内容回顾

- OpenSSL密码学库
  - 编译: g++ -o {可执行文件} {源程序} [-I库头文件所在目录] [-L库所在目录] **-lcrypto**
  - 大数运算库: BIGNUM, BN_new(), BN_free(),...

- RSA加密和签名
  - RSA参数: **(e, d, N, p, q)**
  - 加密(M^e **mod** N), 解密(C^d **mod** N), 签名(M^d **mod** N), 验证(s^e **mod** N)
  - 基于OpenSSL的RSA密码实现: RSA_generate_key(), RSA, RSA_public_encrypt(), RSA_private_decrypt()

对称加密技术实验

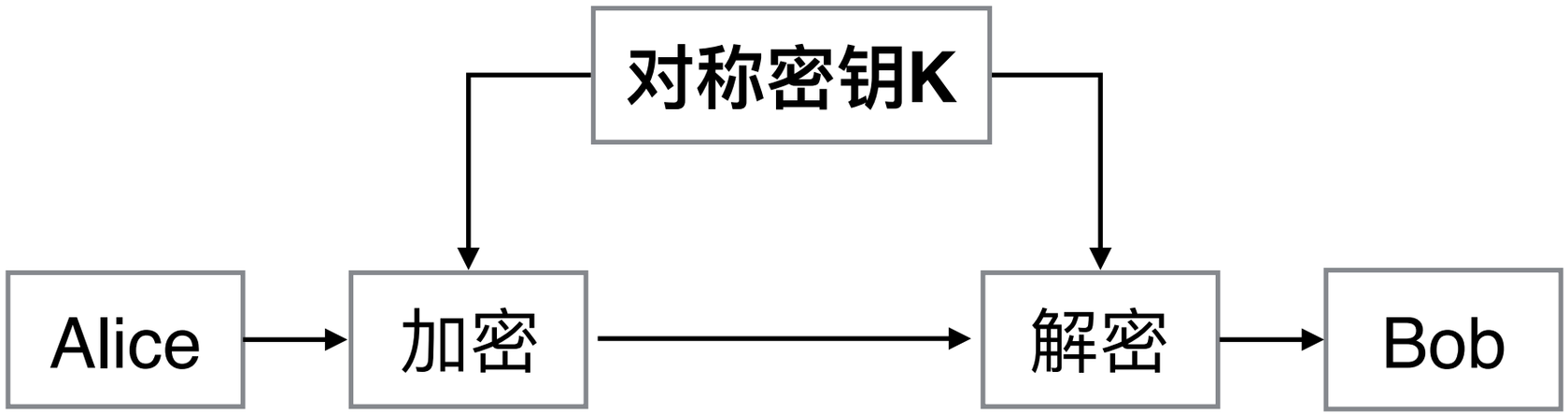
流密码: RC4

分组密码: DES

哈希函数: SHA-1

对称加密

- 加解密使用**相同的密钥**
  - ➡ 解决**安全通信**问题



- ➡ 优点: 加/解密速度快, 密钥管理简单, 适合一对一加密传输
- ➡ 缺点: 密钥分发困难, 不适合一对多加密传输

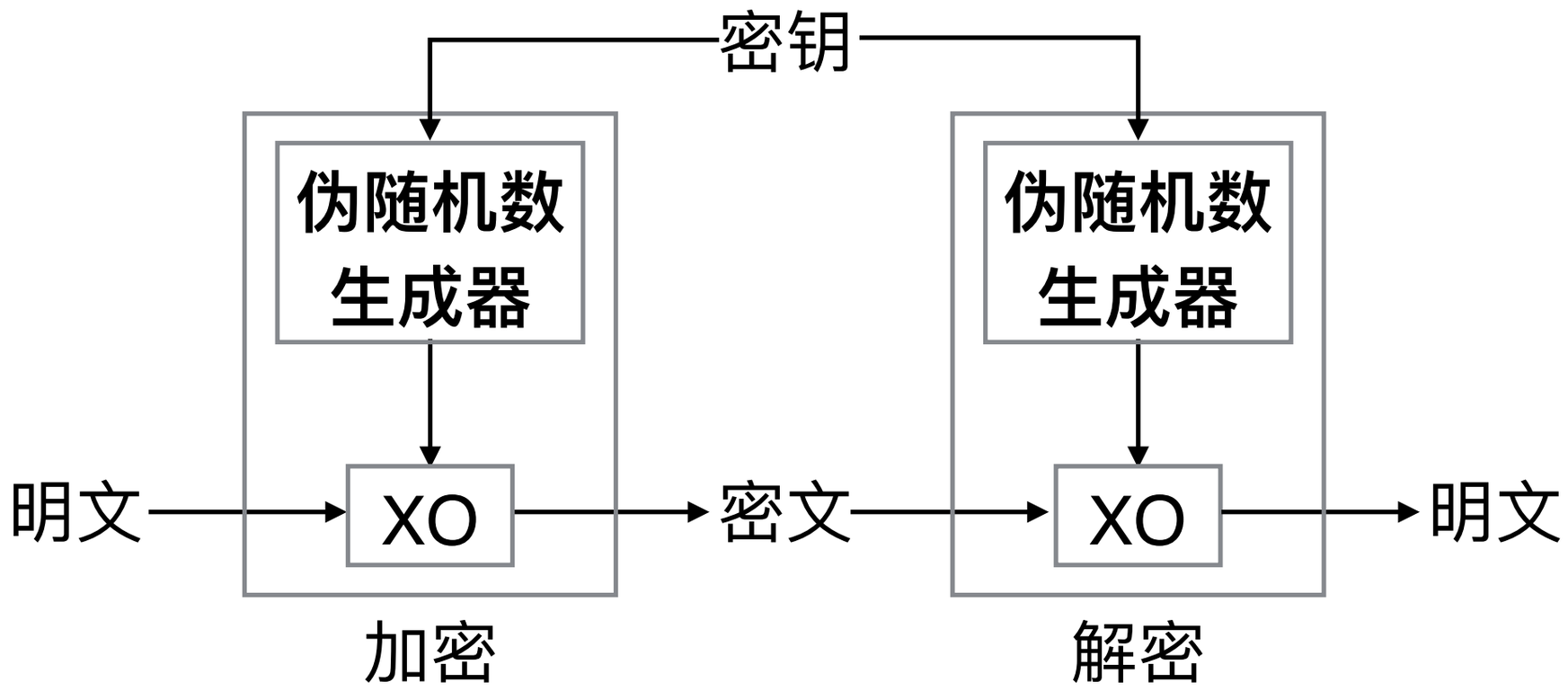
流密码历史

- **一次一密**: **E**(K, M) = K XOR M
  - **完美隐私性**(perfect secrecy)
  - 问题: 密钥长度|K| = |M|

- 流密码: **使用伪随机数替代K**
  - 伪随机数生成器**G**: $\{0,1\}^k \rightarrow \{0,1\}^*$
  - 加密函数**E(K, M) = G(K) XOR M**
    - K为有限长度, **G**(K)可为任意长度

流密码

- 流密码(序列密码): 一种**对称加密**算法, 加解密双方产生相同**伪随机流**, 明文与伪随机流**按位异或(XOR)加密**



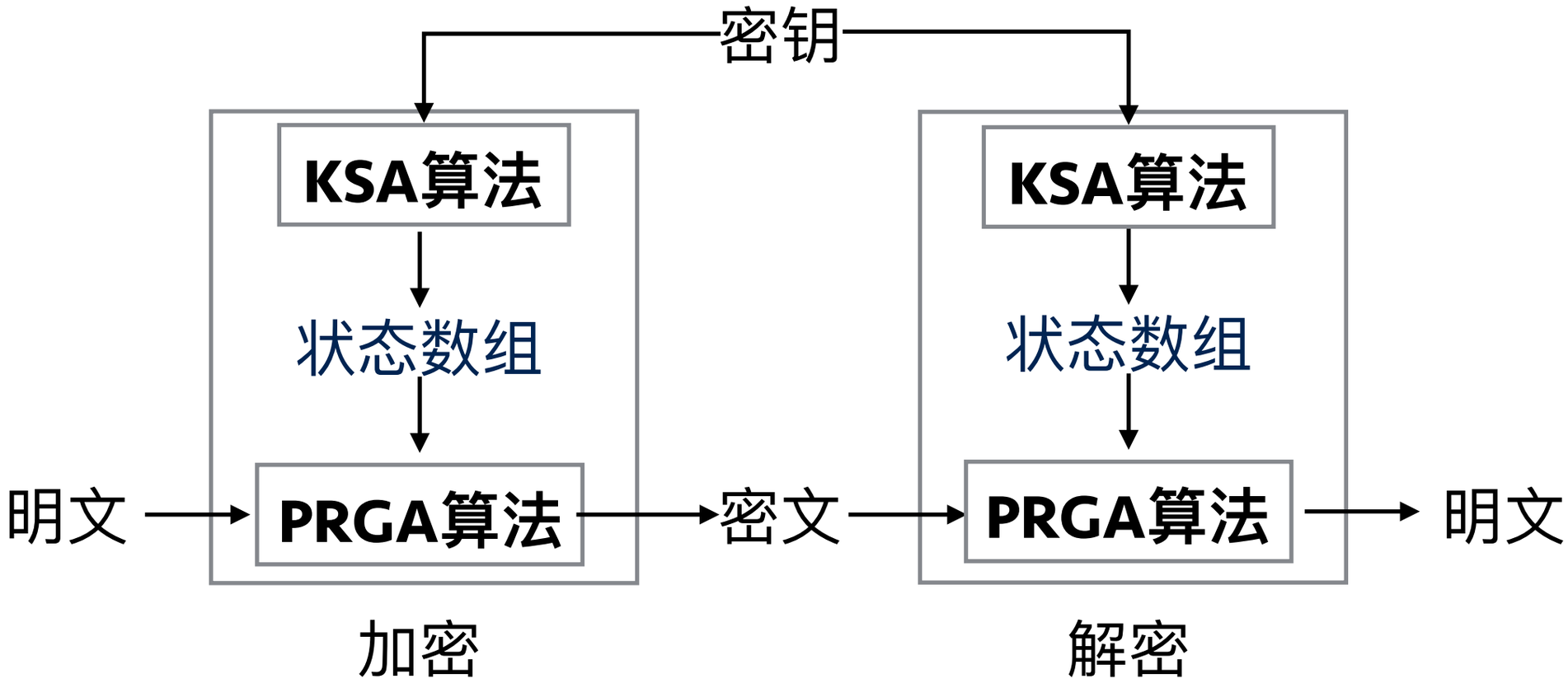
- 伪随机数生成器: 线性同余算法, BBS算法, ANSI X9.17,...

流密码特点

- 加密单位: **比特**
  - ➡ 分组长度为**1比特**的分组密码
- 优点: 低错误传播, 硬件实现简单
  - ➡ 适用于较高传输错误的通信环境
- 缺点: 扩散度低

RC4原理

- RC4: 一种具有**可变密钥长度(1～255字节)**的流密码
  - **KSA算法**: 基于输入对称密钥K, **置换**状态数组
  - **PRNG算法**: **扩充**状态数组, 加密明文数据



KSA算法

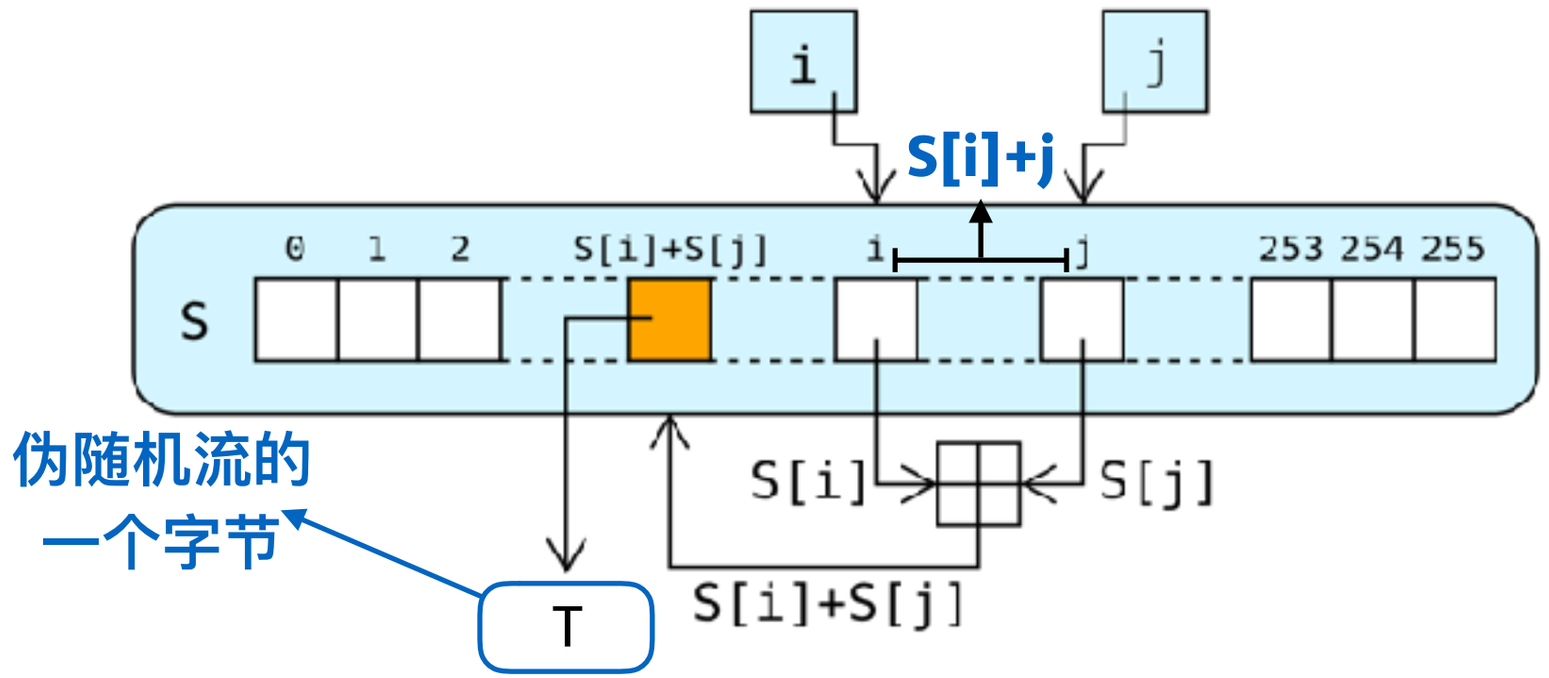
- 目标: 基于输入密钥K, 产生状态数组S的一个**置换**
  - ➡ 初始化S为单位数组(即S[i] = i)
  - ➡ 对S执行256轮置换

```
1:  for i from 0 to 255
2:      S[i] = i                                          // 初始化S为单位数组
3:  end for
4:  j = 0
5:  for i from 0 to 255                                   // 256轮置换
6:      j = (j + S[i] + K[i mod len(K)]) mod 256          // 产生基于K的下标索引
7:      swap values of S[i] and S[j]                      // 置换
8:  end for
```

PRNG算法

- 目标: 产生伪随机流, 并加密明文

```
1:   i = 0, j = 0
2:   for k from 0 to len(M)
3:       i = (i + 1) mod 256
4:       j = (j + S[i]) mod 256
5:       swap values of S[i] and S[j]
6:       T[k] := S[(S[i] + S[j]) mod 256
7:       C[k] := T[k] XOR M[k]
8:   end for
```



RC4示例-KSA算法

- 状态数组: **S[0..7]**, 密钥: **K[0] = 5, K[1] = 6, K[2] = 7**
- 初始化状态数组: S[0] = 0, S[1] = 1,..., S[7] = 7
- 循环8次更新S
  - 例如i = 0, j = 0时, j = (j + S[0] + K[0]) **mod** 8 = 5; 交换S[0]和S[5]
  - 循环结束时, S更新为: S[0] = 5, S[1] = 4, S[2] = 0, S[3] = 7, S[4] = 1, S[5] = 6, S[6] = 3, S[7] = 2

RC4示例-PRNG算法

- 基于S扩充伪随机流T: i = 0, j = 0
  - ➡ T[0]计算
    - ✳ i = (i + 1) **mod** 8 = 1
    - ✳ j = (j + S[i]) **mod** 8 = 4
    - ✳ 交换S[1]和S[4]后, 产生T[0] = S[(S[1] + S[4]) **mod** 8] = S[6] = 3
  - ➡ T[1]计算
  - ➡ … 循环上述过程直至T与明文M具有相同长度
- 加密: T XOR M

RC4安全性

- **Biased outputs** (01)
  - ➡ 区分攻击
- **Roos's biases** (95)
  - ➡ The **first byte of the T** is correlated to **first three bytes of K**
  - ➡ **重构密钥攻击** (08~09)
- **NOMORE attack** (15)
  - ➡ 75小时内获得明文

Fluhrer, Mantin and Shamir attack, **Klein's attack**,……