



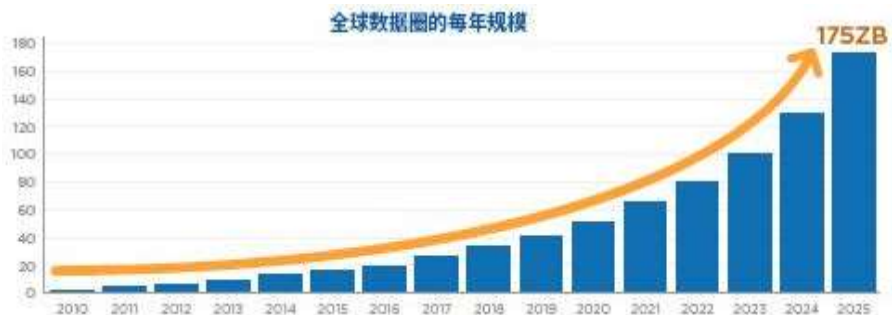
数据加密

李琦
清华大学网研院



大数据时代的到来

21世纪是数据信息大发展的时代，移动互联、电子商务、物联网等信息技术的发展极大拓展了互联网的便捷和应用范围，随之而来的是各种数据的迅速膨胀



数据来源：IDC《数据时代2025》

- 根据IDC监测，人类产生的数据量正在呈指数级增长。全球数据量在2025年预计将增加至**175ZB**
- 1ZB相当于1万亿GB
- TB→PB→EB→ZB

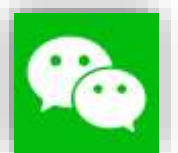
社交网络



电子商务



移动互联





全球大数据战略布局持续深化

美国：强化机构协同，深挖数据资源价值

- 2019年12月，美国发布国家级战略规划《联邦数据战略与2020年行动计划》，《战略》明确提出出将数据作为战略资源
- 2021年10月，美国管理和预算办公室（OMB）发布2021年行动计划，该计划在《战略》的基础上进一步强化了在数据治理、规划和基础设施方面的活动



英国：细化国家战略数据，强调数据应用

- 2020年9月，英国政府发布《国家数据战略》，指出了政府许优先执行释放数据价值、加强数据保护、确保数据基础架构安全性等五项任务
- 2021年5月，英国政府发布《政府对于国家数据战略咨询的回应》，强调2021年工作中心是“深入执行《国家数据战略》”



欧盟：稳步执行欧盟数据战略，打造单一数据市场

- 2020年2月，欧盟委员会退出《欧盟数据战略》，勾画出欧盟未来十年的数据战略行动纲要，注重加强成员国之间的数据流通和使用，构建单一数据市场
- 2021年10月，欧盟成员国表决通过了《欧盟数据治理法案》，旨在确保符合欧洲共同立意和数据提供者合法权益的条件下，实现数据更广泛的国际共享





讨论

畅所欲言

只要应用了密码技术，网络安全就能保证，你同意吗？



数据安全问题频发



2020年6月，科技巨头甲骨文公司的数据管理平台BlueKai因数据无密码保护导致数十亿人的数据记录泄露

1



2019年7月，中国智能家居公司欧瑞博（Orvibo）的产品数据库无密码保护导致超过 20 亿条日志泄露

2



2019年1月，LinkedIn服务器曾遭黑客窃听，1.59亿的用户信息被窃取

3

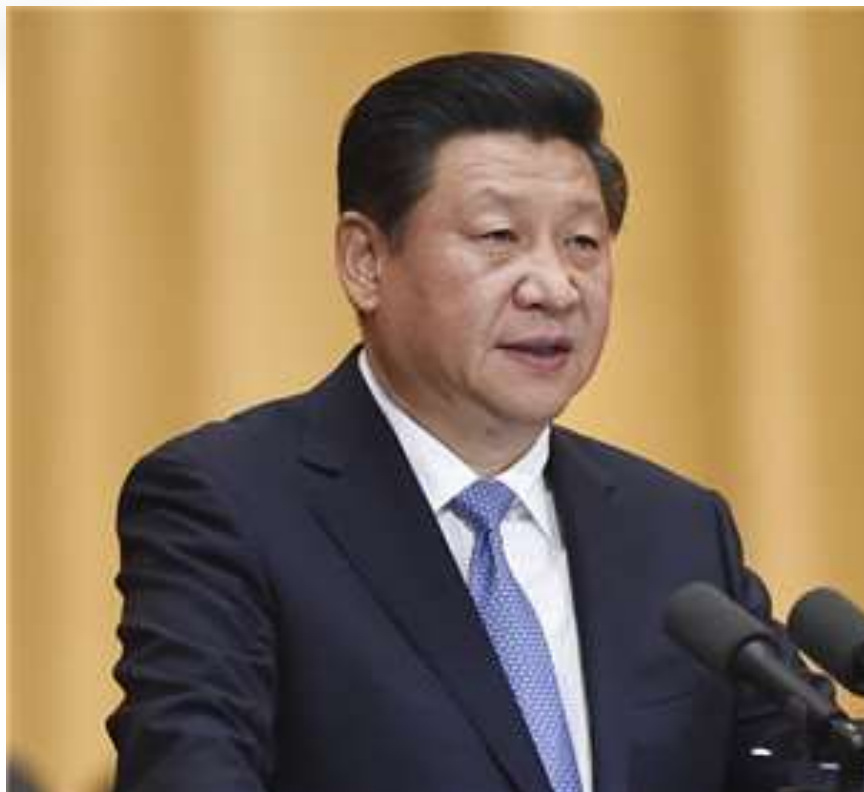


2020年4月有 2.67 亿 Facebook 用户信息被盗，包括姓名、邮箱地址、电话、社会身份、性别等

4



国家高度重视维护数据安全



“随着信息技术和人类生产生活交汇融合，互联网快速普及，全球数据呈现爆发增长、海量集聚的特点，对经济发展、社会治理、国家管理、人民生活都产生了重大影响 ”

——2017年12月 中共中央政治局第二次集体学习

“完善重点领域安全保障体系和重要专项协调指挥体系，强化经济、重大基础设施、金融、网络、**数据**、生物、资源、核、太空、海洋等**安全保障体系建设** ”

——2022年10月 中国共产党第二十次全国代表大会

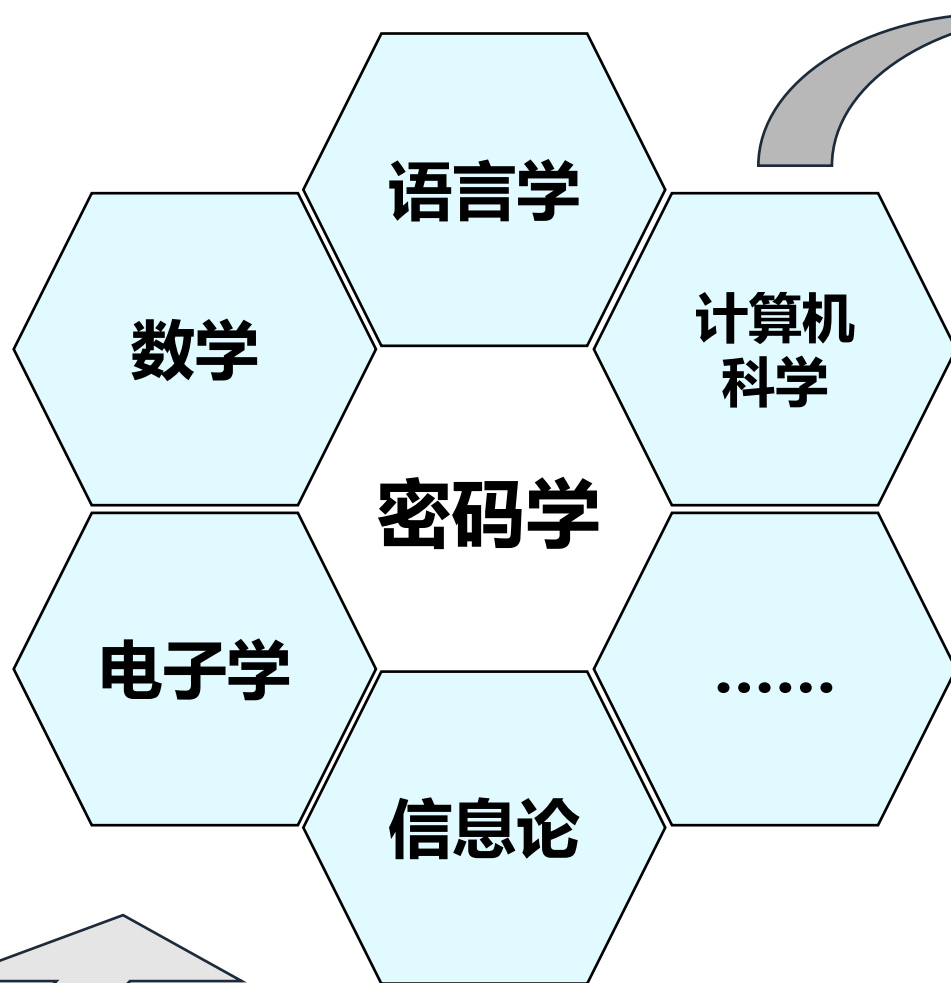
维护数据安全的本质是什么，如何构建数据安全保障体系，不同的数据安全技术如何在不同场景下发挥作用？



使用密码保护数据



- 研究密码活动本质和规律
- 指导密码实践
- 探索密码编码和分析的一般规律



- 通信加密
- 身份认证
- 消息认证
- 数字签名
-



本章的内容组织



第一节 密码学简史

- 古典密码
- 近代密码
- 现代密码

回顾历史，了解
密码学演进方向

为密码算法的优化提
供历史经验



第二节 对称密码

- 分组密码
- DES算法
- 流密码

总结对称密码的
加密解密原理



第三节 公钥密码

- 密钥分发问题
- 公钥密码
- RSA算法
- 实际应用

剖析公钥密码的
来龙去脉



第四节 摘要与签名

- 散列函数
- 消息认证码
- 数字签名

介绍校验数据
完整性的工具方法



第五节 密码分析技术

- 唯密文攻击
- 已知明文攻击
- 选择明文攻击
- 选择密文攻击
- 选择密钥攻击

梳理密码分析领域
的技术体系

对密码算法的设计具
有理论指导意义

密码学方法为网络空间的安全
稳定保驾护航



第1节 密码学简史

- ✓ 古典密码
- ✓ 近代密码
- ✓ 现代密码



隐写术与密码学

- 隐写术是一门关于信息隐藏的技巧与科学
- 隐写术的英文叫做Steganography，来源于Histaiaeus的一本讲述密码学与隐写术的著作《Steganographia》，该书书名源于希腊语，意为“隐秘书写”



隐写术与密码学有何区别？



密码学

- 密码学 (cryptography)
- 源于希腊语kryptós “隐藏的”，和gráphein “书写”
- 研究如何隐密地传递信息的学科



“密码学是关于如何在敌人存在的环境中通讯”

——著名的密码学者Ron Rivest (RSA作者，图灵奖获得者)



Ron Rivest



密码学的基本概念

- **密码编码**: 通过信息编码使信息保密
- **密码分析**: 用分析方法解密信息
- **基本术语**:



- 明文(plain text), 密文(cipher text)
- 加密(encrypt, encryption), 解密(decrypt, decryption)
- 密码算法(Algorithm), 密码(Cipher):用来加密和解密的数学函数

$$c=E(m), \quad m=D(c), \quad D(E(m))=m$$

- 密钥(Key): 密码算法中的一个变量

$$c=E_{K_e}(m), \quad m=D_{K_d}(c), \quad D_{K_d}(E_{K_e}(m))=m$$



密码编码

- 把明文转换成密文的两种操作
 - **代替**: each element in the plaintext is mapped into another element
 - **换位**: elements in the plaintext are rearranged
- 密钥的数量
 - Symmetric, secret key, conventional encryption
 - Asymmetric, two keys, public key encryption
- 对明文加密的方式
 - Block cipher
 - Stream cipher



密码技术的主要用途

- **数据保密—数据加密/解密**
 - 数据加密(存储和传输)
- **认证技术**
 - 实体身份认证
 - 数据源发认证
- **信息完整性保护**
 - 传输过程中没有被插入、篡改、重发
- **数字签名和抗抵赖**
 - 源发抗抵赖
 - 交付抗抵赖





密码学简史

这一时期的密码学更像是一门艺术，其核心的密码编码方法归根结底主要有两种，即置换和代换

古典密码

1949年以前

近代密码

香农发表了“保密系统的通信理论”为密码系统建立了理论基础，是密码发展史上的第一次飞跃，密码学迈入近代密码时期



密码专家迪菲和赫尔曼发表文章“密码学的新方向”提出的公钥密码建立在数学问题难解性问题之上，是密码学的一次真正的革命

现代密码

1975年~至今





古典密码

- 古典密码

- 代替密码 (Substitution Cipher)
- 换位密码 (Transposition Cipher)
- 代替密码与换位密码的组合
- 密码体制的安全性在于保持**算法**本身的保密性

- 缺陷

- 受限算法的缺陷
 - 不适合大规模使用
 - 不适合较大的或者人员变动较大的组织
 - 用户无法了解算法的安全性



Caesar密码

凯撒(Caesar)密码是公元前一世纪在高卢战争时被使用的，它是将英文字母向前移动k位

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

凯撒密码明密文对应表

明文: i went to school this morning

密文: n vjsz zu yhmuuq zmny ruxsnsl

CAESAR 密码: $c = (m + 5) \text{ MOD } 26$



猪圈密码

猪圈密码是一种以格子为基础的简单替代式密码，格子如下：

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~~

示例：

密文为：

□ ◻ ∇

└ ┘ □

◁ ◻ <

明文为：

H O W

A R E

Y O U



纵行换位密码

- 换位密码就是一种早期的加密方法，明文的字母保持相同，但顺序被打乱了
- 广泛应用于美国南北战争

输入方向 →

C	A	N	Y
O	U	U	N
D	E	R	S
T	A	N	D

输出方向 ↓

明文: Can you understand

密文:
C
o
d
t
a
u
e
a
n
u
r
n
y
n
s
d



戚继光 反切码

反切码源自反切拼音注音方法，是用两个字为另一个字注音，取上字的声母和下字的韵母，“切”出另外一个字的读音



诗篇1 声母

1	2	3	4	5		6	7	8	9	10		11	12	13	14	15						
l	b	q	q	d		b	t	zh	r	sh		y	m	y	ch	x		d	zh	y	j	zh
柳	边	求	气	低	,	波	他	争	日	时	,	莺	蒙	语	出	喜	,	打	掌	与	君	知

诗篇2 韵母

1	2	3		4	5	6		7	8	9	10	11	12	13		14	15	16	17	18	19
un	ua	iang		iu	an	ai		ia	in	uan	e	u	in	ei		u	eng	uang	ui	ao	in
春	花	香	,	秋	山	开	,	嘉	宾	欢	歌	须	金	杯	,	孤	灯	光	辉	烧	银
20		21	22	23		24	25	26		27	28	29	30	31		32	33	34	35	36	
ang		i	ong	iao		uo	i	iao		i	eng	ui	u	ian		i	ei	ai	e	ou	
缸	。	之	东	郊	,	过	西	桥	,	鸡	声	催	初	天	,	奇	梅	歪	遮	沟	。



戚继光 反切码

反切码源自反切拼音注音方法，是用两个字为另一个字注音，取上字的声母和下字的韵母，“切”出另外一个字的读音

举例：

密文：

5-25-2

5
d
低

+

25
i
西

di

明文： 敌

诗篇1
声母

诗篇2
韵母

声调
2声

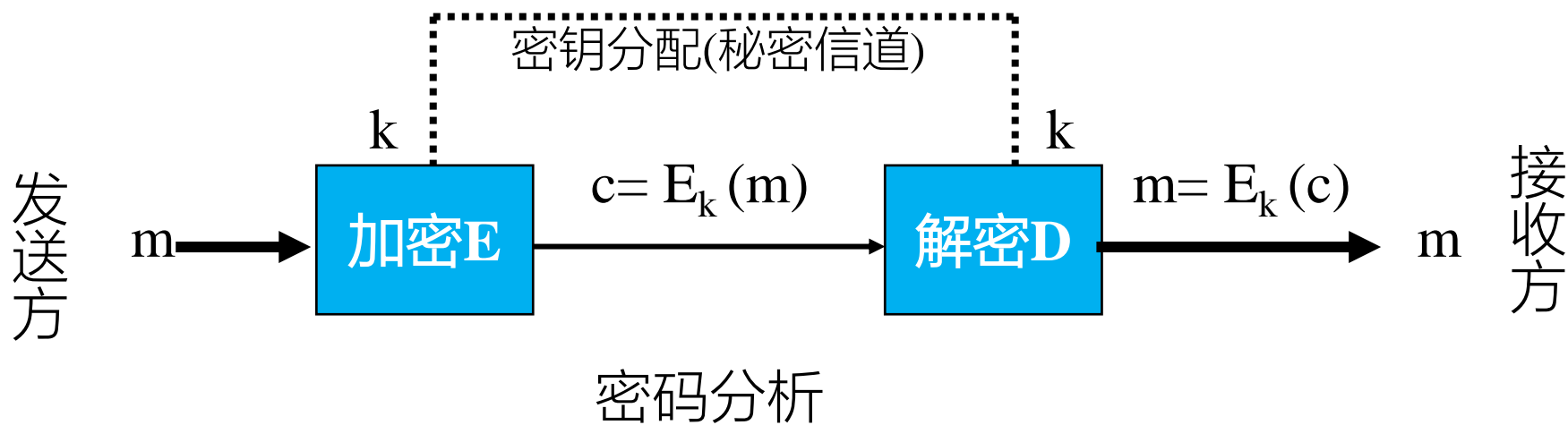




近代密码

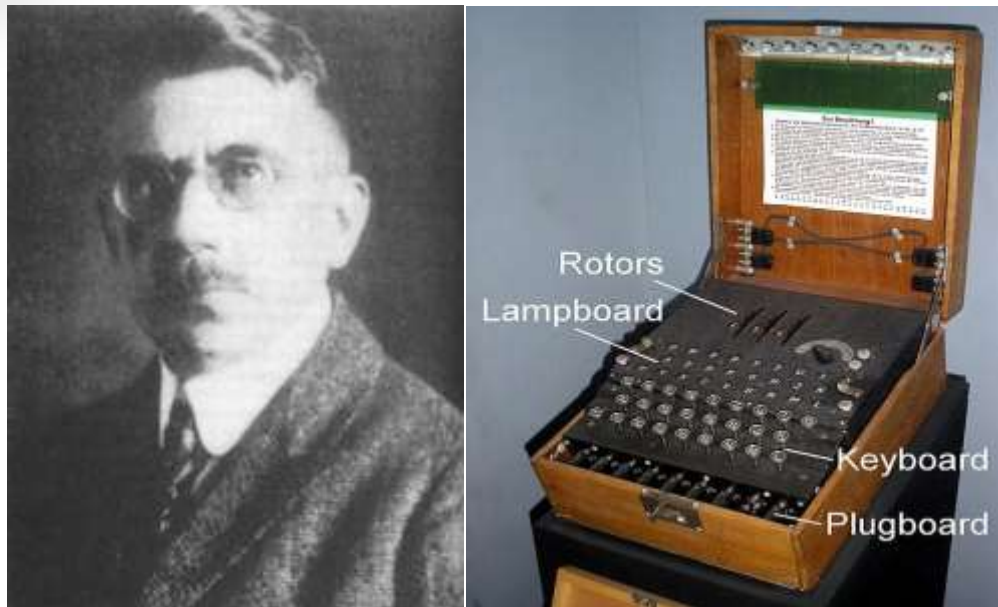
1949年，香农发表的“Communication Theory of Secrecy Systems”为密码系统建立了理论基础，是密码发展史上的**第一次飞跃**，使密码技术由艺术变成了科学：

- 将**算法**和**密钥**分开
- 密码算法可以公开，密钥保密
- 密码系统的安全性在于保持密钥的保密性





ENIGMA机



Arthur Scherbius与转轮密码机ENIGMA

- 转轮密码机ENIGMA，由Arthur Scherbius于1919年发明，1926年装备德军



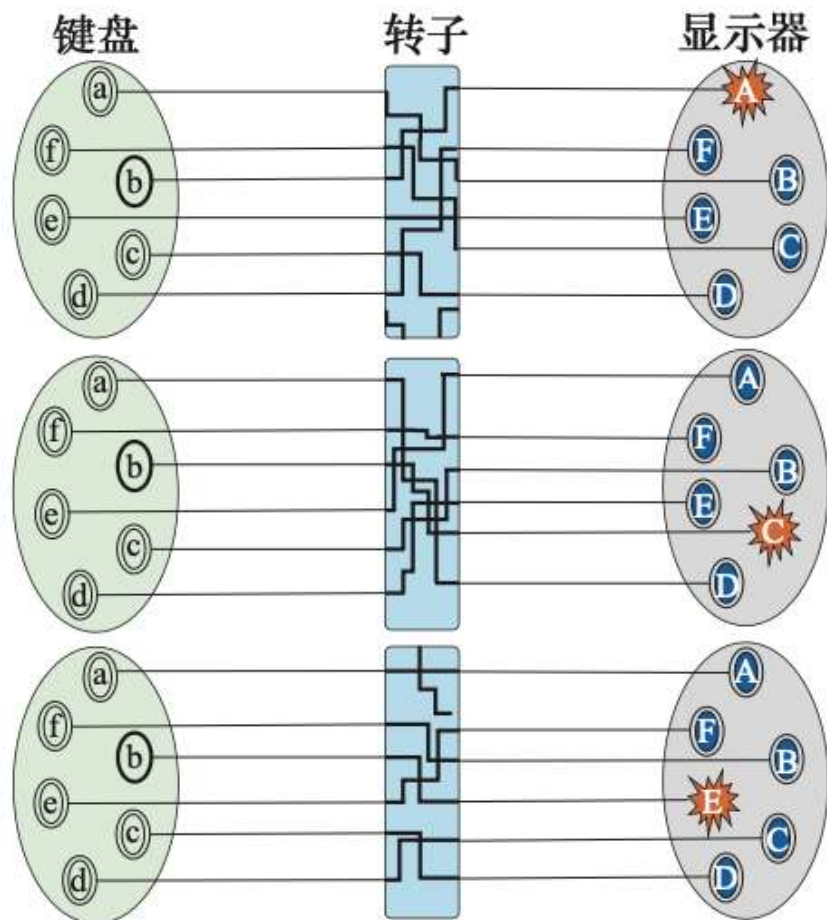
TYPEX打字密码机

- 英国的TYPEX打字密码机，是德国3轮ENIGMA的改进型密码机，它在英国通信中广泛使用

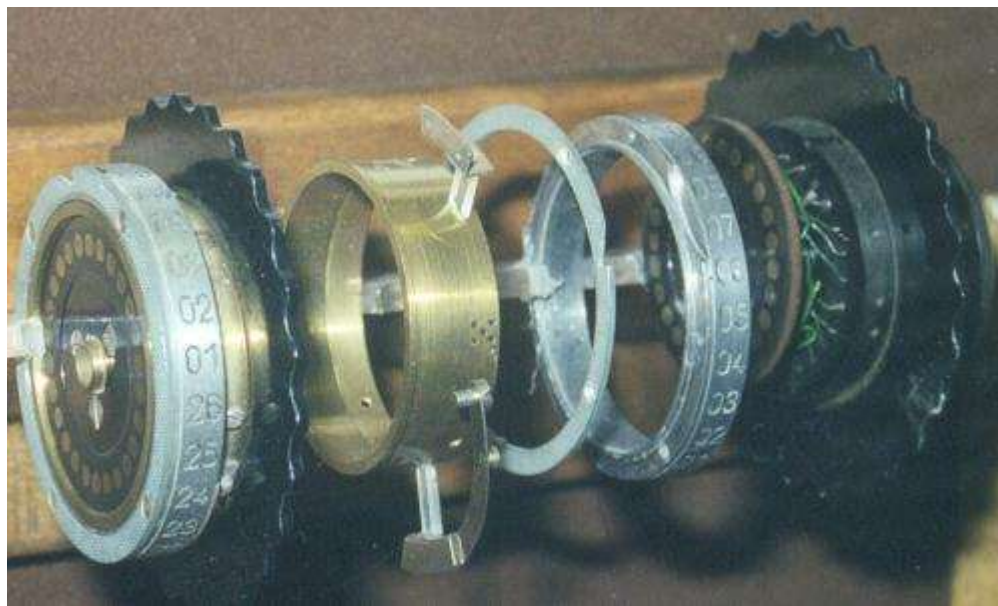


ENIGMA机工作原理

连续键入三个b:

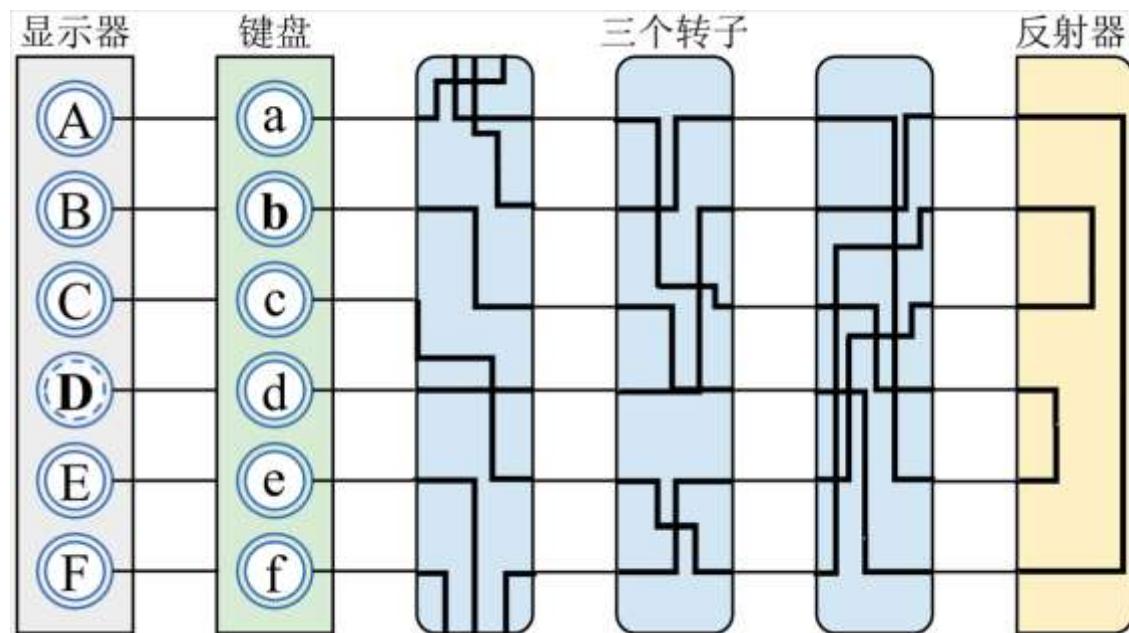
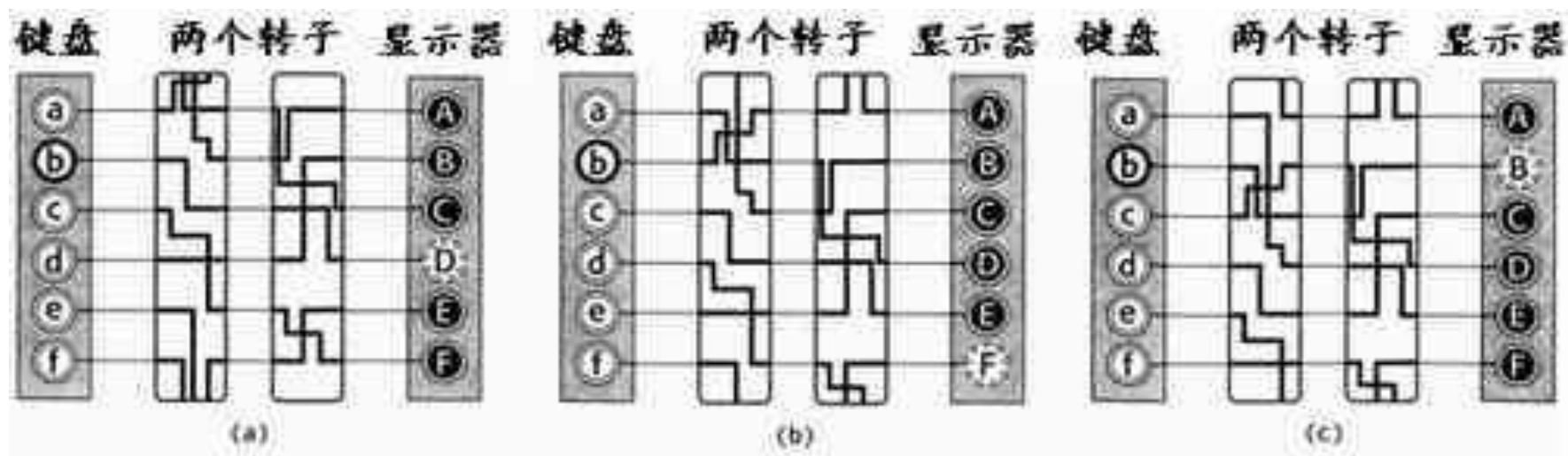


实物图:



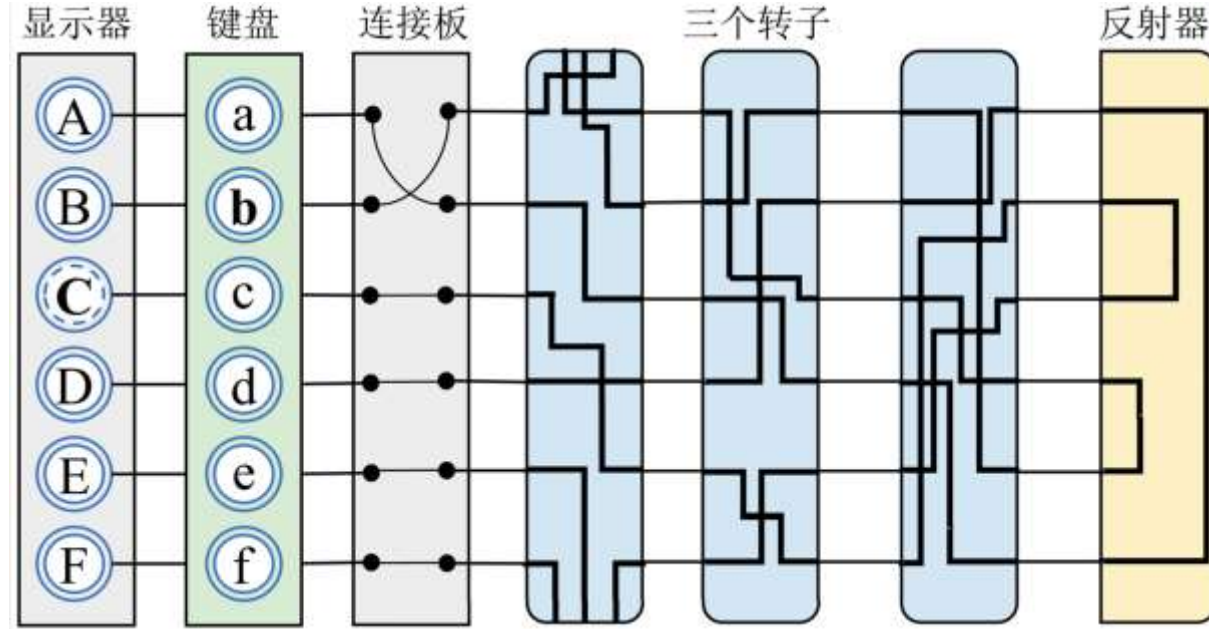


ENIGMA机工作原理





ENIGMA机工作原理



- 转子的初始设置，转子之间的相互位置，以及连接板连线的状况组成了所有可能的密匙
- 三个转子不同的方向组成了 $26 \times 25 \times 26 = 16900$ 种不同可能性
- 三个转子间不同的相对位置为 6 种可能性
- 连接板上两两交换 6 对字母的可能性为 $C(26, 6) \times C(20, 6) \times 6! / 64 = 100391791500$
- 共有 $16900 \times 6 \times 100391791500$ ，约为一亿亿种可能性



ENIGMA工作过程

- **根据密码本取得当日密钥**
- **首先发送一个新的密钥**
 - 随机地选择三个字母，比如说PGH
 - 把PGH在键盘上连打两遍，加密为比如说KIVBJE（注意两次PGH被加密为不同的形式）
 - 把KIVBJE放在在电文的最前面
- **重新调整三个转子的初始方向到PGH**
- **正式对明文加密**



ENIGMA破解



Marian Adam Rejewski



Jerzy Witold Różycki



Henryk Zygalski

- 20世纪30年代，马里安·雷耶夫斯基、杰尔兹·罗佐基与亨里克·佐加尔斯基等人一同进行了对德国Enigma密码的破译工作，并称为密码研究领域的“波兰三杰”
- 2000年7月17日，波兰政府向雷耶夫斯基、罗佐基和佐加尔斯基追授波兰最高勋章



ENIGMA破解

- 阿兰·麦席森·图灵

Alan Mathison Turing

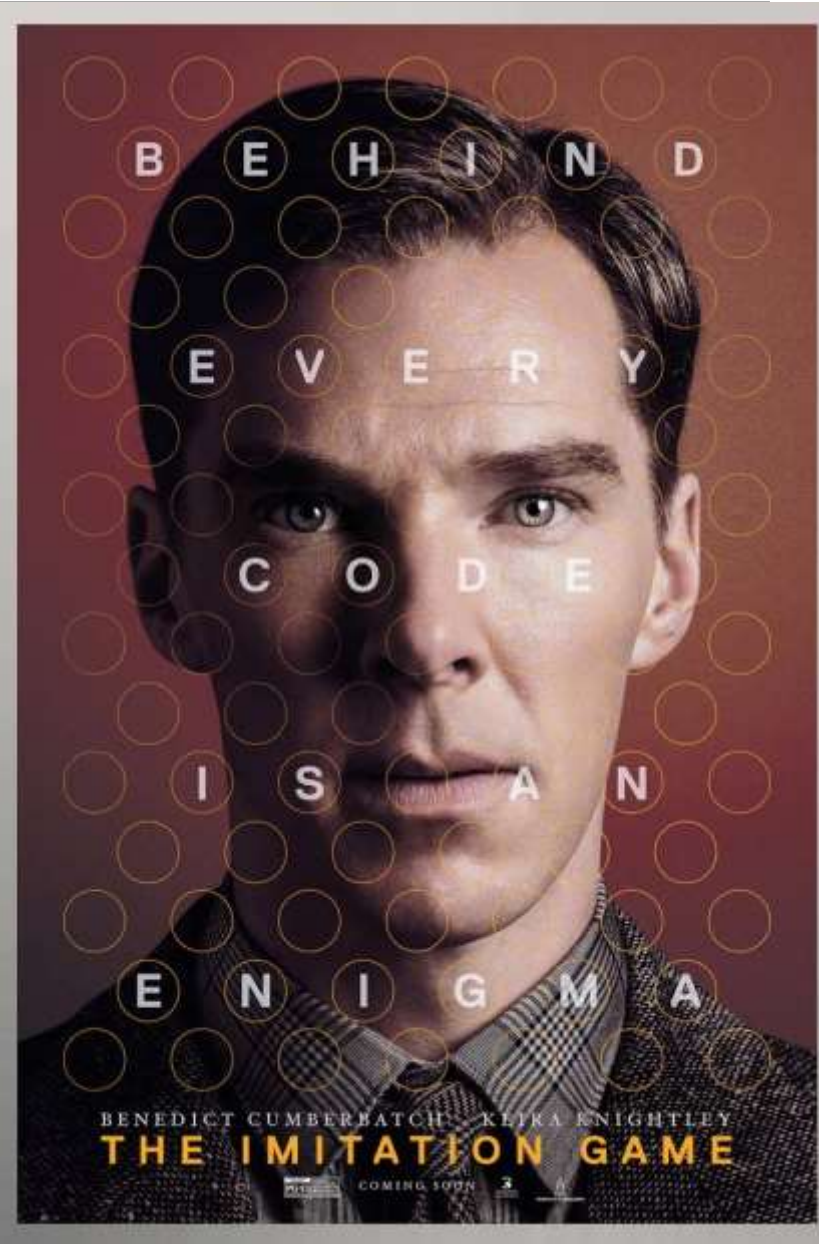
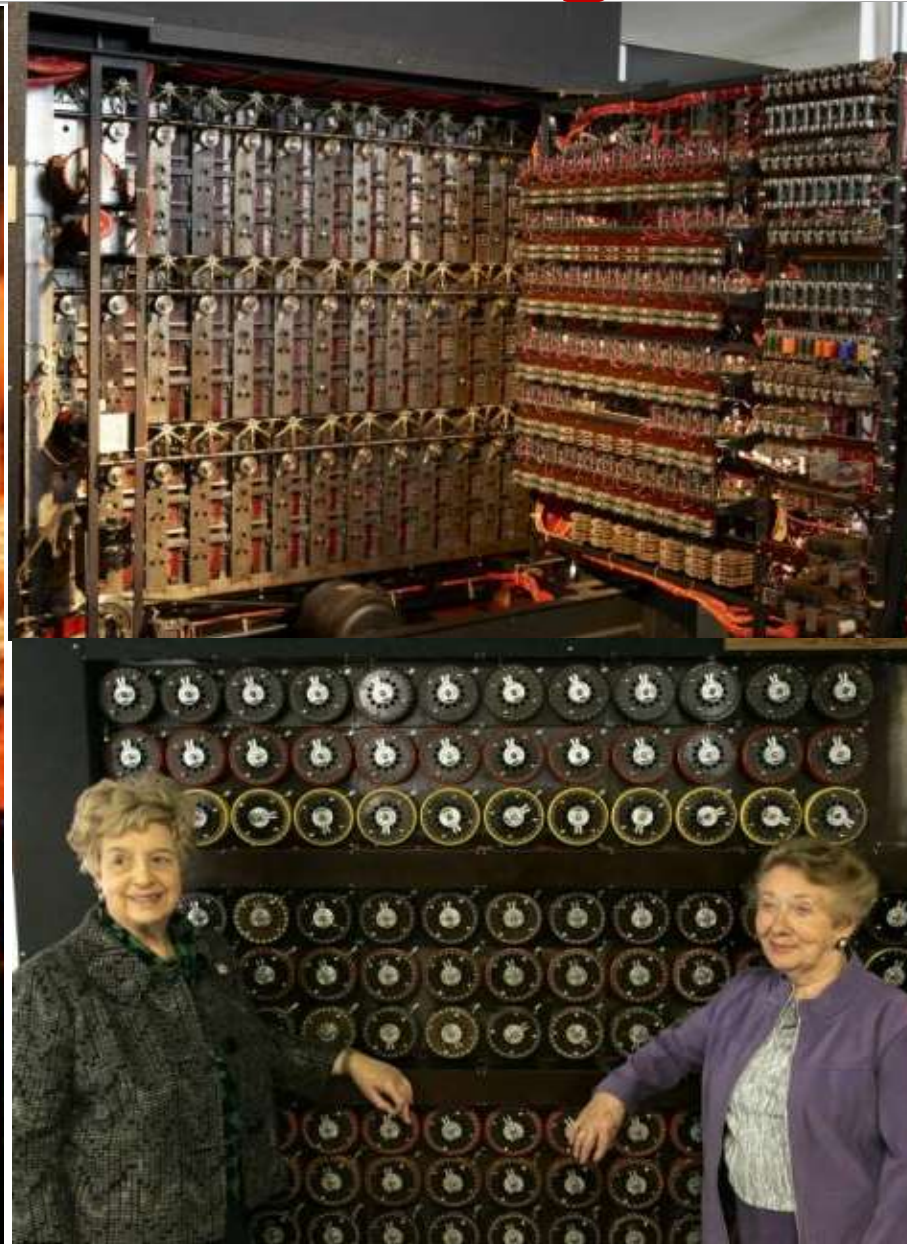
1912.6.23—1954.6.7

- 英国数学家、逻辑学家，他被视为计算机之父
- 1931年图灵进入剑桥大学国王学院，毕业后到美国普林斯顿大学攻读博士学位，二战爆发后回到剑桥，后曾协助军方破解德国的著名密码系统Enigma，帮助盟军取得了二战的胜利





ENIGMA破解-Turing Bombe





现代密码

- 对称密钥密码系统的缺陷

- 密钥必须经过安全的信道分配
- 无法用于数字签名
- 密钥管理复杂 $O(n^2)$

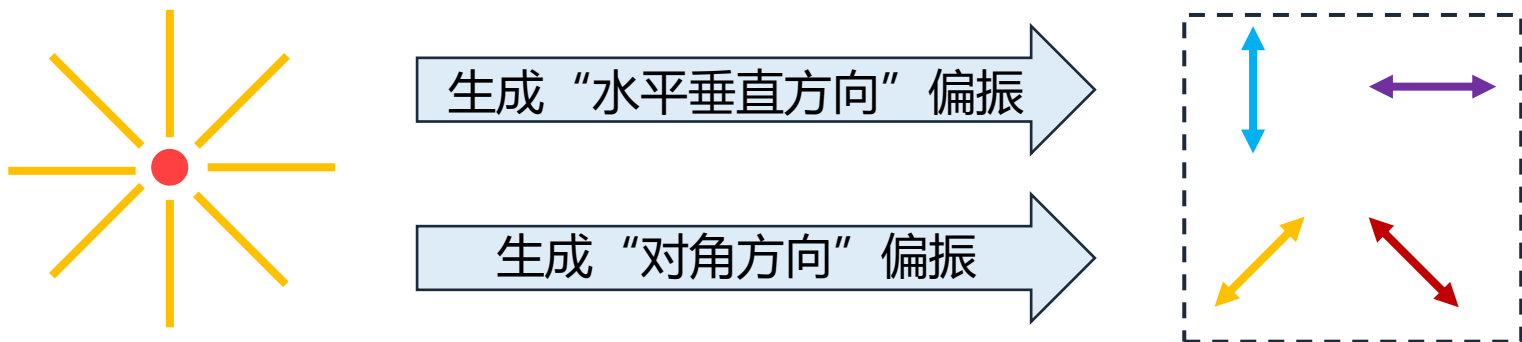


- 1976 年, Diffie和Hellman发表了《*New Direction in Cryptography*》提出了“公钥密码”的概念, 获得2015年图灵奖
- 使用**两个**密钥, 对于密钥分配、数字签名、认证等有深远影响
- 基于**数学函数**而不是代替和换位, 密码学历史上唯一的一次真正的革命, 实现了密码学发展史上的**第二次飞跃**

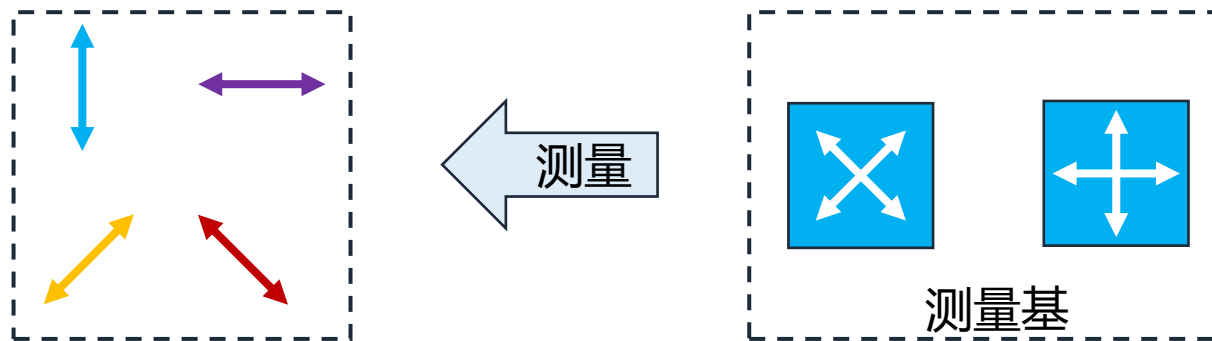


量子密码

- 量子密码体系采用量子态作为信息载体，经由量子通道在合法的用户之间传送密钥，其安全性由量子力学原理所保证
- 光的偏振：



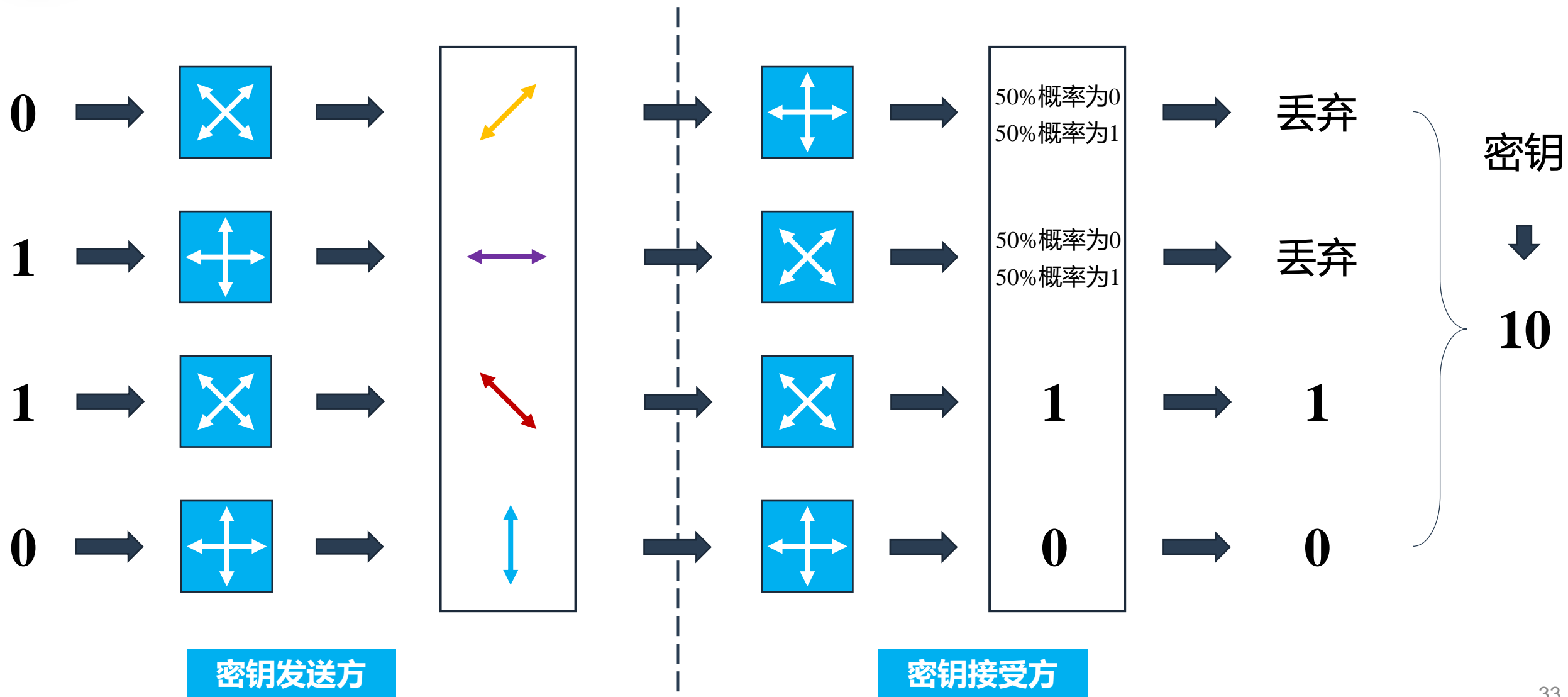
- 偏振方向测量：





量子密码

量子密钥生成方法：





第2节 对称密码

- ✓ 分组密码
- ✓ DES算法
- ✓ 流密码



对称密码

- 加密和解密使用**相同的密钥**: $K_E = K_D$
- 密钥必须使用**秘密的信道**分配
- 常用对称密钥密码算法
 - DES (Data Encryption Standard)及其各种变体
 - AES (Advanced Encryption Standard)





对称密码基本设计原则

- 对称加密算法对称密码算法的两条基本设计原则：

(1) 扩散 (Diffusion)

重新排列消息中的每一个比特，使明文中的冗余度能够扩散到整个密文，将每一个比特明文的影响尽可能作用到较多的输出密文位中

(2) 扰乱 (Confusion)

密文和密钥之间的统计特性关系尽可能复杂化。如果密钥的一位发生变化，密文的绝大多数位也发生变化

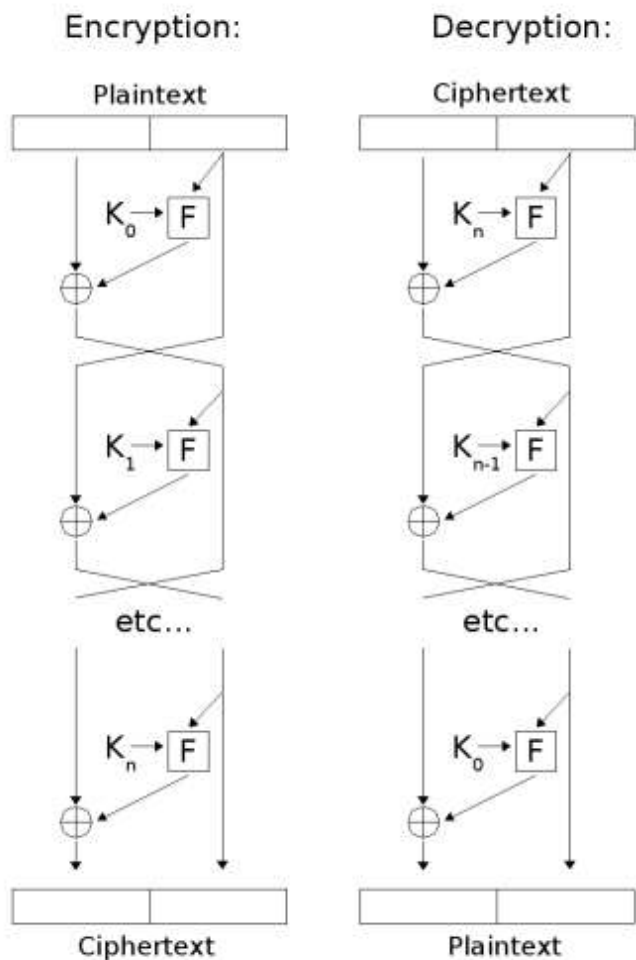
- 多次迭代是基本设计方法



克劳德·艾尔伍德·香农
信息论创始人



对称密码基本设计原则



Feistel Cipher

- **霍斯特·菲斯特尔 (Horst Feistel)**
分组密码之父
- 1973年他在 Scientific American 杂志上发表了Cryptography and Computer Privacy一文，提出了Feistel网络
- 多轮迭代结构
- 加密解密对称



分组密码

- 分组密码(block cipher)是每次只能处理特定长度的一块数据的一类密码算法
- 对不同分组主要有5种迭代模式
 - ECB模式: Electronic CodeBook mode (电子密码模式)
 - CBC模式: Cipher Block Chaining mode (密码分组链接模式)
 - CFB模式: Cipher FeedBack mode (密文反馈模式)
 - OFB模式: Output FeedBack mode (输出反馈模式)
 - CTR模式: CounTeR mode (计数器模式)



ECB模式

ECB模式的全称Electronic CodeBook 模式(电子密码模式)。在ECB中，明文分组加密后的结果直接作为密文分组

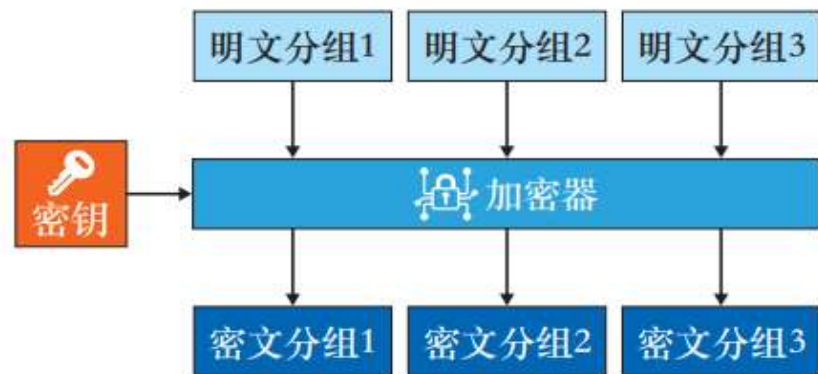
优点:

- (1)每个分组的处理相互独立，可以并行操作
- (2)一个密文块传输错误不会影响后续密文解密，是分组密码的标准工作模式

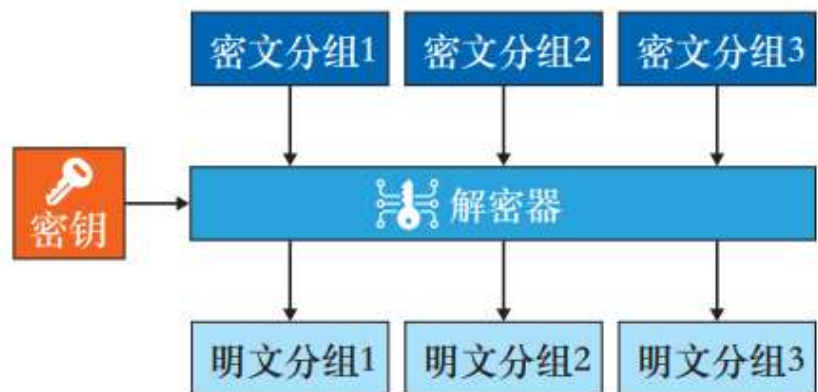
缺点:

- (1)因分组相互独立，信息块可被替换、重排、删除、重放，所以对明文的主动攻击是可能的
- (2)无法纠正传输中的同步错误

ECB模式的加密:



ECB模式的解密:





CBC模式

CBC 模式的全称Cipher Block Chaining模式(密码分组链接模式)

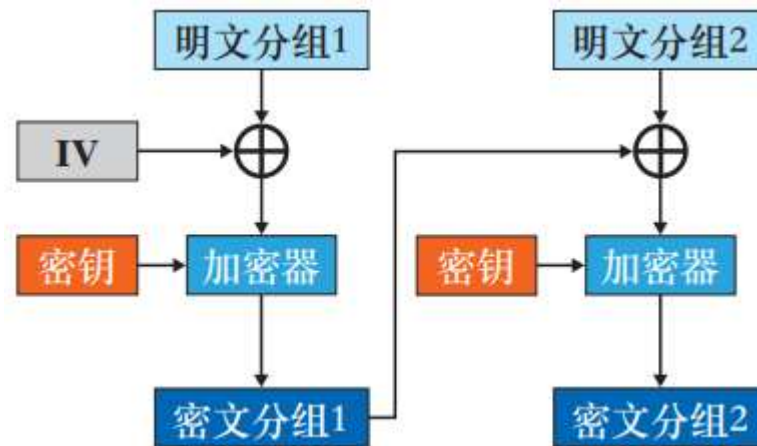
优点:

- (1)能够隐藏明文的数据模式,相同的明文可对应不同的密文
- (2)在一定程度上抵抗主动攻击
- (3)误差有限传递

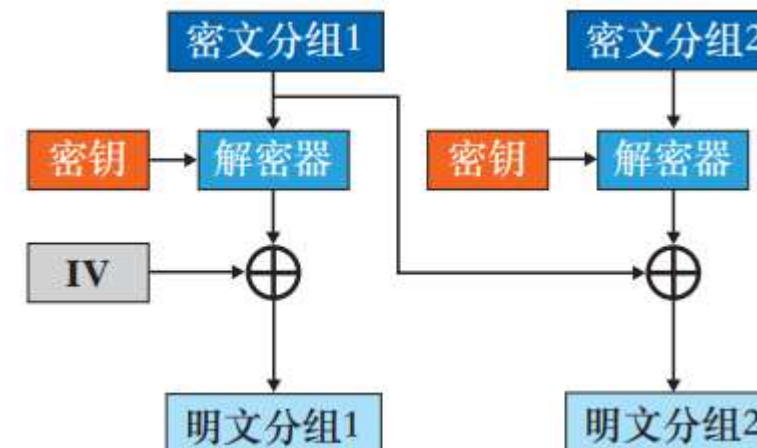
缺点:

加密不支持并行计算

CBC模式的加密:



CBC模式的解密:





CFB模式

CFB模式全称Cipher FeedBack 模式(密文反馈模式)

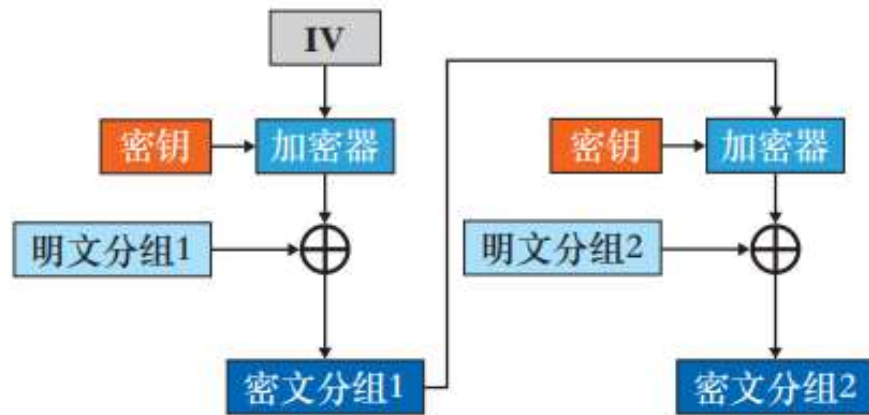
优点:

- (1) 能够隐藏明文的数据模式
- (2) 在一定程度上抵抗主动攻击
- (3) 解密支持并行计算

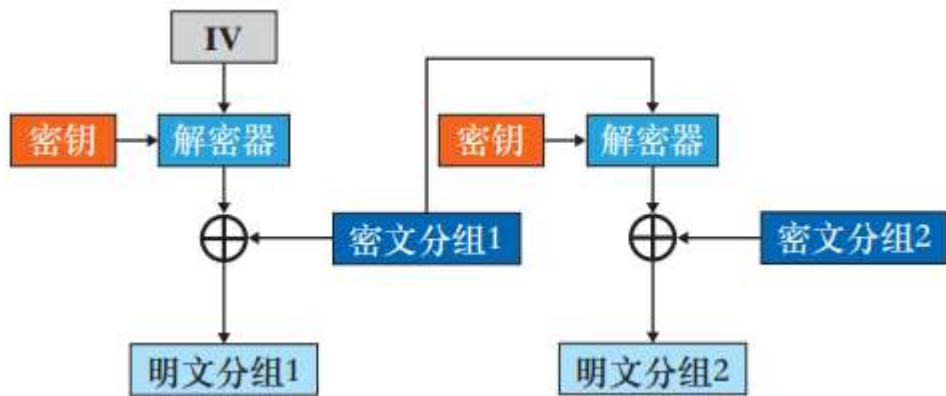
缺点:

- (1) 加密不支持并行计算
- (2) 一个密文块传输错误会影响后续密文解密

CFB模式的加密:



CFB模式的解密:





OFB模式

OFB全称 Output FeedBack 模式 (输出反馈模式)。在OFB中，密码算法的输出会反馈至密码算法的输入中，即上一轮加密算法的输出会作为下一轮加密算法的输入

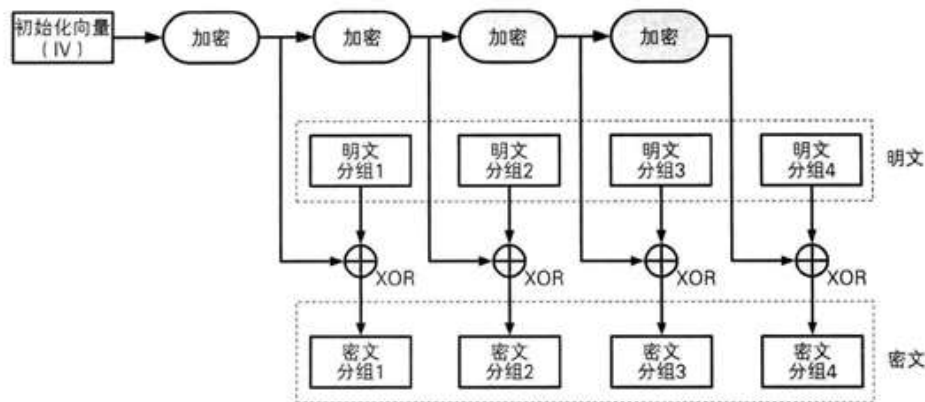
优点:

没有错误传播，一个密文单元的损坏只影响本单元的解密

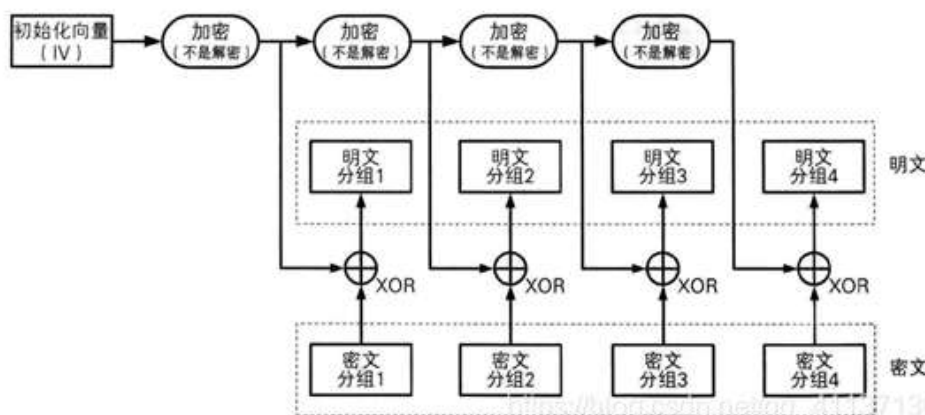
缺点:

- (1)对密文的主动攻击难以检测
- (2)加密时不支持并行计算

OFB模式的加密



OFB模式的解密





CTR模式

CTR的全称 Counter模式(计数器模式)。CTR 是一种通过将逐次累加的计数器进行加密来生成密钥流的分组密码

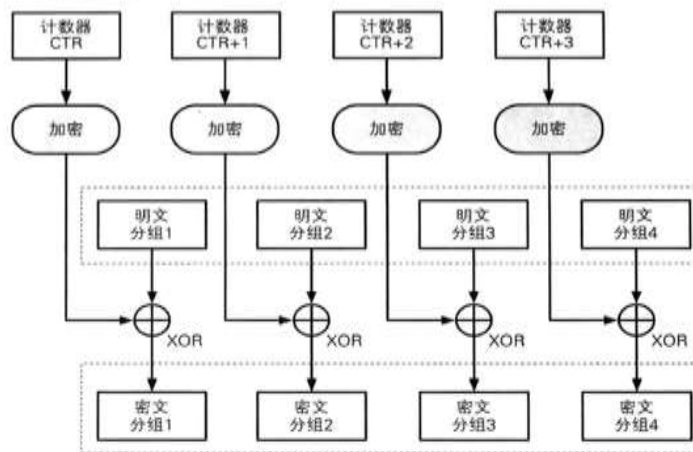
优点:

- (1) 可以随机地对任意一个密文分组进行解密，对该密文分组的处理与其他密文处理无关
- (2) 能并行加解密处理，多个分组同时进行，效率高

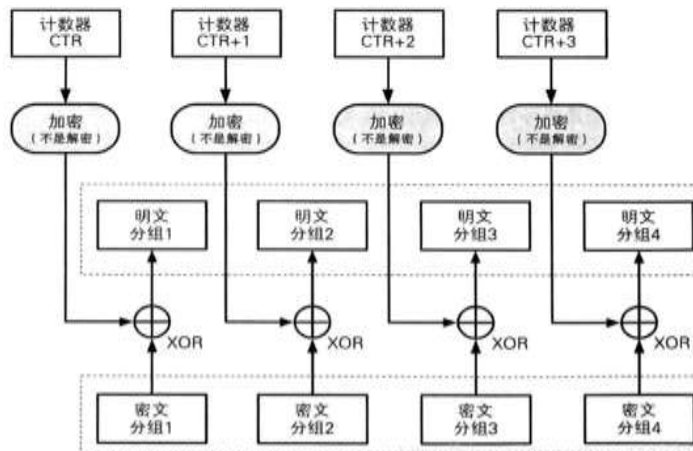
缺点:

对密文的主动攻击难以检测

CTR模式的加密



CTR模式的解密





DES算法原理

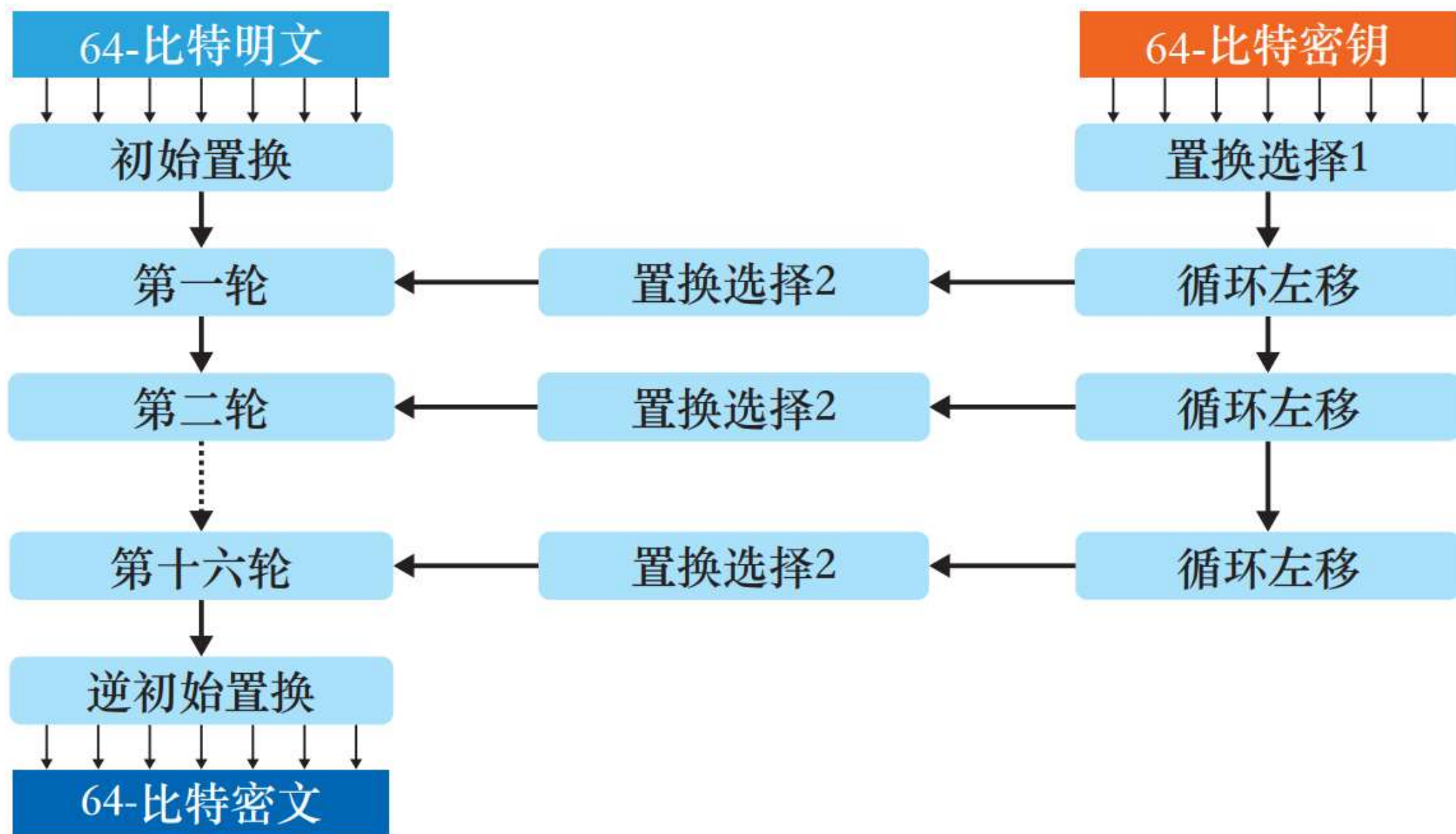
- IBM 公司，70年代初提出
- DES是一种对称密钥算法，密钥长度为56bits (加上奇偶校验，通常写成64bits)
- 是一种分组加密算法，64 bits为一个分组
- 使用标准的**算术**和**逻辑**运算
- 首先把明文分成以64 bit为单位的块 m ，对于每个 m ，执行如下操作：

$$\text{DES}(m) = \text{IP}^{-1} \cdot \text{T16} \cdot \text{T15} \cdot \dots \cdot \text{T2} \cdot \text{T1} \cdot \text{IP}(m)$$

- 初始置换，IP
- 16轮迭代， T_i ， $i=1,2,\dots,16$
- 末置换， IP^{-1}

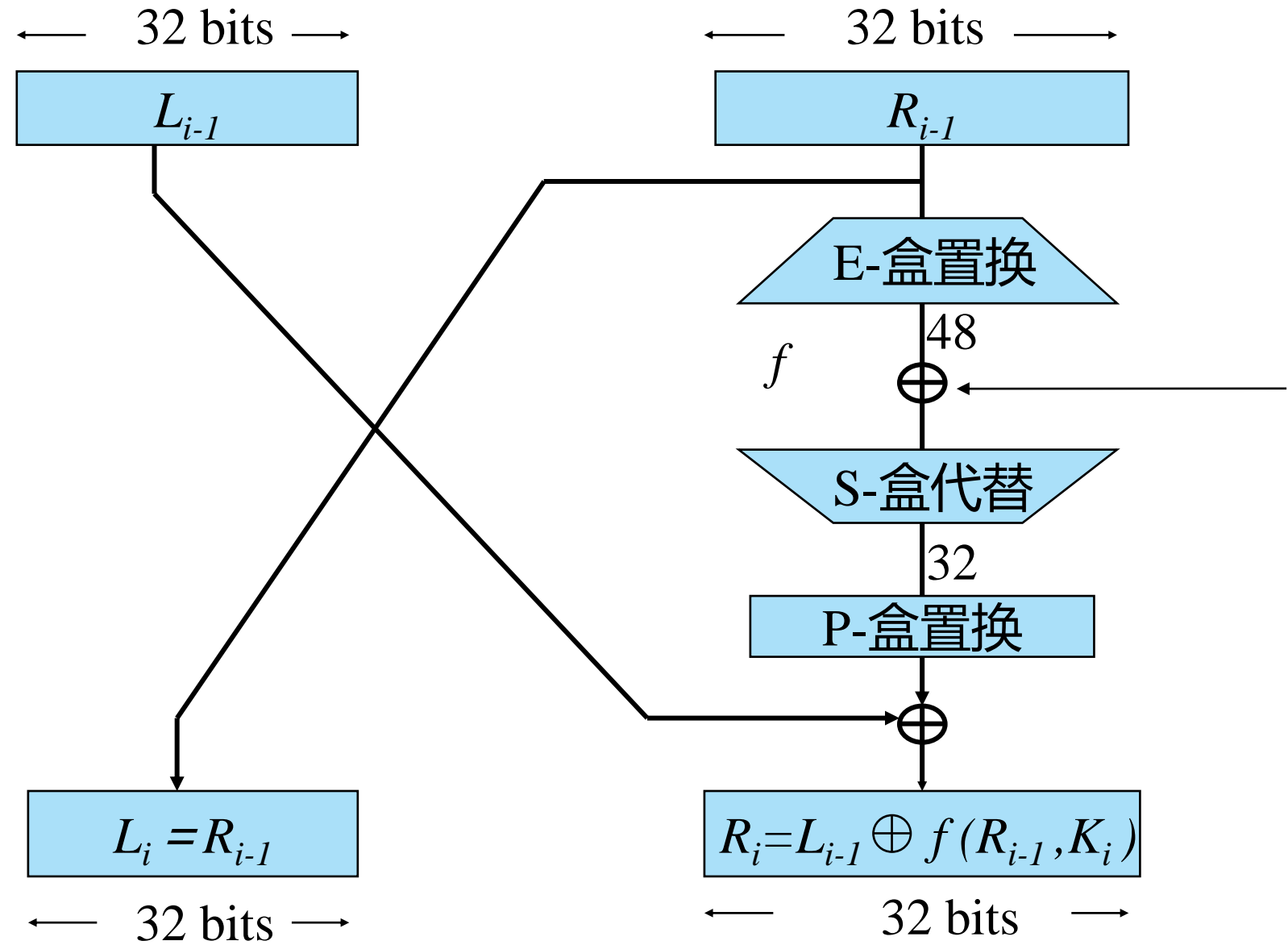


对称密码基本设计原则





一轮迭代





关键组件: 初始置换

- 初始置换(IP)

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

$$M = m_1 m_2, \dots, m_{62} m_{63} m_{64}$$

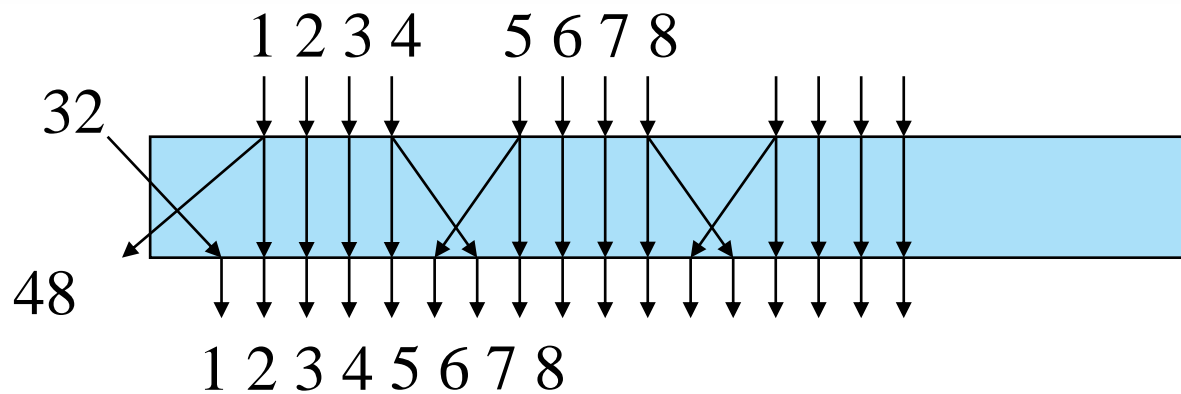
IP(M)

$$M' = m_{58} m_{50}, \dots, m_{23} m_{15} m_7$$



关键组件: 扩展置换 (E-盒置换)

- 将 R_i 从32位扩展到48位, 便于和密钥操作
- 目的:** 输入的一位影响下一步的两个替换, 使得输出对输入的依赖性传播得更快, 密文的每一位都依赖于明文的每一位

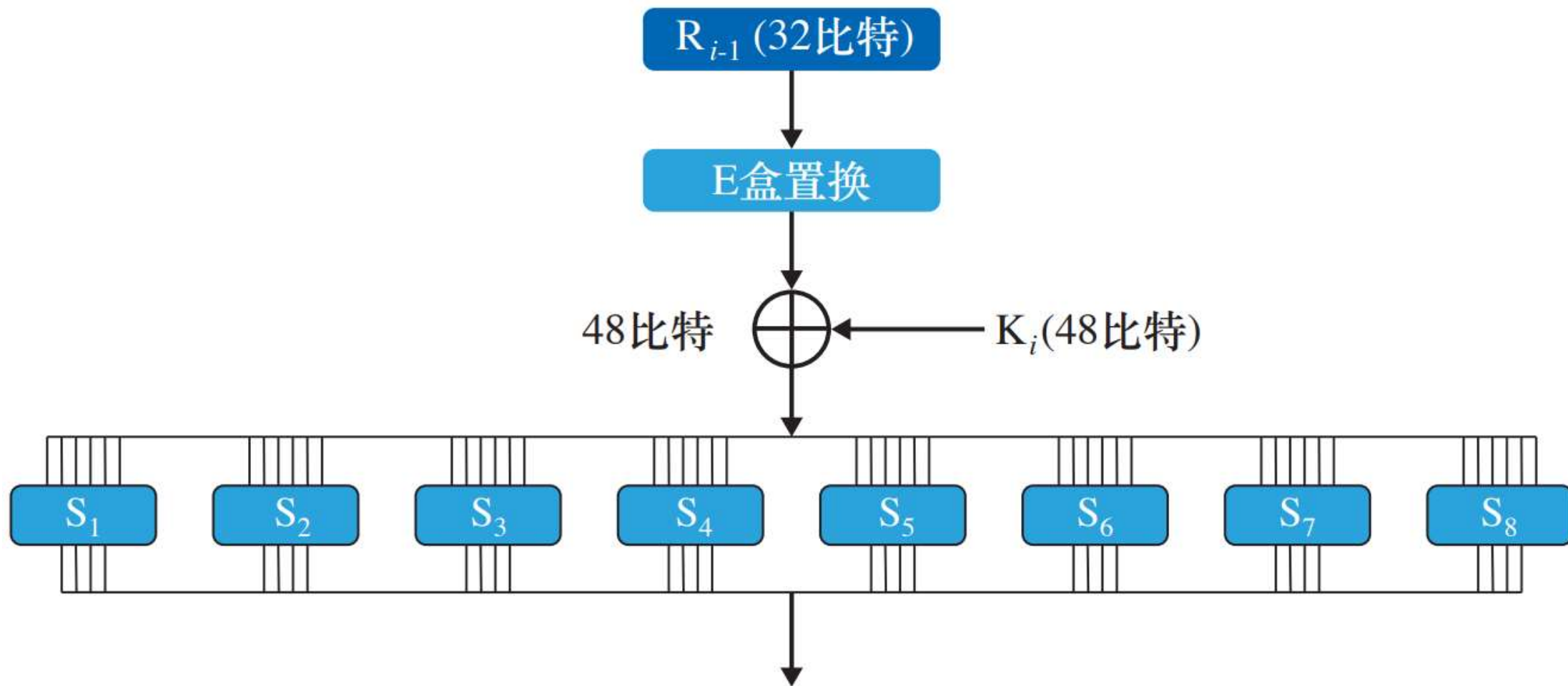


1	2	3	4	5	6	7	8	9	10	11	12	13	14	46	47	48
32	1	2	3	4	5	4	5	6	7	8	9	8	9	...	31	32	1



关键组件: S-盒代替

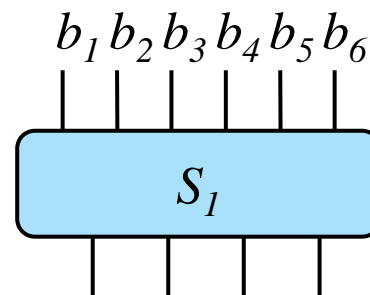
- 将48比特压缩成32比特





S-盒代替

- 输入6比特: $b_1b_2b_3b_4b_5b_6$
- 输出4比特: $S(b_1b_6, b_2b_3b_4b_5)$



		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S1	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S2	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1										
	2																
	3																

举例: $S_1(100110) = 1000$



关键组件: P盒置换

- S盒代替运算的32位输出按照P盒进行置换
- 该置换把输入的每位映射到输出位, 任何一位不能被映射两次, 也不能被略去, 映射规则如下表:

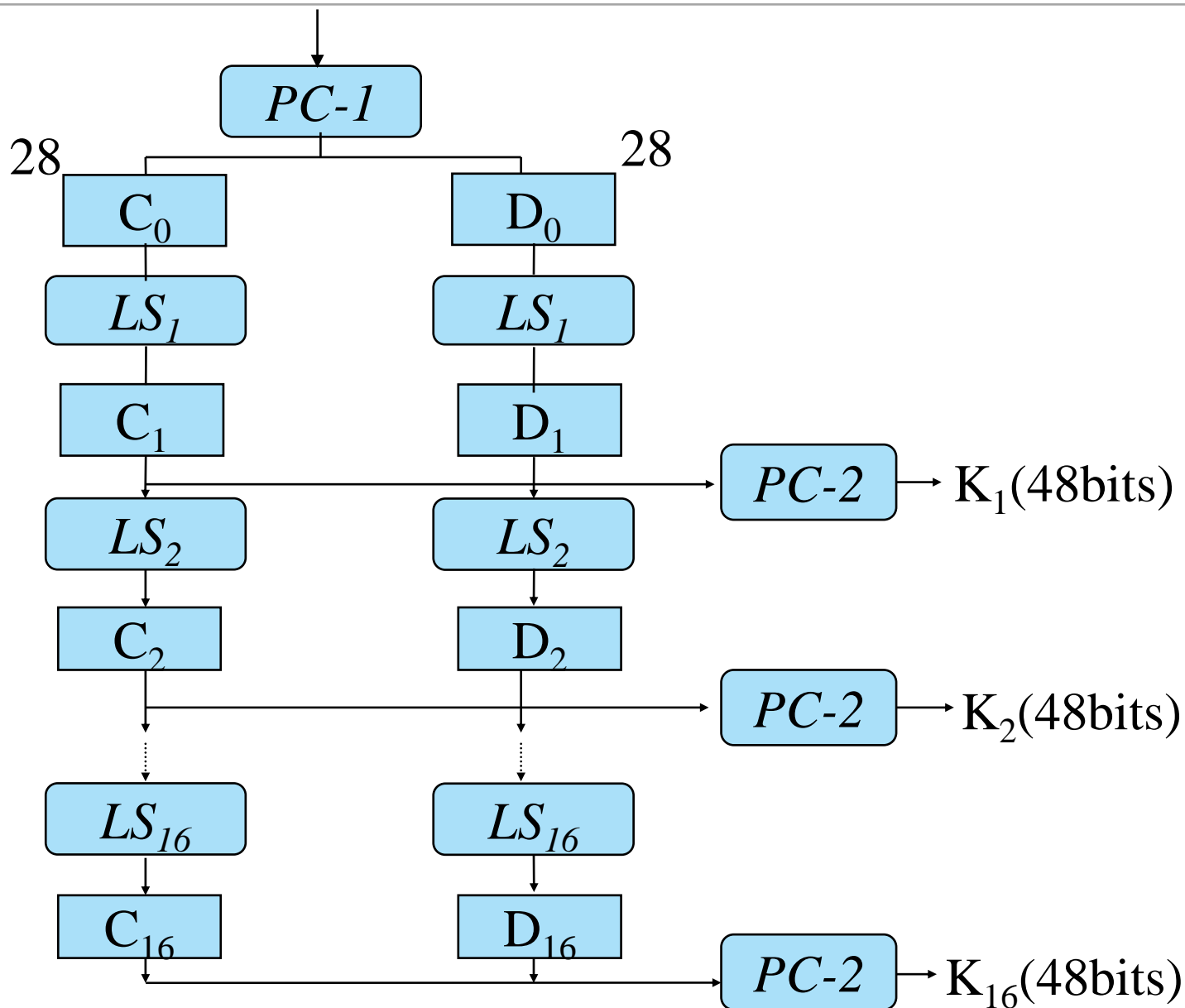
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

- 表中的数字代表原数据中此位置的数据在新数据中的位置, 即原数据块的第16位放到新数据的第1位, 第7位放到第2位,依此类推, 第25位放到第32位



子密钥生成

- DES算法会先对**64位密钥**进行处理生成**48位子密钥**后再参与到算法的轮操作中，在每一轮的迭代过程中，使用不同的子密钥
- 其中的处理包括置换选择 (PC-1)、循环左移 (LS)、压缩置换 (PC-2)





子密钥生成

- 拆分：56 bits 的密钥分成两部分， C_i , D_i ，各28bits
- 循环左移：根据迭代的轮数，分别左移一位或两位

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

- 压缩置换(置换选择)：从56bits中选择48bits

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32



子密钥生成

- 末置换

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

- 初始置换

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

$$IP^{-1}(IP(M))=M$$



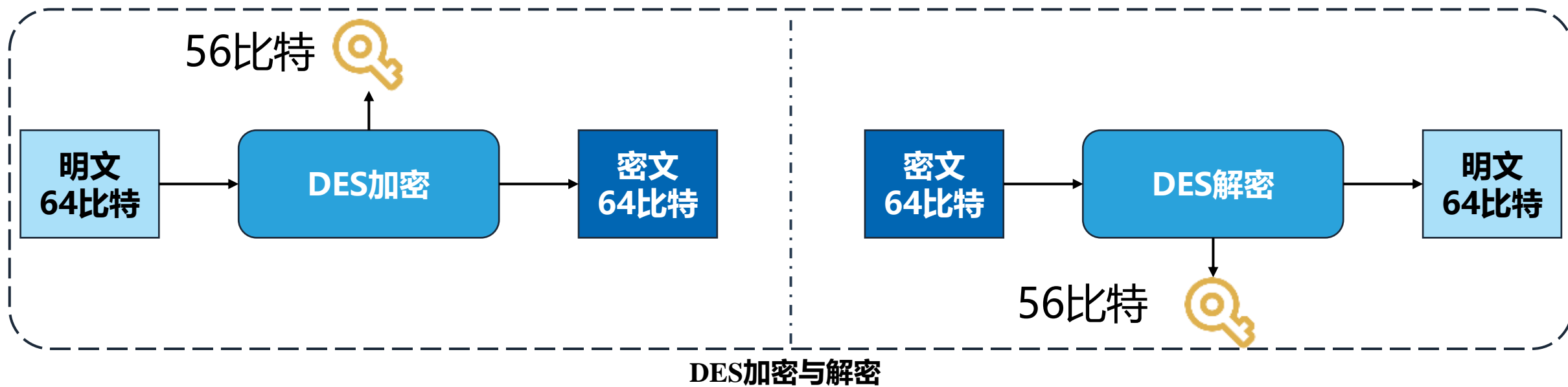
DES解密过程

- DES解密过程与加密过程完全相似，只不过将16次迭代的子密钥顺序倒过来，即

$$m = DES^{-1}(c) = IP^{-1} \cdot T_1 \cdot T_2 \cdot \dots \cdot T_{15} \cdot T_{16} \cdot IP(c)$$

- 可以证明，

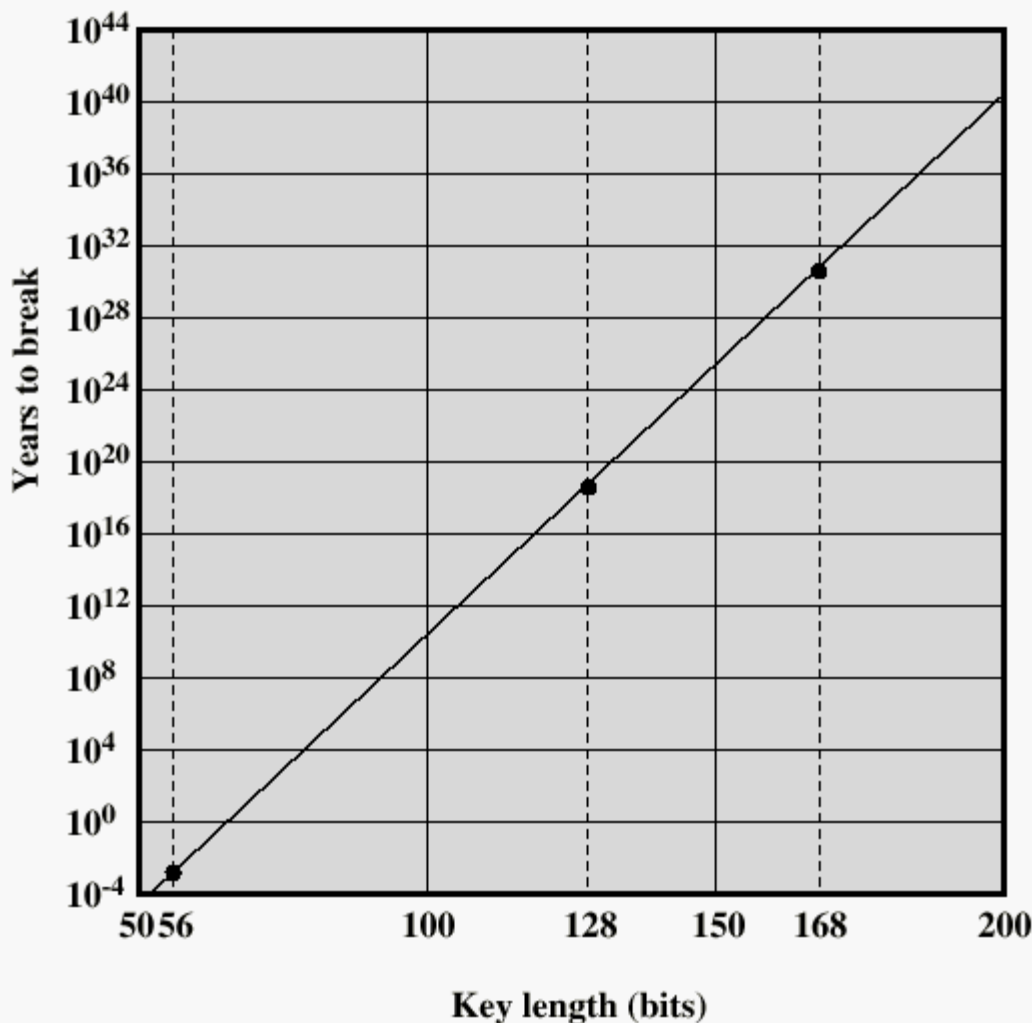
$$DES^{-1}(DES(m)) = m$$





软硬件实现与破译

- 硬件实现
 - 商业DES芯片或者FPGA实现
 - 目前可支持1.5Gbps以上的加解密速度
- 软件实现
 - 80486, CPU 66MHz, 每秒加密43000个DES分组, 336K Bytes/s
 - HP 9000/887, CPU 125 MHz, 每秒加密196,000个分组, 1.53M Bytes/s
- 破译速度
 - 56位密钥长度需要20小时即可破译
 - 128位密钥长度需要 5.4×10^{18} 年破译
 - 168位密钥长度需要 5.9×10^{30} 年破译



Time to break a code (10⁶ decryptions/μs)



DES的一般设计准则

随机性

输出与输入间是无规律的

雪崩效应

改变输入中的1位，平均导致大约一半的输出位被改变

非线性

每个输出位都是所有输入位的复杂函数

完全性

加密函数对于任何密钥值都是非线性的

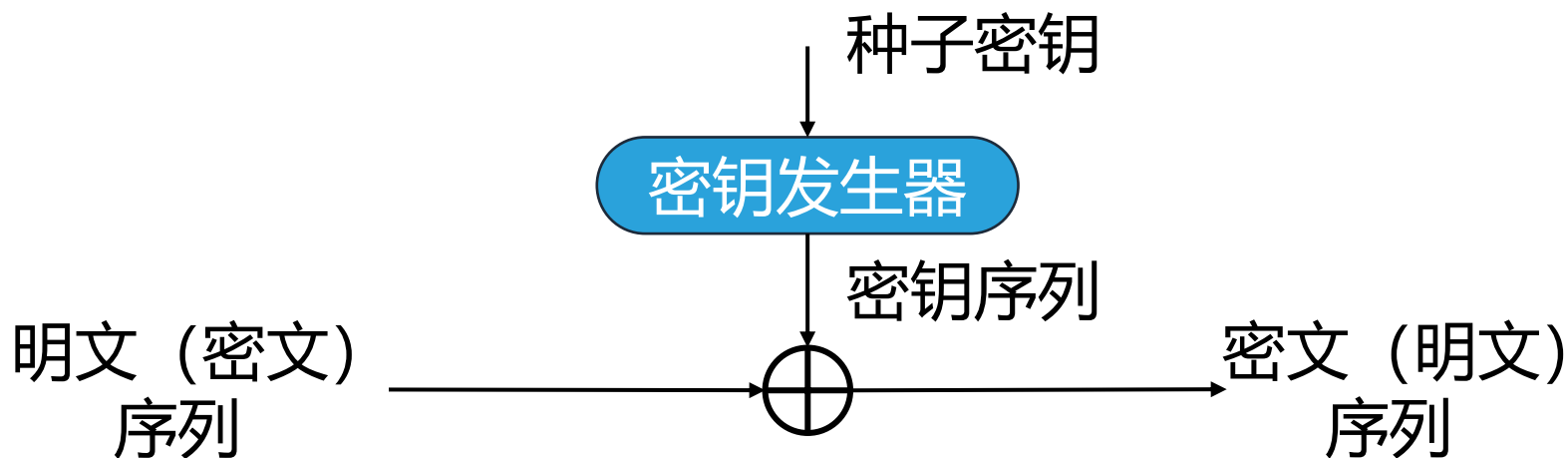
相关关系免疫性

输出统计上独立于任何输入位的子集



流密码

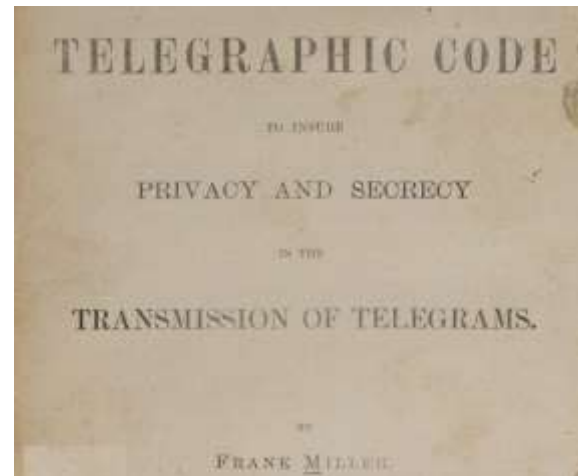
- 流密码是对数据流进行连续处理的一类密码算法
- 流密码中一般以1比特、8比特或32比特等单位进行加密或解密
- 流密码是对一串数据流进行连续处理，因此需要保持内部状态





一次性密码本

一次性密码本(One-Time Pad, OTP)在1882年被弗兰克·米勒(Frank Miller)发现, 并沿用至今



- 首先将明文morning这个字符串通过ASCII进行编码并产生一串字符串:

m	o	r	n	i	n	g	
01101101	01101111	01110010	01101110	01101001	01101110	01100111	morning

- 然后, 产生一个与明文长度相同的随机比特序列:

01111101	00001111	01101101	01111110	01101011	01111110	01100101	密钥
----------	----------	----------	----------	----------	----------	----------	----



一次性密码本

1.加密过程

将明文与密钥进行异或操作，得到的结果即为OTP的密文

01101101	01101111	01110010	01101110	01101001	01101110	01100111	morning
\oplus 01111101	00001111	01101101	01111110	01101011	01111110	01100101	密钥
00010000	01100000	00011111	00010000	00000010	00010000	00000010	密文

2.解密过程

解密是加密的反向运算。解密OPT的过程即是将密文和密钥进行异或运算的过程

00010000	01100000	00011111	00010000	00000010	00010000	00000010	密文
\oplus 01111101	00001111	01101101	01111110	01101011	01111110	01100101	密钥
01101101	01101111	01110010	01101110	01101001	01101110	01100111	明文
m	o	r	n	i	n	g	morning



一次性密码本

3. 无法被破译

即使能够解密出morning这个字符，我们也无法判断它是否是正确的明文

4. 局限性

密钥分发

若可安全地分发密钥，也可以用同样的方法来发送明文

密钥保存

若可安全地存储密钥，也可以用同样的方法来存储明文

密钥重用

密钥无法重用，一旦密钥泄露，过去所有的机密通信内容将全部被解密

密钥同步

发送者和接收者的密钥的比特序列不允许任何错位

密钥生成

需要生成大量的无重复性的随机数，密钥生成成本高

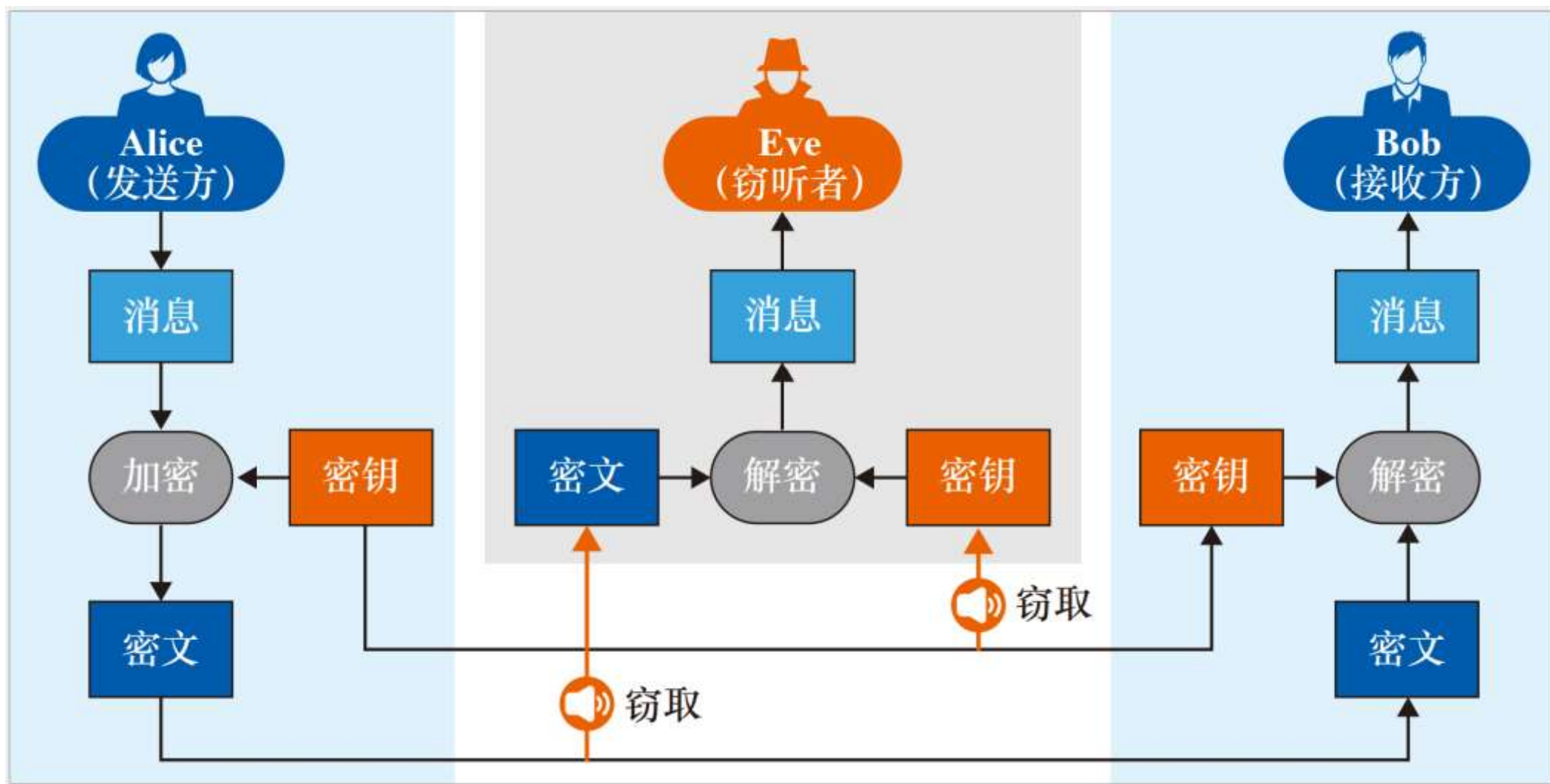


第3节 公钥密码

- ✓ 密钥分发问题
- ✓ 公钥密码
- ✓ RSA算法
- ✓ 实际应用



密钥分发问题





解决密钥分发问题

- **通过事先共享密钥来解决**

在加密通信以前，事先用安全的方式将密钥交给接收方

- **通过密钥分配中心来解决**

当参与加密通信的人过多，可以使用密钥分配中心(Key Distribution Center, KDC)。当需要进行加密通信时，密钥分配中心会生成一个通信密钥，每个人只需和KDC事先共享密钥即可

- **通过Diffie-Hellman密钥交换来解决**

进行加密通信的双方仅需要交换一些信息，该信息即使被窃听也无法对算法进行解密，而通信双方则可以通过该信息各自生成相同的密钥

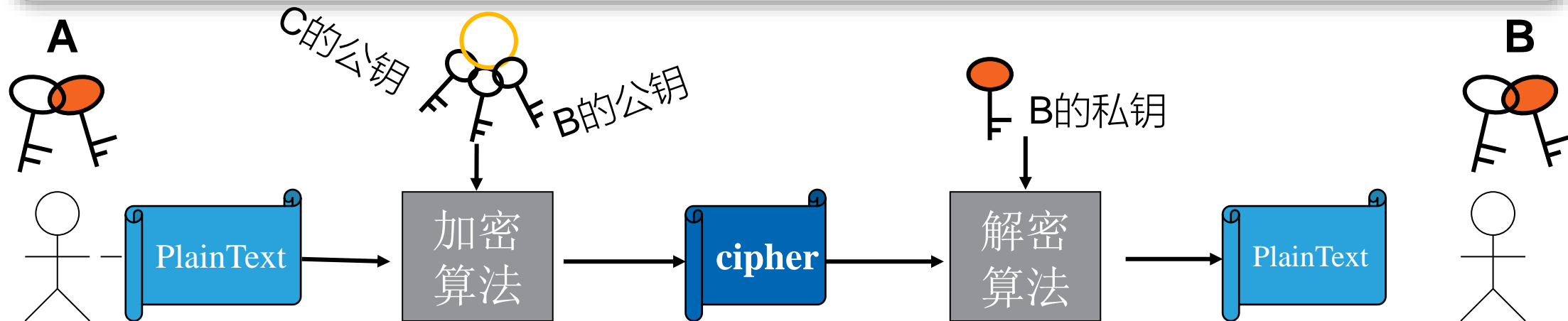
- **通过公钥密码来解决**

由于加解密使用不同密钥，故无需进行密钥分发



公钥密码系统的加密原理

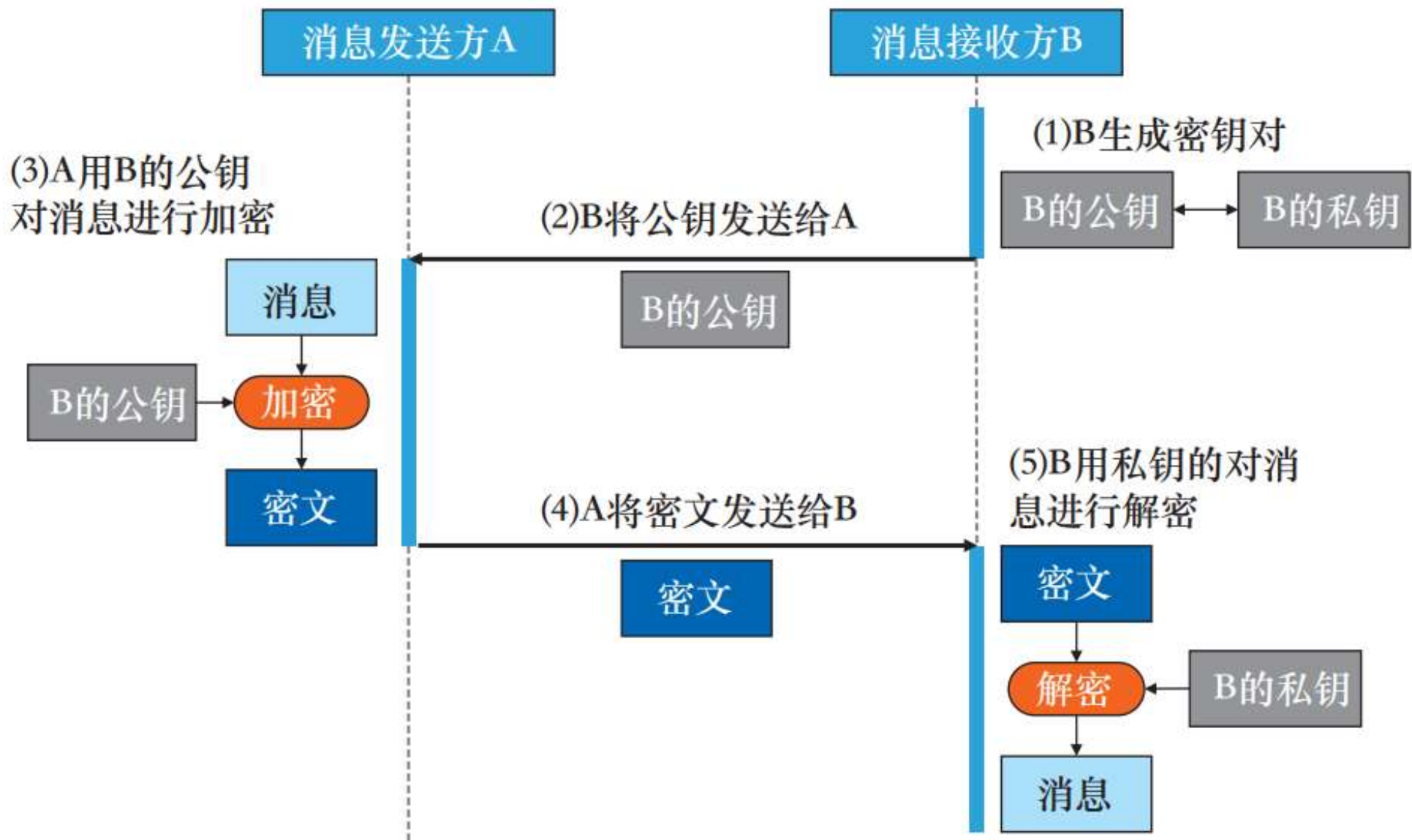
- 每个通信实体有一对密钥(公钥, 私钥)
 - 公钥公开, 用于加密和验证签名, 私钥保密, 用作解密和签名
- A向B 发送消息, 用B的公钥加密
- B收到密文后, 用自己的私钥解密



- 任何人向B发送信息都可以使用同一个密钥(B的公钥)加密
- 没有其他人可以得到B的私钥, 所以只有B可以解密

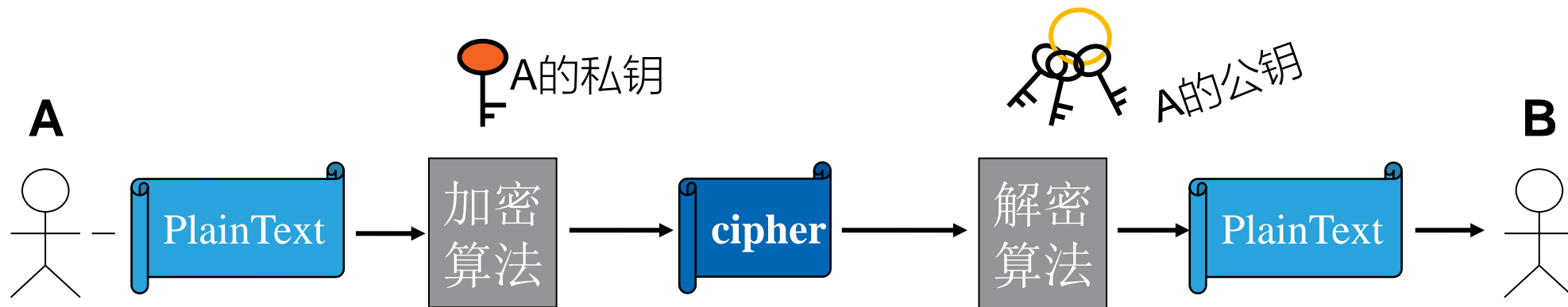


公钥密码系统的加密原理





公钥密码系统的签名原理



- A向B发送消息，用A的私钥加密(签名)
- B收到密文后，用A的公钥解密(验证)



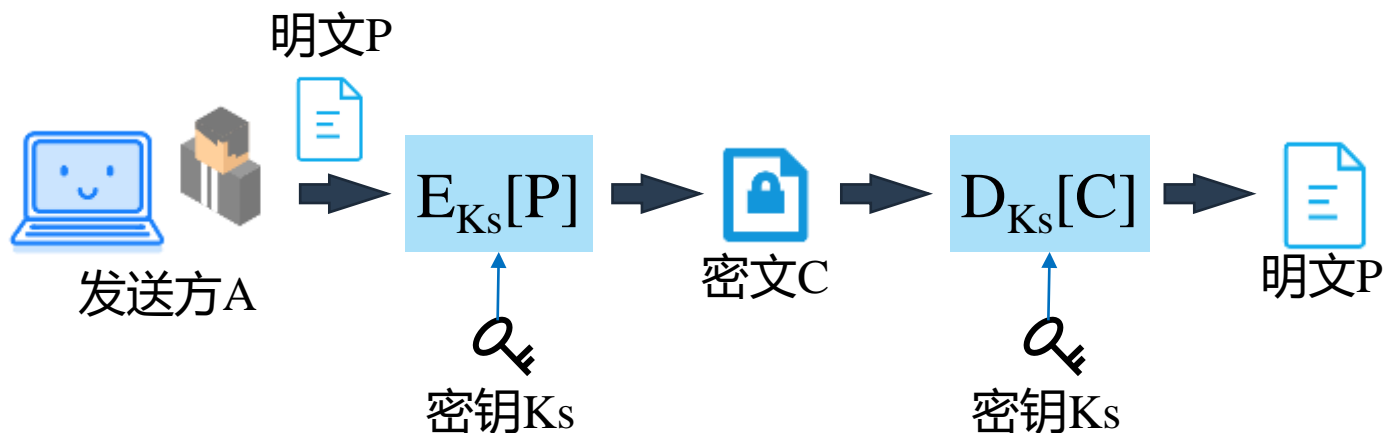
公钥密码算法的表示

- 对称密钥密码

- 密钥：会话密钥(K_s)

- 加密函数： $E_{K_s}[P]$

- 对密文 C ，解密函数： $P=D_{K_s}[C]$

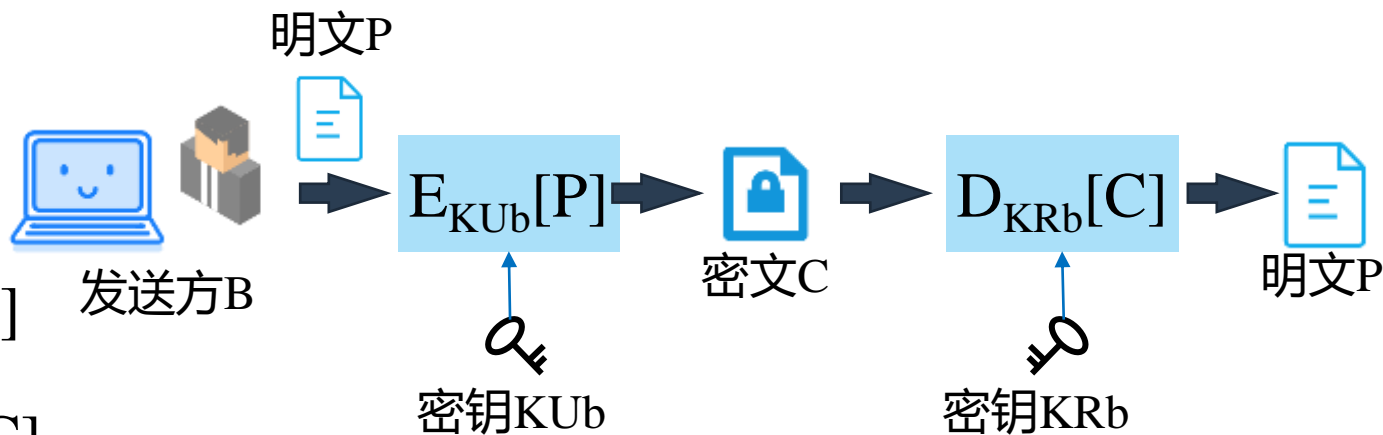


- 公开密钥

- (K_{Ua} , K_{Ra})

- 加密/签名： $C = E_{K_{Ub}}[P], E_{K_{Ra}}[P]$

- 解密/验证： $P = D_{K_{Rb}}[C], D_{K_{Ua}}[C]$





对公开密钥密码算法的要求

- 参与方B容易产生密钥对(K_{Ub} , K_{Rb})

- 已知 K_{Ub} , A的加密操作是容易的

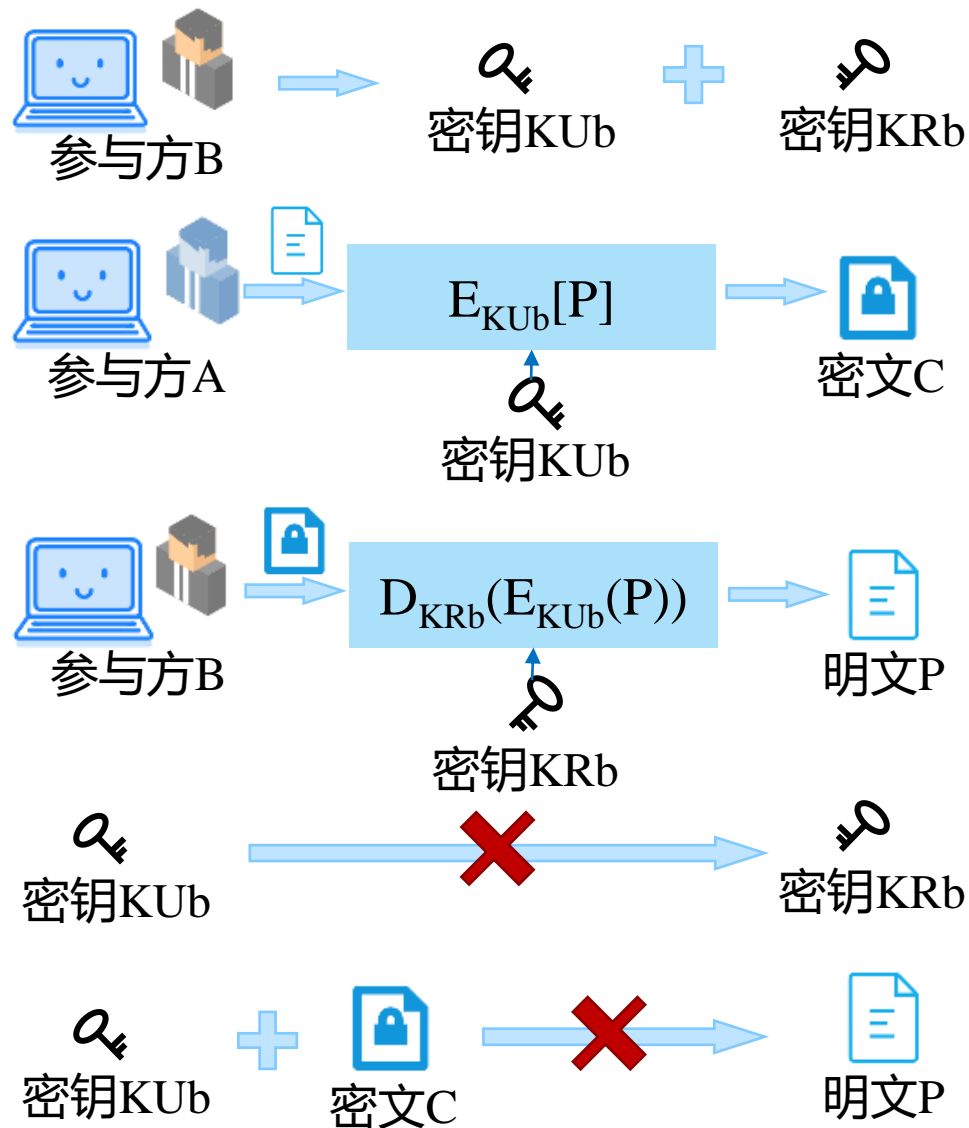
$$C = E_{K_{Ub}}(P)$$

- 已知 K_{Rb} , B的解密操作是容易的

$$P = D_{K_{Rb}}(C) = D_{K_{Rb}}(E_{K_{Ub}}(P))$$

- 已知 K_{Ub} , 求 K_{Rb} 是计算上不可行的

- 已知 K_{Ub} 和C, 欲恢复P是计算上不可行的





对公钥密码算法的误解

- 公开密钥算法比对称密钥密码算法更安全
 - 任何一种算法都依赖于密钥长度、破译密码的工作量，从抗分析角度，**没有一方更优越**
- 公开密钥算法使对称密钥成为过时的技术
 - 公开密钥很慢，只能用在密钥管理和数字签名，对称密钥密码算法将**长期存在**
- 使用公开密钥加密，密钥分配非常简单
 - 事实上的密钥分配既不简单，也不有效



RSA算法简介

- Ron Rivest, Adi Shamir , Leonard Adleman

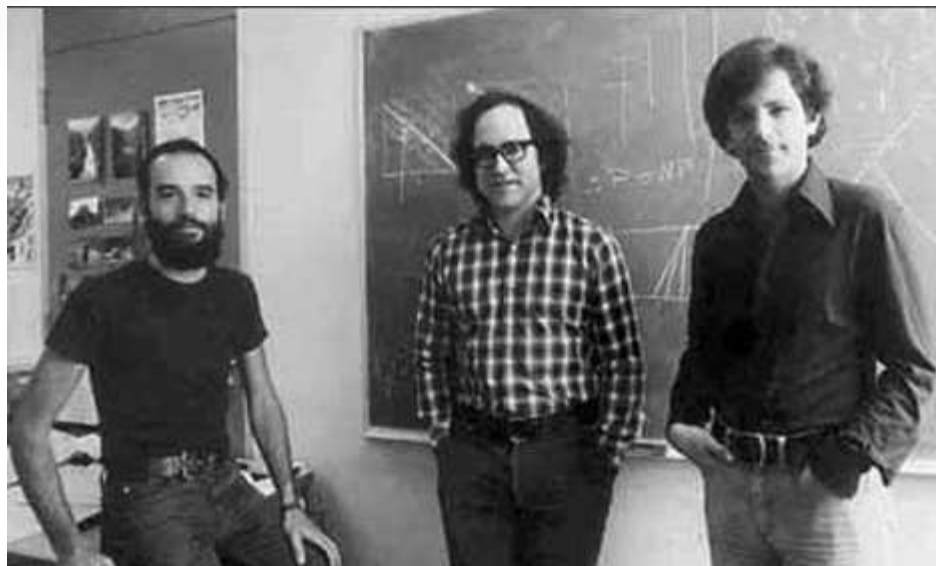
——获得2002年图灵奖

- RSA的安全性基于大数分解的难度
- RSA在美国申请了专利(已经过期), 在其他国家没有
- RSA已经成了事实上的工业标准, 在美国除外
- RSA算法既可以用于公钥密码也可以用于数字签名

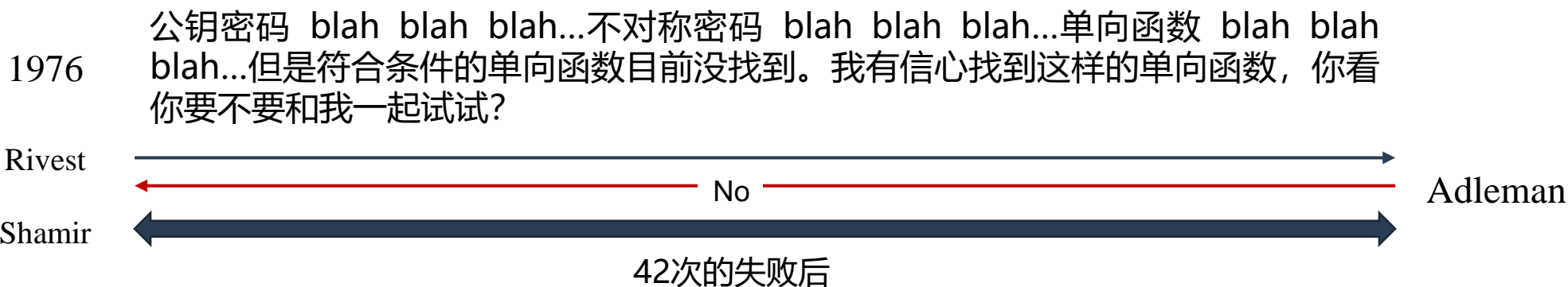




RSA算法诞生



- Ron Rivest、Adi Shamir 和 Leonard Adleman
三个人风格迥异，组成了一个技能互补的完美团队
- 1977年，三人在MIT合作发表了一篇完整描述RSA算法的论文，RSA也正是由三人姓氏的首字母组成



作者：RL Rivest - 1978 - 被引用次数：21052 - 相关文章

A method for obtaining digital signatures and public-key cryptosystems, Published by ACM 1978



RSA论文的评审意见

A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. According to the (very short) introduction, this paper purports to present a practical implementation of Diffie and Hellman's public-key cryptosystem for applications in the electronic mail realm. If this is indeed the premise, **the paper should be rejected both for a failure to live up to it and for its irrelevance.**

I doubt that a system such as this one will ever be practical. The paper does a poor job of convincing the reader that practicality is attainable.

The introduction is only two paragraphs long, the relevant literature is not presented or cited, and there is virtually no comparison with the relevant work in the area. In summary, it looks as if this paper is a mathematical exercise **with little originality** (the authors claim that most of their ideas come from other papers), **too far from practical applicability, running against the established standards**, and **with a declared application area of dubious feasibility**. Not the kind of material our readers like to see in the journal. **Reject.**



RSA算法的性能表现

- **安全性**

- 蛮力攻击：对所有密钥都进行尝试
- 数学攻击：等效于对两个素数乘积(n)的因子分解

- **大数的因子分解**是数论中的一个难题

因子分解的进展

十进制数字位数	近似比特数	得到的数据	MIPS年
100	332	1991	7
110	365	1992	75
120	398	1993	830
129	428	1994	5000
130	431	1996	500

- **速度**

- 软件实现比DES慢100倍
- 硬件实现比DES慢1000倍

8位公开密钥的RSA

	512位	768位	1024位
加密 (秒)	0.03	0.05	0.08
解密 (秒)	0.16	0.48	0.93
签名 (秒)	0.16	0.52	0.97
验证 (秒)	0.02	0.07	0.08



RSA算法操作过程

- 密钥产生

1. 取两个大素数 p, q , **保密**

2. 计算 $n=pq$, 公开 n

3. 计算欧拉函数 $\varphi(n) = (p-1)(q-1)$

4. 任意取一个与 $\varphi(n)$ 互素的小整数 e , 即

$$\gcd(e, \varphi(n)) = 1; 1 < e < \varphi(n)$$

5. 寻找 d , $d < \varphi(n)$, 使得 $de \equiv 1 \pmod{\varphi(n)}$, $de = k\varphi(n) + 1$

公开 $(e, n) = (5, 119)$

将 d 保密, 丢弃 p, q

$$p=7, q=17$$

$$n=119$$

$$\varphi(n)=96$$

$$\text{取 } e=5$$

$$5d = k \times 96 + 1$$

取 $k=4$, 得到
 $d=77$



RSA算法加密/解密

- 公开密钥: $KU=\{e, n\}$
- 秘密密钥: $KR=\{d, n\}$
- 加密过程
 - 把待加密的内容分成k比特的分组
 $k \leq \log_2 n$, 并写成数字, 设为M, 则
 $C = M^e \bmod n$
- 解密过程

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

Remember:

$$ed = k\varphi(n) + 1$$

$$M^{k\varphi(n)+1} \equiv M \bmod n$$

$$\{5, 119\}$$

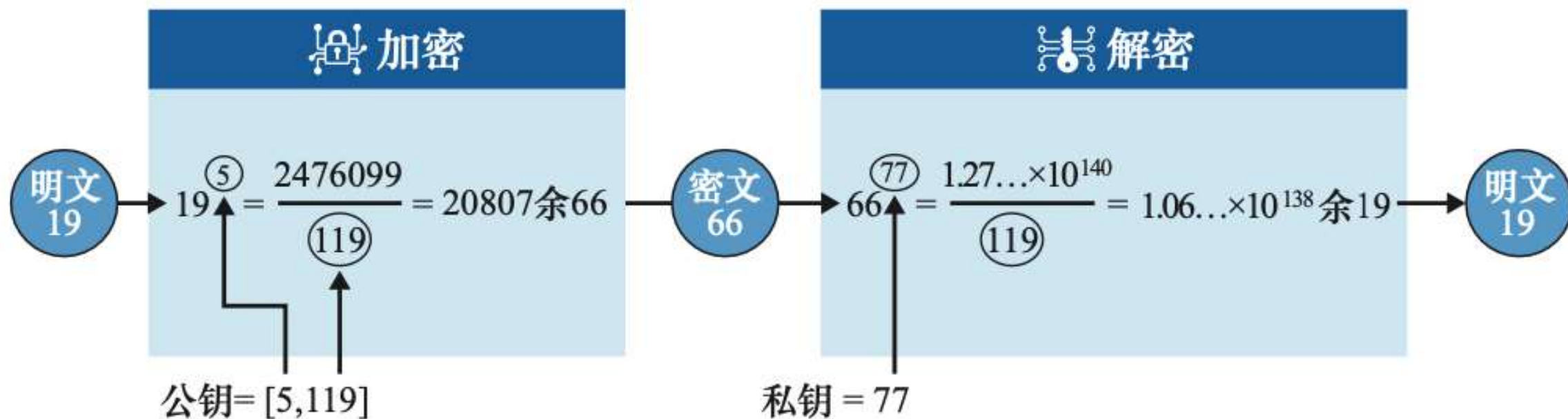
$$\{77, 119\}$$

$$c = m^5 \bmod 119$$

$$m = c^{77} \bmod 119$$



RSA算法举例





公钥密码系统的应用

- 加密/解密
- 数字签名：发送方用自己的私钥签署报文，接收方用对方的公钥验证对方的签名
- 密钥交换：双方协商会话密钥

算法	加密/解密	数字签名	密钥交换
RSA	Y	Y	Y
Diffie-Hellman	N	N	Y
DSA	N	Y	N



第4节 摘要与签名

- ✓ 散列函数
- ✓ 消息认证码
- ✓ 数字签名



数字签名与摘要

- 将数据比作一个幼儿
- 数据加密即使给幼儿“穿衣服”
- 数字签名在衣服上留下“印记”
- 摘要是孩子的“健康证明”



健康证

姓名: _____

性别: _____

年龄: _____

编号: _____

有效期至 _____ 年 _____ 月 _____ 日

照片

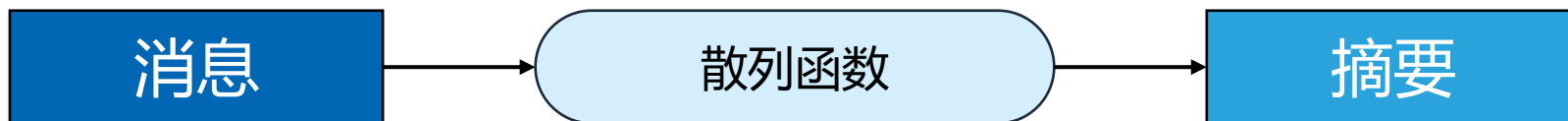


散列函数

- 散列函数又叫哈希 (Hash) 函数
- 把任意长度的消息压缩为固定长度的二进制串：散列值(hash value):

$$h = H(m)$$

- 散列函数是进行消息认证的基本方法，主要用于消息完整性检测和数字签名





散列函数

- 纵向的奇偶校验码

	比特1	比特2	比特n
分组1	b_{11}	b_{21}	...	b_{n1}
分组2	b_{12}	b_{22}	...	b_{n2}

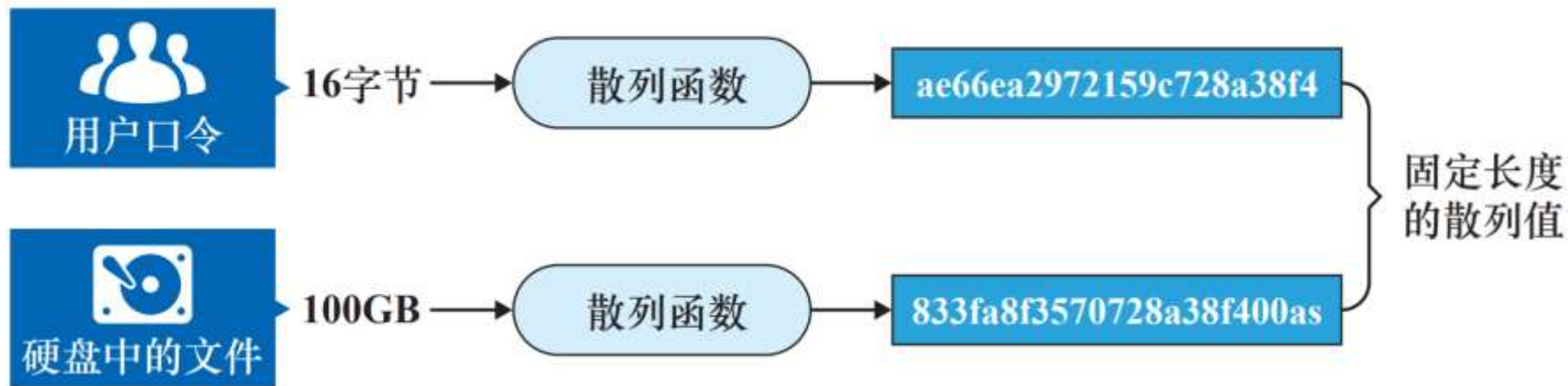
分组m	b_{1m}	b_{2m}		b_{nm}
散列码	C_1	C_2		C_n

$$C_i = b_{i1} \oplus b_{i2} \dots \oplus b_{im}$$



散列函数的性质

1. 根据任意长度的消息计算出固定长度的散列值



2. 能够快速计算出散列值

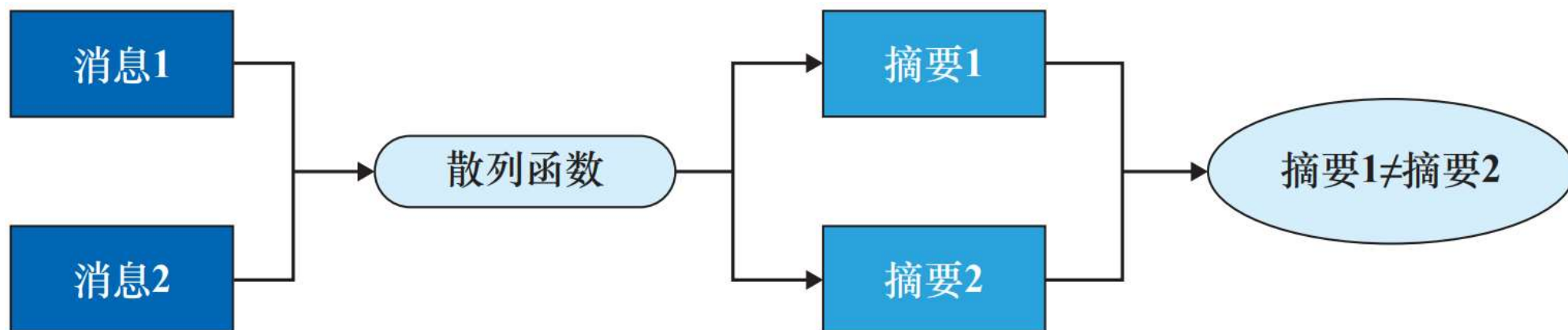
3. 单向性





散列函数的性质

4.消息不同散列值也不同





安全性：HASH 碰撞

- 2005年5月国际密码学会议（Eurocrypt'2005）**王小云院士**发表论文“ How to break MD5 and other hash functions”
- 成功破译**MD5、HAVAL-128、MD4和RIPEMD**算法
- 该次会议的总结报告写道：“我们该怎么办？MD5被重创了，它即将从应用中淘汰。SHA-1仍然活着，但也看到了它的末日，现在就得开始更换SHA-1了”



王小云院士

How to Break MD5 and Other Hash Functions

Xiaoyun Wang and Hongbo Yu

Shandong University, Jinan 250100, China

xywang@sdu.edu.cn

yhb@mail.sdu.edu.cn

Abstract. MD5 is one of the most widely used cryptographic hash functions nowadays. It was designed in 1992 as an improvement of MD4, and its security was widely studied since then by several authors. The best known result so far was a semi free-start collision, in which the initial value of the hash function is replaced by a non-standard value, which is the result of the attack. In this paper we present a new powerful attack on MD5 which allows us to find collisions efficiently. We used this attack to find collisions of MD5 in about 15 minutes up to an hour computation time. The attack is a differential attack, which unlike most differential attacks, does not use the exclusive-or as a measure of difference, but instead uses modular integer subtraction as the measure. We call this kind of differential a *modular differential*. An application of this attack to MD4 can find a collision in less than a fraction of a second. This attack is also applicable to other hash functions, such as RIPEMD and HAVAL.

1 Introduction

People know that digital signatures are very important in information security. The security of digital signatures depends on the cryptographic strength of the underlying hash functions. Hash functions also have many other applications in cryptography such as data integrity, group signature, e-cash and many other cryptographic protocols. The use of hash functions in these applications not only ensure the security, but also greatly improve the efficiency. Nowadays, there are two widely used hash functions – MD5 [18] and SHA-1 [12].

MD5 is a hash function designed by Ron Rivest as a strengthened version of MD4 [17]. Since its publication, some weaknesses has been found. In 1993, B. den Boer and A. Bosselaers [3] found a kind of pseudo-collision for MD5 which consists of the same message with two different sets of initial values. This attack discloses the weak avalanche in the most significant bit for all the chaining variables in MD5. In the rump session of Eurocrypt'96, H. Dobbertin [8] presented a semi free-start collision which consists of two different 512-bit messages with a chosen initial value IV'_0 .

$a_0 = 0x12ac2375$, $b_0 = 0x3b341042$, $c_0 = 0x5f62b97c$, $d_0 = 0x4ba763ed$

A general description of this attack was published in [9].

R. Cramer (Ed.): EUROCRYPT 2005, LNCS 3494, pp. 19–[5] 2005.
© International Association for Cryptologic Research 2005



SHA1碰撞的理论计算上界

- 2005年2月7日，美国国家标准技术研究院NIST对外宣称：SHA-1还没有被攻破，并且也没有足够的理由怀疑它会很快被攻破
- **2005年8月，王小云教授再度令世界密码学界大跌眼镜——SHA-1也被她攻破了！王教授提出了复杂度为 2^{69} 的理论攻击**
- Wang, X., Yin, Y.L., Yu, H. (2005). **Finding Collisions in the Full SHA-1**. In: Shoup, V. (eds) *Advances in Cryptology – CRYPTO 2005*. CRYPTO 2005. Lecture Notes in Computer Science, vol 3621. Springer, Berlin, Heidelberg

Finding Collisions in the Full SHA-1

Xiaoyun Wang^{1,*}, Yiqun Lisa Yin², and Hongbo Yu³

¹ Shandong University, Jinan 250100, China
xywang@sdu.edu.cn

² Independent Security Consultant, Greenwich CT, US
yyin@princeton.edu

³ Shandong University, Jinan 250100, China
yhb@mail.sdu.edu.cn

Abstract. In this paper, we present new collision search attacks on the hash function SHA-1. We show that collisions of SHA-1 can be found with complexity less than 2^{69} hash operations. This is the first attack on the full 80-step SHA-1 with complexity less than the 2^{80} theoretical bound.

Keywords: Hash functions, collision search attacks, SHA-1, SHA-0.

1 Introduction

The hash function SHA-1 was issued by NIST in 1995 as a Federal Information Processing Standard [5]. Since its publication, SHA-1 has been adopted by many government and industry security standards, in particular standards on digital signatures for which a collision-resistant hash function is required. In addition to its usage in digital signatures, SHA-1 has also been deployed as an important component in various cryptographic schemes and protocols, such as user authentication, key agreement, and pseudorandom number generation. Consequently, SHA-1 has been widely implemented in almost all commercial security systems and products.

In this paper, we present new collision search attacks on SHA-1. We introduce a set of strategies and corresponding techniques that can be used to remove some major obstacles in collision search for SHA-1. Firstly, we look for a near-collision differential path which has low Hamming weight in the “disturbance vector” where each 1-bit represents a 6-step local collision. Secondly, we suitably adjust the differential path in the first round to another possible differential path so as to avoid impossible consecutive local collisions and truncated local collisions. Thirdly, we transform two one-block near-collision differential paths into a two-block collision differential path with twice the search complexity. We show that, by combining these techniques, collisions of SHA-1 can be found with complexity less than 2^{69} hash operations. This is the first attack on the *full* 80-step SHA-1 with complexity less than the 2^{80} theoretical bound.

* Supported by the National Natural Science Foundation of China (NSFC Grant No.90304009) and Program for New Century Excellent Talents in University.

V. Shoup (Ed.): *Crypto 2005*, LNCS 3621, pp. 17–36, 2005.
© International Association for Cryptologic Research 2005





安全性：HASH碰撞

2017年2月23号，Google在其安全博客上公布了其找到了世界首例的SHA1碰撞，标志着SHA1不再安全了

Collision Attack: Two Different Documents, But Same SHA-1 Hash Fingerprint

SHattered

The first concrete collision attack against SHA-1
<https://shattered.io>



Marc Stevens
Pierre Karpman

Elie Bursztein
Ange Albertini
Yarik Markov

SHattered

The first concrete collision attack against SHA-1
<https://shattered.io>



Marc Stevens
Pierre Karpman

Elie Bursztein
Ange Albertini
Yarik Markov

```
└─ sha1sum *.pdf
38762cf7f55934b34d179ae6a4c80cadccbb7f0a 1.pdf
38762cf7f55934b34d179ae6a4c80cadccbb7f0a 2.pdf
└─ /tmp/sha1
└─ sha256sum *.pdf
2bb787a73e37352f92383abe7e2902936d1059ad9f1ba6daaa9c1e58ee6970d0 1.pdf
```

0.64G 8-11h



哈希算法实例

算法名称	输出长度(位)	简述
MD4	128	由美国密码学家Rivest于1990年针对32位操作系统所设计的哈希算法，虽然于1996年被破解，但对后来的MD5，SHA系列和RIPEMD算法产生了深远影响
MD5	128	1991年由Rivest对MD4算法改进而来，它在MD4的基础上增加了“安全带”的概念。2004年其强抗碰撞性被王小云院士团队攻破
SHA-1	160	由NIST（美国国家标准技术研究所）MD5的基础上改进而来，其强抗碰撞性于2005年被王小云院士团队攻破
SHA-256	256	SHA-2系列的一个版本，由NIST设计，用到了8个哈希初值以及32个哈希常量。该算法被用于生成比特币地址
SHA-512	512	SHA-2系列的一个版本，由NIST设计，用到了8个哈希初值和64个哈希常量
SHA-3	任意长度	在2005年SHA-1的强抗碰撞性被攻破的背景下，NIST开始设计下一代哈希函数SHA-3。2012年10月，NIST选择Keccak算法作为SHA-3的标准算法



散列函数的应用: 比特币



- 比特币是一种P2P形式的虚拟的加密数字货币，使用了SHA256、RIPEMD160
- SHA256
 - 输出：256位的哈希值
 - 用途：区块的头部信息、交易数据、工作量证明、比特币地址等
- RIPEMD160
 - 输出：160位的哈希值
 - 用途：比特币地址生成，可以让地址更短



SHA256算法

SHA256是SHA-2标准下细分出的一种散列函数。算法的实现主要分为三个步骤：实现可以分为三个部分：常量初始化，信息预处理，生成摘要





SHA256算法

1. 常量初始化

SHA256算法中用到了8个哈希初值以及64个哈希常量。其中8个哈希初值，分别是最开始的连续8个素数（即2,3,5,7,11,13,17,19）的平方根的二进制表示的小数部分的前32位

$h0 := 0x6a09e667$	$h1 := 0xbb67ae85$	$h2 := 0x3c6ef372$	$h3 := 0xa54ff53a$
$h4 := 0x510e527f$	$h5 := 0x9b05688c$	$h6 := 0x1f83d9ab$	$h7 := 0x5be0cd19$

还要初始化64个值，分别是最开始的连续64个素数（从2到311）的立方根的二进制表示的小数部分的前32位

428a2f98	71374491	b5c0fbcf	e9b5dba5	3956c25b	59f111f1	923f82a4	ab1c5ed5
d807aa98	12835b01	243185be	550c7dc3	72be5d74	80deb1fe	9bdc06a7	c19bf174
e49b69c1	efbe4786	0fc19dc6	240ca1cc	2de92c6f	4a7484aa	5cb0a9dc	76f988da
983e5152	a831c66d	b00327c8	bf597fc7	c6e00bf3	d5a79147	06ca6351	14292967
27b70a85	2e1b2138	4d2c6dfc	53380d13	650a7354	766a0abb	81c2c92e	92722c85
a2bfe8a1	a81a664b	c24b8b70	c76c51a3	d192e819	d6990624	f40e3585	106aa070
19a4c116	1e376c08	2748774c	34b0bcb5	391c0cb3	4ed8aa4a	5b9cca4f	682e6ff3
748f82ee	78a5636f	84c87814	8cc70208	90befffa	a4506ceb	bef9a3f7	c67178f2



SHA256算法

2. 信息预处理

信息的预处理分为两个步骤：附加**填充比特**和**附加长度**。以信息“abc”为例进行预处理演示

- 填充比特：第一个填充的比特为1，其后的填充的比特都为0，直到长度满足对512取模后余数是448
- 附加长度：将原始数据的长度信息补到已经进行了填充操作的消息之后

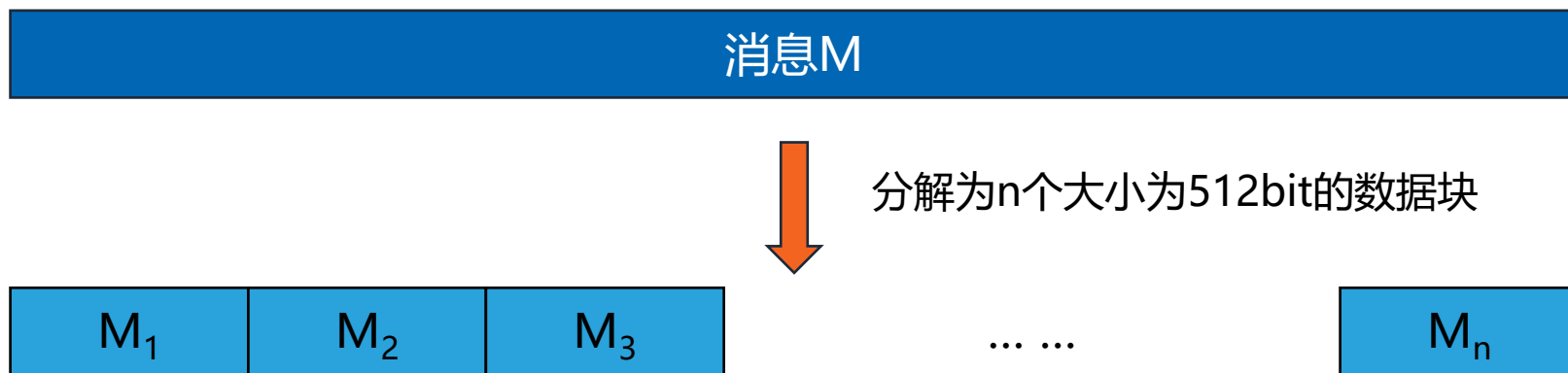
信息	a	b	c
ASCII码（二进制）	01100001	01100010	01100011
补位（第一位补“1”）	01100001 01100010 01100011 1		
补位（其后位补“0”）	01100001 01100010 01100011 10000000 00000000 ... 00000000(423个0)		
比特填充结果（十六进制）	61626380 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000		
附加长度结果（十六进制）	61626380 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000018		



SHA256算法

3. 生成摘要

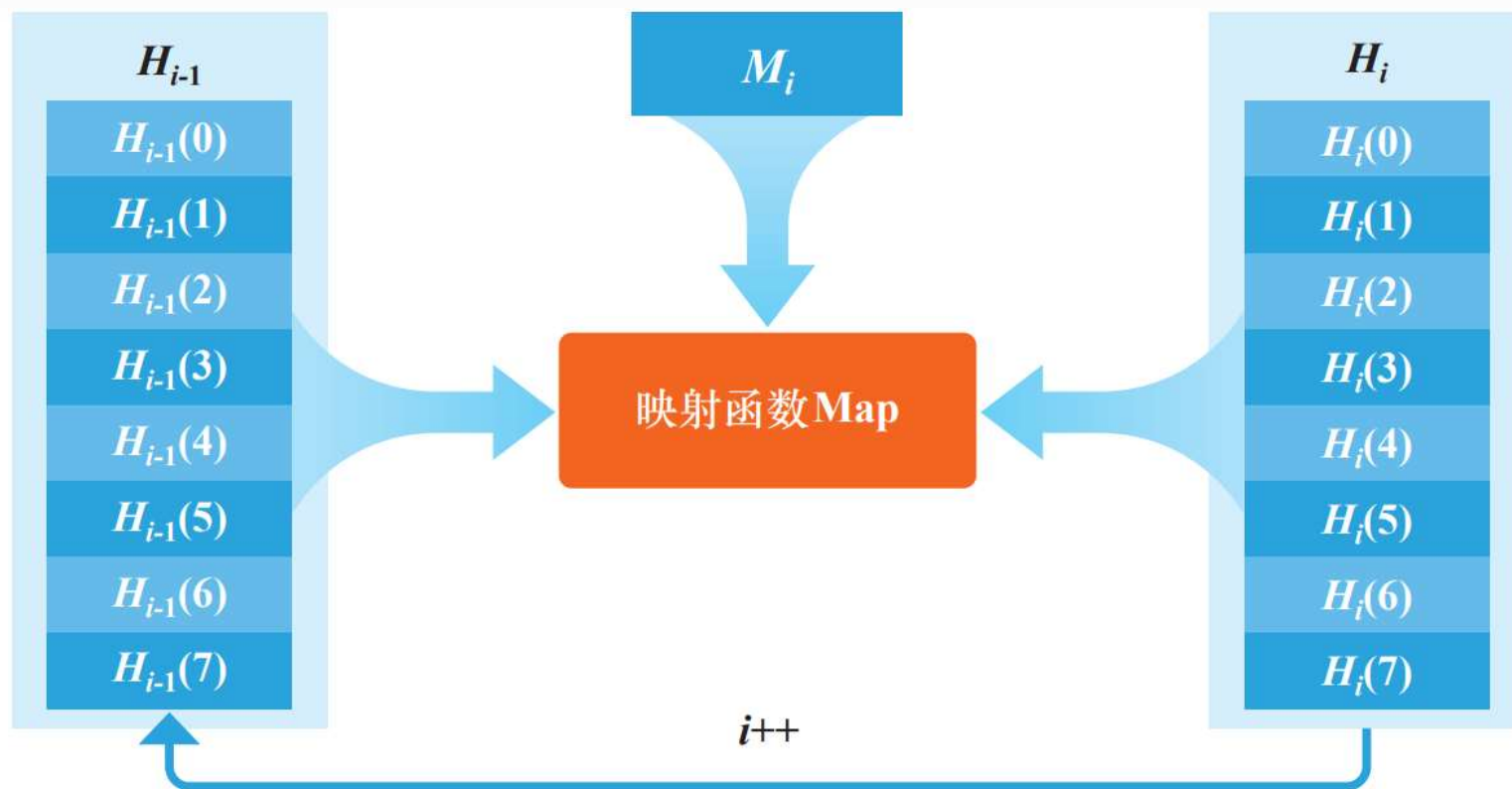
假设消息M可以被分解为n个块，于是整个算法需要做的就是完成n次迭代，n次迭代的结果就是最终的哈希值，即256bit的数字摘要





SHA256算法

- 一个256-bit的摘要的初始值 H_0 ，经过第一个数据块进行运算，得到 H_1 ，即完成了第一次迭代， H_1 经过第二个数据块得到 H_2 ，依次处理，最后得到 H_n ， H_n 即为最终的256-bit消息摘要
- 将每次迭代进行的映射用 $Map(H_{i-1}) = H_i$ 表示，于是迭代可以更形象的展示为：



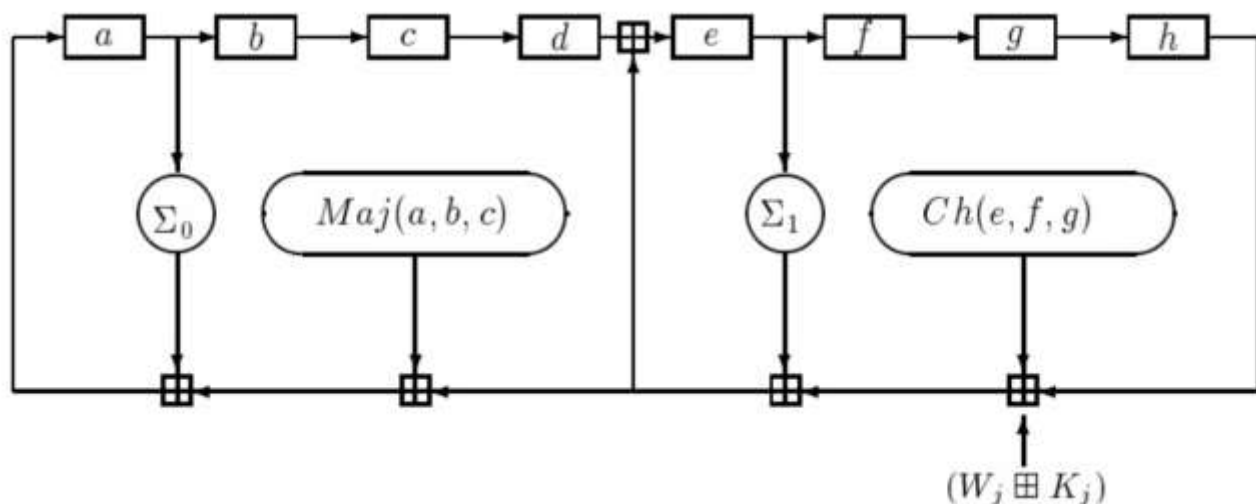


SHA256中的映射函数

Step1: 构造64个32bit字 (word) , 将每一块分解为16个32-bit的big-endian的字, 记为 $w[0], \dots, w[15]$, 前16个字直接由消息的第 i 个块分解得到, 其余的字由如下迭代公式得到:

$$W_t = \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16}$$

Step2: 进行64次循环



$$Ch(x, y, z) = (x \wedge y) \oplus (\neg x \wedge z)$$

$$Maj(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$$

$$\Sigma_0(x) = S^2(x) \oplus S^{13}(x) \oplus S^{22}(x)$$

$$\Sigma_1(x) = S^6(x) \oplus S^{11}(x) \oplus S^{25}(x)$$

$$\sigma_0(x) = S^7(x) \oplus S^{18}(x) \oplus R^3(x)$$

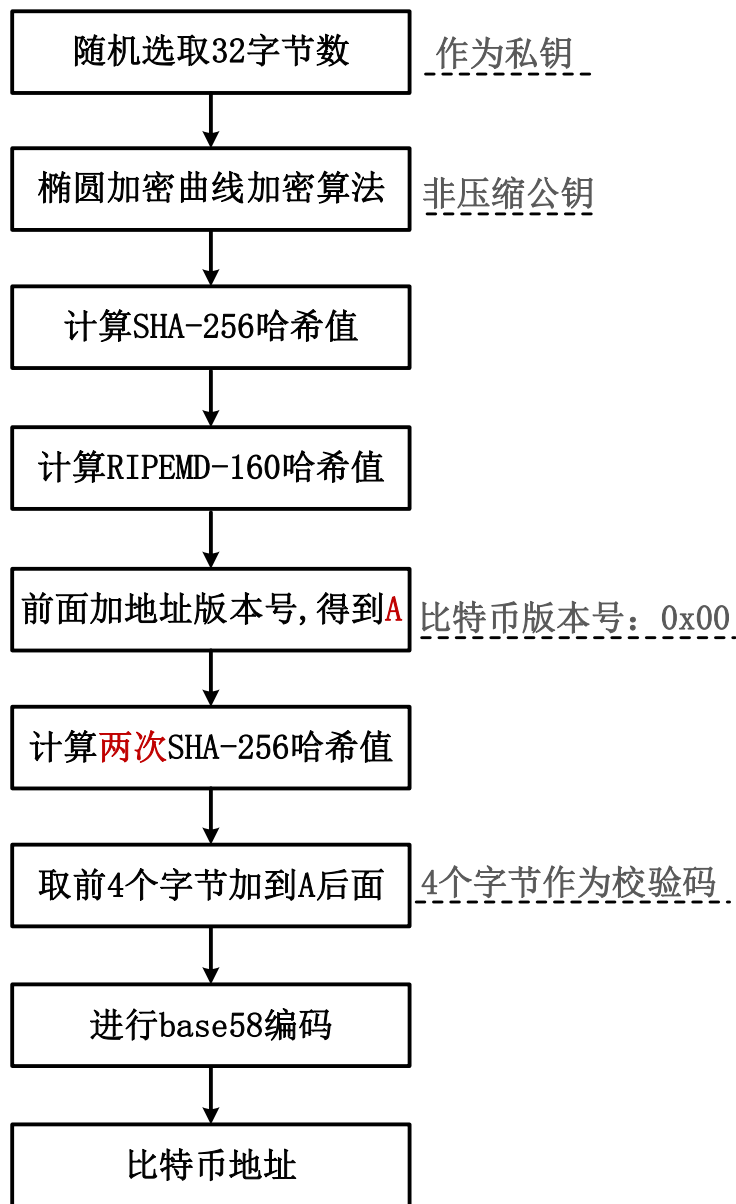
$$\sigma_1(x) = S^{17}(x) \oplus S^{19}(x) \oplus R^{10}(x)$$

\oplus	bitwise XOR
\wedge	bitwise AND
\vee	bitwise OR
\neg	bitwise complement
$+$	mod 2^{32} addition
R^n	right shift by n bits
S^n	right rotation by n bits



比特币地址的生成

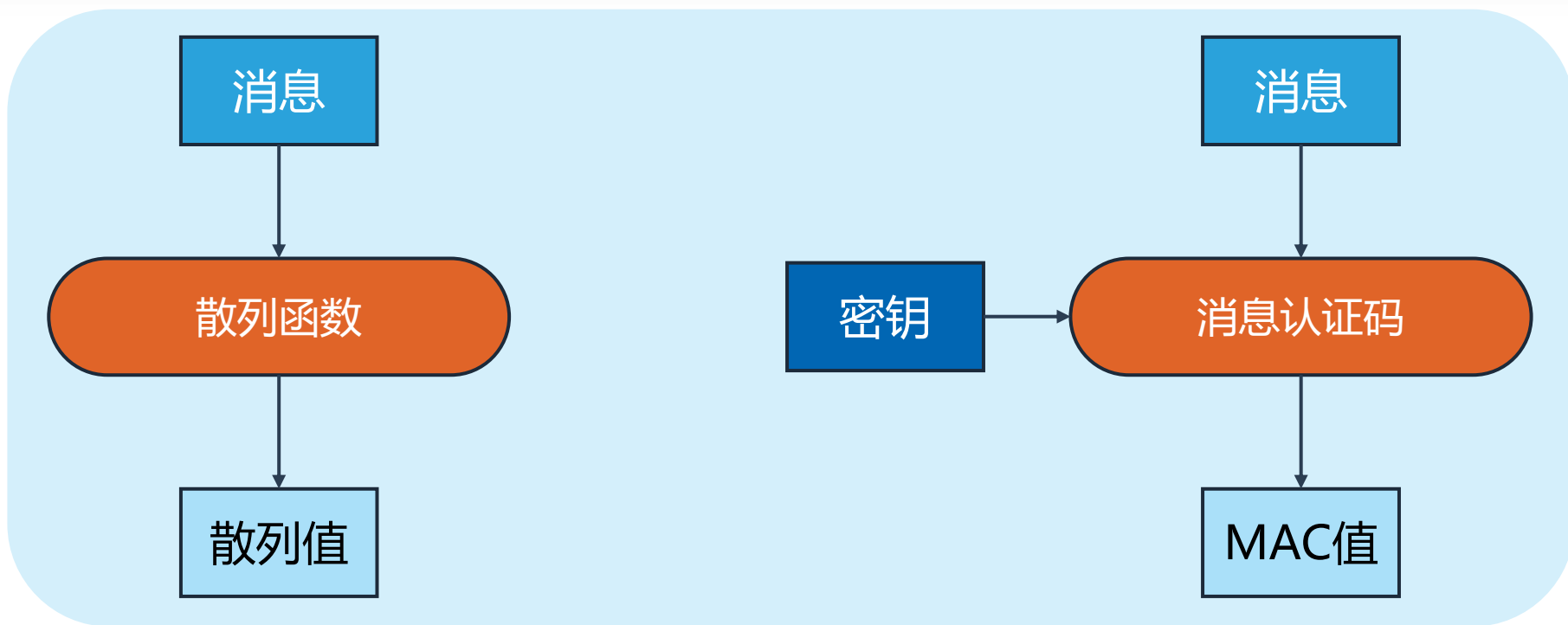
- 比特币地址的生成：
 1. 随机选取32字节的数作为私钥
 2. 椭圆曲线加密算法得到非压缩公钥
 3. 依次计算SHA-256、RIPEMD-160值
 4. 上一结果前加地址版本号 (0x00)
 5. 计算两次SHA-256
 6. 取前4个字节加到第4步结果后面
 7. 进行base58编码
 8. 获得比特币地址





消息认证码

消息认证码 (Message Authentication Code) 的输入包括任意长度的消息和一个发送者与接受者之间共享的密钥，它可以输出固定长度的数据，这个数据称为**MAC值**

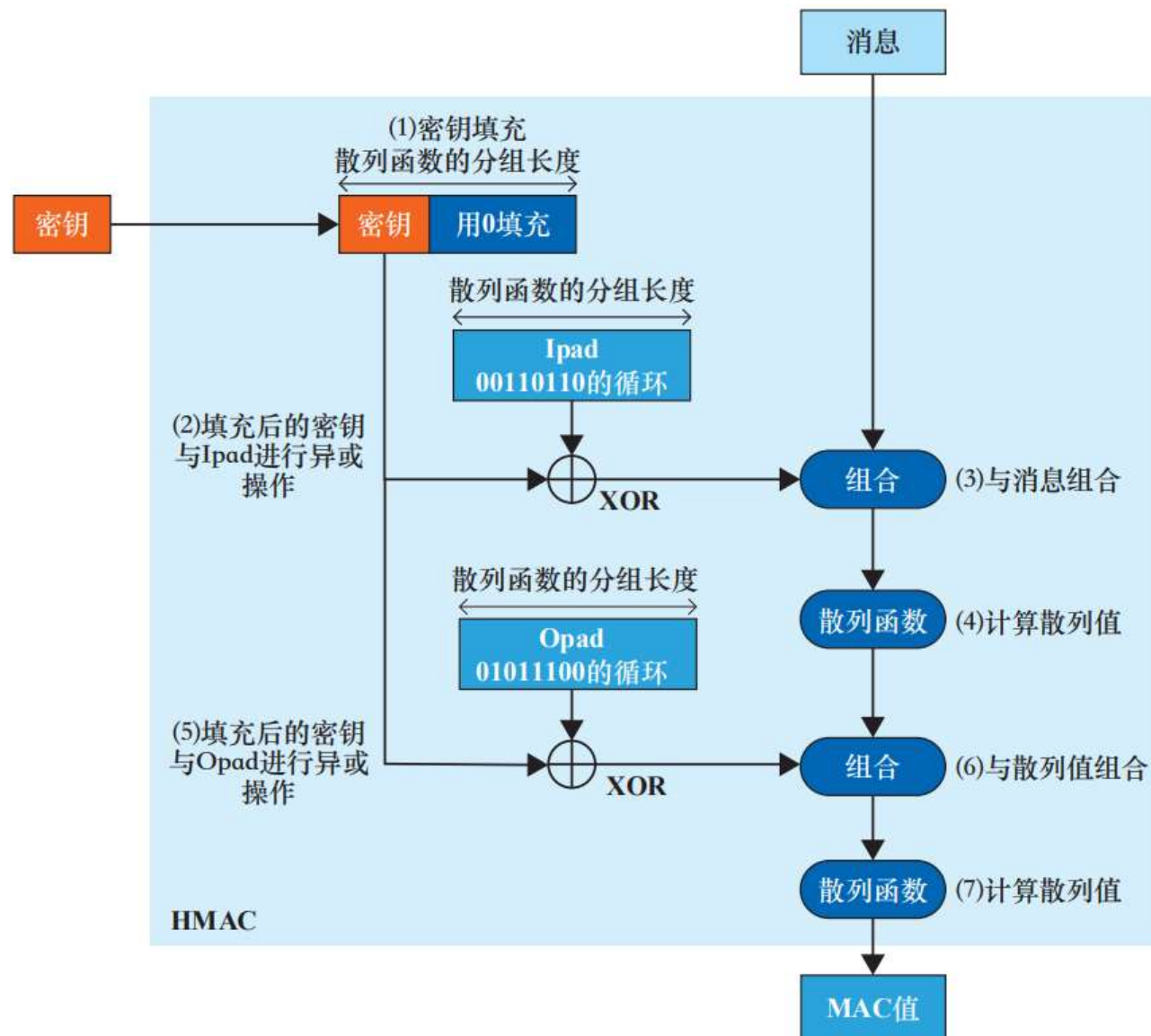


散列函数与消息认证码的比较



HMAC

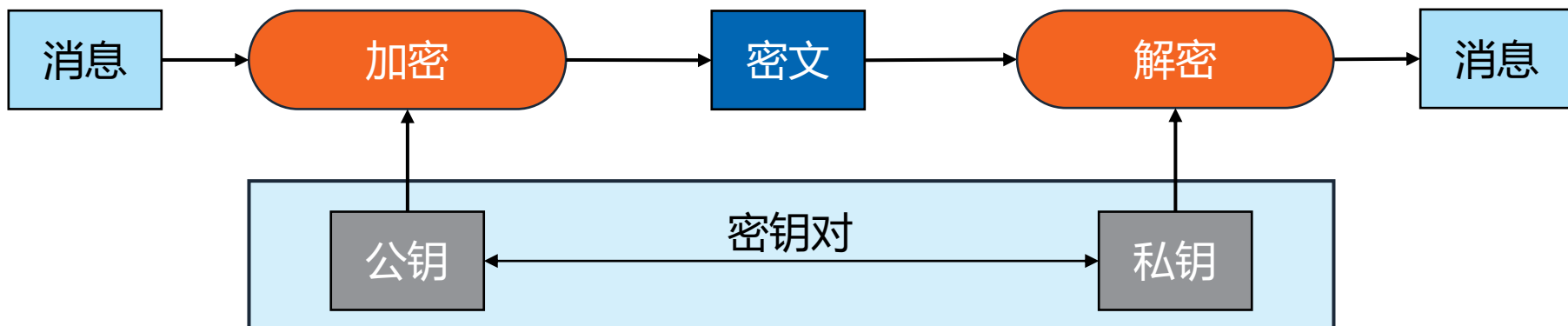
- HMAC是一种使用散列函数来构造消息认证码的方法
- 该方法所使用的散列函数不仅限于一种，任何高强度的散列函数都可以被用在HMAC
- 使用SHA-1, SHA-256, 所构造的HMAC分别称为HMAC-SHA1, HMAC-SHA256



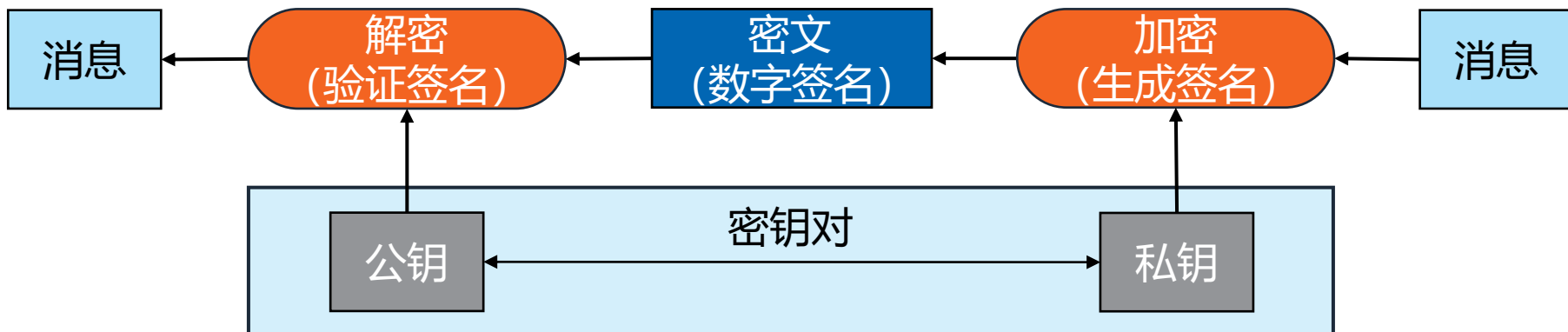


公钥密码与数字签名

(1) 公钥密码



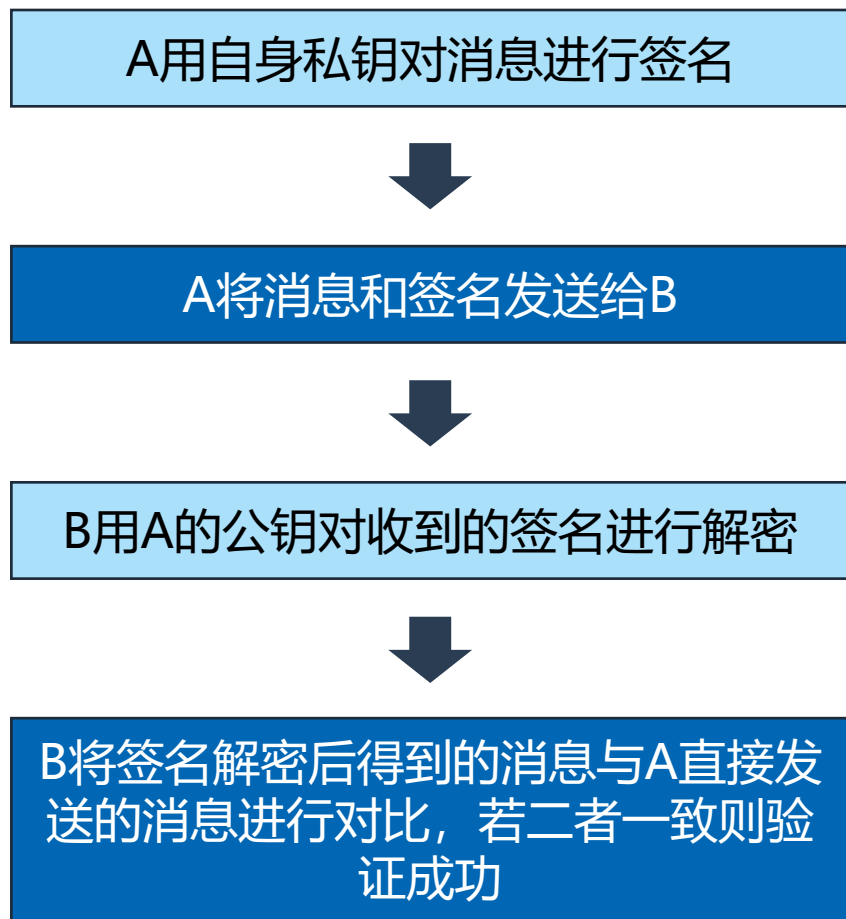
(2) 数字签名



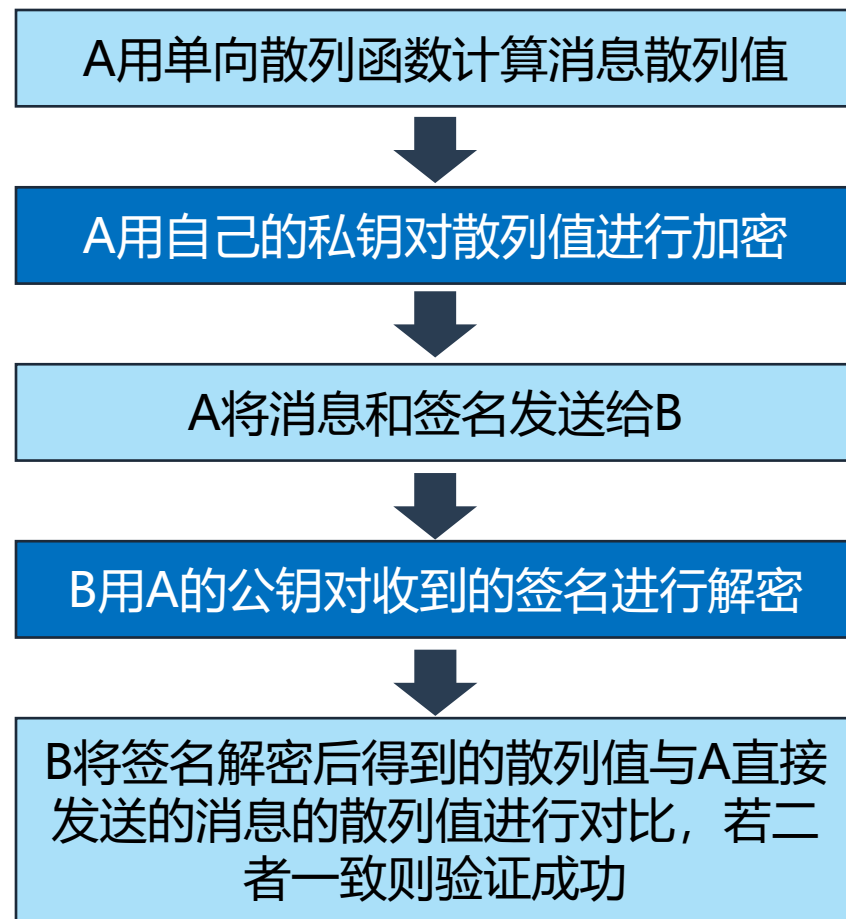


数字签名的方法

- 直接对消息签名的方法



- 对消息的散列值签名的方法





RSA数字签名方案

数字签名方案包括3个过程：系统的初始化、产生签名和验证签名

系统的初始化过程

选择两个保密的大素数 p 和 q ，计算 $n = p \times q$ ， $l = \text{lcm}(p-1, q-1)$ ，选择整数 e ，满足 $1 < e < l$ ，且 $\text{gcd}(e, l) = 1$ ；计算 $de \equiv 1 \pmod l$ ；以 $\{e, n\}$ 为公钥， $\{d, n\}$ 为私钥

签名产生过程

设消息为 M ，对其签名为 $S \equiv M^d \pmod n$ ，并将 (M, S) 发送给签名验证者

签名验证过程

接收方在收到消息 M 和签名 S 后，验证 $M \equiv S^e \pmod n$ 是否成立。若成立，则发送方的签名有效；若不成立，则签名无效



数字签名的分类

盲签名

签名者不知道代签名文件内容时使用的数字签名

门限签名

如果一个群体中有 n 个人，那么至少需要 p 个人签名才视为有效签名($n > p$)

群签名

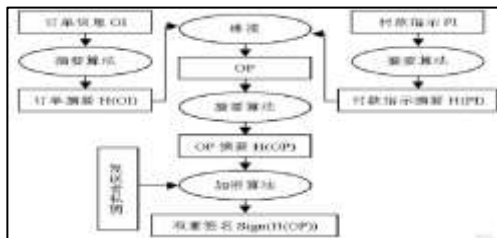
一个群体由多个成员组成，某个成员可以代表整个群体来进行数字签名，而且该成员作为签名者可以被验证

代理签名

密钥的所有者可以将签名权利授予第三方，获得权力的第三方可以进行数字签名

双重签名

签名者希望有个中间人在他与验证者之间进行验证授权操作





数字签名的应用

1. 网站认证

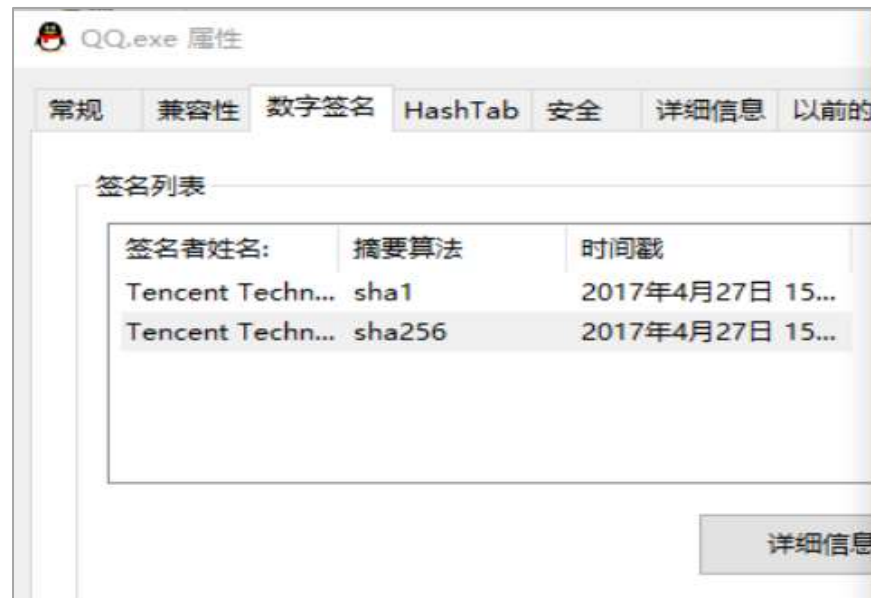
通过对网站域名信息、主体身份信息、域名权属信息等进行严格鉴证审核，并利用PKI数字签名技术形成不可篡改的认证标识，在互联网终端以安全可靠的方式进行展示，使网民更直观确认网站的真实身份

2. 比特币

比特币是一种完全匿名的数字货币，它的身份认证是基于ECDSA，比特币的账户地址就是对公钥计算摘要得到的，并用私钥确认账户拥有者

3. 代码签名

如果Windows上的可执行程序程序来源于正规公司，通常它会有代码签名，用于确保其来源可靠且未被篡改





第5节 密码分析技术

- ✓ 唯密文攻击
- ✓ 已知明文攻击
- ✓ 选择明文攻击
- ✓ 选择密文攻击
- ✓ 选择密钥攻击



密码分析技术

在未知密钥的前提下，从密文恢复出明文、或者推导出密钥，对密码进行分析的尝试

攻击方法分类(根据已知信息量的多少):

- 唯密文攻击(Ciphertext only)
- 已知明文攻击(Known Plaintext)
- 选择明文攻击(Chosen Plaintext)
- 选择密文攻击(Chosen Ciphertext)
- 选择密钥攻击(Chosen Key)





密码分析技术

唯密文攻击(Ciphertext only)

- 分析者有一些消息的密文，都是用同一算法加密的
- 分析者的目标是恢复尽量多的明文或者推算出密钥

已知明文攻击(Known Plaintext)

- 密码分析者不仅可以得到一些消息的密文，而且也知道这些消息的明文
- 分析者的任务是得到加密的密钥或者得到一个算法，该算法可以解密用同样的密钥加密的消息

选择明文攻击(Chosen Plaintext)

- 分析者不仅知道一些消息的明文和密文，而且可以选择被加密的明文，这比已知明文攻击更加有效
- 分析者的任务是得到加密的密钥或者得到一个算法，该算法可以解密用同样的密钥加密的消息

选择密文攻击(Chosen Ciphertext)

- 密码分析者能选择不同的被加密的密文，并可以得到对应的解密的明文，密码分析者的任务是推出密钥
- 主要针对公钥算法

相关密钥攻击(Related Key)

- 攻击者可以得到被两个不同的钥匙所加密（或解密）得到的密文（或明文）。攻击者不知道这两个钥匙的数值，但知道这两个钥匙之间的关系，比如两个钥匙之间相差一个比特



密码分析技术

- Success in dealing with unknown ciphers is measured by these four things in the order named, **perseverance, careful methods of analysis, intuition, luck**. The ability at least to read the language of the original text is very desirable but not essential.

——Colonel Parker Hitt “The Manual for the Solution of Military Ciphers”

- **毅力、审慎的分析方法、直觉、运气**





密码分析技术

- 密码算法的相对安全性
 - 破解算法的代价大于加密数据本身的价值
 - 破解算法的时间超过了信息的生命期

Average Time Required for Exhaustive Key Search

Key Size (bits)	Number of Alternative Keys	Time required at 10^6 Decryption/ μ s
32	$2^{32} = 4.3 \times 10^9$	4.3 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	20 hours
128	$2^{128} = 3.4 \times 10^{38}$	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	5.9×10^{30} years



第6节 总结和展望



数据生命周期与密码技术

你认为 密码技术在数据生命周期中可以发挥哪些作用 吗？

根据数据分级分类的要求，采用基于密码技术的数据标签，确保数据生命周期中处理过程可溯可查

采用密码技术解决存储安全性问题；利用公钥密码技术或哈希函数验证远程数据的完整性

采用非对称公私钥对构建可信身份；利用数字签名技术进行用户身份认证，实现数据确权

数据采集

数据传输

数据存储

数据处理

数据交换

数据销毁

数字签名、数据摘要等技术可防止数据在传输过程中经由骨干网络节点转发时被未经授权的查看、篡改或破坏

采用全同态加密等技术解决利用云计算过程中导致的数据所有权和管理权分离所带来的数据安全问题

对云存储中的数据进行多次加密，当需要销毁数据时，可直接删除密钥，提高数据销毁效率



总结

围绕数据加密的主题，介绍了保障网络安全的核心技术——密码学，纵观密码学的发展历程，阐述了密码学的理论基础，总结了密码学的应用现状



第一节 密码学简史

- 古典密码
- 近代密码
- 现代密码

回顾历史，了解
密码学演进方向



第二节 对称密码

- 分组密码
- DES算法
- 流密码

总结对称密码的加
密解密原理



第三节 公钥密码

- 秘钥分发问题
- 公钥密码
- RSA算法
- 实际应用

剖析公钥密码的来
龙去脉



第四节 摘要与签名

- 散列函数
- 消息认证码
- 数字签名

介绍校验数据完整
性的工具方法



第五节 密码分析技术

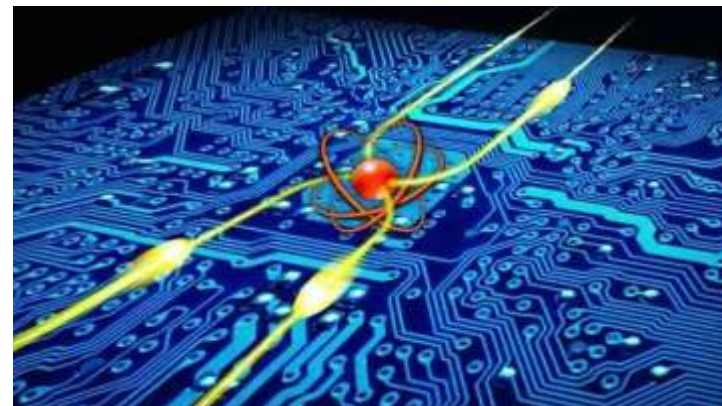
- 唯密文攻击
- 已知明文攻击
- 选择明文攻击
- 选择密文攻击
- 选择密钥攻击

梳理密码分析领域
的技术体系



展望

密码技术发展为解决数据安全问题提出新的技术思路



- **基于编码的密码体制、基于格的密码体制、多变量密码体制**等后量子密码体制突飞猛进，为数据安全技术发展注入新动力
- **抵抗量子攻击的密码体制**受到关注



展望

突破受制于人的密码技术，构建内生安全体系，保障国家数据安全



- 开展**自主可控密码体制**开发设计，以及软硬件基础设施、网络加密协议、应用等由美国主导控制下的**极限对抗研究**
- 构建覆盖**人工智能、区块链、5G、量子通信、云计算**等新型信息技术的数据加**密体制**