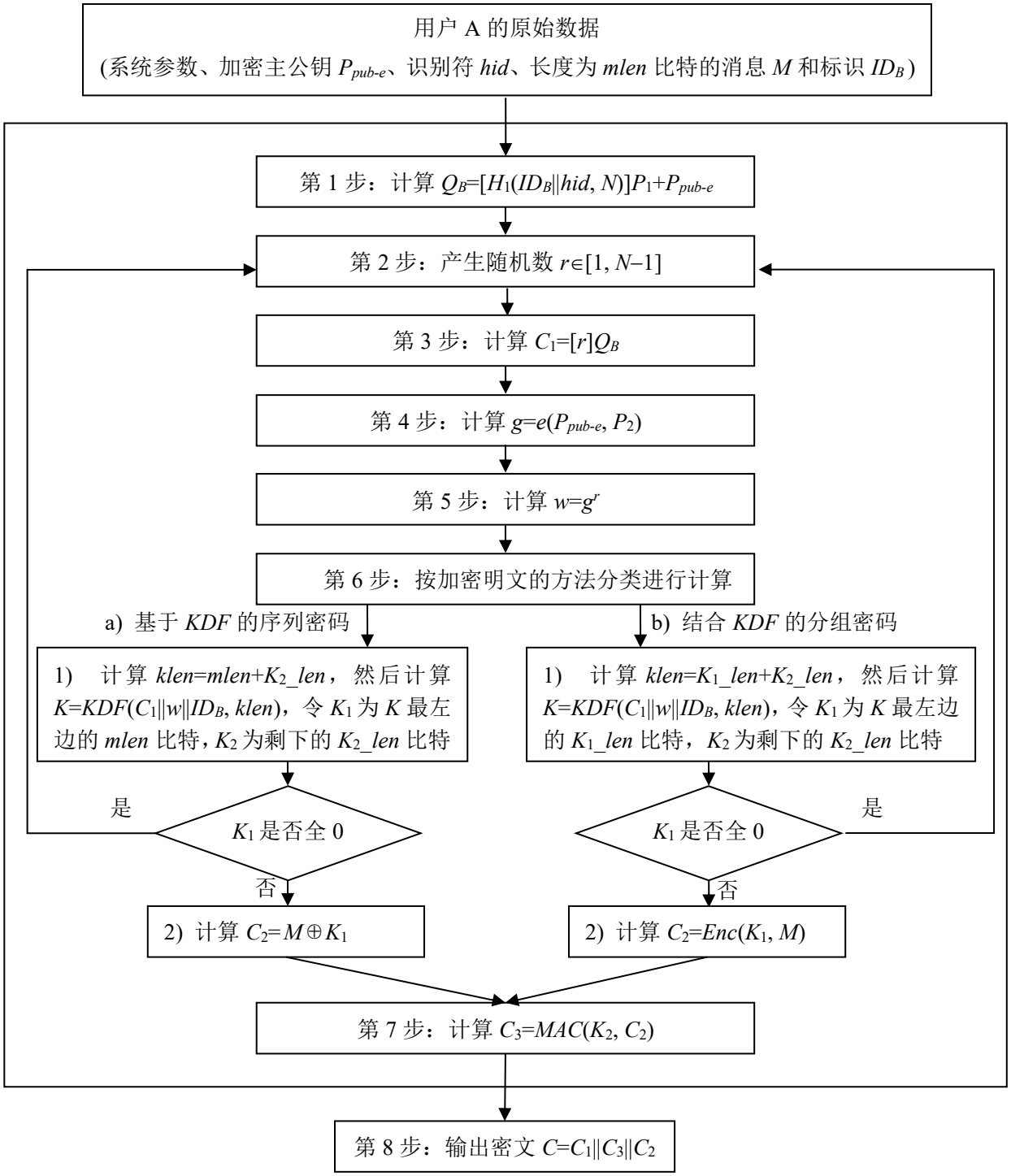

图 3　加密算法流程

5.2　解密算法及流程

5.2.1　解密算法

设 $mlen$ 为密文 $C=C_1\|C_3\|C_2$ 中 C_2 的比特长度，K_1_len 为分组密码算法中密钥 K_1 的比特长度，K_2_len

为函数 $MAC(K_2, Z)$中密钥 K_2 的比特长度。

为了对 C 进行解密，作为解密者的用户 B 应实现以下运算步骤：

B1： 从 C 中取出比特串 C_1，将 C_1 的数据类型转换为椭圆曲线上的点，验证 $C_1 \in G_1$ 是否成立，若不成立则报错并退出；

B2： 计算群 G_T 中的元素 $w'=e(C_1, de_B)$，将 w'的数据类型转换为比特串；

B3： 按加密明文的方法分类进行计算：

a) 如果加密明文的方法是基于密钥派生函数的序列密码算法，则

1) 计算整数 $klen=mlen+K_2\_len$，然后计算 $K'=KDF(C_1\|w'\|ID_B, klen)$。令 K_1'为 K'最左边的 $mlen$ 比特，K_2'为剩下的 K_2_len 比特，若 K_1'为全 0 比特串，则报错并退出；

2) 计算 $M'=C_2 \oplus K_1'$。

b) 如果加密明文的方法是结合密钥派生函数的分组密码算法，则

1) 计算整数 $klen=K_1\_len+K_2\_len$，然后计算 $K'=KDF(C_1\|w'\|ID_B, klen)$。令 K_1'为 K'最左边的 K_1_len 比特，K_2'为剩下的 K_2_len 比特，若 K_1'为全 0 比特串，则报错并退出；

2) 计算 $M'=Dec(K_1', C_2)$。

B4： 计算 $u=MAC(K_2', C_2)$，从 C 中取出比特串 C_3，若 $u \neq C_3$，则报错并退出；

B5： 输出明文 M'。

5.2.2 解密算法流程

解密算法流程如图4。

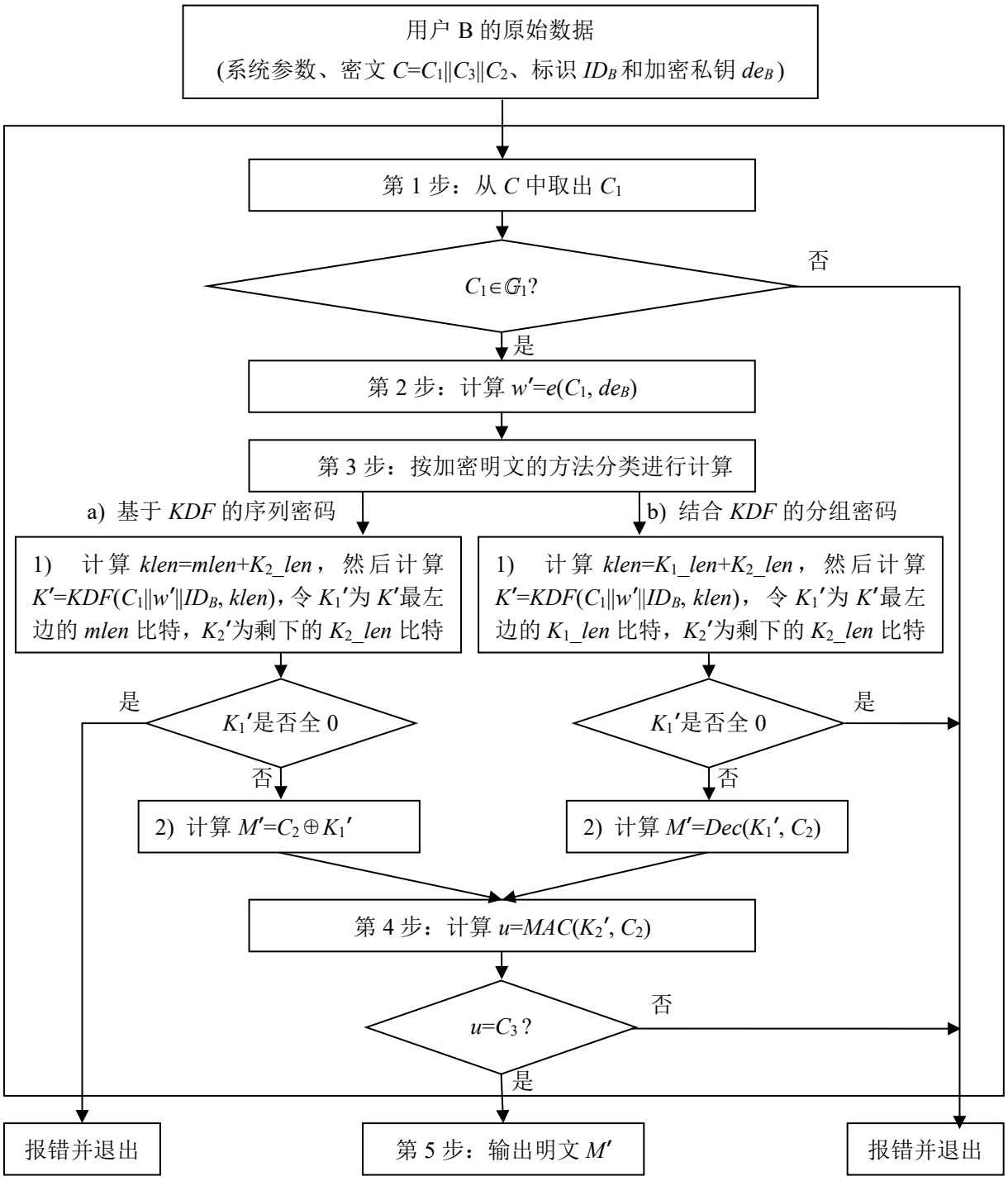
用户 B 的原始数据
(系统参数、密文 $C=C_1\|C_3\|C_2$、标识 ID_B 和加密私钥 de_B)
第 1 步：从 C 中取出 C_1
$C_1\in\mathbb{G}_1$?
否
是
第 2 步：计算 $w'=e(C_1, de_B)$
第 3 步：按加密明文的方法分类进行计算
a) 基于 KDF 的序列密码
b) 结合 KDF 的分组密码
1) 计算 $klen=mlen+K_2\_len$，然后计算 $K'=KDF(C_1\|w'\|ID_B, klen)$，令 K_1'为 K'最左边的 $mlen$ 比特，K_2'为剩下的 K_2_len 比特
1) 计算 $klen=K_1\_len+K_2\_len$，然后计算 $K'=KDF(C_1\|w'\|ID_B, klen)$，令 K_1'为 K'最左边的 K_1_len 比特，K_2'为剩下的 K_2_len 比特
是
K_1'是否全 0
否
2) 计算 $M'=C_2\oplus K_1'$
是
K_1'是否全 0
否
2) 计算 $M'=Dec(K_1', C_2)$
第 4 步：计算 $u=MAC(K_2', C_2)$
$u=C_3$?
否
是
报错并退出
第 5 步：输出明文 M'
报错并退出

图 4　解密算法流程