

*Project16: implement sm2 2P decrypt with real network communication

说明:

本项目要求用实际网络通信实现 sm2 解密，示意图如下：

SM2 two-party decrypt

- Public key: $P = [(d_1 d_2)^{-1} - 1]G$
- Private key: $d = (d_1 d_2)^{-1} - 1$



(1) Generate sub private key $d_1 \in [1, n - 1]$,

(2) get ciphertext $C = C_1 || C_2 || C_3$

- Check $C_1 \neq 0$
- Compute $T_1 = d_1^{-1} \cdot C_1$

(4) Recover plaintext M'

- Compute $T_2 - C_1 = (x_2, y_2) = [(d_1 d_2)^{-1} - 1] \cdot C_1 = kP$
- Compute $t = KDF(x_2 || y_2, klen)$
- Compute $M'' = C_2 \oplus t$
- Compute $u = Hash(x_2 || M'' || y_2)$
- If $u = C_3$, output M''

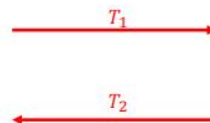
• Encrypt:

- $C_1 = kG = (x_1, y_1)$ where $k \in [1, n - 1]$
- $kP = (x_2, y_2)$
- $t = KDF(x_2 || y_2, klen)$
- $C_2 = M \oplus t$
- $C_3 = H(x_2 || M || y_2)$



(1) Generate sub private key $d_2 \in [1, n - 1]$

(3) compute $T_2 = d_2^{-1} \cdot T_1$,



用户 A 的加密如下：

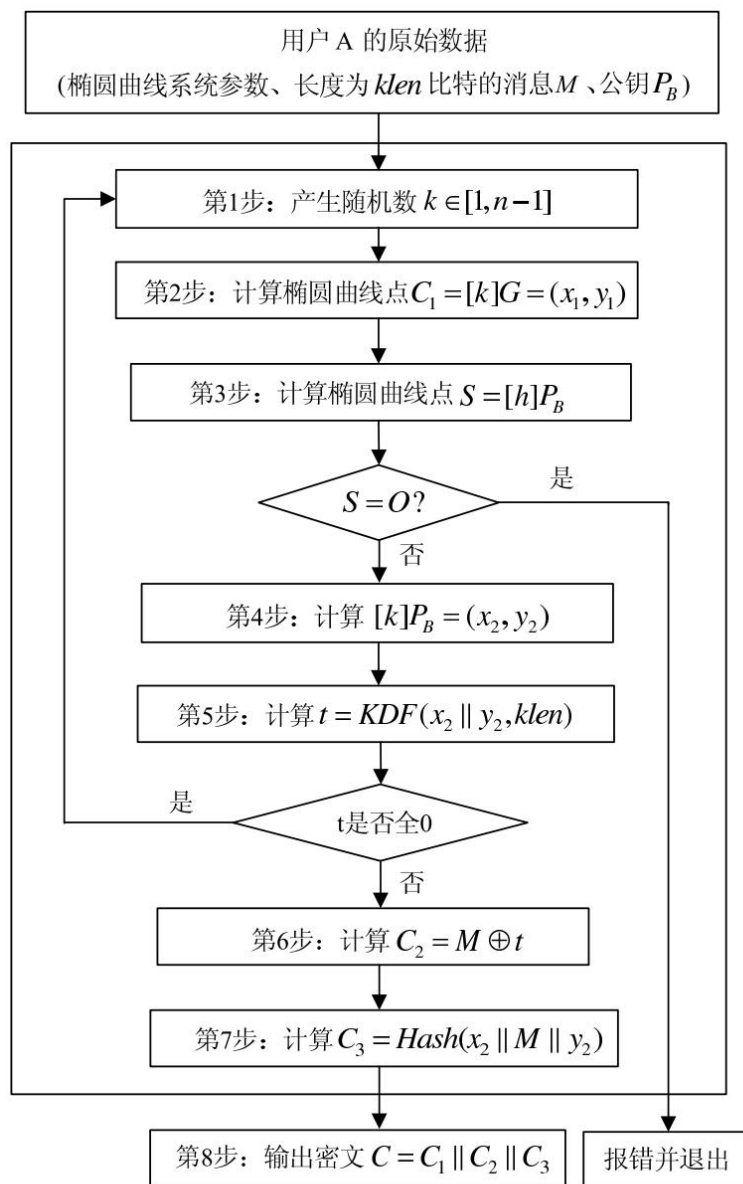


图1 加密算法流程

用户 B 的解密如下：

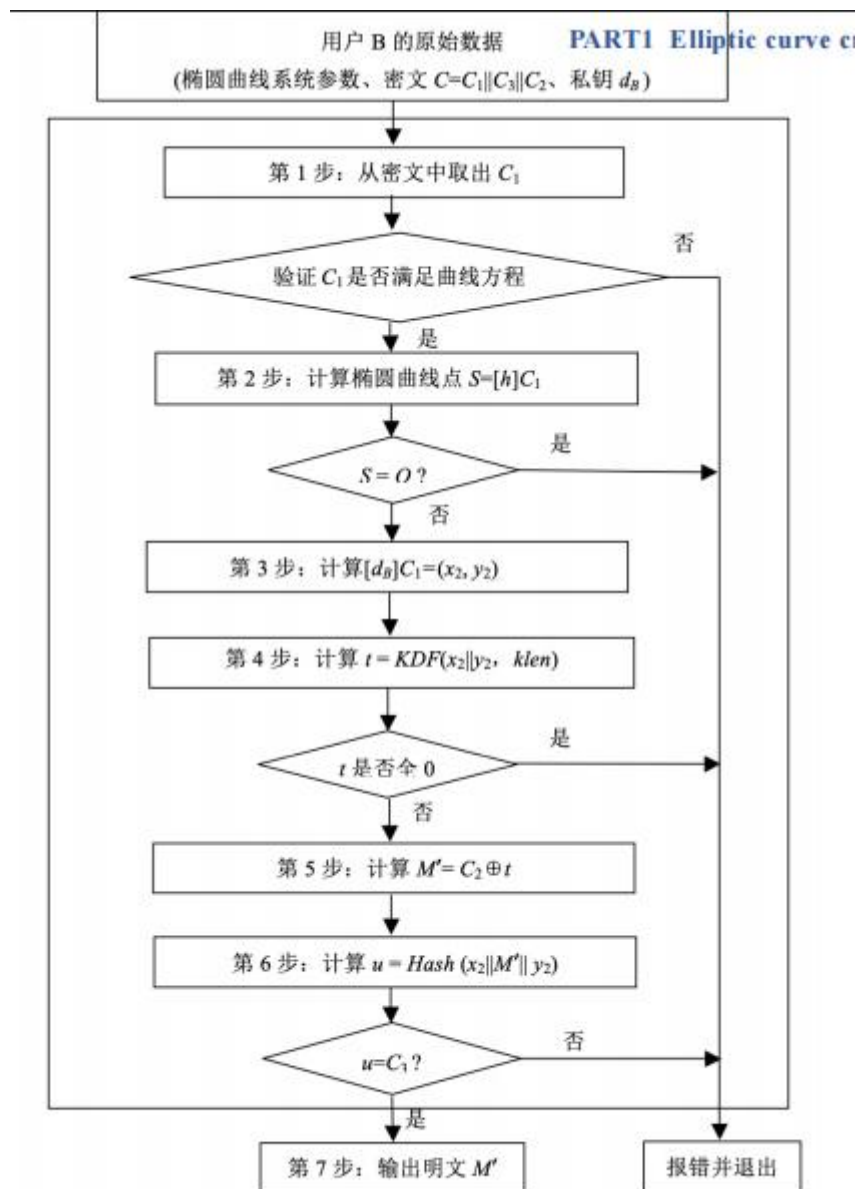


图 2 解密算法流程

实际网络通信可以用以下等代码实现：

```

1. client.sendto(x.encode('utf-8'), address)
2. client.sendto(y.encode('utf-8'), address)

```

结果：

服务端：

```
===== RESTART: E:\project16\服·
===
等待建立连接...
连接已关闭
```

客户端:

```
===== RESTART: E:\project16\客户端.
===
连接建立
696C6F7665796F75
696C6F7665796F75
```

实现方式: python

分工: 自己独立完成