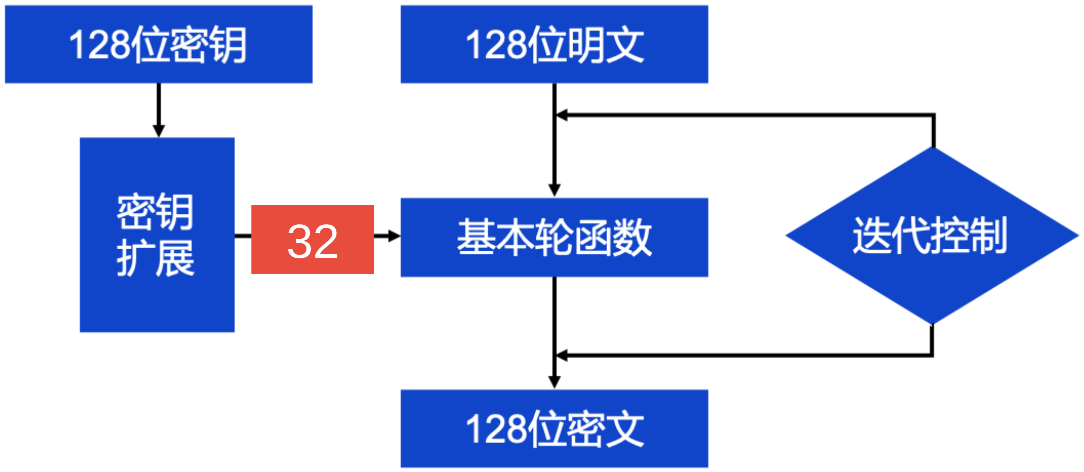
密 码 学

第五章 公钥密码算法

网络空间安全学院
胡 伟　朱丹
weihu/zhudan@nwpu.edu.cn

知识回顾–SM4密码算法概述

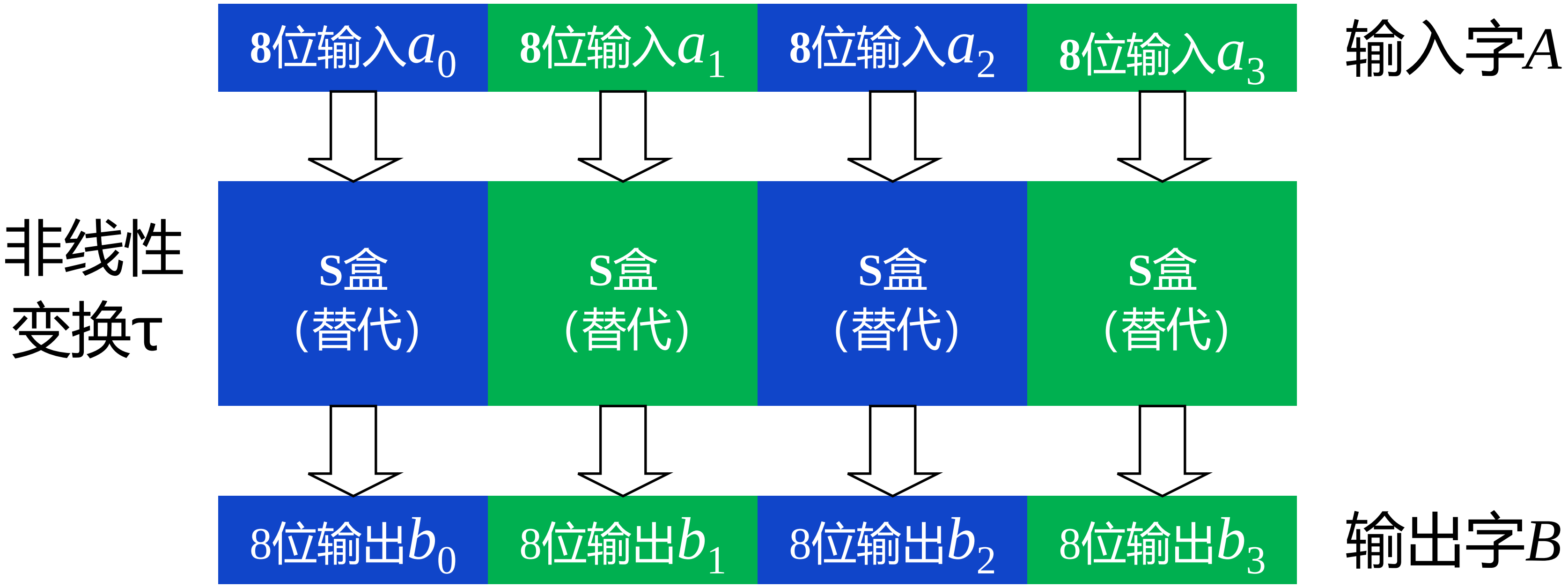
- 分组密码
  - 由数据分组（明文，密文）长度为128位、密钥长度为128位
  - 迭代轮数：32轮
  - 数据处理单位：字节（ 8位），字（32位）
- 密码算法特点
  - 对合运算：解密算法与加密算法相同
  - 密钥生成算法与加密算法结构类似

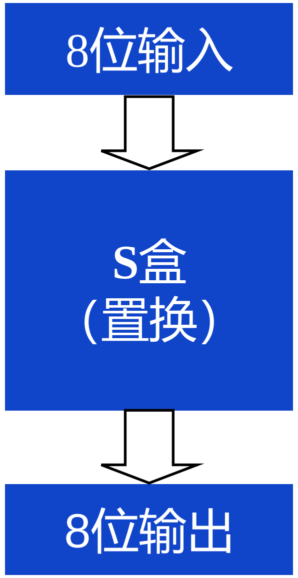


知识回顾–非线性字变换

Non-linear Word Transform

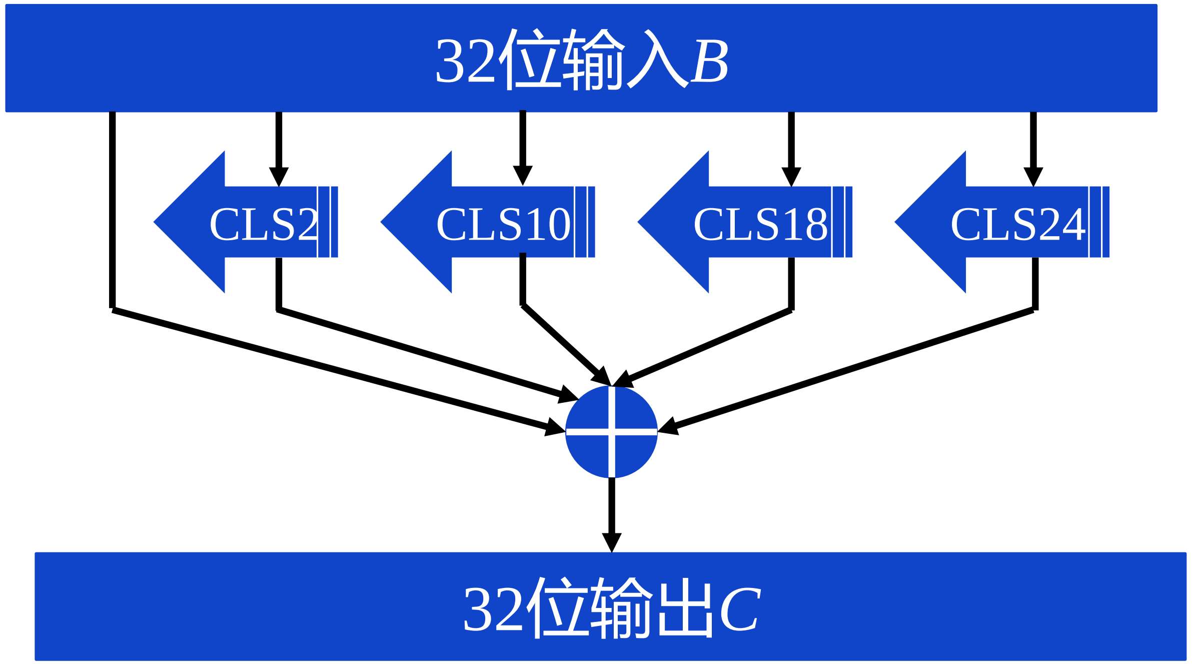
- 非线性字变换：**32位字的非线性变换**
  - 4个S盒并行替代
  - 设输入字A = (a_0, a_1, a_2, a_3)，输出字B = (b_0, b_1, b_2, b_3)
  - $B = \tau(A) = (\text{S-Box}(a_0), \text{S-Box}(a_1), \text{S-Box}(a_2), \text{S-Box}(a_3))$



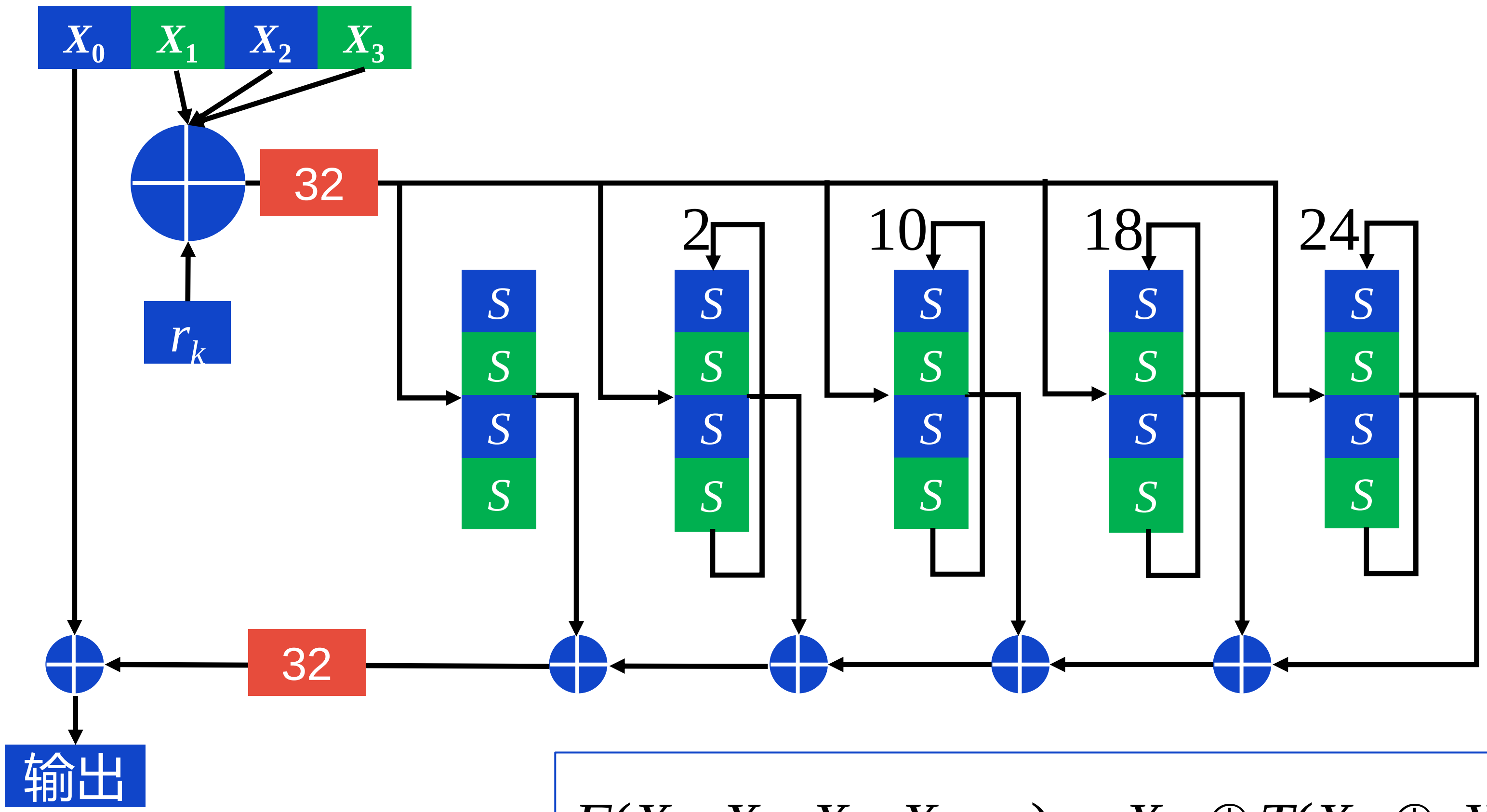


知识回顾–线性L变换

- 线性变换，32位输入，32位输出
  - 设输入为B，输出为C，表为：$C = L(B)$
  - $C = L(B) = B \oplus (B<<<2) \oplus (B<<<10) \oplus (B<<<18) \oplus (B<<<24)$



知识回顾–轮函数



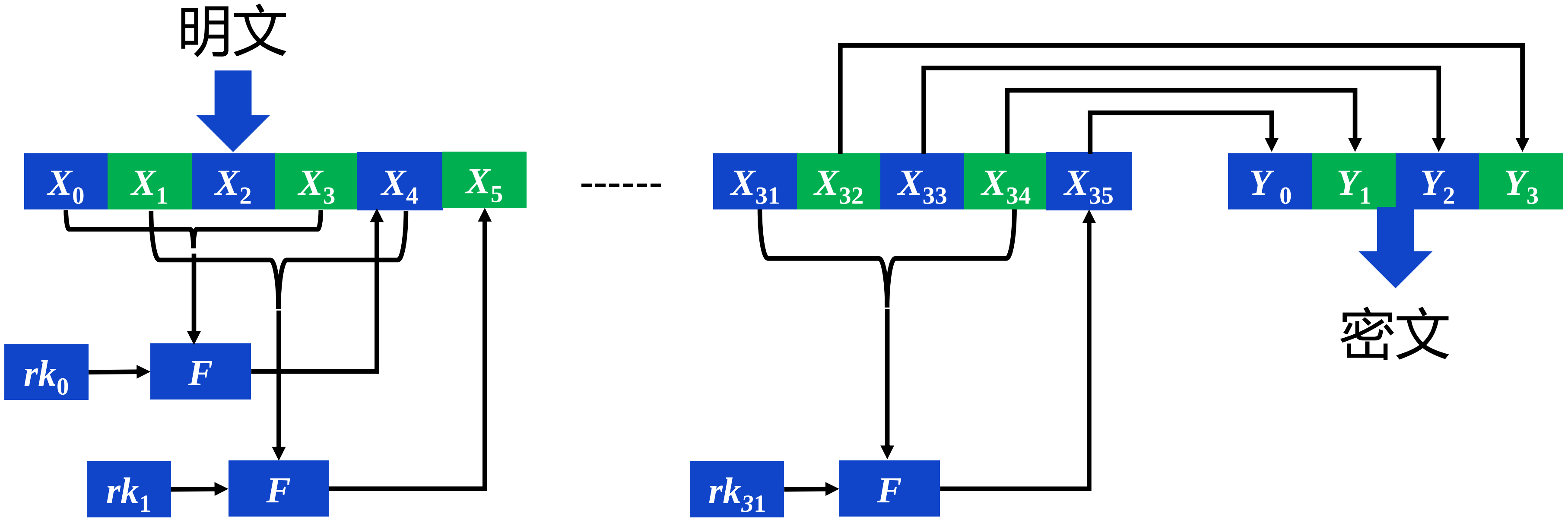
$$F(X_0, X_1, X_2, X_3, r_k) = X_0 \oplus T(X_1 \oplus X_2 \oplus X_3 \oplus r_k),\ T(X) = L(\tau(X))$$

$$B = \tau(A) = (\text{S-Box}(a_0), \text{S-Box}(a_1), \text{S-Box}(a_2), \text{S-Box}(a_3))$$

$$C = L(B) = B \oplus (B<<<2) \oplus (B<<<10) \oplus (B<<<18) \oplus (B<<<24)$$

知识回顾–SM4加密算法

- 滑动窗口迭代结构（广义 Feistel结构）



知识回顾–SM4解密算法

- SMS4密码算法是对合的，因此解密与加密算法相同，只是轮密钥的使用顺序相反
- 输入密文：(Y_0, Y_1, Y_2, Y_3)，128位，四个32位字
- 输入轮密钥：r_{ki} $(i = 31, 30, \ldots, 1, 0)$，32位字，共32个轮密钥
- 输出明文：(X_0, X_1, X_2, X_3)，128位，四个32位字

$$
\begin{cases}
X_{i+4} = F(X_i, X_{i+1}, X_{i+2}, X_{i+3}, rk_i) \\
\qquad = X_i \oplus T(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus rk_i), \quad i = 31, 30, \ldots, 1, 0 \\
\qquad = X_i \oplus L(\tau(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus rk_i)), \quad i = 31, 30, \ldots, 1, 0 \\
(Y_0, Y_1, Y_2, Y_3) = (X_{35}, X_{34}, X_{33}, X_{32})
\end{cases}
$$

知识回顾–SM4密钥扩展算法

- 输入加密密钥：$MK = (MK_0, MK_1, MK_2, MK_3)$
- 输出轮密钥：$rk_i, i = 0, 1, 2, \ldots, 30, 31$
- 中间数据：$K_i, i = 0, 1, 2, \ldots, 34, 35$
- 扩展算法流程：
  - $(K_0, K_1, K_2, K_3) = (MK_0 \oplus FK_0, MK_1 \oplus FK_1, MK_2 \oplus FK_2, MK_3 \oplus FK_3)$
  - For $i = 0, 1, 2, \ldots, 30, 31$ Do

$$rk_i = K_{i+4} = K_i \oplus T'(K_{i+1} \oplus K_{i+2} \oplus K_{i+3} \oplus CK_i)$$

  - 说明：T' 变换与加密算法轮函数中的T 相似，只将其中的线性变换L 修改为以下：L'

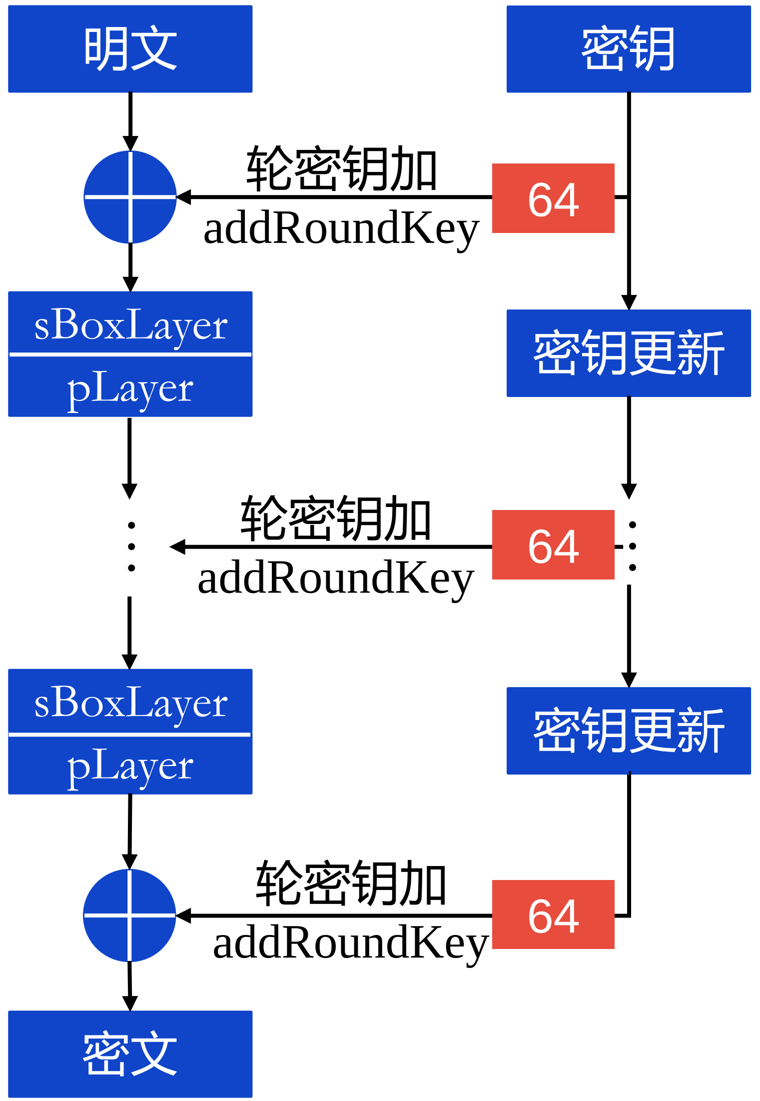
$$L'(B) = B \oplus (B <<< 13) \oplus (B <<< 23)$$

FK_0 = (A3B1BAC6), FK_1 = (56AA3350), FK_2 = (677D9197), FK_3 = (B27022DC), CK_i 也是常数

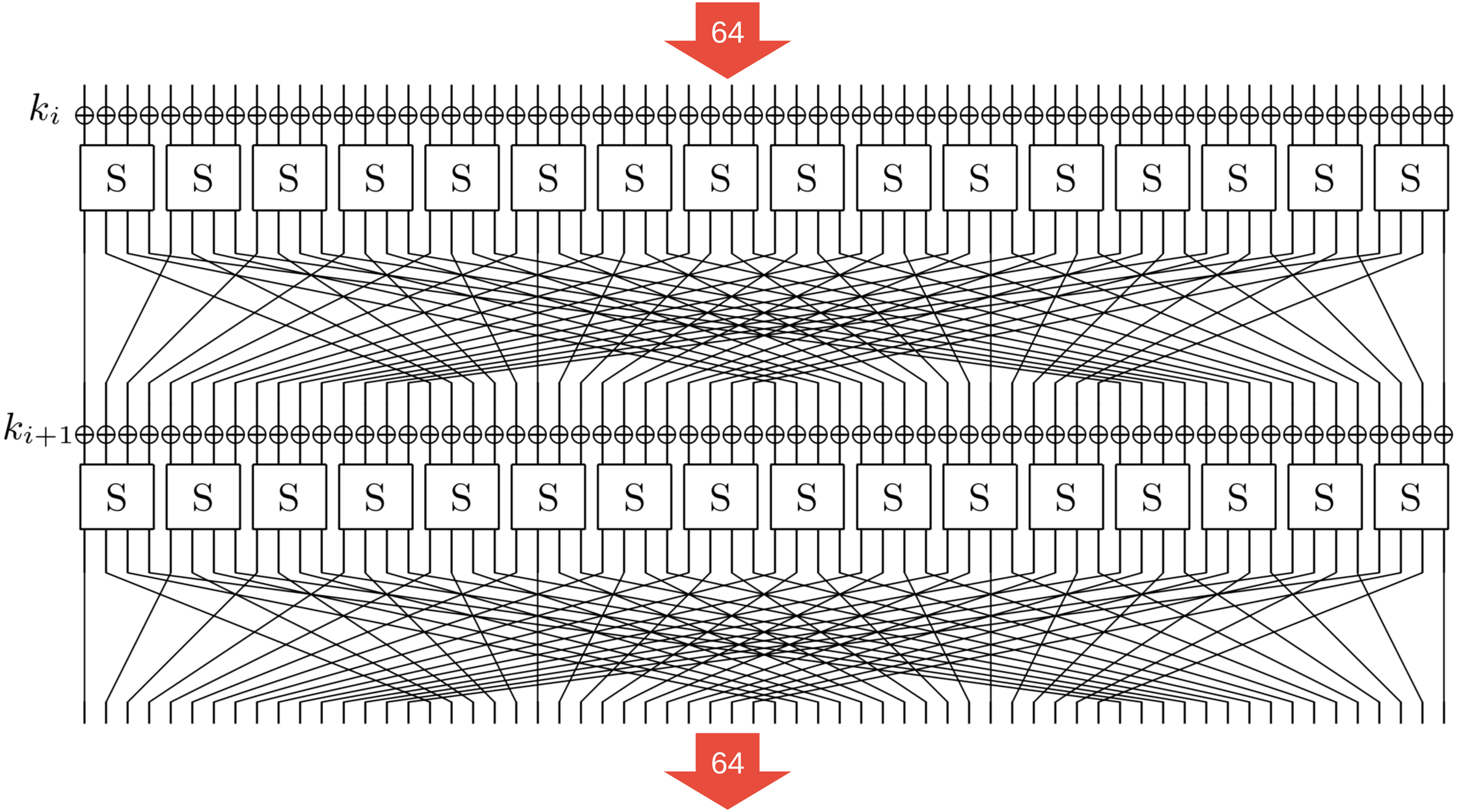
知识回顾–PRESENT密码算法

- 基本运算
  - 轮密钥加 (addRoundKey)
  - S盒代换层 (sBoxLayer)
  - P置换层 (pLayer)
  - 32个子密钥(加密31轮)，目的是使结果白化

```
generateRoundKey(key)
for i = 1 to 31 do
    addRoundKey(State, K_i)
    sBoxLayer(State)
    pLayer(State)
end for
addRoundKey(State, K_32)
```



知识回顾–PRESENT密码算法结构



练 习 题

密码算法	DES	AES	SM4	PRESENT
网络结构	Feistel网络	S-P网络	滑动窗口 广义Feistel网络	S-P网络
分组长度	64	128	128	64
密钥长度	64	128/192/256	128	80/128
子密钥长度	48	128	32	64
轮数	16	10/12/14	32	31
S盒规模	6进4出	8进8出	8进8出	4进4出
结构特点	对合	对称	对合	非对称

章节安排

Outline

章节安排

Outline

- 公钥密码的产生背景
- 公钥密码的原理
- RSA公钥密码算法

5.1(1) 密码算法分类

密码算法

对称密码算法

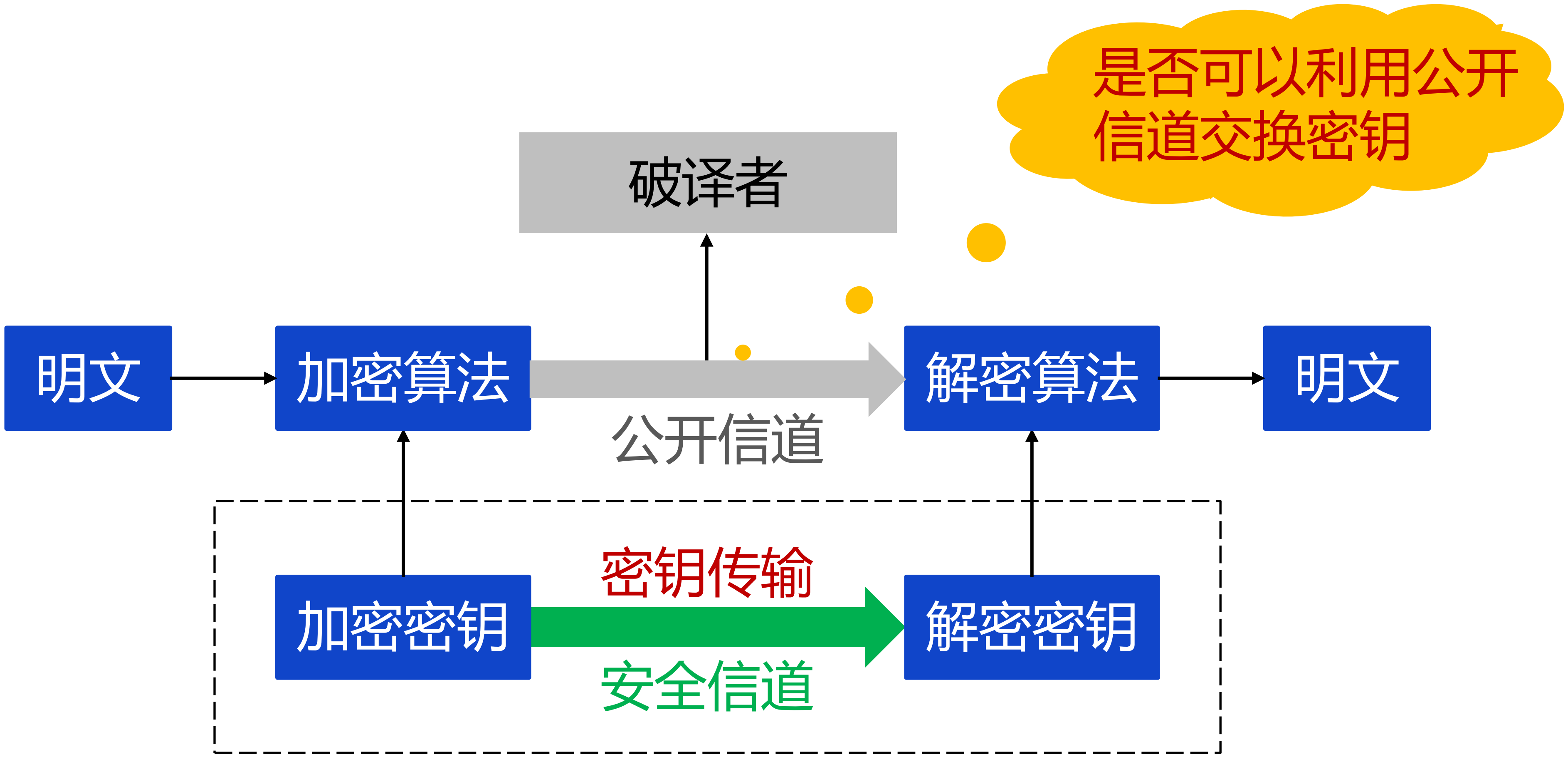
加密密码和解密密钥相同，或本质上相同

- 分组密码算法
- 序列密码算法

非对称密码算法

加密密钥和解密密钥不同，且不能由加密密钥推导出解密密钥

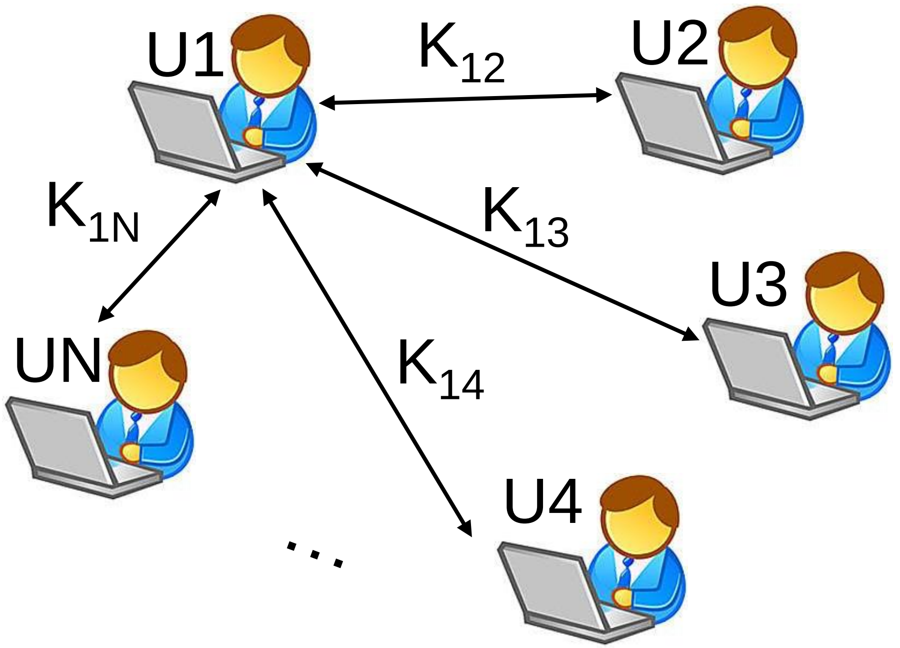
5.1(2) 公钥密码的产生背景



传统密码体制的不足：密钥难共享

5.1(2) 公钥密码的产生背景

- N个实体通信
- 每对实体通信都需要一个共享密钥
- N个实体的网络中，一共需要N*(N-1)/2个密钥
- 还需要同样数量的保密信道用于密钥传输



传统密码体制的不足：密钥难管理

5.1(2) 公钥密码的产生背景

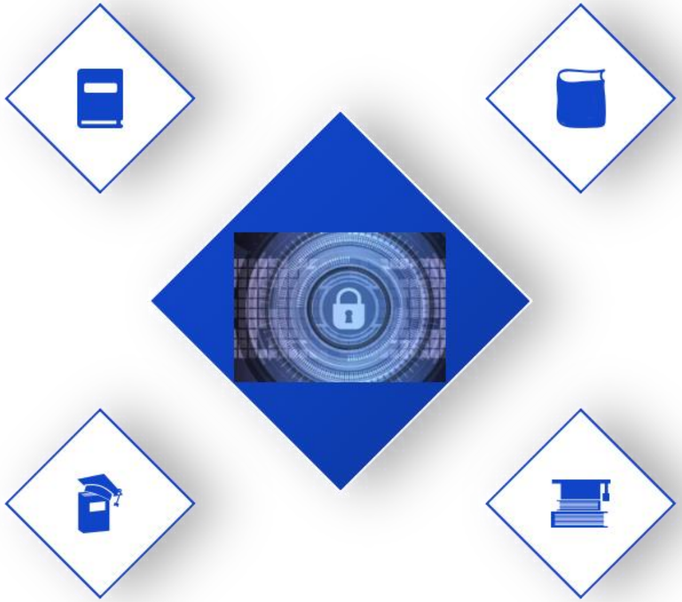
- 传统密码体制的不足：难以解决签名和认证问题
  - 消息接收方可伪造原文
  - 消息发送方可否认所发消息
  - 未解决消息的真实性和不可否认性问题

机　密　性

防止敏感信息泄漏

完　整　性

防止关键信息被篡改



真　实　性

防止身份或数据假冒

不 可 否 认 性

防止攻击行为抵赖

5.1(2) 公钥密码的产生背景

- 20世纪70年代，斯坦福大学Diffie和Hellman研究了密钥分发问题，提出了一种通过公开信道共享密钥的方案，即Diffie-Hellman密钥交换协议
- 1976年，Diffie和Hellman发表了题为《密码学的新方向》（New Directions in Cryptography）的论文，论文提出了公钥密码的思想

美国计算机协会（ACM）将2015年的图灵奖授予Sun Microsystems的前首席安全官惠特菲尔德·迪菲（Whitfield Diffie）以及斯坦福大学电气工程系名誉教授马丁·赫尔曼（Martin Hellman），以表彰他们在现代密码学中所起的至关重要的作用。

章 节 安 排

Outline

5.2(1) 公钥密码的基本思想

- 将密钥 K一分为二：K_e和 K_d。K_e专门加密，K_d专门解密，$K_e \neq K_d$
- 由K_e不能简单地计算出K_d，于是可将K_e公开，使密钥K_e分配简单
- 由于$K_e \neq K_d$且由K_e不能计算出K_d，所以K_d便成为用户的指纹，可方便地实现数字签名和身份认证

称上述密码体制为公开密钥密码，简称为公钥密码

5.2(2) 公钥密码的基本条件

- ① 安全条件：$K_e \neq K_d$且由K_e不能计算出K_d
- ② 保密条件：E和D互逆；$D(E(M)) = M$
- ③ 使用条件：E和D都高效
- ④ 保真条件：$E(D(M)) = M$

- 如果满足①②③可用于保密
- 如果满足①③④可用于保真
- 如果①②③④都满足，可同时用于保密和保真

5.2(3) 公钥密码的理论模型

- 单向函数: 设函数$y = f(x)$，如果满足以下两个条件，则称为单向函数:
  - 如果对于给定的x，要计算出$y = f(x)$很容易
  - 而对于给定的y，要计算出$x = f^{-1}(y)$很难

单向函数是否
适于构造数据加密算法

- 利用单向函数构造加密函数
  - 用正变换作加密，加密效率高
  - 用逆变换作解密，安全，敌手不可破译

合法的消息接收者也无法解密

5.2(3) 公钥密码的理论模型

- 单向陷门函数：设函数$y = f(x)$，且f具有陷门，若满足以下条件，则称为单向陷门函数：
  - 如果对于给定的x，要计算出$y = f(x)$很容易
  - 而对于给定的y，如果不掌握陷门要计算出$x = f^{-1}(y)$很难
  - 而如果掌握陷门要计算出$x = f^{-1}(y)$就很容易

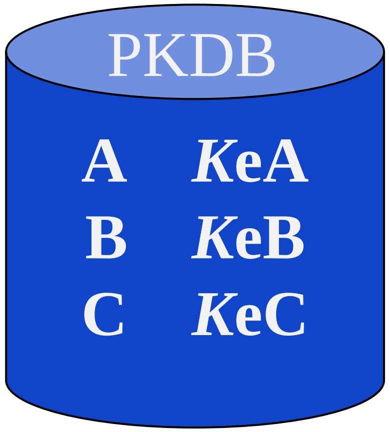
- 利用单向陷门函数构造加密函数
  - 用正变换作加密，加密效率高
  - 用逆变换作解密，安全，敌手难以破译
  - 把陷门信息作为密钥，且只分配给合法用户。确保合法用户能够方便地解密，而非法用户不能破译

5.2(4) 单向函数的研究现状

- 理论上：尚不能证明单向函数一定存在
- 实际上：密码学认为只要函数单向性足够应用即可
- ① 大合数的因子分解问题
- 大素数的乘积容易计算($p\times q=n$)，而大合数的因子分解困难($n=p\times q$)
- ② 有限域上的离散对数问题
- 有限域上大素数的幂乘容易计算($a^b=c$)，而对数计算困难($\log_a c=b$)
- ③ 椭圆曲线离散对数问题
- 设d是正整数，G是解点群的基点，计算$dG=Q$是容易的，而由Q求出d是困难的

5.2(5) 公钥密码的工作方式

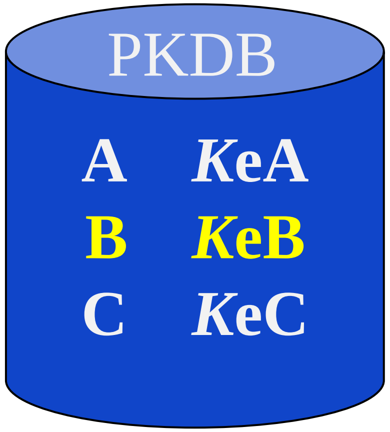
- 基本概念
  - 设M为明文，C为密文，E为加密算法，D为解密算法
  - 每个用户都配置一对密钥：K_e为公开的加密密钥，K_d为保密的解密密钥
  - 将所有用户公开的加密密钥K_e存入共享的密钥库PKDB
  - 保密的解密密钥K_d由用户妥善保管



5.2(5) 公钥密码的工作方式

Working Flow

- 确保数据机密性
- 发方
  - ① A首先查PKDB，查到B的公开的加密密钥K_{eB}
  - ② A用K_{eB}加密M得到密文C：$C = E(M，K_{eB})$
  - ③ A发C给B



- 收方
  - ① B接收C
  - ② B用自己的K_{dB}解密，得到明文$M = D(C, K_{dB}) = D(E(M，K_{eB}), K_{dB})$

5.2(5) 公钥密码的工作方式

- 确保数据机密性（安全性分析）：
  - ① 只有B才有K_{dB}，因此只有B才能解密，所以确保了数据的机密性
  - ② 任何人都可查PKDB得到B的K_{eB}，所以任何人都可冒充A给B发送数据。不能确保数据的真实性

A

B

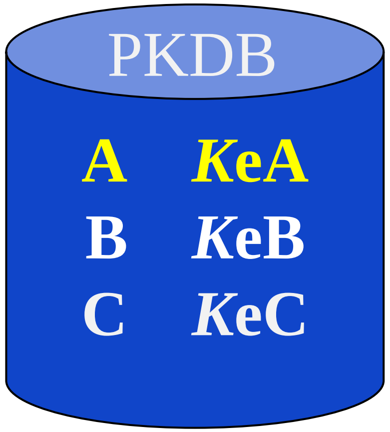
$$C = E(M,\ K_{eB})$$

$\longrightarrow$

$$M = D(C, K_{dB}) = D(E(M,\ K_{eB}), K_{dB})$$

5.2(5) 公钥密码的工作方式

- 确保数据真实性
- 发方
  - ① A首先用自己的K_{dA}对M加密，得到$C = D(M, K_{dA})$
  - ② A发C给B
- 收方
  - ① B接收C
  - ② B查PKDB查到A的公开的加密密钥K_{eA}
  - ③ B用K_{eA}加密C，得到明文$M = E(C, K_{eA}) = E(D(M, K_{dA}), K_{eA})$



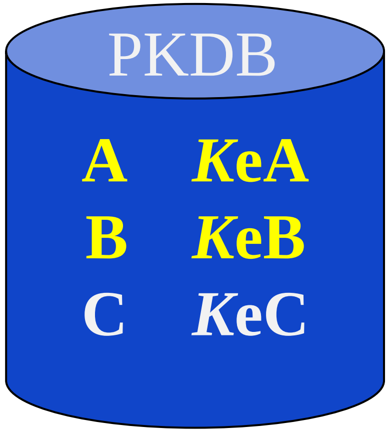
5.2(5) 公钥密码的工作方式

- 确保数据真实性（安全性分析）：
  - ① 只有A才有K_{dA}，因此只有A才能加密产生C，所以确保了数据的真实性
  - ② 任何人都可查PKDB得到A的K_{eA}，所以任何人都可解密得到明文。不能确保数据的机密性

$C = D(M, K_{dA})$ → $M = E(C, K_{eA}) = E(D(M, K_{dA}), K_{eA})$

5.2(5) 公钥密码的工作方式

- 同时确保数据**机密性**和**真实性**
- 发方
  - ① A首先用自己的K_{dA}对M加密，得到S：$S = D(M, K_{dA})$
  - ② A查PKDB，查到B的公开的加密密钥K_{eB}
  - ③ A用K_{eB}加密S得到C：$C = E(S, K_{eB})$
  - ④ A发C给B
- 收方
  - ① B接收C
  - ② B用自己的K_{dB}解密C，得到S：$S = D(C, K_{dB})$
  - ③ B查PKDB，查到A的公开的加密密钥K_{eA}
  - ④ B用A的公开密钥K_{eA}解密S，得到M：$M = E(S, K_{eA})$



5.2(5) 公钥密码的工作方式

- 同时确保数据机密性和真实性(安全性分析)：
  - ① 只有A才有K_{dA}，因此只有A才能加密产生S，所以确保了数据的真实性
  - ② 只有B才有K_{dB}，因此只有B才能解密获得明文，所以确保了数据的机密性

A

$$C = E(S, K_{eB}) = E(D(M, K_{dA}),\ K_{eB})$$

B

$$M = E(S, K_{eA}) = E(D(C, K_{dB}), K_{eA})$$

章节安排

Outline

- 公钥密码的产生背景
- 公钥密码的原理
- **RSA公钥密码算法**

5.3 RSA公钥密码算法

- 1977年由美国麻省理工学院的Ron Rivest、Adi Shamir和Len Adleman提出，1978年正式公布
- 算法建立在大整数因子分解的困难性之上
- 目前应用最广泛的公钥密码算法之一
- 既可用于加密，又可用于数字签名
- 目前常使用的密钥长度1024、2048、4096
- RSA的计算量远大于DES和AES, 加密速度慢

5.3 RSA公钥密码算法

加解密算法

- 随机地选择两个大素数p和q，而且保密
- 计算$n = p * q$，将n公开
- 计算$\varphi(n) = (p-1) * (q-1)$，对$\varphi(n)$保密
- 随机地选取一个正整数e，$1 < e < \varphi(n)$且$(e, \varphi(n)) = 1$，将 e公开
- 根据 $ed = 1 \bmod \varphi(n)$，求出d，并对d保密
- 加密运算：$C = M^e \bmod n$
- 解密运算：$M = C^d \bmod n$
- 公开密钥$K_e = <e, n>$，保密密钥$K_d = <p, q, d, \varphi(n)>$

- 欧拉函数：对正整数n，欧拉函数$\varphi(n)$是小于或等于n的正整数中与n互质的数的数目，当n为素数时$\varphi(n)$为n - 1
- 欧拉定理（也称费马-欧拉定理）：是一个关于同余的性质。若n, a为正整数，且n, a互素，则:

$$a^{\varphi(n)} \equiv 1 \, mod \, n$$

5.3 RSA公钥密码算法

- 解密算法*E*和加密算法*D*的可逆性
  - 要证明：$D(E(M)) = M$，即要证明

$$M = C^d = (M^e)^d = M^{ed} \bmod n$$

  - 由$ed = 1 \bmod \varphi(n)$，有$ed = t\varphi(n) + 1$，其中t为整数。所以，

$$M^{ed} = M^{t\varphi(n)+1} \bmod n$$

  - 因此要证明 $M^{ed} = M \bmod n$，只需证明

$$M^{t\varphi(n)+1} = M \bmod n$$

  - 在$(M, n) = 1$的情况下，根据数论(Euler定理)，

$$M^{t\varphi(n)} = 1 \bmod n$$

  - 于是$M^{t\varphi(n)+1} = M \bmod n$

$$M^{ed} \downarrow M^{t\varphi(n)+1}$$

$$M^{\varphi(n)} = 1 \bmod n$$

5.3 RSA公钥密码算法

- 解密算法*E*和加密算法*D*的可逆性
  - $(M, n) \neq 1$的情况下，分两种情况：
  - 第一种情况：$M \in \{1, 2, 3, \ldots, n-1\}$
  - 因为$n = pq$, p和q为素数，$M \in \{1, 2, 3, \ldots, n-1\}$，且$(M, n) \neq 1$
  - 这说明M必含p或q之一为其因子，而且不能同时包两者，否则将有$M \geq n$，与 $M \in \{1, 2, 3, \ldots, n-1\}$矛盾
  - 不妨设$M = ap$。又因q为素数，且M不包含q，故有$(M, q) = 1$，于是有，

$$M^{\varphi(q)} = 1 \bmod q$$

Euler定理

- 解密算法*E*和加密算法*D*的可逆性
  - 进一步有$M^{t(p-1)\varphi(q)} = 1 \bmod q$。因为$q$是素数，$\varphi(q) = (q-1)$，所以有$t(p-1)\varphi(q) = t\varphi(n)$，

$$\mathrm{M}^{t\varphi(n)} = 1 \bmod q$$

  - 于是，$M^{t\varphi(n)} = bq + 1$，其中b为整数。两边同乘M，$\mathrm{M}^{t\varphi(n)+1} = bqM + M$。因为$M = ap$（根据上一页假设），故

$$\mathrm{M}^{t\varphi(n)+1} = bqap + M = abn + \mathrm{M}$$

  - 取模n得，$\mathrm{M}^{\varphi(n)+1} = M \bmod n$
  - 第二种情况：$\mathrm{M} = 0$
  - 当$\mathrm{M} = 0$时，直接验证，可知命题成立

5.3 RSA公钥密码算法

- 加密和解密运算的可交换性
  - $D(E(M)) = (M^e)^d = M^{ed} = (M^d)^e = E(D(M)) \bmod n$
  - 因此，**RSA**密码可同时确保数据的机密性和真实性

- 加解密算法的有效性
  - **RSA**密码的加解密运算是模幂运算，是比较高效的

5.3 RSA公钥密码算法

- 在计算上由公开的加密钥不能求出解密钥
  - 大合数的因子分解却是十分困难的
  - 大合数的因子分解的时间复杂度下限目前尚无定论
  - 迄今为止的各种因子分解算法提示人们这一时间下限将不低于 $O(EXP(\ln N*\ln\ln N)^{1/2})$
  - 可见，只要合数足够大，进行因子分解是相当困难的

5.3 RSA公钥密码算法

- 在计算上由公开的加密密钥不能求出解密密钥
  - 假设攻击者截获了密文C，想求出明文M
  - 他知道$M \equiv C^d \mod n$，因为n是公开的
  - 要从C中求出明文M，必须先求出d，而d是保密的
  - 但他知道，$ed \equiv 1 \mod \varphi(n)$，$e$是公开的
  - 要从中求出d，必须先求出$\varphi(n)$，而$\varphi(n)$是保密的

5.3 RSA公钥密码算法

- 在计算上由公开的加密钥不能求出解密密钥
  - 但他又知道，$\varphi(n) = (p - 1)(q - 1)$，要从中求出$\varphi(n)$，必须先求出$p$和$q$，而$p$和$q$是保密的
  - 但他知道，$n = pq$，要从n求出p和q，只有对n进行因子分解。而当n足够大时，这是困难的
  - 只要能对n进行因子分解，便可攻破RSA密码
  - 此可以得出，破译RSA密码的困难性 ≤ 对n因子分解的困难性。目前尚不能证明两者是否能确切相等
  - 尚不能确知除了对n进行因子分解的方法外，是否还有别的更简捷的破译方法

5.3 RSA公钥密码算法

- 在计算上由公开的加密密钥不能求出解密密钥
  - 公开密钥$K_e = <e, n>$
  - 保密密钥$K_d = <p, q, d, \varphi(n)>$
  - 假设攻击者截获密文C，*需恢复明文*

属于哪类攻击?

$$n = pq$$

$$\varphi(n) = (p - 1)(q - 1)$$

$$ed \equiv 1 \bmod \varphi(n)$$

$$M \equiv C^d \bmod n$$

$$C$$

课后阅读与实践

- RSA Calculator，
  https://www.cs.drexel.edu/~jpopyack/IntroCS/HW/RSAWorksheet.html
- RSA在线计算器，http://nmichaels.org/rsa.py
- 欧拉函数，https://baike.baidu.com/item/欧拉函数/1944850?fr=Aladdin
- 欧拉定理，https://baike.baidu.com/item/欧拉定理/891345?fr=Aladdin
- https://www.di-mgt.com.au/crt_rsa.html


感谢聆听!
THANK YOU FOR YOUR ATTENTION!