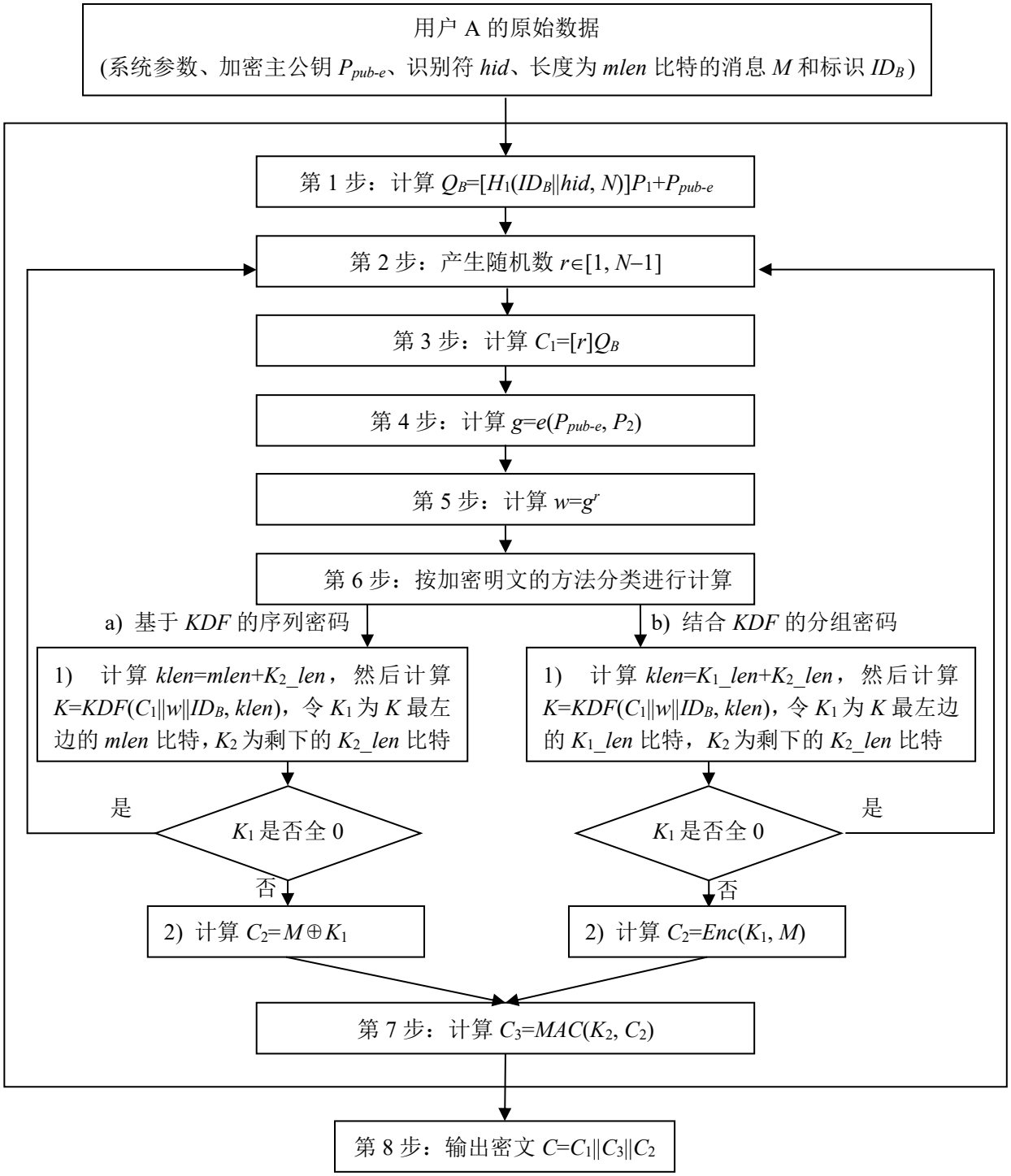

图 3　加密算法流程

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。

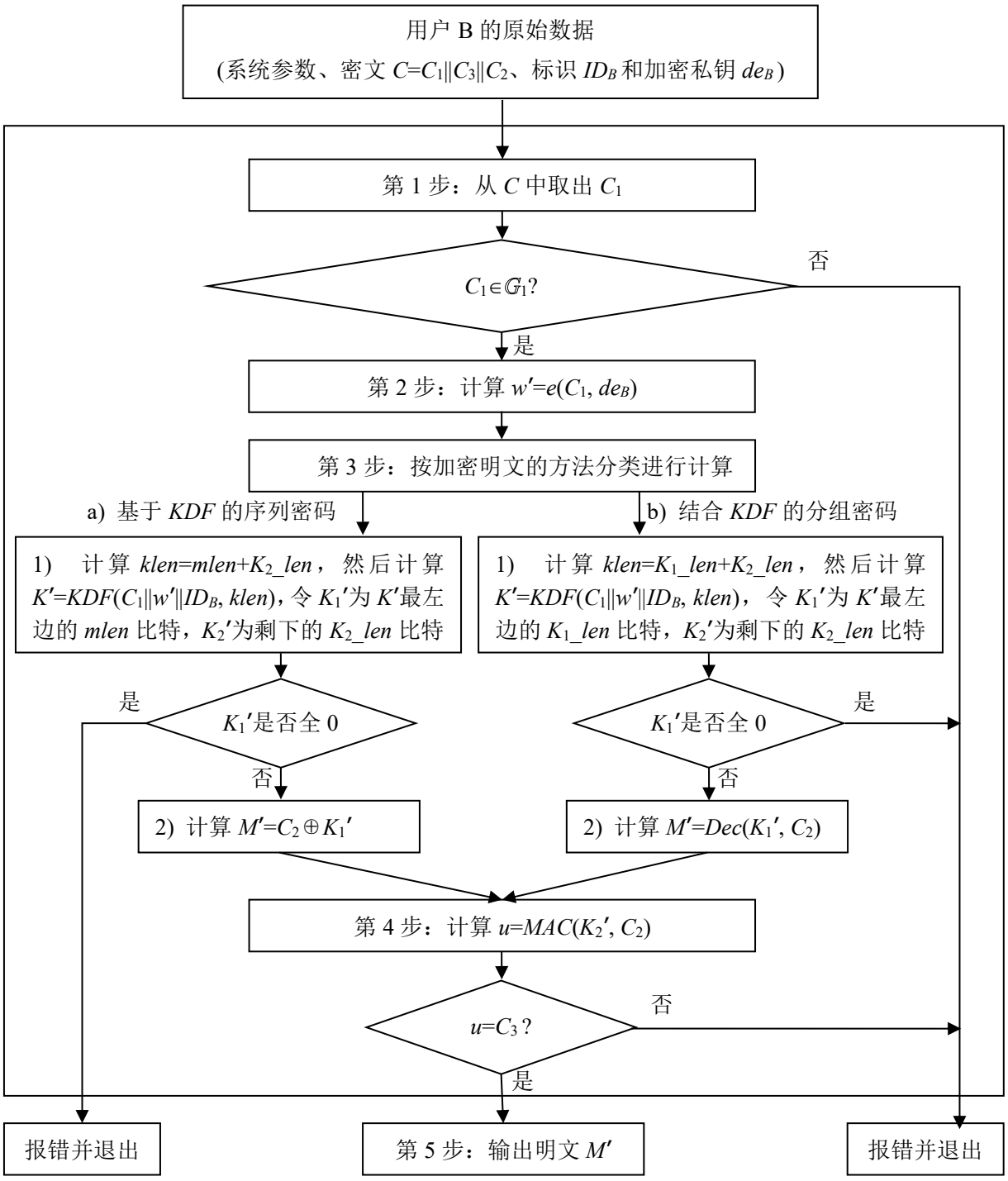
用户 B 的原始数据
(系统参数、密文 $C=C_1\|C_3\|C_2$、标识 ID_B 和加密私钥 de_B)
第 1 步：从 C 中取出 C_1
$C_1\in\mathbb{G}_1$?
否
是
第 2 步：计算 $w'=e(C_1, de_B)$
第 3 步：按加密明文的方法分类进行计算
a) 基于 KDF 的序列密码
b) 结合 KDF 的分组密码
1) 计算 $klen=mlen+K_2\_len$，然后计算 $K'=KDF(C_1\|w'\|ID_B, klen)$，令 K_1'为 K'最左边的 $mlen$ 比特，K_2'为剩下的 K_2_len 比特
1) 计算 $klen=K_1\_len+K_2\_len$，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