安全协议设计与分析

第二讲：安全协议中的密码算法

李晖 网络空间安全学院

本讲内容

1. 密码学的基本概念
2. 古典密码
3. 密码体制的安全性要素
4. 对称密码体制的概念与分类
5. 非对称密码体制
6. 签名算法
7. Hash算法

1. 密码学的基本概念

- 专业术语
  - **发送者**和**接收者**
  - **明文**和**密文**
  - 密码体制
    - 所有可能的明文的集合P，称为明文空间；
    - 所有可能的密文的集合C，称为密文空间；
    - 所有可能的密钥的集合K，称为密钥空间；
    - **加密算法**：$E: P \times K \to C, (m,k) \mapsto E_k(m),$
    - **解密算法**：$D: C \times K \to P, (c,k) \mapsto D_k(c),$

对 $\forall m \in P, k \in K$,有$D_k(E_k(m)) = m$ 。

五元组（P，C，K，E，D）称为一个密码体制。

1. 密码学的基本概念

- 常见的**密码分析攻击**有四类：
  - **唯密文攻击**（ciphertext-only attack）指密码分析者仅根据截获的密文进行的密码攻击 。
  - **已知明文攻击**（know-plaintext attack）指密码分析者已经掌握了一些相应的明、密文对，根据这些明、密文对对密码体制进行的攻击。
  - **选择明文攻击**（chosen-plaintext attack）密码分析者暂时获得对加密机的访问权限，可以选择一些明文，并可取得相应的密文。
  - **选择密文攻击**（ chosen-ciphertext attack ）密码分析者暂时获得对解密机的访问权限，可以选择一些密文，并可取得相应的明文。

1. 密码学的基本概念

根据其攻击方式，分为

- **主动攻击**
  - **中断**。如破坏计算机硬件、网络或文件管理系统。

    --〉攻击可用性
  - **篡改**。如修改数据文件中的数据、替换某一程序使其执行不同的功能、修改网络中传送的消息内容等。

    --〉攻击完整性
  - **伪造**。如在网络中插入伪造的消息或在文件中插入伪造的记录。

    --〉攻击真实性
- **被动攻击**
  - **窃听**。如搭线窃听、对文件或程序的非法复制等，以获取他人的信息。

    --〉攻击保密性

2. 古典密码

- **移位密码**（恺撒密码）
  - 加密函数：E(m)=(m+k)%q
- **仿射密码**
  - 加密函数：$e(x) = ax + b \pmod{m}$

    其中：a和m互质，m是字母数目
- **维吉利亚密码**
- **置换密码**
  - 根据一定的规则重新排列明文，以便打破明文的结构特性。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

维吉利亚密码表格

- **总结**：算法简单，算法复杂度不高，难以抵抗穷举搜索攻击

3. 密码体制的安全性要素

- **理论安全性**，完善保密性或无条件安全性
  - 当攻击者不知道密钥时，知道对应的密文对于估计明文没有任何帮助
  - 攻击者具有无穷的计算资源，也无法破解
- **实际安全性**，实际的不可破译性
  - 不能在希望的时间内或实际可能的条件下求解

- 一个密码体制的安全性涉及两方面的因素：
  - 所使用的密码算法的**保密强度**
  - **密码算法之外**的不安全因素

4 对称密码体制的概念与分类

4.1 密码算法分类

- 按照保密的内容分:
  - **受限制的（restricted)算法**: 算法的保密性基于保持算法的秘密。
  - **基于密钥（key-based)的算法**: 算法的保密性基于对密钥的保密。

4.1 密码算法分类-ii

- 基于密钥的算法，按照密钥的特点分类：
  - **对称密码算法**（symmetric cipher）
    - 又称传统密码算法，或秘密密钥算法或单密钥算法。
    - 加密密钥和解密密钥相同，或实质上等同。
  - **非对称密码算法**（asymmetric cipher）
    - 又称**公开密钥算法**（public-key cipher）。
    - 加密密钥和解密密钥不相同，从一个很难推出另一个。
    - 用一个密钥进行加密，而用另一个进行解密。
      - **公开密钥**(public key)，简称**公钥**，可以公开，用于加密和签名验证；
      - **私人密钥**(private key)，简称**私钥**，必须保密，用于解密和生成签名。

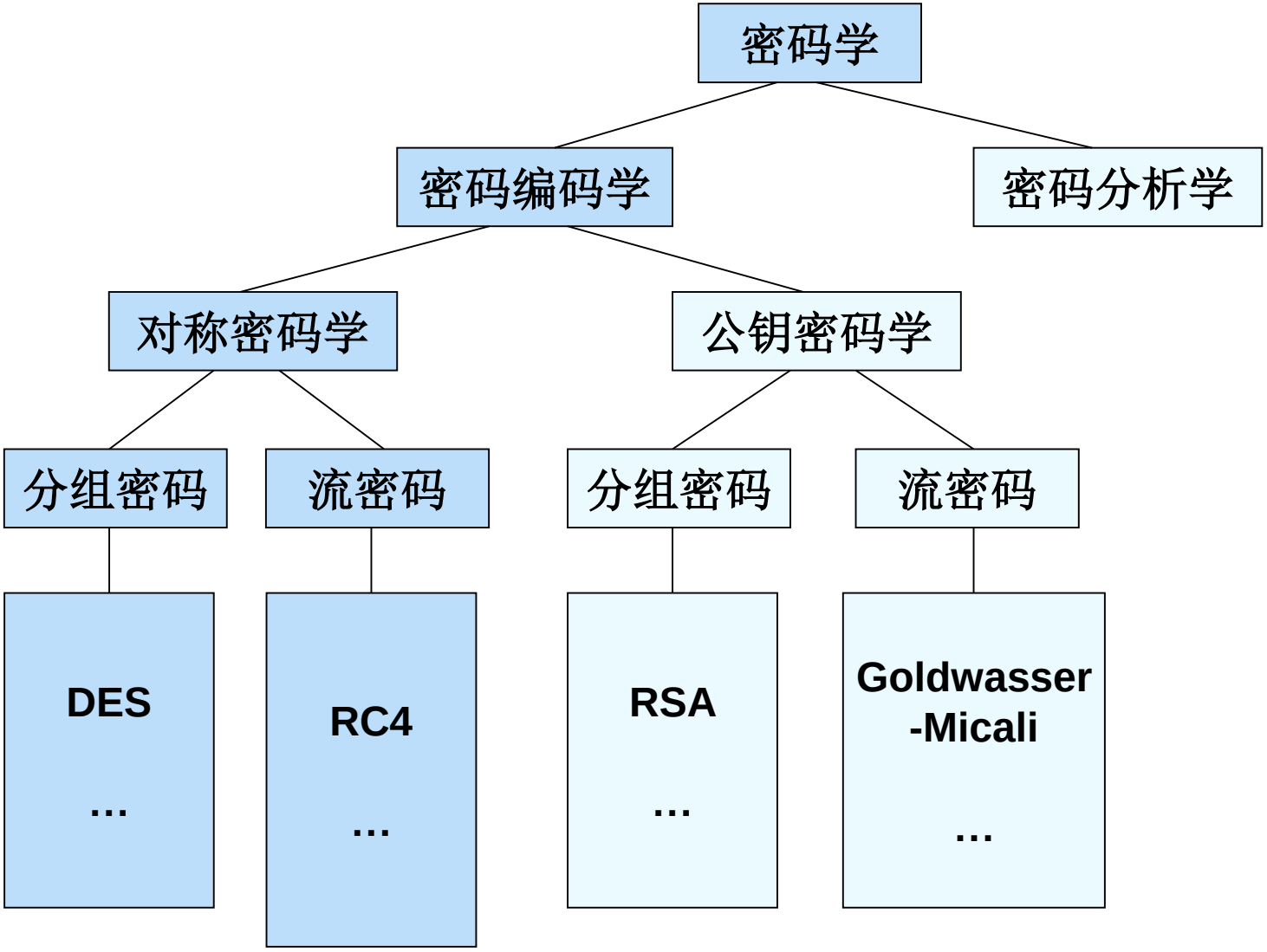
4.1密码算法分类-iii

- 按照明文的处理方法：
  - **分组密码（block cipher）**:将明文分成固定长度的组，用同一密钥和算法对每一块加密，输出也是固定长度的密文。
  - **流密码（stream cipher）**:又称**序列密码。** 序列密码每次加密一位或一字节的明文，也可以称为流密码。

序列密码

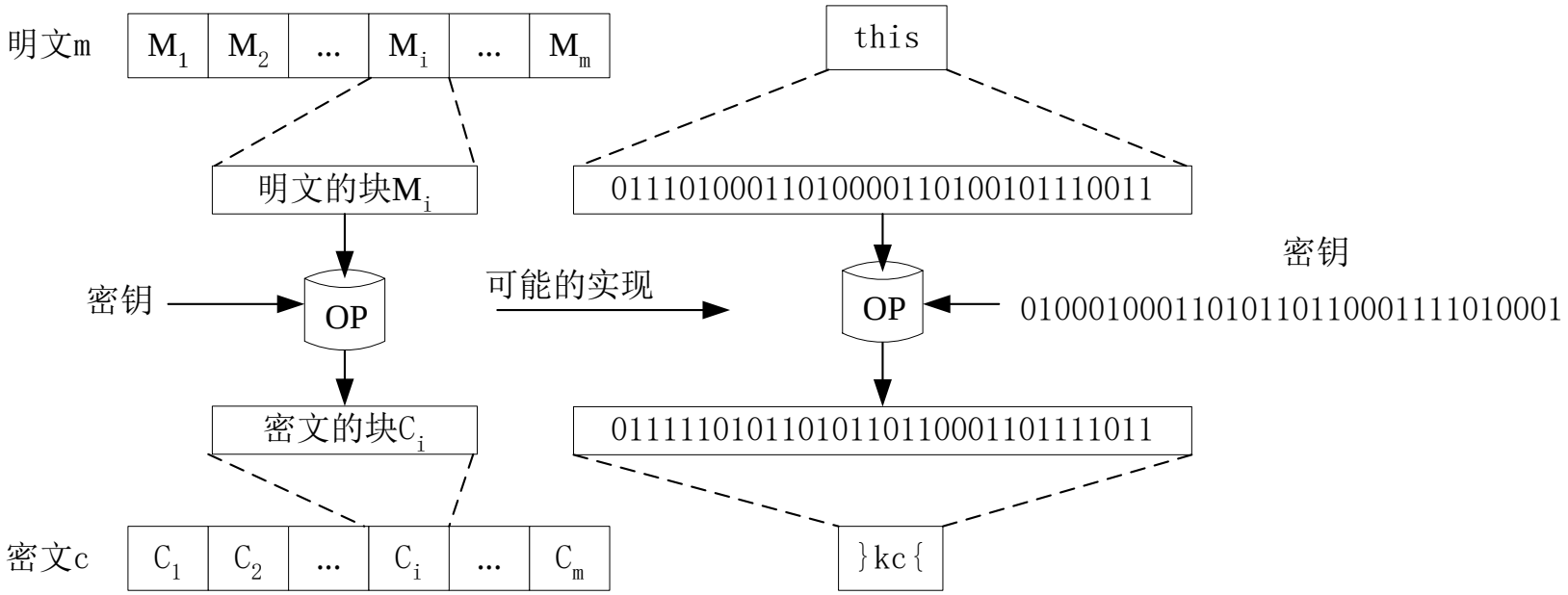
流密码是属于对称密码体制，
还是非对称密码体制？

4.1 密码算法分类


密码学
密码编码学
密码分析学
对称密码学
公钥密码学
分组密码
流密码
分组密码
流密码
DES
…
RC4
…
RSA
…
Goldwasser
-Micali
…

4.2 分组密码的概念

- **分组密码**（Block Cipher），也称之为**块密码**，将明文消息编码表示后的数字序列，划分成固定大小的组（或位块），各组分别在密钥的控制下变换成等长的输出数字序列。



4.2 分组密码的分类

- 分组密码分类
  - 加密方式的不同，分组密码又可分为三类：
    - 代替密码
    - 移位密码
    - 乘积密码
  - 根据加密和解密密钥密钥是否相同，分为：
    - 对称分组密码
    - 非对称分组密码

4.3 流密码的定义

- 流密码又称为序列密码，是指明文消息按字符（如二元数字）逐位地加密的一类密码算法。
  - 与它对应的是分组密码，是指将明文消息（含多个字符），逐组地进行加密。
  - 流密码算法：RC4，A5，SEAL，PKZIP…

4.3 流密码原理

- 将明文划分成字符（如单个字母），或其编码的基本单元（如0，1数字），分别与密钥流进行加密运算，解密时以同步产生的同样的密钥流实现。

设**明文**为 $x=x_0x_1x_2\ldots \quad x_i\in GF(2),i\geq 0$

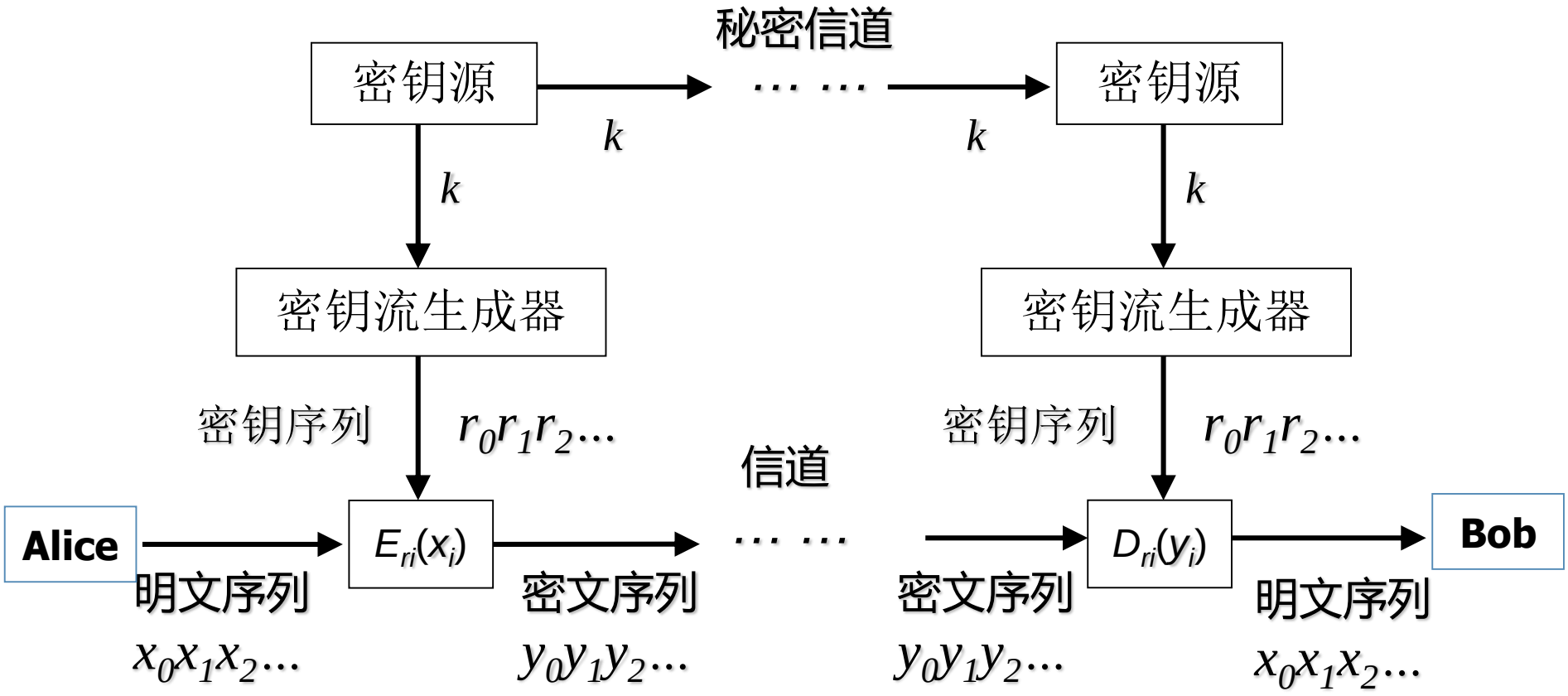
设**密钥**为 $r=r_0r_1r_2\ldots \quad r_i\in GF(2),i\geq 0$

设**密文**为 $y=y_0y_1y_2\ldots \quad y_i\in GF(2),i\geq 0$

则**加密变换**为 $y_i=\boldsymbol{E}_{\boldsymbol{r_i}}(x_i)\quad i\geq 0$

则**解密变换**为 $x_i=\boldsymbol{D}_{\boldsymbol{r_i}}(y_i)\quad i\geq 0$

4.4基于流密码体制的加密通信模型



分组密码与流密码的区别

- 定义上：
  - 分组密码是对大的明文数据块分组进行变换的操作
  - 流密码是对单个明文位的随时间变换的操作
- 实现上：
  - 分组密码易于用软件实现
  - 流密码易于用硬件实现
- 其它：
  - 相同的明文经过相同的分组密码将转换成为相同的密文
  - 相同的明文经过相同的流密码将转换成为不同的密文

4.5 分组密码的工作模式

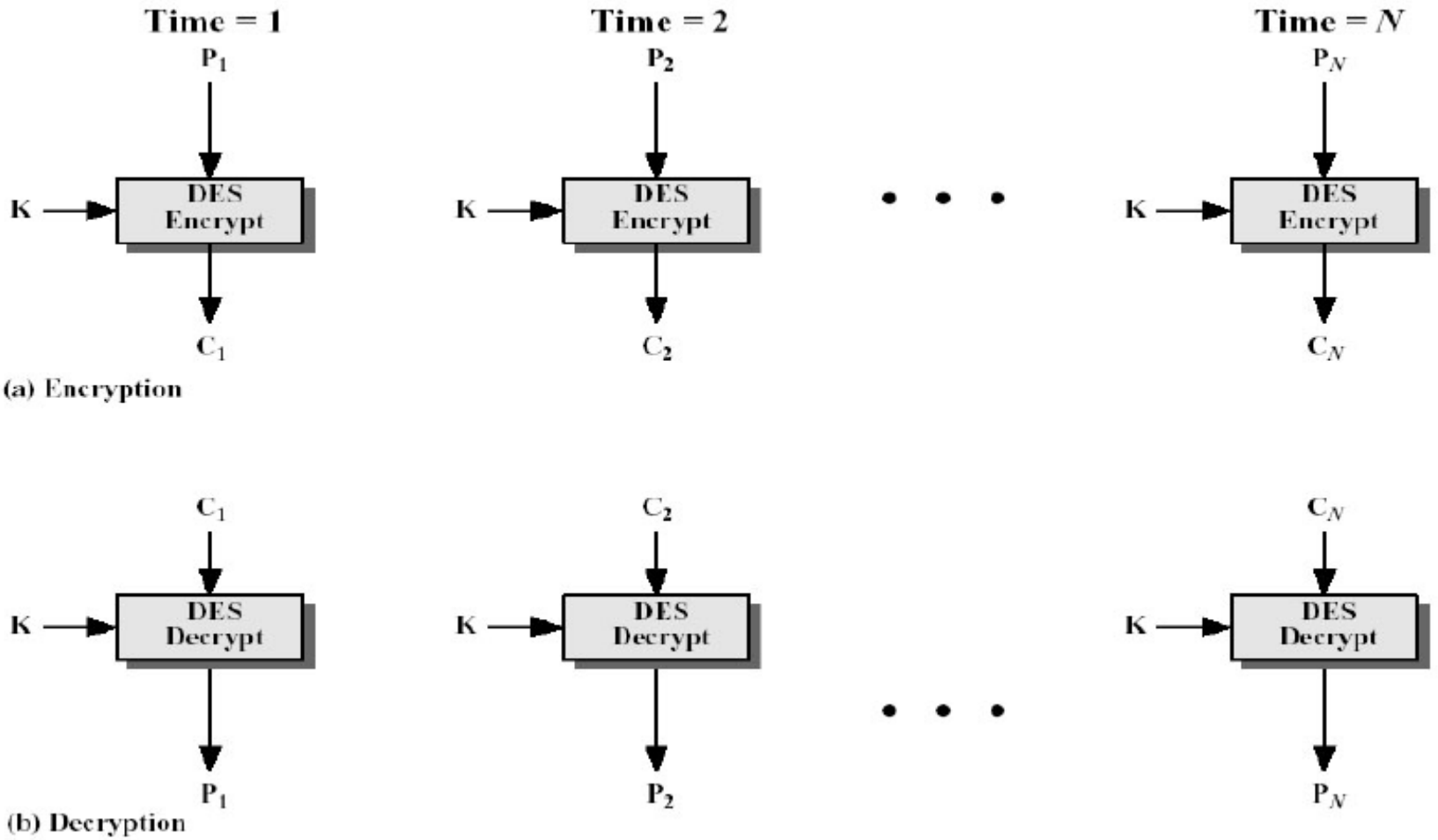
- 分组密码的工作模式，也称为密码模式，通常是由基本密码算法、一些反馈和一些简单运算组合而成。
- 特点：
  - 其安全性依赖于基本密码
  - 模式的效率将不会明显地低于基本密码
  - 不同的模式有不同的特点，适用于不同场合

4.5 分组密码的工作模式

- 电子密码本ECB (**E**lectronic **C**ode**B**ook)
- 密码分组链接CBC (**C**ipher **B**lock **C**haining)
- 密码反馈CFB (**C**ipher **F**eed**B**ack)
- 输出反馈OFB (**O**utput **F**eed**B**ack)
- 计数器模式CTR (**C**ou**T**e**R**)
- 带偏移的密码本OCB (**O**ffset **C**ode**B**ook)
- CCM模式 (**C**TR +**C**BC-**M**AC)
- GCM模式 (**G**alois/**C**ounter **M**ode)

电子密码本（ECB）

- $C_i = E_K(P_i) \Leftrightarrow P_i = D_K(C_i)$

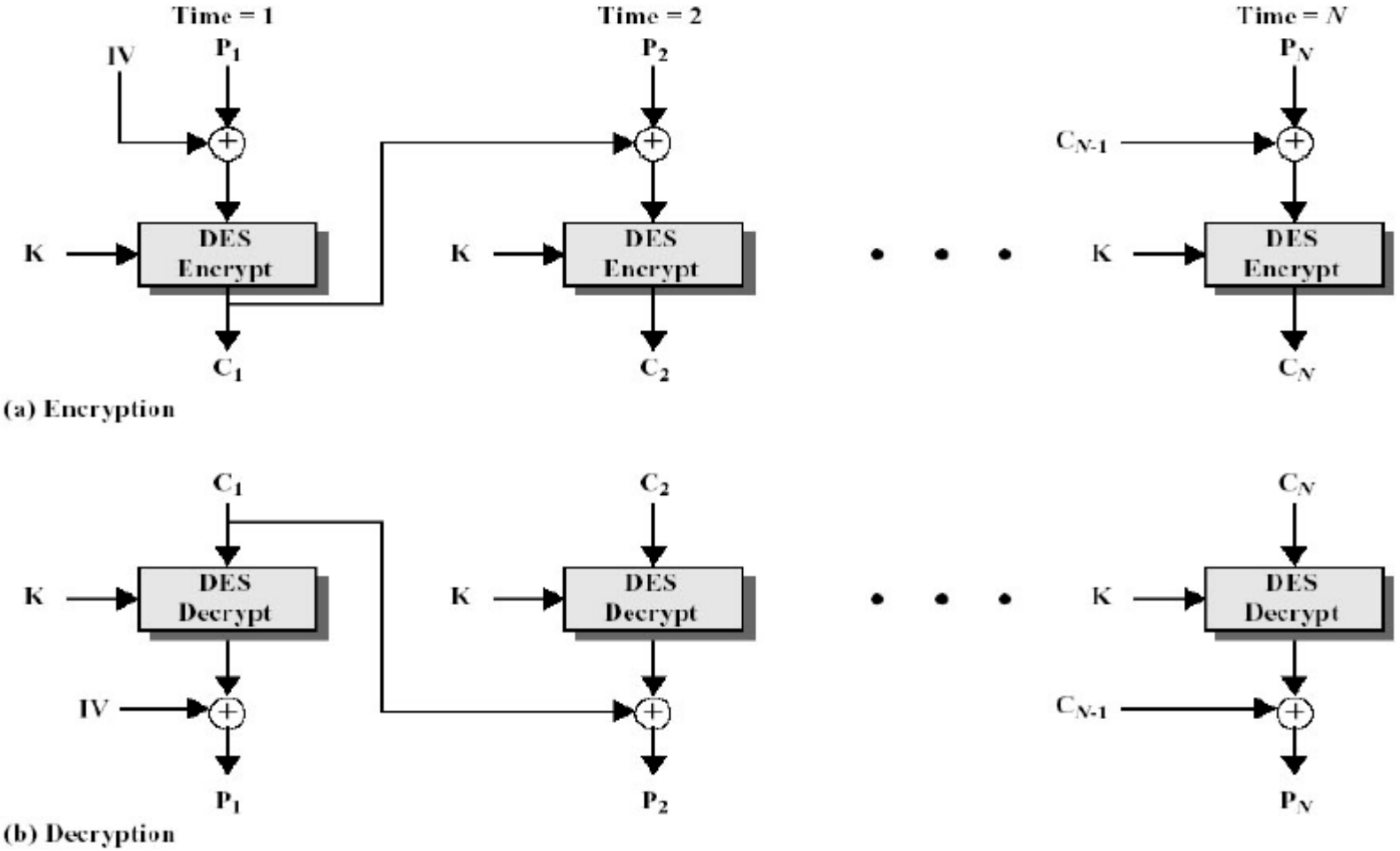


ECB特点

- 简单和有效
- 可以并行实现
- 不能隐藏明文的模式信息
  - 相同明文 ⇒ 相同密文
  - 同样信息多次出现造成泄漏
- 对明文的主动攻击是可能的
  - 信息块可被替换、重排、删除、重放
- 误差传递：密文块损坏 ⇒ 仅对应明文块损坏
- 适合于传输短信息

密码分组链接CBC

- $Ci=E_K(Ci-1 \oplus Pi) \Leftrightarrow Pi=D_K(Ci) \oplus Ci-1$



CBC特点

- 没有已知的并行实现算法
- 能隐藏明文的模式信息
  - 需要共同的初始向量IV
  - 相同明文➡不同密文
  - 初始向量IV可以用来改变第一块
  - 定期更换IV（如果需要传递IV，要保证IV的完整性）
- 对明文的主动攻击是不容易的
  - 信息块不容易被替换、重排、删除、重放
  - 误差传递：密文块损坏➡两明文块损坏
- 安全性好于ECB
- 适合于传输长度大于64位的报文

密码反馈CFB

- CFB:分组密码 ⟹ 自同步流密码

Si 为移位寄存器, j为流单元宽度

加密: $C_i = P_i \oplus (E_K(S_i)$的高j位)

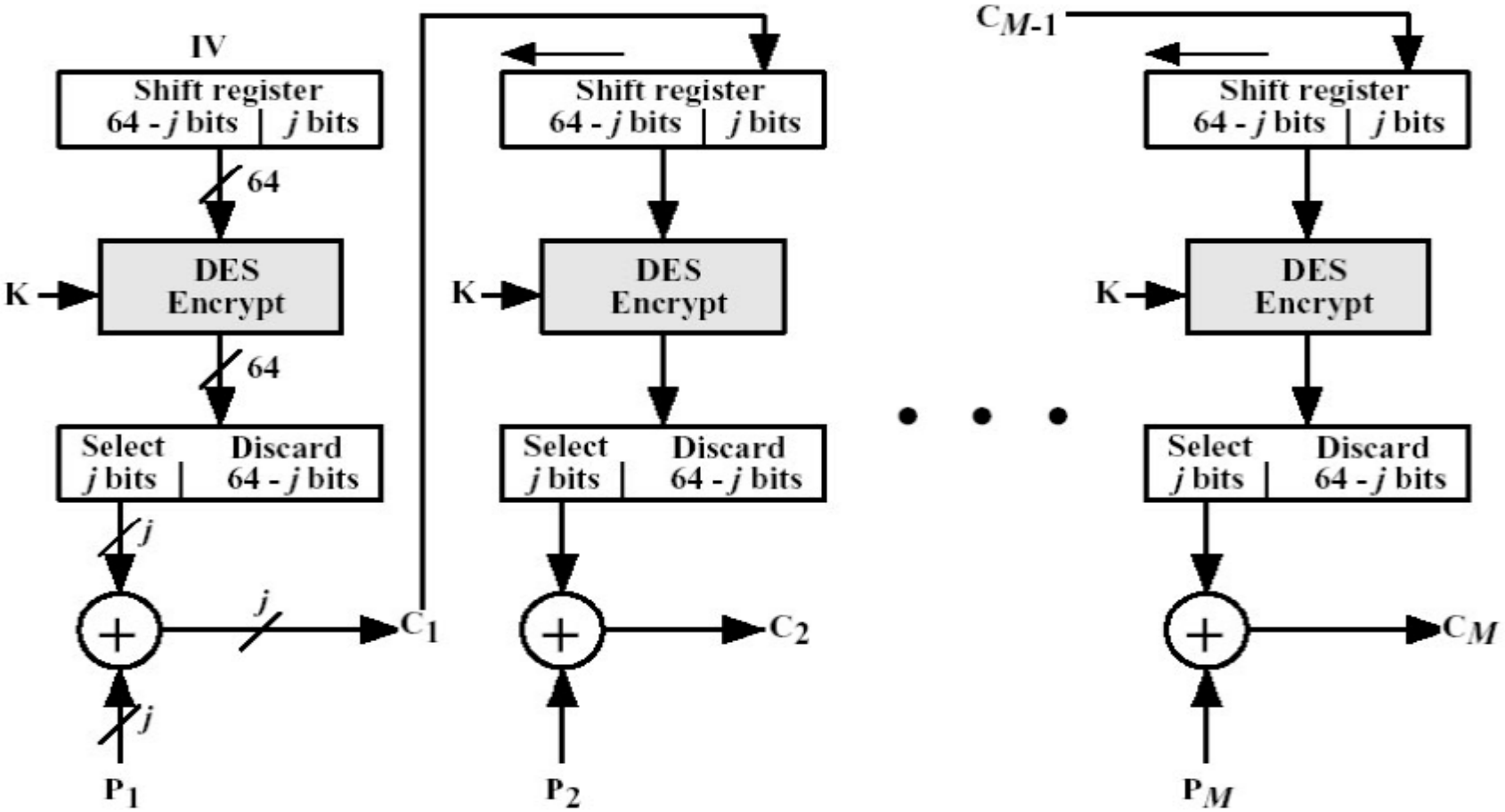
$S_{i+1} = (S_i << j) | C_i$

解密: $P_i = C_i \oplus (E_K(S_i)$的高j位)

$S_{i+1} = (S_i << j) | C_i$

CFB加密示意图

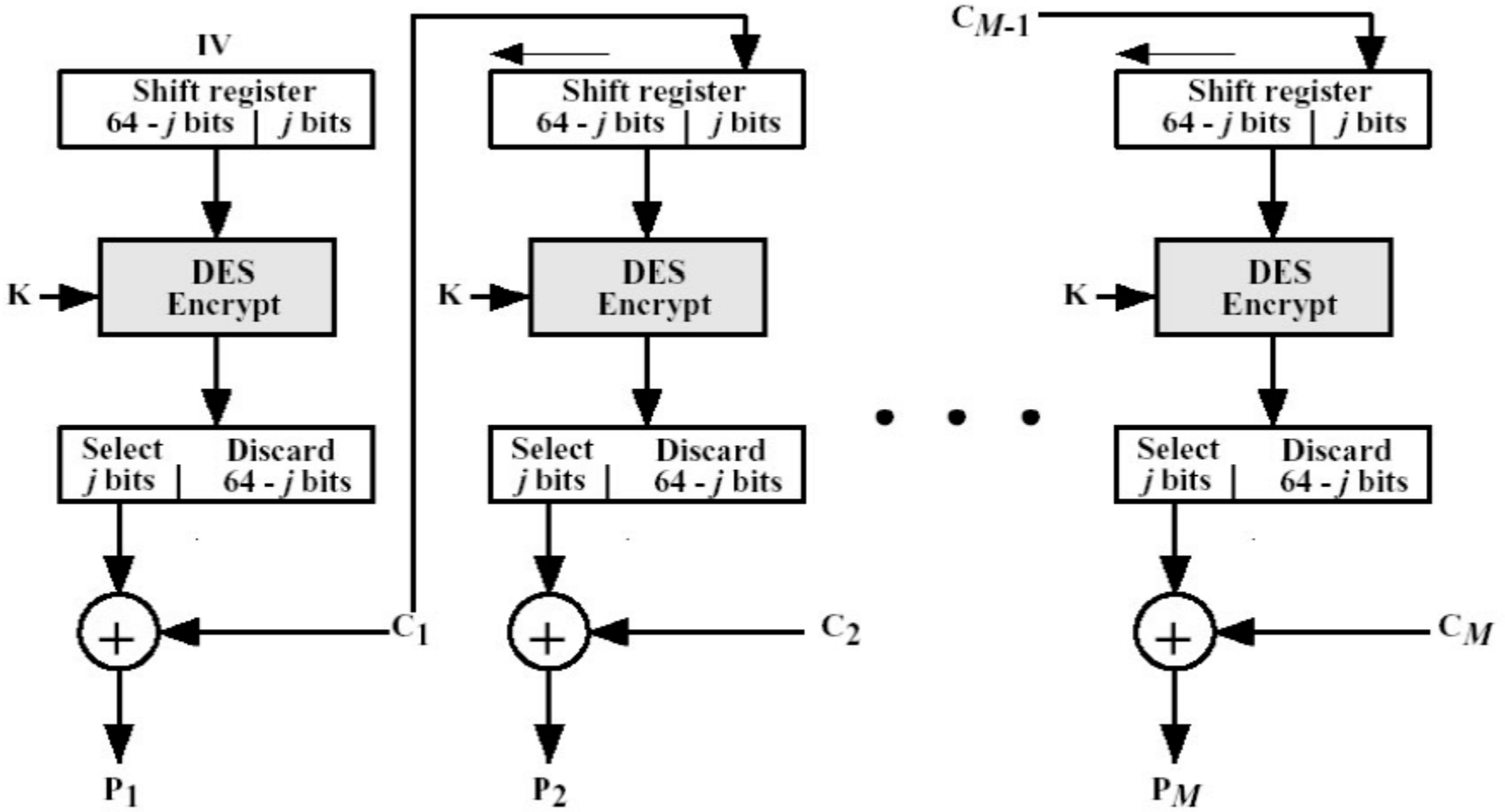
- $C_i = P_i \oplus (E_K(S_i)$的高j位$)$；$S_{i+1} = (S_i << j) | C_i$



(a) Encryption

CFB解密示意图

- Pi=Ci⊕(EK(Si)的高j位);



(b) Decryption

CFB特点

- 分组密码 ⟹ 自同步流密码
- 没有已知的并行实现算法
- 关于IV
  - 为进行正常的加密和解密，发送与接收需要共同IV
  - IV应具有唯一性（即在密钥不变时，每次加密使用不同的IV）
- 优点：隐藏了明文模式
- 缺点：
  - 误差传递，一个单元损坏影响多个单元

输出反馈OFB

- OFB:分组密码 ⟹ 同步流密码

  Si 为移位寄存器, j为流单元宽度

  加密: $C_i = P_i \oplus (E_K(S_i)$的高j位)

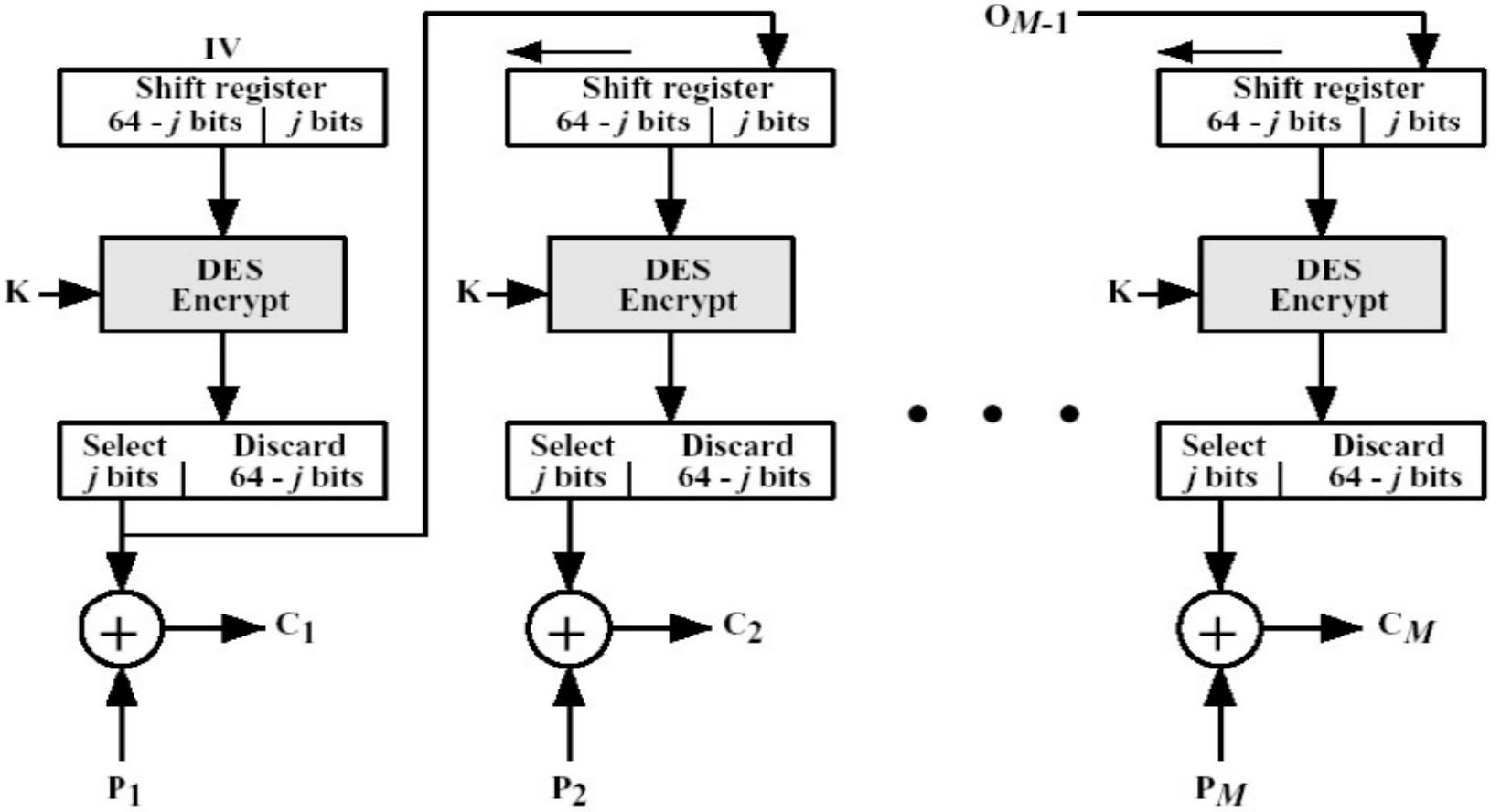
  $S_{i+1} = (S_i << j) | (E_K(S_i)$的高j位)

  解密: $P_i = C_i \oplus (E_K(S_i)$的高j位)

  $S_{i+1} = (S_i << j) | (E_K(S_i)$的高j位)

OFB加密示意图

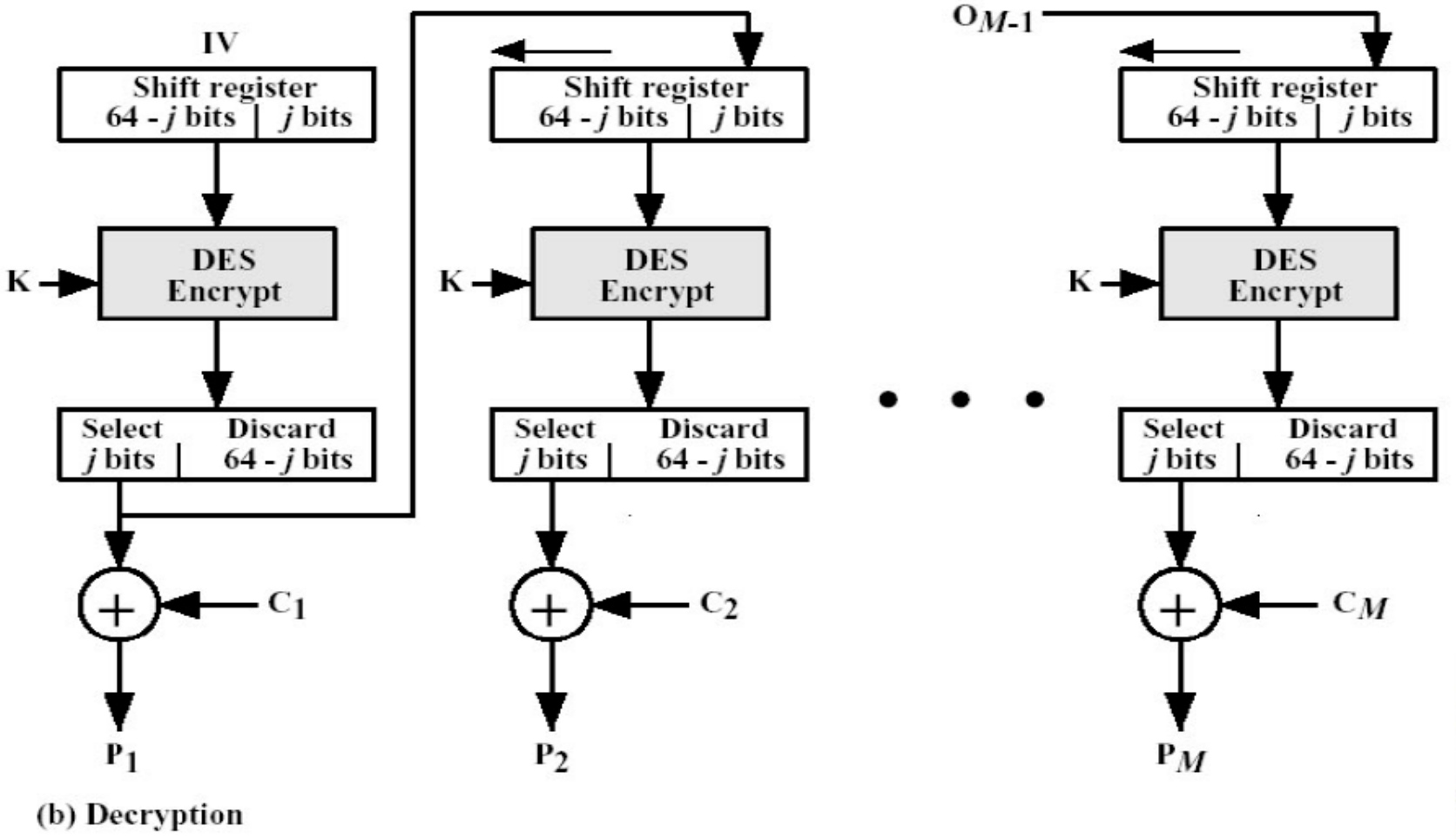
Ci =Pi ⊕ (EK(Si)的高j位);Si+1=(Si<<j)|(EK(Si)的高j位)



(a) Encryption

OFB解密示意图

$P_i=C_i \oplus (E_K(S_i)$的高j位$)$; $S_{i+1}=(S_i<<j)|(E_K(S_i)$的高j位$)$



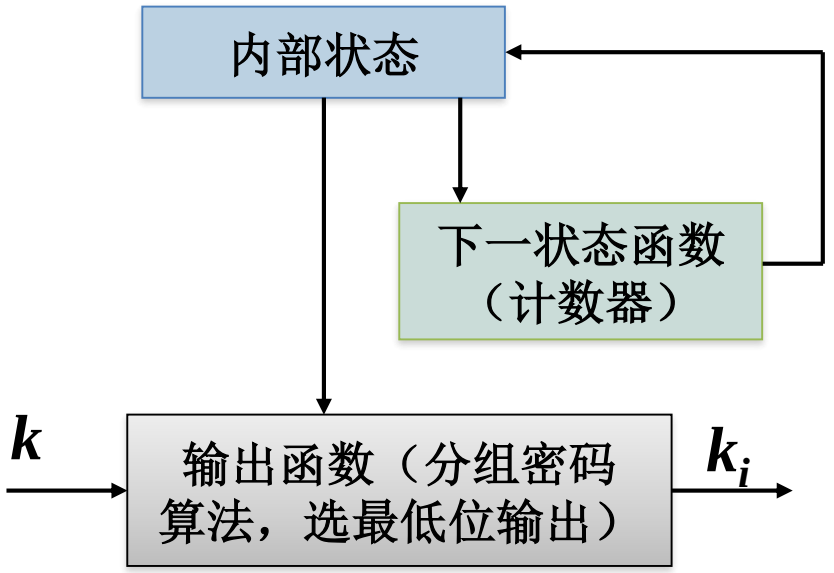
(b) Decryption

OFB特点

- OFB:分组密码 ⟹ 同步流密码
- 没有已知的并行实现算法
- 优点：
  - 隐藏了明文模式
  - 没有误差传递：一个单元损坏只影响对应单元
- 缺点：
  - 不具有自同步能力，要求系统要保持严格的同步
  - 重新同步时需要新的IV，IV可以用明文形式传送
- 对明文的主动攻击是可能的
  - 信息块可被替换、重排、删除、重放
- 安全性较CFB差

计数器模式(Counter Mode)

- Diffie等人在1979年提出
- 将一个计数器输入到寄存器中。每一个分组完成加密后，计数器都要增加某个常数，典型值是1。
- 该模式的同步和错误扩散特性同OFB模式完全一样。
- 可直接生成第i 个密钥比特ki；保密随机访问数据文件时是非常有用



CTR模式

- **特点**
  - 硬件效率：允许同时处理多块明/密文
  - 软件效率：允许并行计算
  - 预处理
  - 随机访问
  - 可证明安全性：能够证明CTR至少和其他模式一样安全
  - 简单性：只需要实现加密算法
  - 无填充：可以高效地作为密钥流使用。

OCB模式

- OCB模式通过使用同一个密钥对数据的一次处理，同时提供了加密和数据完整性检测。
- WPA2.0中使用基于128位的AES在OCB（offset CodeBook）模式，实现WRAP中的数据机密性与完整性保护机制。
- OCB模式优点:
  - 并行处理
  - 非常高效
  - 可以证明是可靠的

OCB模式

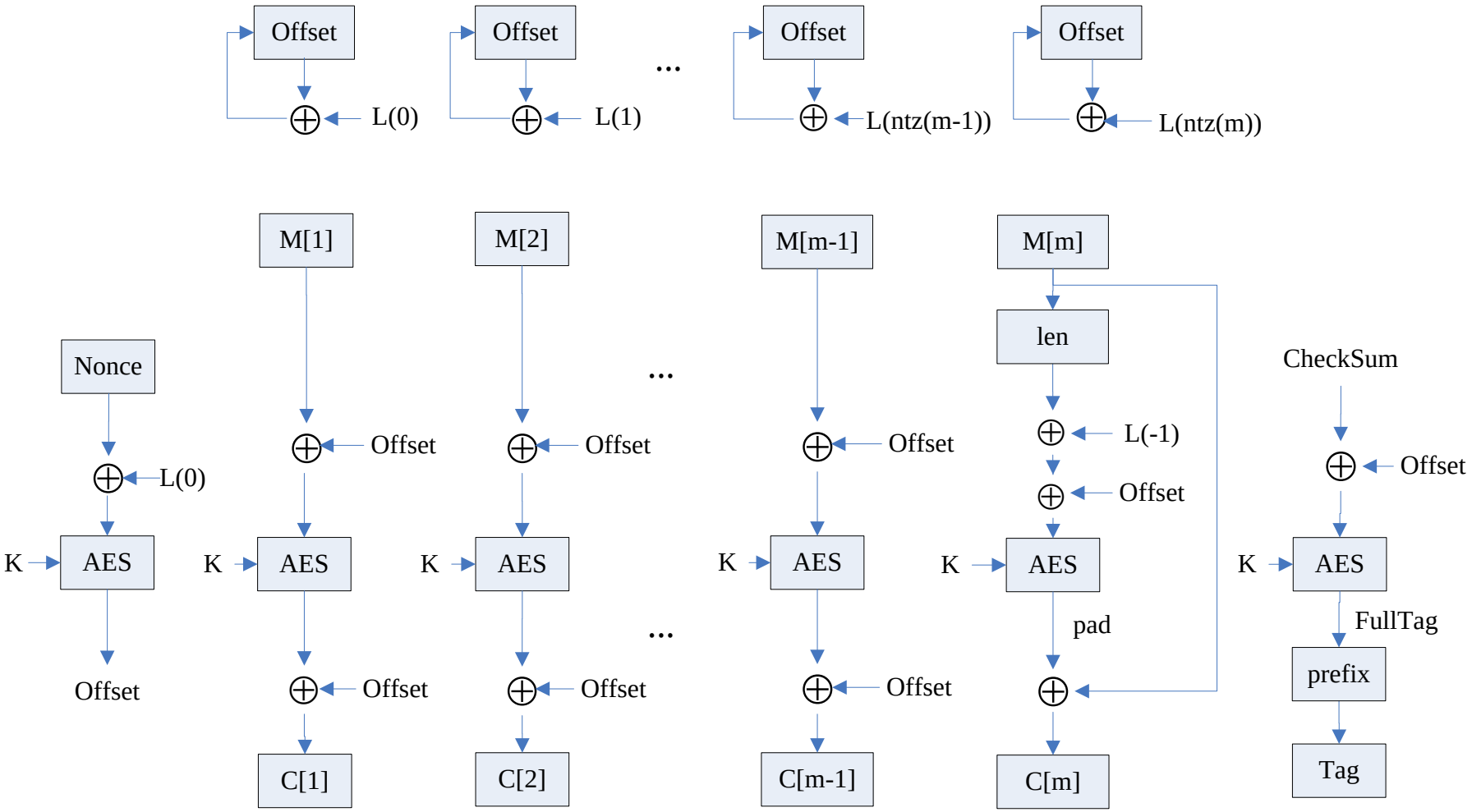
预先计算：

1) $L(0) = E_k(0)$;

2) $L(-1) = lsb(L(0))?(L(0) >> 1) \oplus const43 : (L(0) >> 1)$;

3) For i>0， L(i)=lsb(L(i-1))?(L(i-1)<<1) ⊕ const87: (L(i-1)<<1);

- 加密



GCM模式

- GCM被用于TLS1.3
- GCM中的基础算法
  - GHASH: 利用“伽罗瓦域算法”计算比特串X 的HASH值；
  - $GCTR_K(ICB,X)$: 对比特串X和初始分组ICB利用密钥K计算其密文；
  - $CIPH_K(X)$: 对于密钥为K,分组为X 的分组密文的输出。

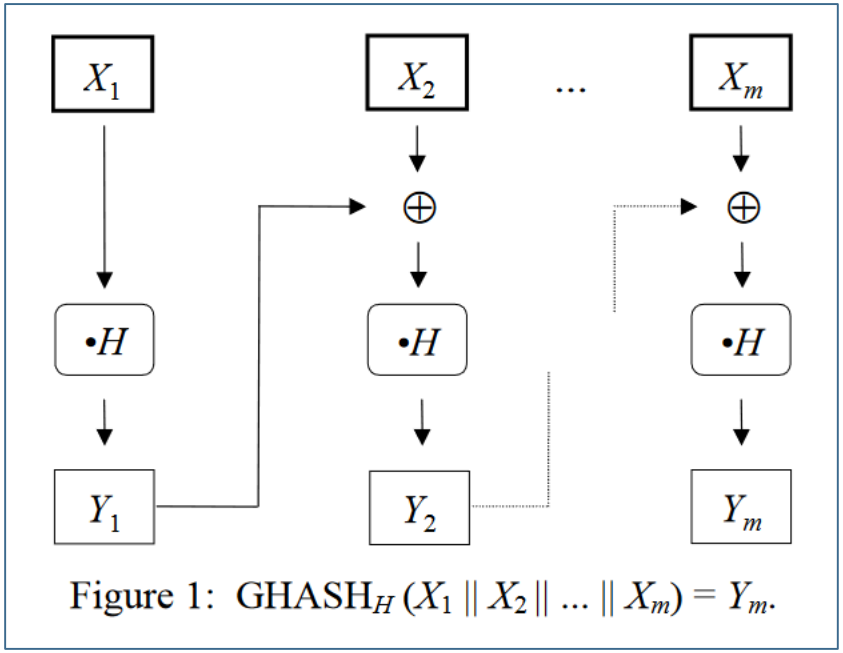


Figure 1: $GHASH_H(X_1 \| X_2 \| \ldots \| X_m) = Y_m$.

GCM模式

- GCM被用于TLS1.3
- GCM中的基础算法
  - GHASH
  - $GCTR_K(ICB,X)$: 对比特串X和初始分组ICB利用密钥K计算其密文；

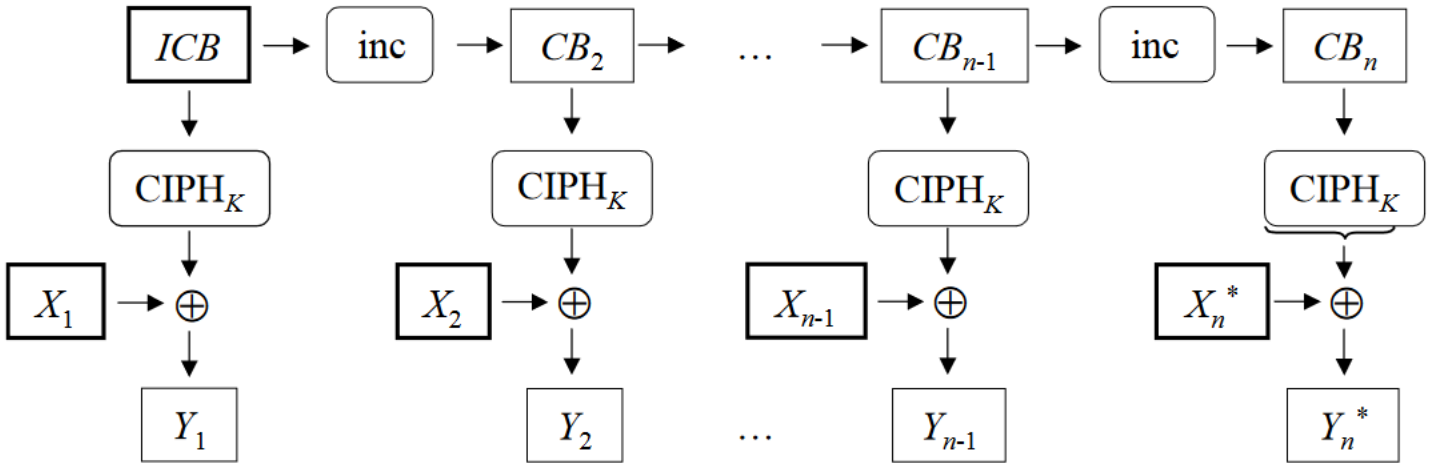


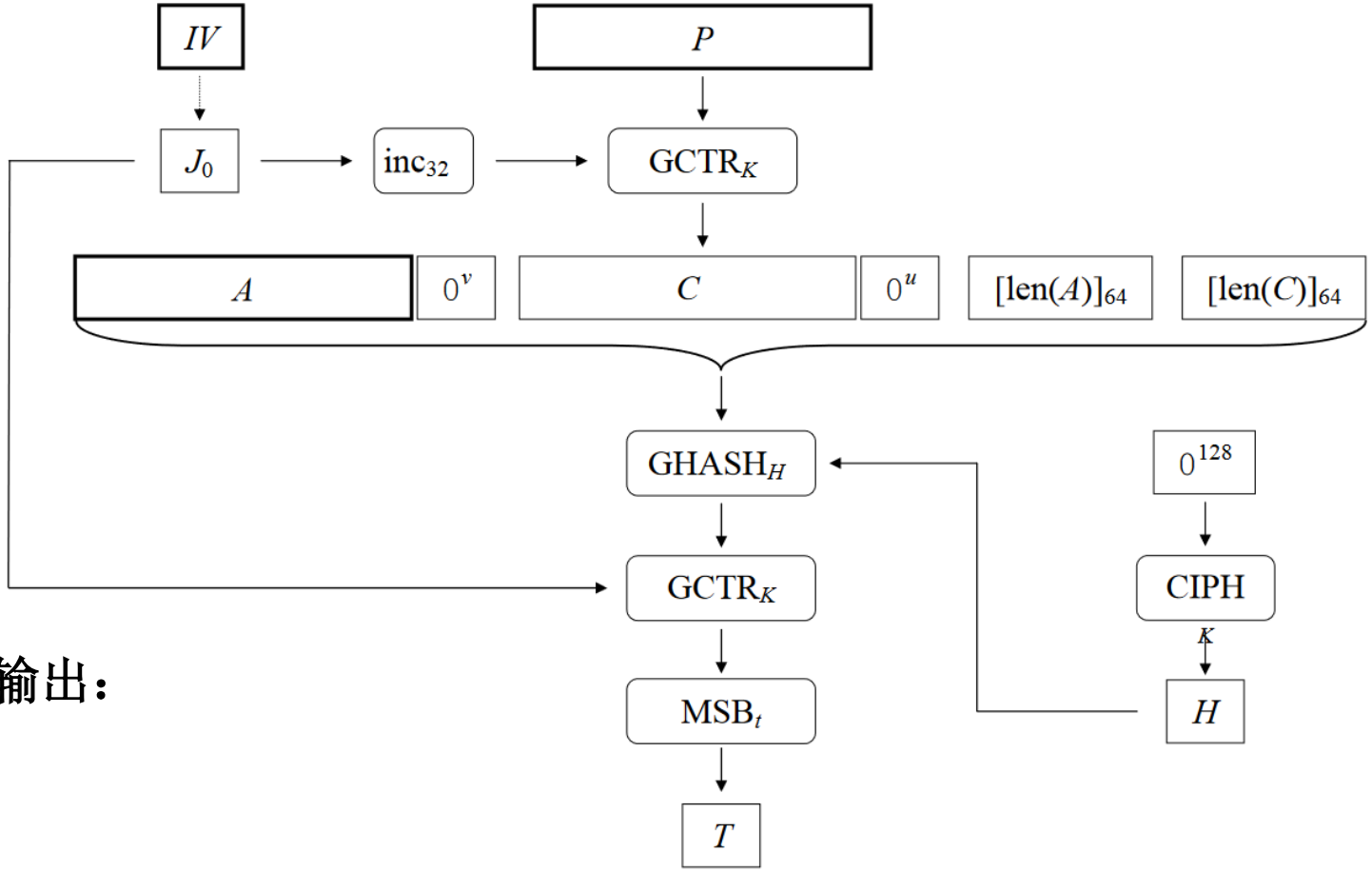
Figure 2: $GCTR_K(ICB, X_1 \| X_2 \| \ldots \| X_n^*) = Y_1 \| Y_2 \| \ldots \| Y_n^*$.

GCM模式

➢**发送方流程，** $CIPH_K(X)$: 对于密钥为K,分组为X 的分组密文的输出。

GCM加密算法的输入：

1. 初始化向量IV
2. 明文P
3. 关联数据A
4. 密钥K



GCM加密算法的输出：

1. 密文C
2. 校验值T

GCM模式

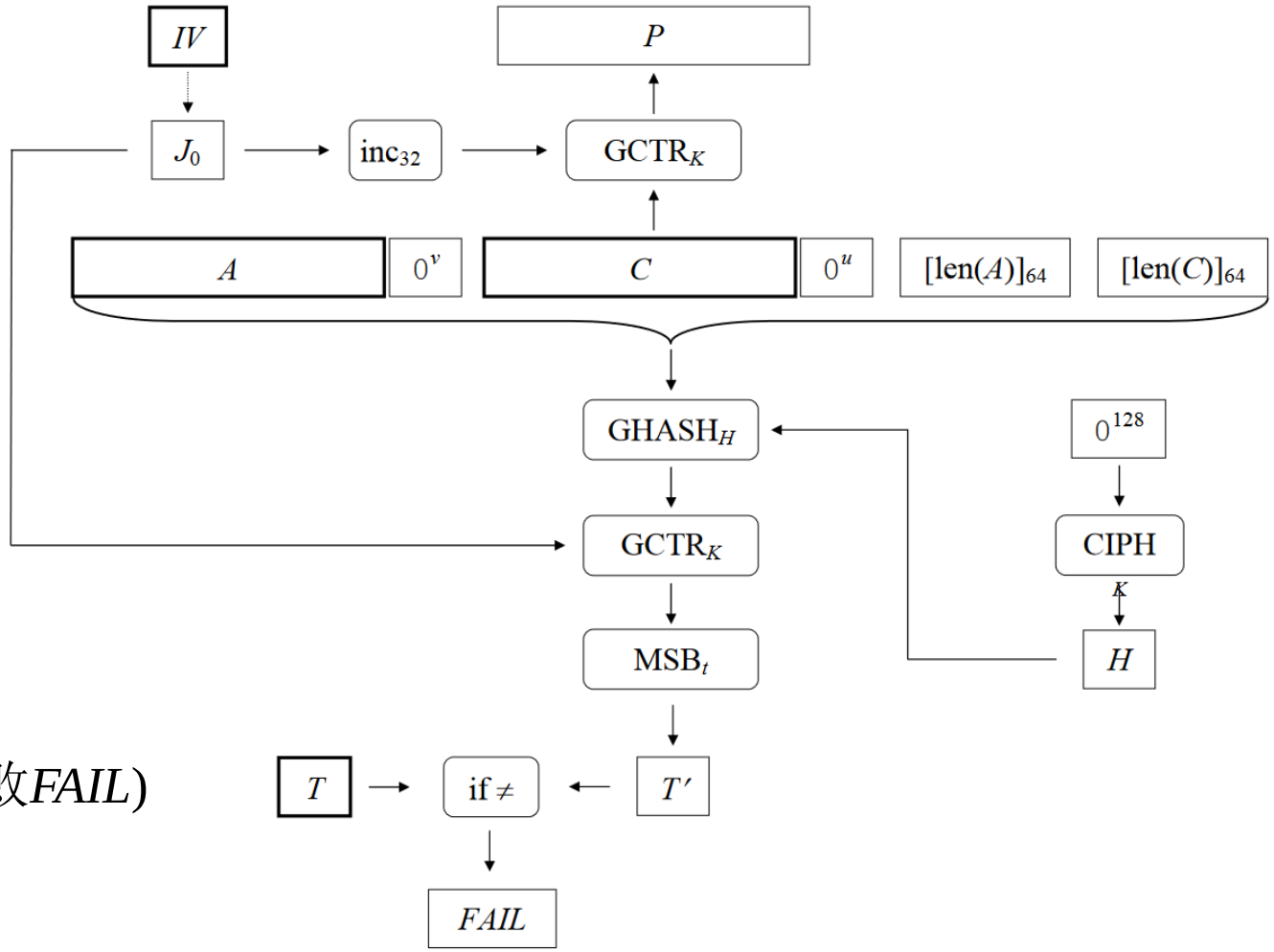
➢接收方流程

GCM解密算法输入：

1. 初始化向量IV
2. 密文C
3. 关联数据A
4. 认证值T
5. 密钥K

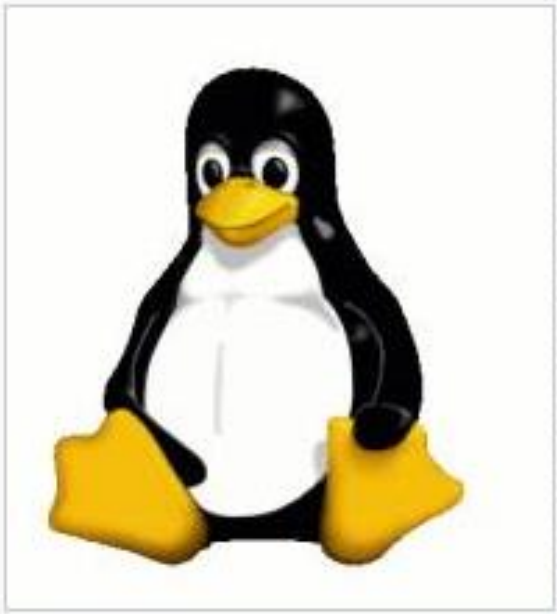
GCM加密算法的输出：

1. 明文P(或者验证失败*FAIL*)

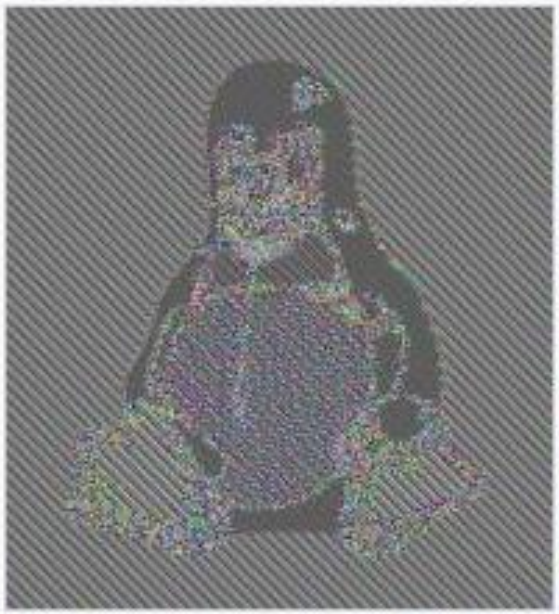


来自：https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf

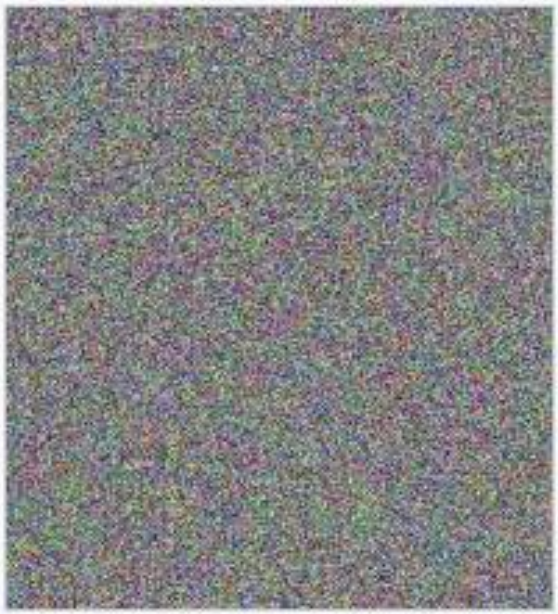
不同模式加密的例子



Original image



Encrypted using ECB mode



Modes other than ECB result in pseudo-randomness

The image on the right is how the image might appear encrypted with CBC, CTR or any of the other more secure modes—indistinguishable from random noise. Note that the random appearance of the image on the right does not ensure that the image has been securely encrypted; many kinds of insecure encryption have been developed which would produce output just as "random-looking".

注：https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

4.6 工作模式选用原则

- **ECB模式**，简单、高速，但最弱，易受重发攻击，一般不推荐；
- **CBC,CFB,OFB,CTR的选择取决于实用特殊考虑**；
- **CBC适用于文件加密**，但较ECB慢，且需要另加移存器和组的异或运算，但安全性加强。软件加密最好选用此种方式；
- **OFB和CFB较CBC慢许多**，每次迭代只有少数bit完成加密。**若可以容忍少量错误扩展，可选CFB。否则，可选OFB或CTR**;
- **在字符为单元的流密码种多选CFB模式**，如终端和主机间通信。而**OFB或CTR用于高速同步系**统，不容忍差错传播。
- **OCB模式**通过使用同一个密钥对数据的一次处理，同时提供了加密和数据完整性检测，**适合保护长数据。**

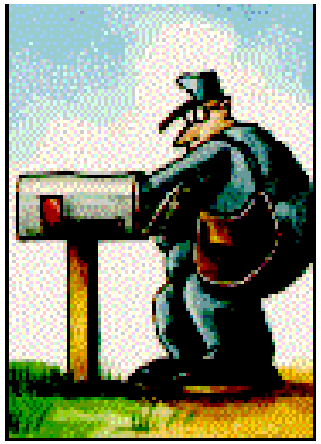
5 非对称密码体制

对称密码体制的缺陷

- **密钥管理问题** 在有多个用户的网络中，任何两个用户之间都需要有共享的秘密钥，当网络中的用户*n*很大时，需要管理的密钥数目是非常大 n（n-1）/2；而公钥体制只需要n对公/私钥；
- **无签名功能** 当主体A收到主体B的电子文挡（电子数据）时，无法向第三方证明此电子文档确实来源于B。

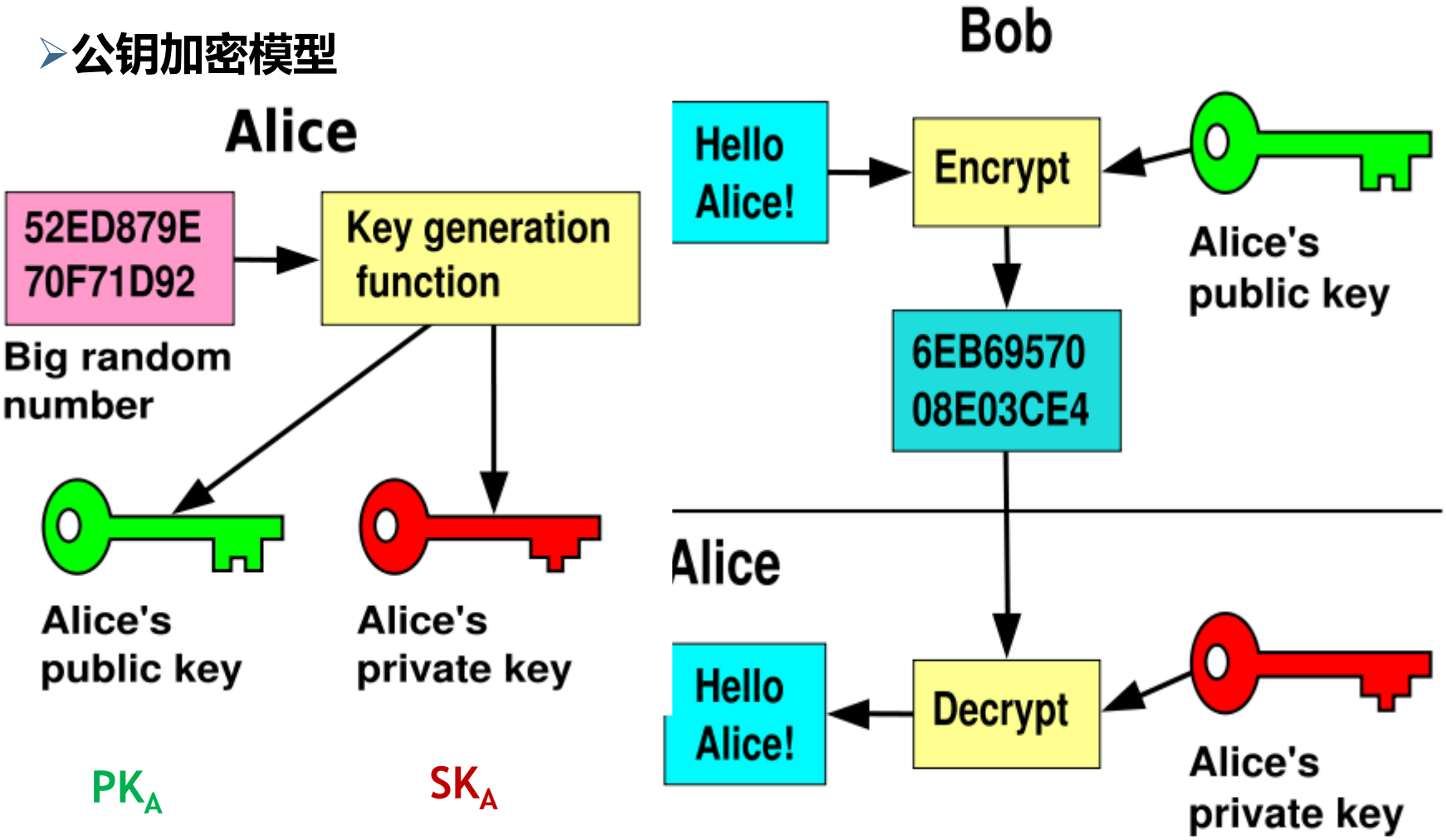
邮箱的例子

- 任何人可以向邮箱投举报信
- 用户(审计人员)才能打开邮箱，读信的内容



5 非对称密码体制

➢公钥加密模型



5 非对称密码体制--原理

参数生成过程：

1）要求接收消息的端系统，产生一对用来加密和解密的密钥，如图中的接收者A，产生一对密钥PK_A，SK_A，其中PK_A是公开密钥，SK_A是秘密密钥.

2）接收者A将加密密钥（图中的PK_A）予以公开，另一密钥被保密 (图中的SK_A).

参数生成需满足的要求：

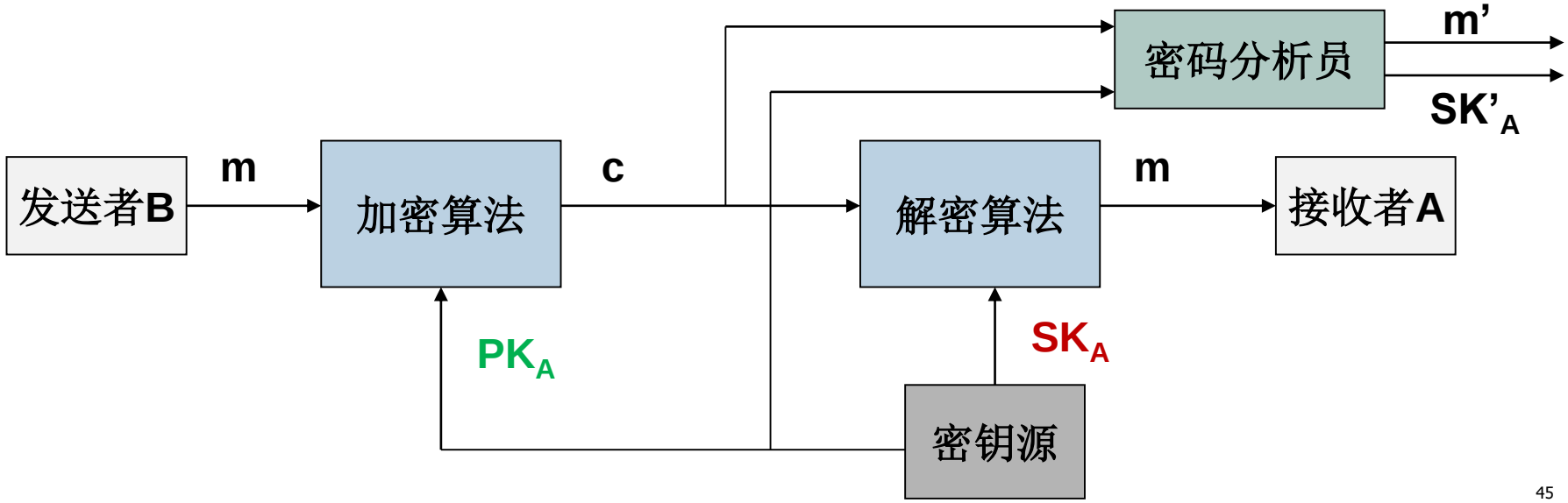
- 由私钥及公开参数容易计算出公开密钥；
- 由公钥及公开参数推导私钥是困难的；

5 非对称密码体制--原理

加解密过程：

1) B要想向A发送消息m，则使用A的公开钥PK_A加密m，表示为$c=E_{PKA}[m]$，其中c是密文，E是加密算法。

2) A收到密文c后，用自己的秘密钥SK_A解密，表示为$m=D_{SKA}[c]$，其中D是解密算法。



公钥加密体制框图

5 非对称密码体制--原理

加解密算法需满足要求：

- 利用公钥及公开参数加密明文容易计算；
- 利用私钥及公开参数解密密文容易计算；
- 只利用公钥解密密文困难；
- 加、解密次序可换，即

$$E_{PKA}[D_{SKA}(m)]=D_{SKA}[E_{PKA}(m)]$$

这一条虽然非常有用，但不是对所有的算法都作要求。

典型公钥密码算法

- RSA算法、ElGamal算法、椭圆曲线密码算法等

6 数字签名

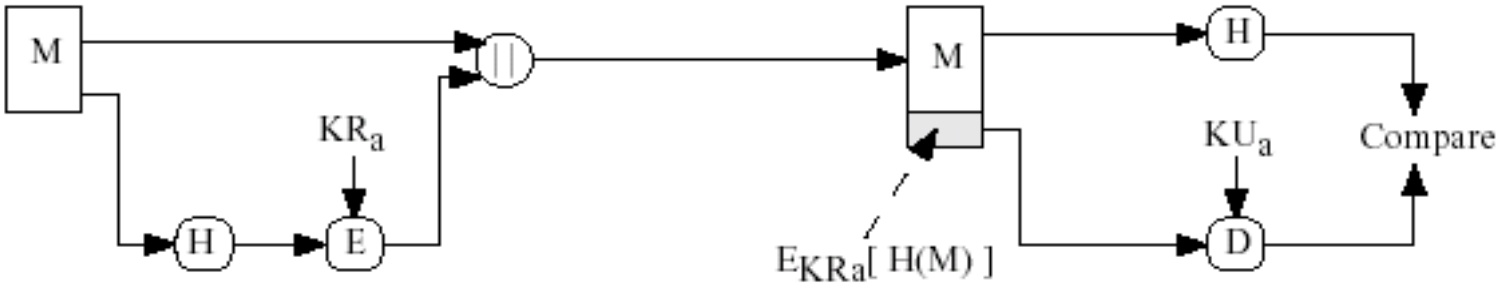
- 传统签名的基本特点:
  - 能与被签的文件在物理上不可分割
  - 签名者不能否认自己的签名
  - 签名不能被伪造
  - 容易被验证
- 数字签名是传统签名的数字化，基本要求:
  - 能与所签文件“绑定”
  - 签名者不能否认自己的签名
  - 签名不能被伪造
  - 容易被验证

6 数字签名方案

- 基本流程
  - 先对消息M作一个摘要H(M)
  - 然后发送方用自己的私钥对H(M)进行加密，得到签名 $E_{KRa}(H(M))$
  - 连同消息M一起，发送出去
  - B收到复合的消息之后，把签名提取出来
  - B用A的公钥对签名解密得到H’
  - B计算所收到消息的摘要H(M’)
  - 如果H’=H(M’)，则消息确实是A产生的
- 问题
  - 公钥的管理，公钥与身份的对应关系
  - 签名的有效性，私钥丢失?

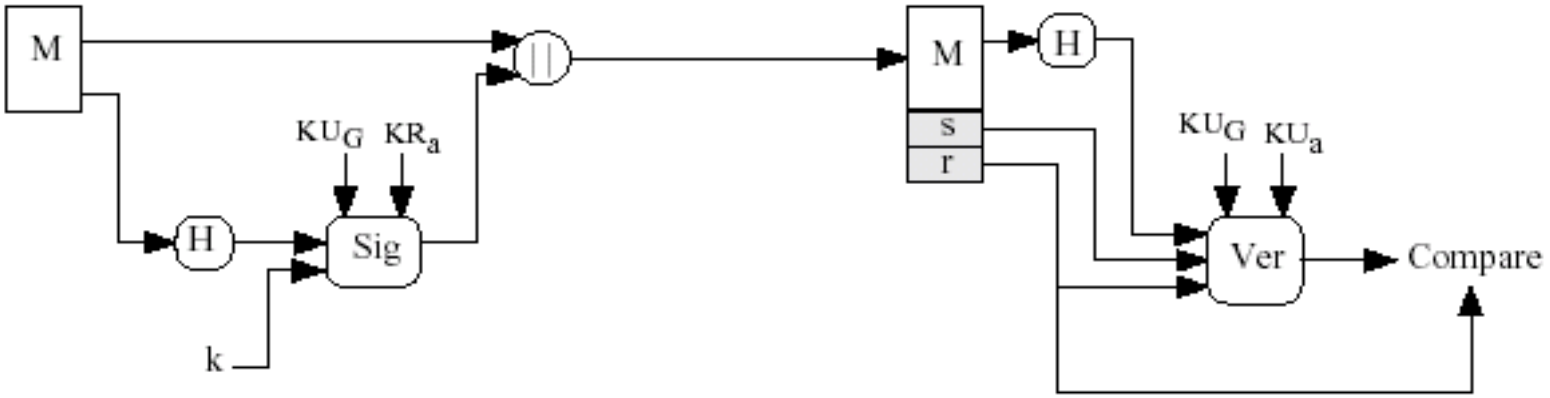
6 两种数字签名方案

A



(a) RSA Approach

A



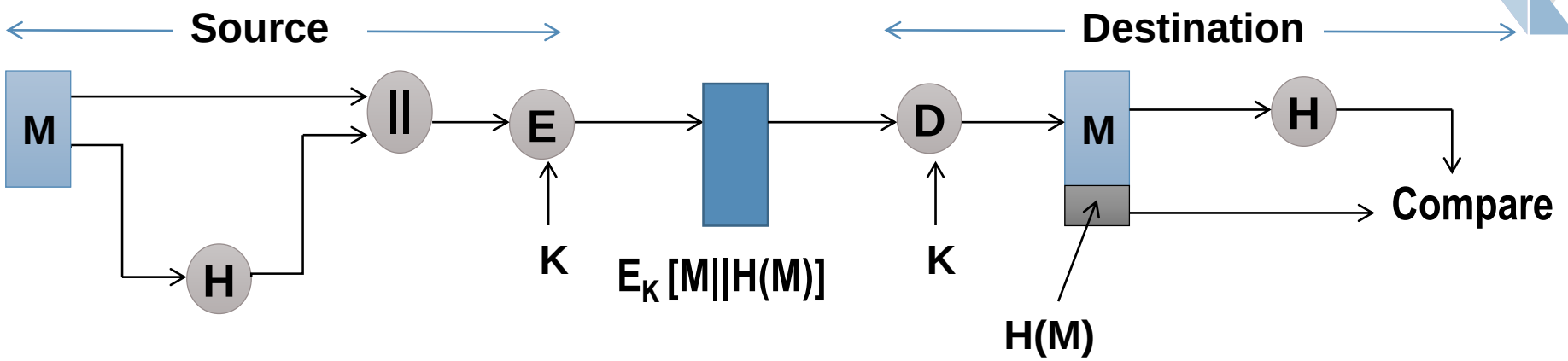
(b) DSS Approach

注：KU_G全局公共密钥；K一个随机数

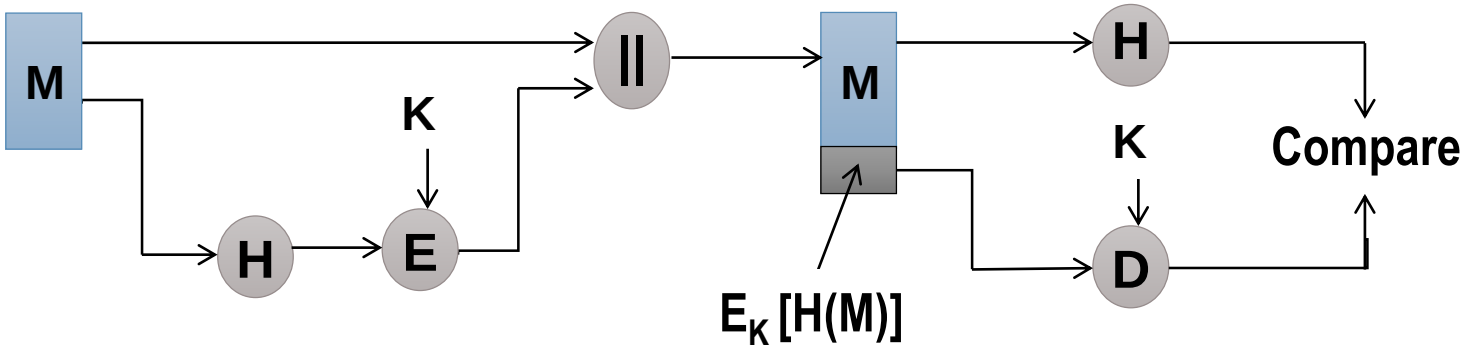
7 散列函数Hash Function

- **H(M):** 输入为任意长度的消息M; 输出为一个固定长度的散列值，称为**消息摘要**(MessageDigest）。
- 这个散列值是消息M的所有位的函数并提供错误检测能力：消息中的任何一位或多位的变化都将导致该散列值的变化。
- 又称为**哈希函数**、**数字指纹**（Digitalfinger print)。

7 散列函数的基本用法(a、b)

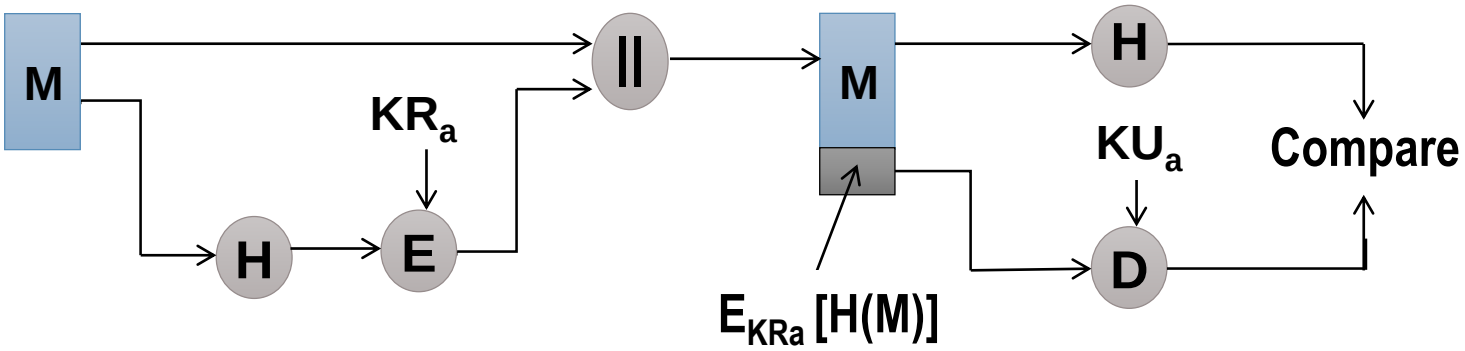


(a) **Provides confidentiality - - only A and B share K**
Provides authentication - -H(M) is cryptographically protected



(b) **Provides authentication - -H(M) is cryptographically protected**

7 散列函数的基本用法(c)

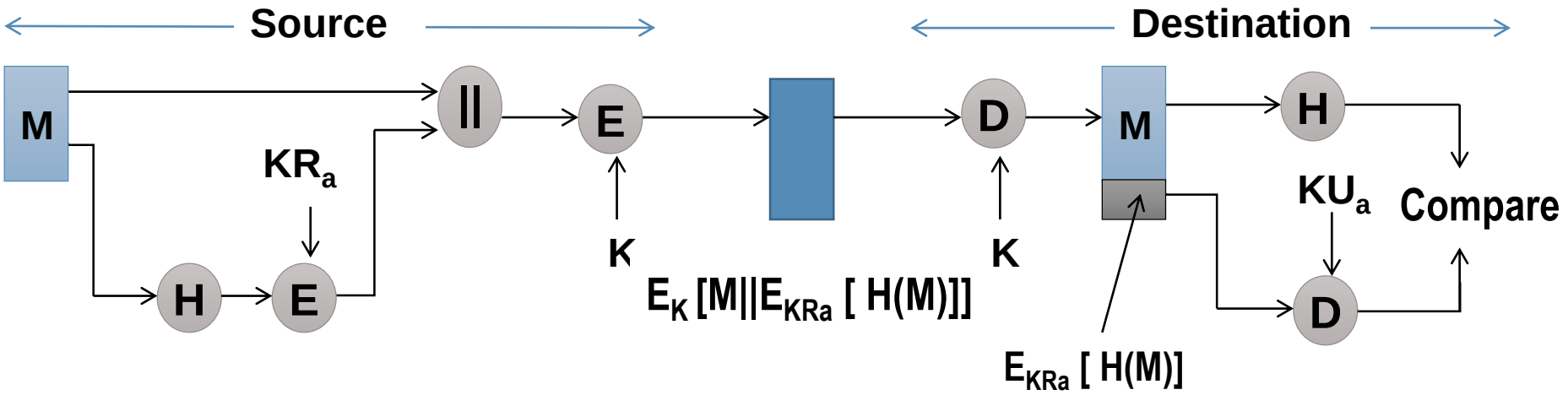


(c) Provides authentication and digital signature

-- H(M) is cryptographically protected

-- only A could create $E_{KRa}[H(M)]$

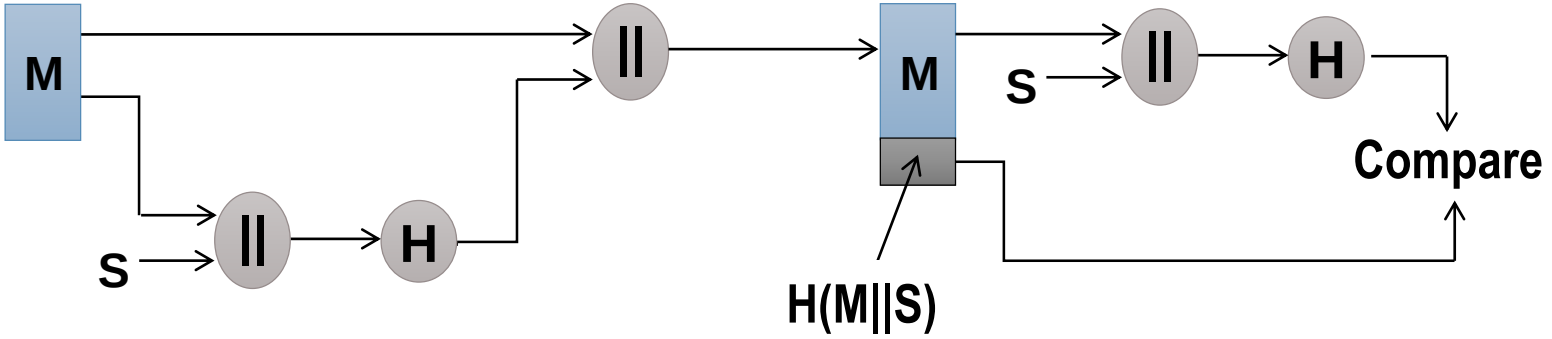
7 散列函数的基本用法(d)



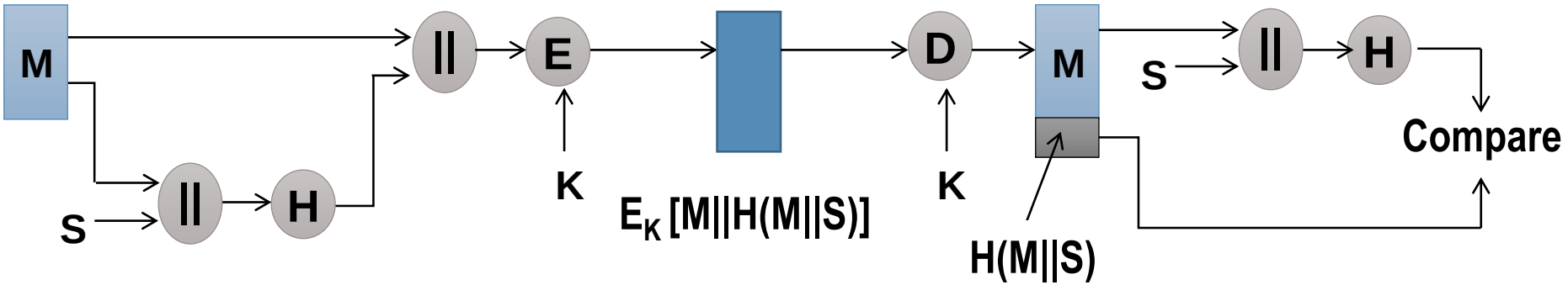
(d) $A \rightarrow B: E_K[M \| E_{KRa}[H(M)]]$

- Provides authentication and digital signature
- Provides confidentiality

7 散列函数的基本用法(e,f)



(e) **Provides authentication - - only A and B share S**



(f) **Provides authentication -- only A and B share S**
Provides confidentiality -- only A and B share K

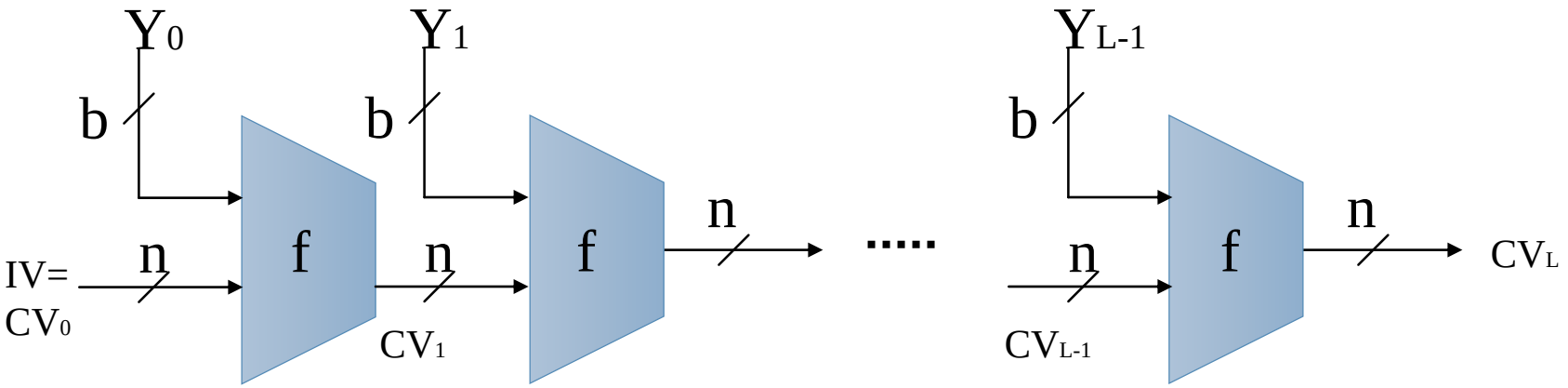
7 Hash函数要求

- Hash函数: h=H(x), 要求:
  - 可作用于任何尺寸数据且均产生定长输出
  - H(x)能够快速计算
  - 单向性: 给定h，找到x使h=H(x)在计算上不可行
  - Weak Collision Resistence(WCR):
    给定x，找到y≠x使H(x)=H(y)在计算上不可行
  - Strong Collision Resistence(SCR): 找到y≠x使H(x)=H(y)在计算上不可行

7 hash函数通用模型

- 由Merkle于1989年提出
- 几乎被所有hash算法采用
- 具体做法:
  - 把原始消息M分成一些固定长度的块Y_i
  - 最后一块padding并使其包含消息M的长度
  - 设定初始值CV_0
  - 压缩函数f, $CV_i=f(CV_{i-1},Y_{i-1})$
  - 最后一个CV_i为hash值

7 hash函数模型图



CV_0=IV= initial n-bit value
$CV_i=f(CV_{i-1}, Y_{i-1})$ $(1 \le i \le L)$
$H(M) = CV_L$

IV = initial value 初始值
CV = chaining value 链接值
Y_i = ith input block (第i 个输入数据块)
f = compression algorithm (压缩算法）
n = length of hash code (散列码的长度)
b = length of input block(输入块的长度)

7 Hash函数例子

- MD5报文摘要算法
- 安全散列算法SHA（Secure Hash Algorithm）
  - SHA-1
  - SHA-2(SHA-224、SHA-256、SHA-384 和 SHA-512)
- RIPEMD-160报文摘要算法
- HMAC

密码算法的使用

1) 保证机密性
   - 只有拥有正确的密钥才能完成正确的解密
2) 提供可认证性
   - 加密消息来源于拥有正确密钥的实体
3) 提供不可否认性
   - 通过公私钥对中的私钥进行签名实现

密码算法: 结论

- 我们通常假设核心密码算法是安全的，需要关注加密算法的**使用**细节
  - 进行机密性保护的模式是否正确使用?
  - 加密多次是否反而会泄露信息?
- 简单的应用方式更能保证方案的安全性等同于密码算法的安全性，越复杂≠越安全.
- 必须明确使用密码算法的目的.

密码算法在协议中的表示

- 对称密码算法
  - 加密：$E_K(M)$ /$\{M\}_K$ ； 解密：$D_K(C)$
- 非对称密码算法
  - 加密：$E_{Pka或KUa}(M)$ / $\{M\}_{Pka或KUa}$； 解密：$D_{KRa或SKa}(C)$
- 签名
  - 签名：$Sig_{SKa}(M)$ / $Sig_{KRa}(M)$
  - 验签：$Ver(Sig_{SKa}(M))$
- 哈希
  - Hash(M) / H(M)

Next lecture

- 课程内容：
  - 安全协议需求分析
  - 协议可能的安全目标

- 课程项目安排
  - 自由分组（5人）
  - 下周上课之前把组队情况和题目选择情况发给助教

谢谢大家！欢迎提问！