### ****工作原理及过程详解****

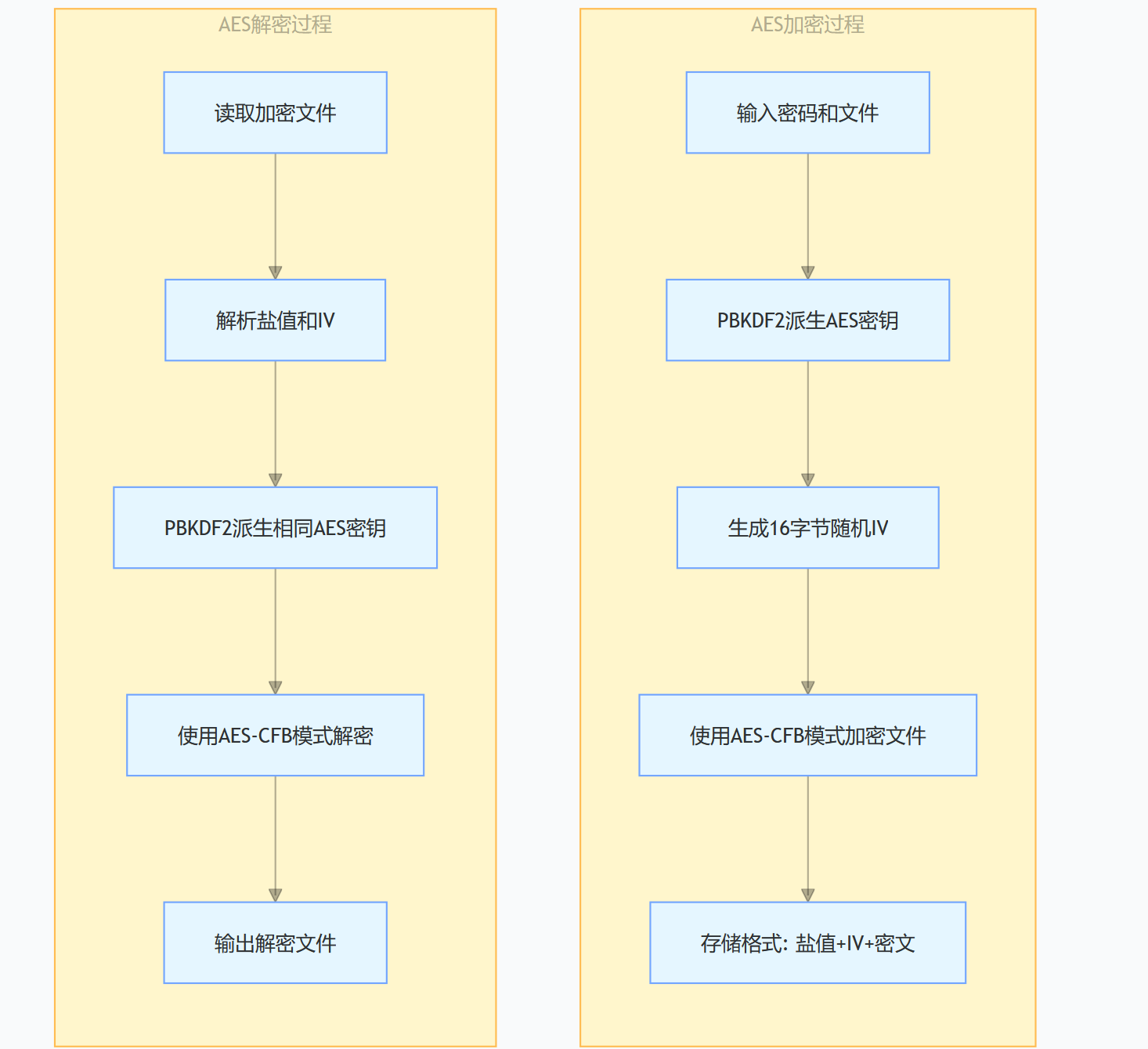
#### ****AES加解密****

1.加密输入密码：使用 PBKDF2-HMAC-SHA256 算法，迭代 10 万次，将用户输入的密码转换为 32 字节（256 位）密钥。

2.AES加密：使用 CFB（密码反馈）模式，支持流式加密，并初始化向量（IV）：随机生成 16字节 IV，确保相同明文生成不同密文。

3.存储加密信息：盐值和 IV 随密文存储（放入前32字节），无需保密但需解密时使用。

4.AES解密：利用存储的盐值和用户输入的密码，生成与加密时相同的密钥。



### ****2.混合加解密****

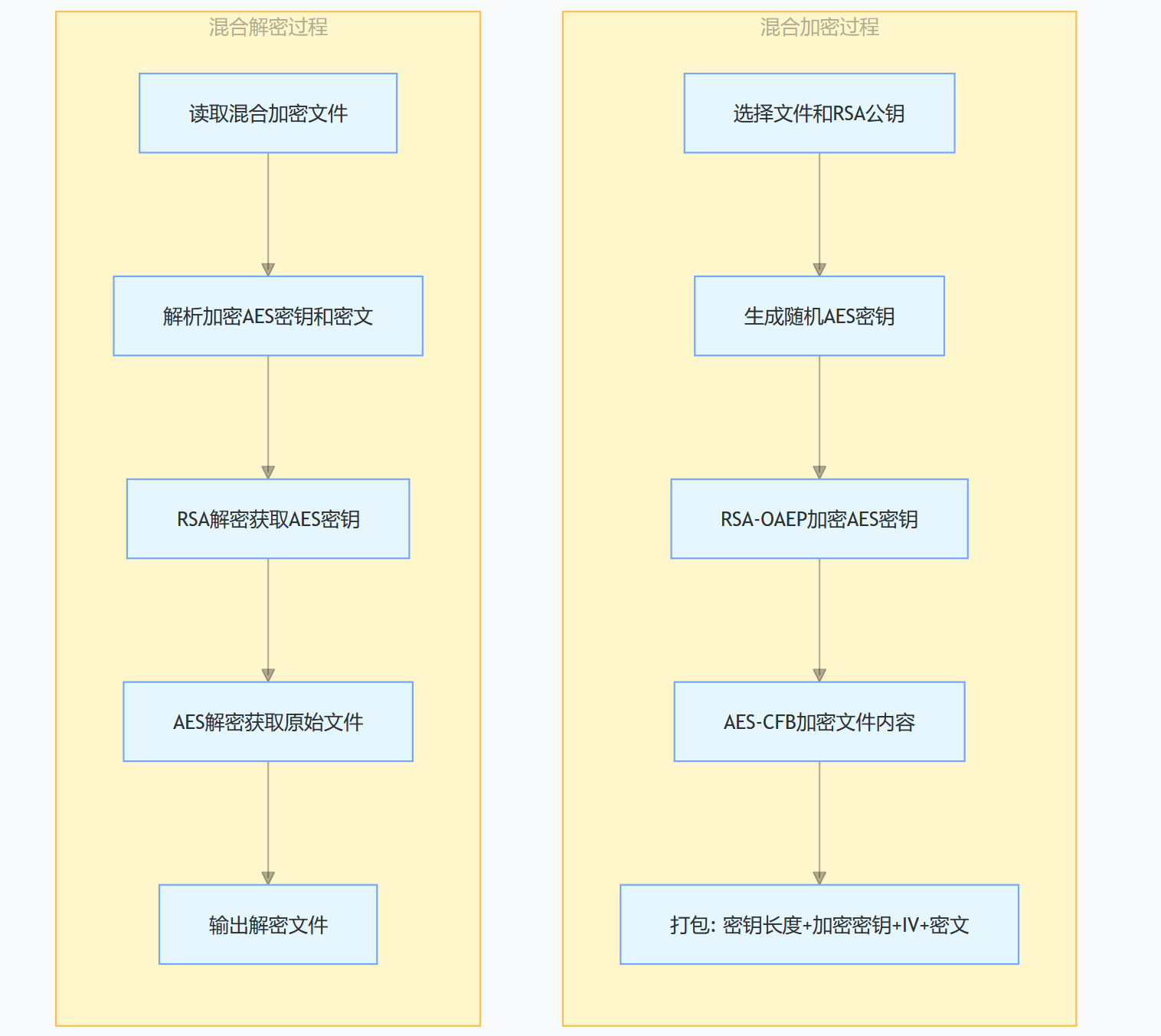
1.生成临时密钥：为每个文件生成一个随机的 AES（AES加解密流程同上）临时密钥。

2.加密会话密钥：使用接收方的 RSA 公钥加密 AES 密钥。

3.加密文件内容：使用 AES 加密文件内容。

4.组合数据：将加密后的 AES 密钥和加密文件内容打包存储。

5.私钥解密：接收方使用与公钥对应的私钥对文件进行解密。



### ****3.签名和验证签名过程详解****

1.读取文件与签名：获取原始文件和对应的签名文件

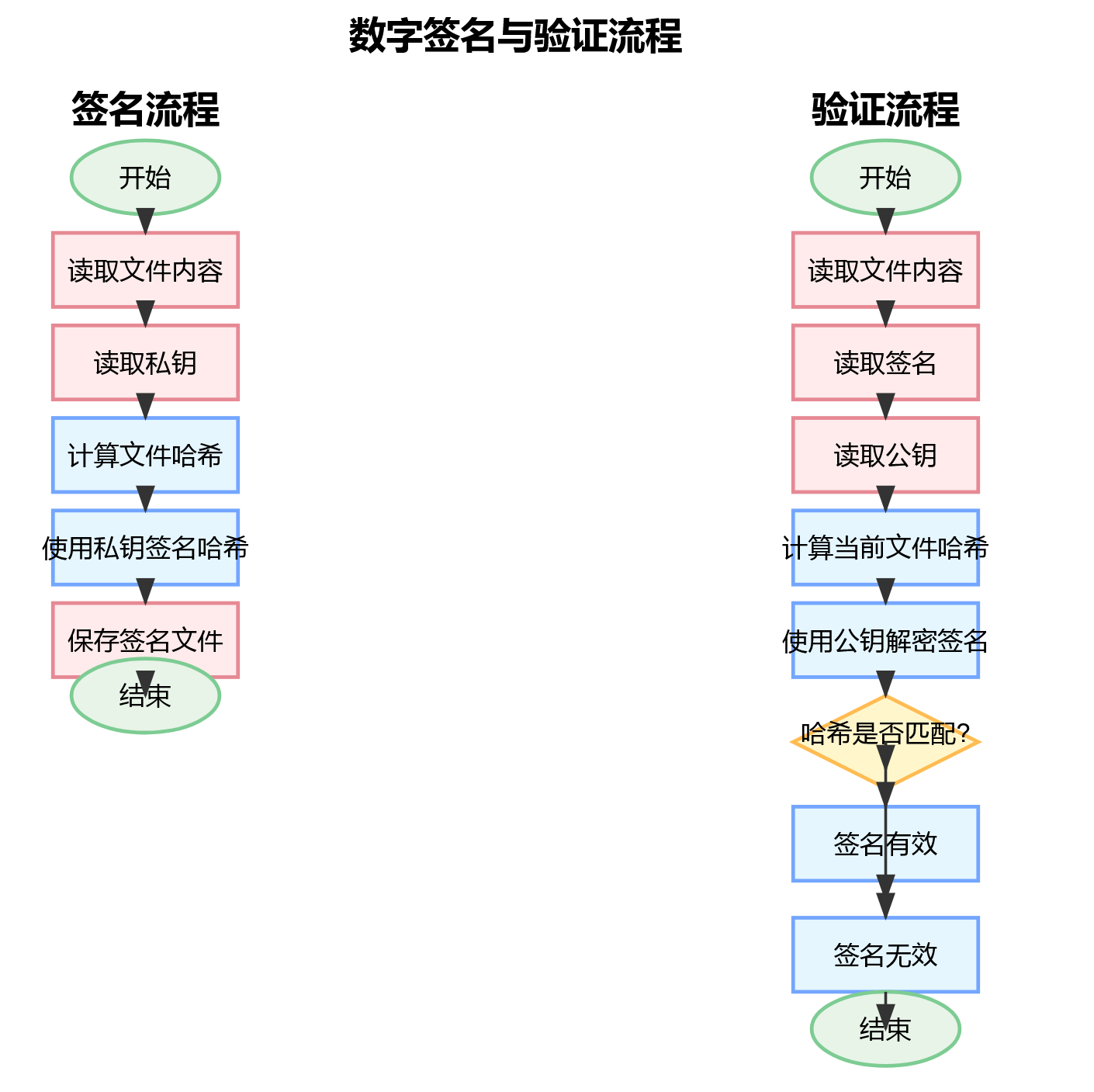
2.计算文件哈希：对原始文件重新计算哈希值（新哈希值 H1）

3.公钥解密签名：用public\_key\_path中的 RSA 公钥解密签名文件，得到原始哈希值（H2）

4.哈希值比对：比较 H1 和 H2 是否一致：

一致 → 签名有效（文件未被篡改且由对应私钥签名）

不一致 → 签名无效（文件被篡改或签名非对应私钥生成）



### ****4.文件传输过程详解****

1.服务器监听：创建 TCP 服务器监听连接请求

2.客户端连接：主动连接到远程服务器

3.文件传输：支持加密和非加密的文件传输

4.多线程处理：使用独立线程处理连接和文件传输

5.状态回调：通过回调函数与 UI 层交互

