密钥封装算法流程如图1。

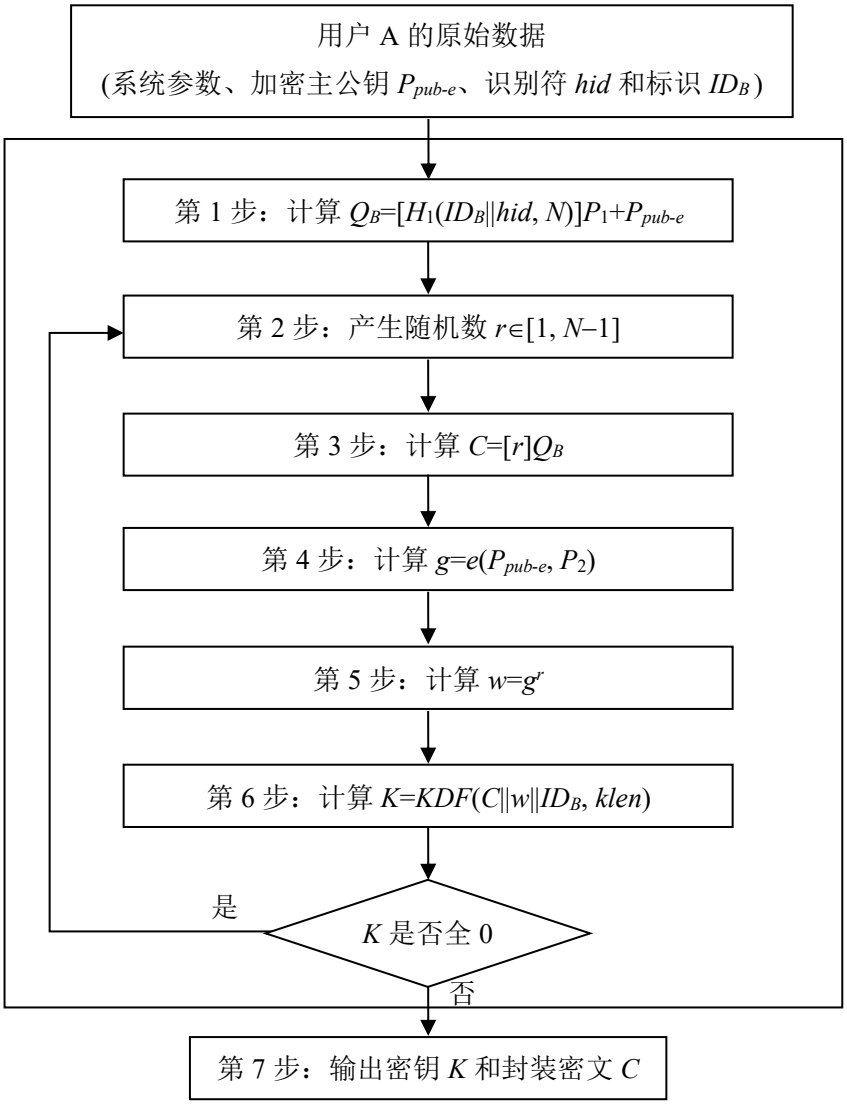

图 1　密钥封装算法流程

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。

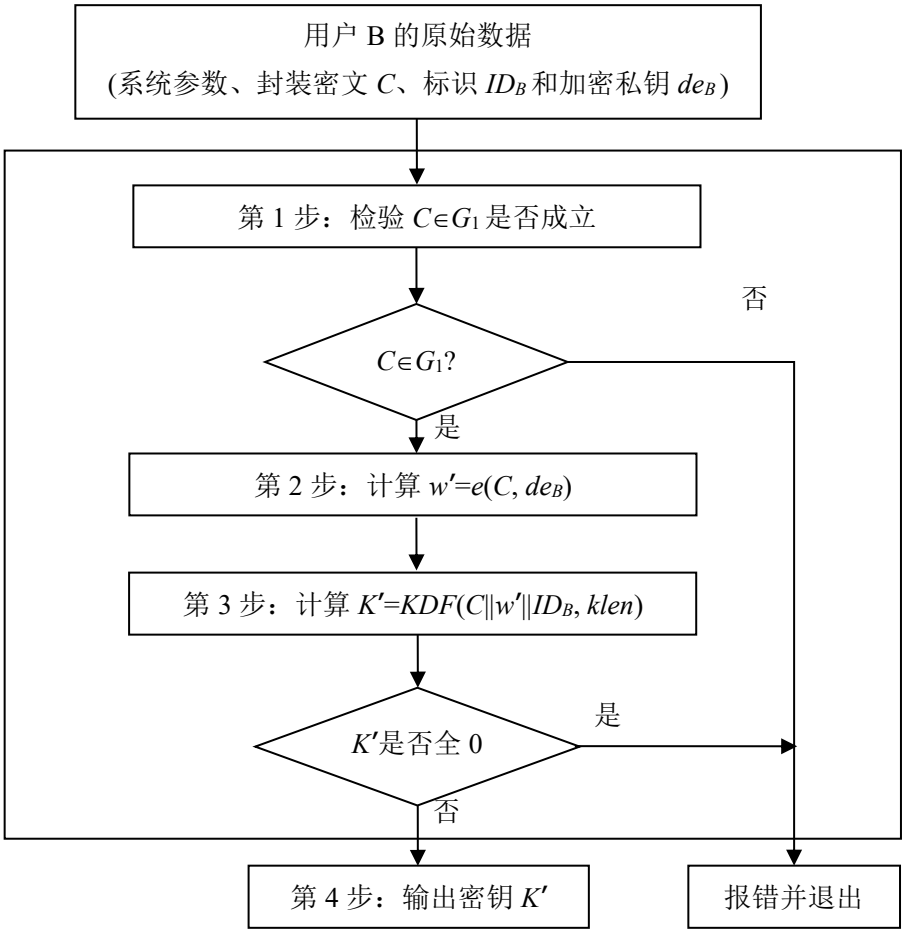

图 2　解封装算法流程

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；