第2章 密码学基础

罗文坚

主要内容

2.1 密码学基础知识

2.2 古典替换密码

2.3 对称密钥密码

2.4 公开密钥密码

2.5 消息认证

2.6 密码学新进展

概述

- 威胁信息完整性的行为主要包括：
  - 伪造：假冒他人的信息源向网络中发布消息；
  - 内容修改：对消息的内容进行插入、删除、变换和修改；
  - 顺序修改：对消息进行插入、删除或重组消息序列；
  - 时间修改：针对网络中的消息，实施延迟或重放；
  - 否认：接受者否认收到消息，发送者否认发送过消息。

- 消息认证是保证信息完整性的重要措施。其目的主要包括：
  - 证明：消息的信源和信宿的真实性；
  - 证明：消息内容是否曾受到偶然或有意的篡改，
  - 证明：消息的序号和时间性是否正确。

消息和信息的关系

- 信息，一般解释为，“事物运动状态或存在方式的不确定性的描述”。

- 消息，一般解释为，“用文字、符号、数据、语言、音符、图片、图像等能够被人们感觉感官所感知的形式，把客观物质运动和主观思维活动的状态表示出来的载体”。

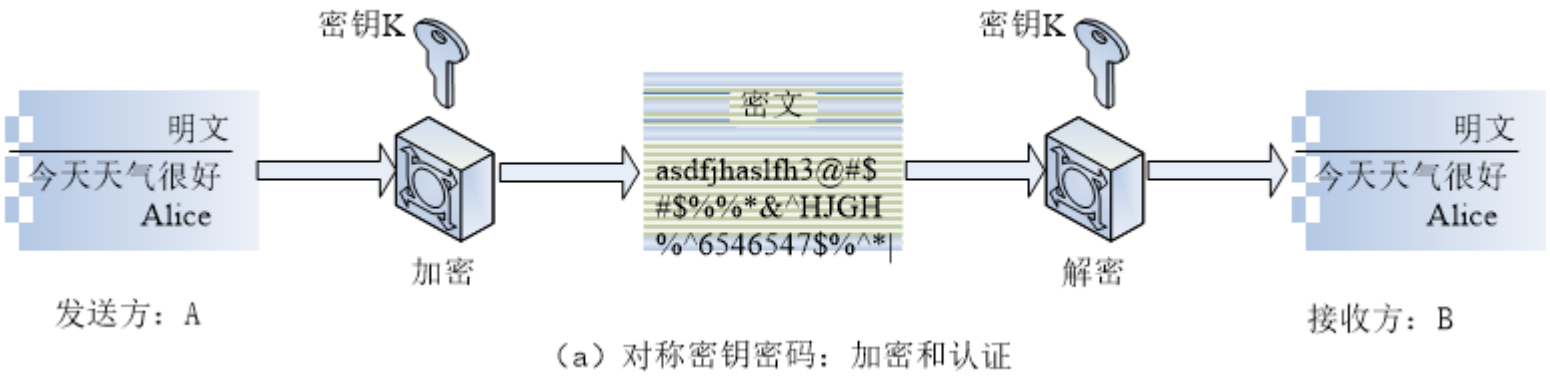
- 消息是信息的载体，信息通过消息来传递；消息是符号形式的，信息则是消息所反映的实质内容。

概述

- 认证技术在功能上可以分为两层。
  - 下层包含一个产生认证符的函数；认证符是一个用来认证消息的值；
  - 上层是以认证函数为原语；接收方可以通过认证函数来验证消息的真伪。

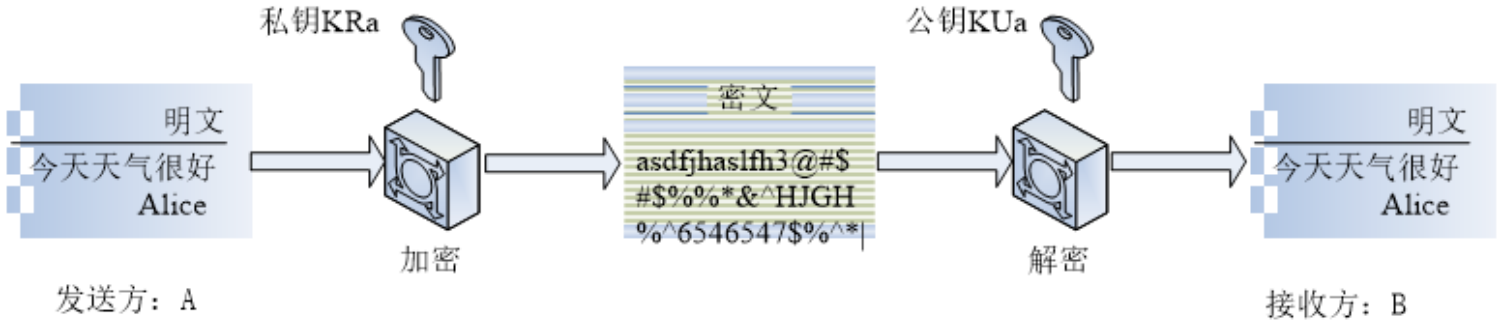
- 消息认证由具有认证功能的函数来实现的。
  - 消息加密，用消息的完整密文作为消息的认证符；
  - 消息认证码MAC（Message Authentication Code），也称密码校验和，使用密码对消息加密，生成固定长度的认证符；
  - 消息编码，是针对信源消息的编码函数，使用编码抵抗针对消息的攻击。

消息加密函数



（a）对称密钥密码：加密和认证

- 对称密钥密码对消息加密，不仅具有机密性，同时也具有一定的可认证性；



（b）公开密钥密码：认证

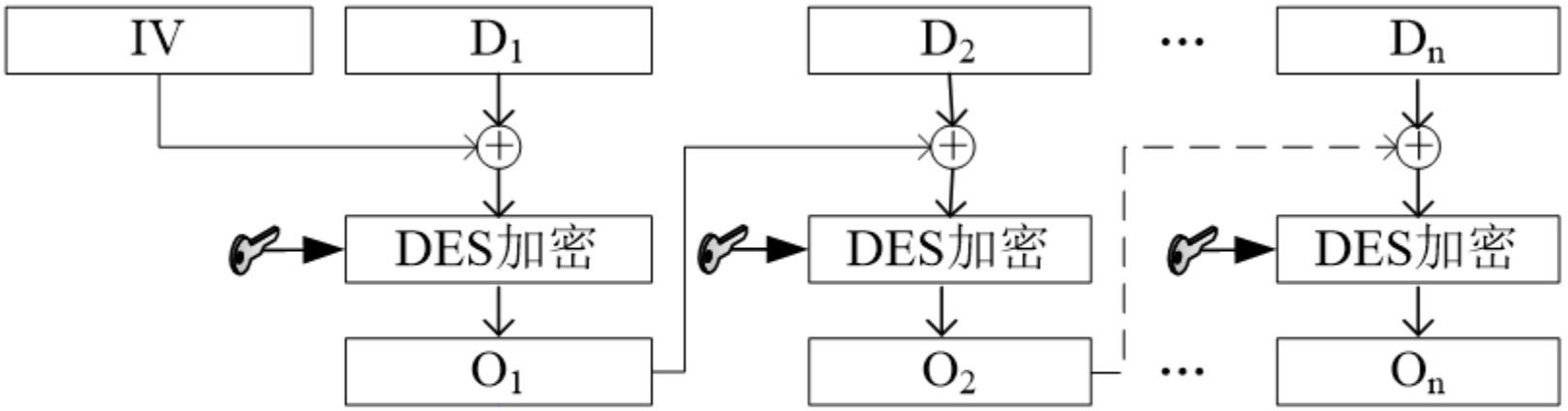
- 公开密钥密码本身就提供认证功能，其具有的私钥加密、公钥解密以及反之亦然的特性；

消息认证码

- 消息认证码MAC的基本思想：
  - 利用事先约定的密码，加密生成一个固定长度的短数据块MAC，并将MAC附加到消息之后，一起发送给接收者；
  - 接收者使用相同密码对消息原文进行加密得到新的MAC，比较新的MAC和随消息一同发来的MAC，如果相同则未受到篡改。
- 生成消息认证码的方法主要包括：
  - 基于加密函数的认证码：使用加密函数生成固定长度的认证符。
  - 消息摘要：将任意长度的消息全文作为单向散列函数的输入，进行散列计算，得到的被压缩到某一固定长度的散列值（即消息摘要）作为认证符。消息摘要的运算过程无需加密算法的参与，其关键是单向散列函数是否具有良好的无碰撞性。

基于DES的消息认证码

- 该算法采用DES加密和CBC模式。
  - 消息认证符可以是整个64位的O_n，也可以是O_n最左边的M位。



基于DES的消息认证码

消息编码

- 使用消息编码对信息进行认证，基本思想来源于信息通信中的差错校验码。
  - 差错校验码是差错控制中的检错方法。数据通信中的噪音可能会使得传输的比特值改变，用校验码可以检测出来。同样，一些人为造成的比特值的改变，使用差错控制也可以检测到。
- 消息编码认证的基本思想：
  - 引入冗余度，使通过信道传送的可能序列集M（编码集）大于消息集S（信源集）。
  - 发送方从M中选出用来代表消息的许用序列L_i，即对信息进行编码；接收方根据编码规则，进行解码，还原出发送方按此规则向他传来的消息。
  - 窜扰者不知道被选定的编码规则，因而所伪造的假码字多是M中的禁用序列，接收方将以很高的概率将其检测出来，并拒绝通过认证。

消息编码举例

信源S / 编码法则L	0	1	禁用序列
L_0	00	10	01，11
L_1	00	11	01，10
L_2	01	10	00，11
L_3	01	11	00，10

- 如果决定采用L_0，则以发送消息“00”代表信源“0”，发送消息“10”代表信源“1”。
  - 在子规则L_0下，消息“00”和“10”是合法的，而消息“01”和“11”在L_0之下不合法，收方将拒收这两个消息。

散列函数

- 散列函数（Hash Function）的目的
  - 将任意长的消息映射成一个固定长度的散列值（hash值），也称为消息摘要。
  - 消息摘要可以作为认证符，完成消息认证。

- 散列函数的健壮性
  - 弱无碰撞特性：散列函数h被称为是弱无碰撞的，是指在消息特定的明文空间X中，给定消息$x \in X$，在计算上几乎找不到不同于x的x'，$x' \in X$，使得h(x)=h(x')。
  - 强无碰撞特性：散列函数h被称为是强无碰撞的，是指在计算上难以找到与x相异的x'，满足h(x)=h(x')，x'可以不属于X。
  - 单向性：散列函数h被称为单向的，是指通过h的逆函数h^{-1}来求得散列值h(x)的消息原文x，在计算上不可行。

散列函数

- 伪造一：
  1. 攻击者得到一个有效签名(x, y)，其中x表示消息原文，y表示经过私钥签名消息摘要，$y=E_{kr}(Z)$，kr为私钥，Z为x的消息摘要，即Z=h(x)，h(x)为散列函数。
  2. 攻击者可以通过公钥计算得到Z=h(x)，还原出Z，然后企图找到一个x'，满足h(x')=h(x)。
  3. 如果他做到这一点，则(x', y)也可以通过认证，即为有效的伪造。

散列函数

- 伪造二：
  - 攻击者首先找到两个消息x和x'，满足h(x)=h(x')。然后，攻击者把x给Bob，且使他对x的摘要h(x)进行签名，从而得到y。
  - 那么，(x', y）也是一个有效的伪造。

- 伪造三：
  - 在某种签名方案中，可伪造一个随机消息摘要Z的签名y，$y=E_{kr}(Z)$。若散列函数h的逆函数h^{-1}是易求的，可算出$x=h^{-1}(Z)$，满足Z=h(x)，则(x, y)为合法签名。

散列值的安全长度

- “生日悖论”：如果一个房间里有23个或23个以上的人，那么至少有两个人的生日相同的概率要大于50%。对于60或者更多的人，这种概率要大于99%。
  - 不计特殊的闰年，计算房间里所有人的生日都不相同的概率。

① 第一个人不发生生日冲突的概率是 $\frac{365}{365}$ ，

② 第二个人不发生生日冲突的概率是 $1-\frac{1}{365}$ ，

③ …，第n个人是 $1-\frac{n-1}{365}$ ，

④ 所有人生日都不冲突的概率是：

① $E=1\times(1-\frac{1}{365})\times\ldots\times(1-\frac{n-2}{365})\times(1-\frac{n-1}{365})$，

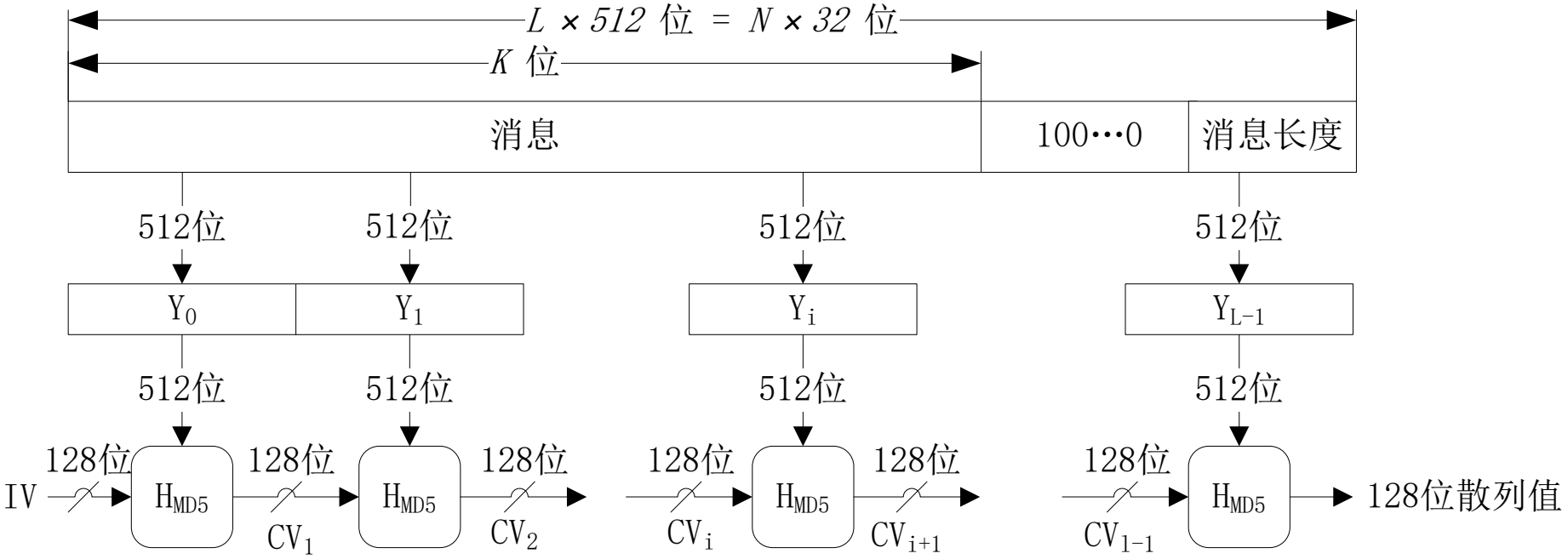
⑤ 而发生冲突的概率P=1-E；当n=23时，P≈0.507；n=100时，P≈0.9999996。

散列值的安全长度

- 生日悖论对于散列函数的意义
  - n位长度的散列值，可能发生一次碰撞的测试次数不是2^n次，而是大约$2^{n/2}$次。
  - 一个40位的散列值将是不安全的，因为大约100万个随机散列值中将找到一个碰撞的概率为50%。
  - 消息摘要的长度不低于为128位。

MD5

- 20世纪90年代初，RSA Data Security Inc.先后方面了MD2、MD3、MD4。1991年，Rivest对MD4的进行改进升级，提出了MD5（Message Digest Algorithm 5），具有更高的安全性。
  - 以512位分组来处理，每组又包含16个32位字。



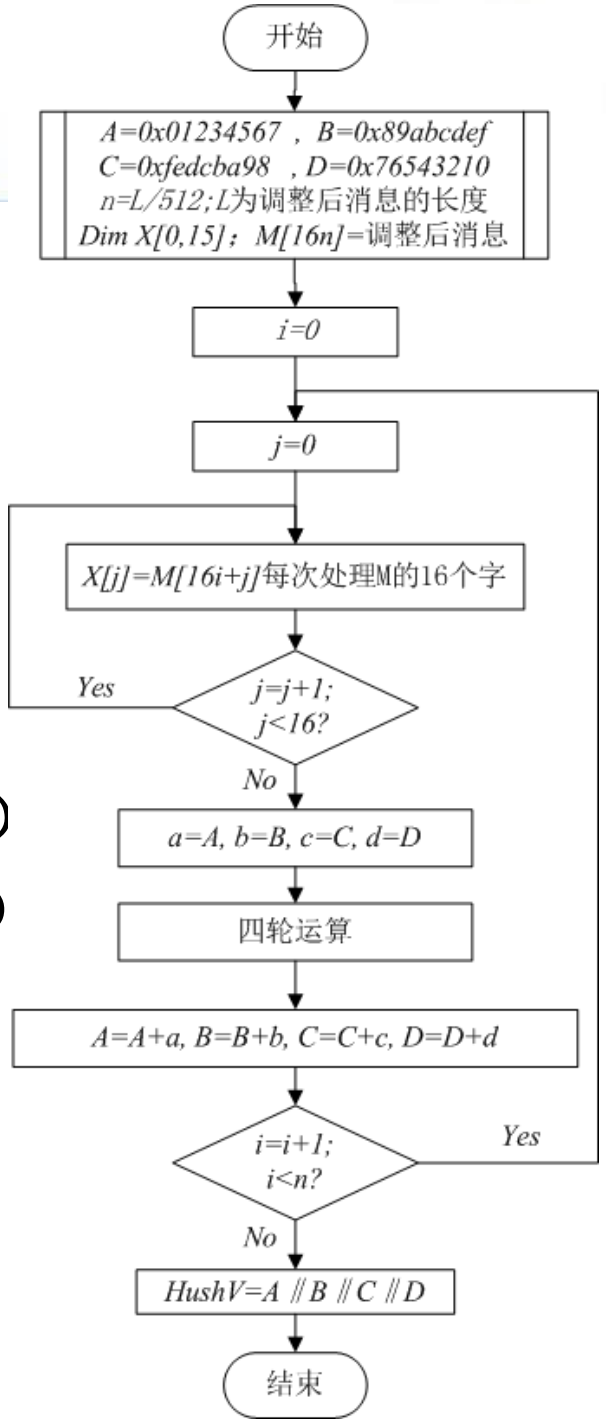
MD5算法

MD5

- A、B、C、D称为链接变量。

- 主循环次数为512位消息分组的数目。
- 循环体内包含四轮运算。

- 四轮运算涉及四个函数：
  - E（X，Y，Z）=（X ∧ Y）∨（（¬X）∧ Z）
  - F（X，Y，Z）=（X ∧ Z）∨（Y ∧（¬Z））
  - G（X，Y，Z）= X ⊕ Y ⊕ Z
  - H（X，Y，Z）= Y ⊕（X ∨（¬Z））

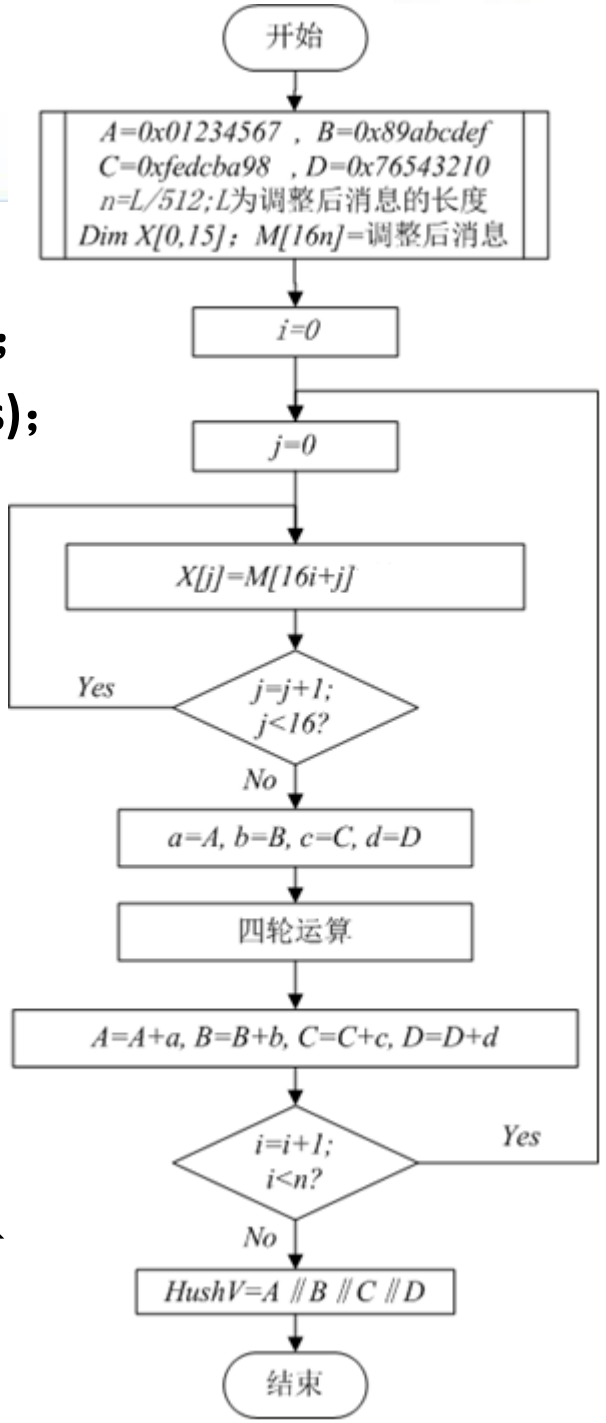


MD5

- 四轮运算：
  - EE(a,b,c,d,M_j,s,t_i)： a = b + ((a+(E(b,c,d)+M_j+t_i)<<s);
  - FF(a,b,c,d,M_j,s,t_i)： a = b + ((a+(F(b,c,d)+ M_j + t_i)<<s);
  - GG(a,b,c,d,M_j,s,t_i) ： a = b + ((a+(G(b,c,d)+ M_j + t_i)<<s);
  - HH(a,b,c,d, M_j,s,t_i) ： a = b + ((a+(H(b,c,d)+ M_j + t_i)<<s);

- M_j对应流程图中的X[j]。
- 常数t_i的计算方法是：
  - 整个四轮操作总共分为64步，在第i步中t_i是2^{32}×abs(sin(i))的整数部分，i的单位是弧度。

- 四轮结束后，用新的A、B、C、D，在下一轮分组上继续运行算法。



第一轮

- **EE(a,b,c,d,M_j,s,t_i)： a = b + ((a+(E(b,c,d)+M_j+t_i)<<s)；**

(1) EE(a,b,c,d,M_0,7,0xd76aa478) —》 a = b + ((a+(E(b,c,d)+M_0+0xd76aa478)<<7)；
(2) EE(d,a,b,c,M_1,12,0xe8c7b756)
(3) EE(c,d,a,b,M_2,17,0x242070db)
(4) EE(b,c,d,a,M_3,22,0xc1bdceee)
(5) EE(a,b,c,d,M_4,7,0xf57c0faf)
(6) EE(d,a,b,c,M_5,12,0x4787c62a)
(7) EE(c,d,a,b,M_6,17,0xa8304613)
(8) EE(b,c,d,a,M_7,22,0xfd469501)
(9) EE(a,b,c,d,M_8,7,0x698098d8)
(10)EE(d,a,b,c,M_9,12,0x8b44f7af)
(11)EE(c,d,a,b,M_{10},17,0xffff5bb1)
(12)EE(b,c,d,a,M_{11},22,0x895cd7be)
(13)EE(a,b,c,d,M_{12},7,0x6b901122)
(14)EE(d,a,b,c,M_{13},12,0xfd987193)
(15)EE(c,d,a,b,M_{14},17,0xa679438e)
(16)EE(b,c,d,a,M_{15},22,0x49b40821)

第二轮

- **FF(a,b,c,d,M_j,s,t_i)： a = b + ((a+(F(b,c,d)+ M_j + t_i)<<s)**

(1) FF(a,b,c,d,M_1,5,0xf61e2562) —》 a = b + ((a+(F(b,c,d)+M_1+0xf61e2562)<<5)；
(2) FF(d,a,b,c,M_6,9,0xc040b340)
(3) FF(c,d,a,b,M_{11},14,0x265e5a51)
(4) FF(b,c,d,a,M_0,20,0xe9b6c7aa)
(5) FF(a,b,c,d,M_5,5,0xd62f105d)
(6) FF(d,a,b,c,M_{10},9,0x02441453)
(7) FF(c,d,a,b,M_{15},14,0xd8a1e681)
(8) FF(b,c,d,a,M_4,20,0xe7d3fbc8)
(9) FF(a,b,c,d,M_9,5,0x21e1cde6)
(10)FF(d,a,b,c,M_{14},9,0xc33707d6)
(11)FF(c,d,a,b,M_3,14,0xf4d50d87)
(12)FF(b,c,d,a,M_8,20,0x455a14ed)
(13)FF(a,b,c,d,M_{13},5,0xa9e3e905)
(14)FF(d,a,b,c,M_2,9,0xfcefa3f8)
(15)FF(c,d,a,b,M_7,14,0x676f02d9)
(16)FF(b,c,d,a,M_{12},20,0x8d2a4c8a))

第三轮

- **GG(a,b,c,d,M_j,s,t_i)： a = b + ((a+(G(b,c,d)+ M_j + t_i)<<s)；**

(1) GG(a,b,c,d,M_5,4,0xEEfa3942) —》 a = b + ((a+(G(b,c,d)+M_5+0xEEfa3942)<<4)；
(2) GG(d,a,b,c,M_8,11,0x8771f681)
(3) GG(c,d,a,b,M_{11},16,0x6d9d6122)
(4) GG(b,c,d,a,M_{14},23,0xfde5380c)
(5) GG(a,b,c,d,M_1,4,0xa4beea44)
(6) GG(d,a,b,c,M_4,11,0x4bdecfa9)
(7) GG(c,d,a,b,M_7,16,0xf6bb4b60)
(8) GG(b,c,d,a,M_{10},23,0xbebfbc70)
(9) GG(a,b,c,d,M_{13},4,0x289b7ec6)
(10)GG(d,a,b,c,M_0,11,0xeaa127fa)
(11)GG(c,d,a,b,M_3,16,0xd4ef3085)
(12)GG(b,c,d,a,M_6,23,0x04881d05)
(13)GG(a,b,c,d,M_9,4,0xd9d4d039)
(14)GG(d,a,b,c,M_{12},11,0xe6db99e5)
(15)GG(c,d,a,b,M_{15},16,0x1fa27cf8)
(16)GG(b,c,d,a,M_2,23,0xc4ac5665)

第四轮

- **HH(a,b,c,d, M_j,s,t_i) ： a = b + ((a+(H(b,c,d)+ M_j + t_i)<<s)；**

(1) HH(a,b,c,d,M_0,6,0xf4292244) —》 a = b + ((a+(H(b,c,d)+M_0+0xf4292244)<<6)；
(2) HH(d,a,b,c,M_7,10,0x432aEE97)
(3) HH(c,d,a,b,M_{14},15,0xab9423a7)
(4) HH(b,c,d,a,M_5,21,0xfc93a039)
(5) HH(a,b,c,d,M_{12},6,0x655b59c3)
(6) HH(d,a,b,c,M_3,10,0x8f0ccc92)
(7) HH(c,d,a,b,M_{10},15,0xEEeEE47d)
(8) HH(b,c,d,a,M_1,21,0x85845dd1)
(9) HH(a,b,c,d,M_8,6,0x6fa87e4f)
(10)HH(d,a,b,c,M_{15},10,0xfe2ce6e0)
(11)HH(c,d,a,b,M_6,15,0xa3014314)
(12)HH(b,c,d,a,M_{13},21,0x4e0811a1)
(13)HH(a,b,c,d,M_4,6,0xf7537e82)
(14)HH(d,a,b,c,M_{11},10,0xbd3af235)
(15)HH(c,d,a,b,M_2,15,0x2ad7d2bb)
(16)HH(b,c,d,a,M_9,21,0xeb86d391)

数字签名

- 数字签名：Digital Signature。
- 在ISO7498-2标准定义为：
  - “附加在数据单元上的一些数据或是对数据单元所作的密码变换，这种数据或变换可以被数据单元的接收者用来确认数据单元来源和数据单元的完整性，并保护数据不会被人（例如接收者）伪造”。
- 美国电子签名标准对数字签名作了如下解释：
  - “数字签名是利用一套规则和一个参数对数据进行计算所得的结果，用此结果能够确认签名者的身份和数据的完整性”。

- 一般来说，数字签名可以被理解为：
  - 通过某种密码运算生成一系列符号及代码，构成可以用来进行数据来源验证的数字信息。

数字签名

- 从签名形式上分，数字签名有两种。
  - 一种是对整个消息的签名。
  - 一种是对压缩消息的签名。
  - 它们都是附加在被签名消息之后或在某一特定位置上的一段数据信息。

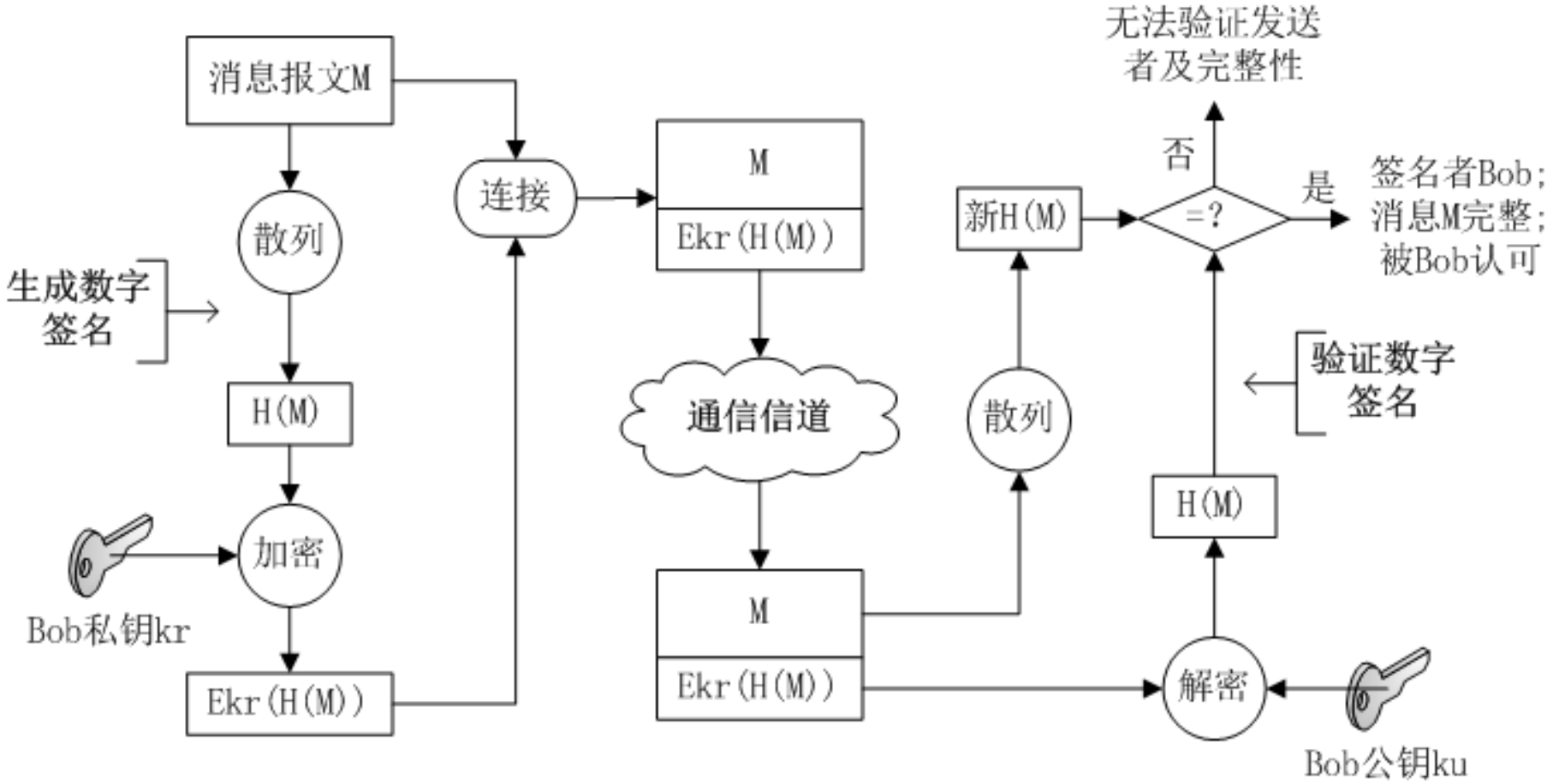
- 数字签名主要目的：
  - 保证接收方能够确认或验证发送方的签名，但不能伪造；发送方发出签名消息后，不能否认所签发的消息。

数字签名

- 设计数字签名必须满足下列条件：
  1. 签名必须基于一个待签名信息的位串模板；
  2. 签名必须使用某些对发送方来说是唯一的信息，以防止双方的伪造与否认；
  3. 必须相对容易生成、识别和验证数字签名；
  4. 伪造该数字签名在计算复杂性意义上具有不可行性。
     - 既包括对一个已有的数字签名构造新的消息，也包括对一个给定消息伪造一个数字签名。

- 数字签名主要采用公钥加密技术来实现。
  - 通常情况下，一次数字签名涉及三个信息，分别是一个哈希函数、发送者的公钥、发送者是私钥。

数字签名的生成及验证


消息报文M
散列
生成数字签名
H(M)
加密
Bob私钥kr
Ekr(H(M))
连接
M
Ekr(H(M))
通信信道
M
Ekr(H(M))
散列
新H(M)
无法验证发送者及完整性
否
=?
是
签名者Bob；消息M完整；被Bob认可
验证数字签名
H(M)
解密
Bob公钥ku

数字签名的一般应用过程

- 发送方：
  - 首先，使用散列函数对消息报文进行散列计算，生成散列值（消息报文摘要），并用自己的私钥对这个散列值进行加密，加密的散列值即为数字签名。
  - 然后，这个加密的散列值将作为消息报文的附件和消息报文一起发送给接收方。

- 接收方：
  - 首先，用与发送方一样的散列函数计算原始消息报文的散列值，接着再用发送方的公钥来对报文附加的数字签名进行解密，得到发送方计算的散列值。
  - 然后，比较两个散列值。如果相同，接收方就可确认消息报文的发送方，并且消息报文是完整的。

数字签名方法

- 基于对称密钥密码体制，也可以依靠其密钥的双方保密的特点来实现数字签名，但使用范围受到局限。
- 目前，数字签名多数还是基于公钥密码体制，常见的数字签名算法有RSA、ElGamal、DSA以及椭圆曲线数字签名算法等。
- 另外，还有一些特殊数字签名方法，如盲签名、代理签名、群签名、门限签名、具有消息恢复功能的签名等，它们与具体应用环境密切相关。

- 美国国家标准技术研究所（NIST）1994年公布了数字签名标准（DSS，Digital Signature Standard），采用的算法是DSA。
  - DSA是Schnorr和ElGamal签名算法变种，是基于有限域上的离散对数问题设计的。DSA算法不是标准的公钥密码，只能提供数字签名，但安全性和灵活性好，被广泛应用于金融等领域。

主要内容

2.1 密码学基础知识

2.2 古典替换密码

2.3 对称密钥密码

2.4 公开密钥密码

2.5 消息认证

2.6 密码学新进展

混沌密码学

- 1989年，英国数学家Matthews，基于混沌的加密技术。
  - 混沌系统具有良好的伪随机特性、轨道的不可预测性、对初始状态及控制参数的敏感性等一系列特性；
  - 传统的密码算法敏感性依赖于密钥，而混沌映射依赖于初始条件和映射中的参数；
  - 传统的加密算法通过加密轮次来达到扰乱和扩散，混沌映射则通过迭代，将初始域扩散到整个相空间；
  - 传统加密算法定义在有限集上，而混沌映射定义在实数域内。

量子密码

- 1970年，威斯纳提出利用单量子态制造不可伪造的“电子钞票”，这个构想由于量子态的寿命太短而无法实现，
- 1984年，IBM的贝内特和加拿大学者布拉萨德，提出了第一个量子密码方案，由此迎来了量子密码学的新时期。

- 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。

- 量子密码的安全性由量子力学原理所保证，被称为是绝对安全的。
  - 所谓绝对安全，是指即使在窃听者可能拥有极高的智商、可能采用最高明的窃听措施、可能使用最先进的测量手段，密钥的传送仍然是安全的，可见量子密码研究具有极其重大的意义。

DNA计算

- 1994年，Adleman等科学家进行了世界上首次DNA计算，解决了一个7节点有向汉密尔顿回路问题。

- 由于DNA计算具有的信息处理的高并行性、超高容量的存储密度和超低的能量消耗等特点，非常适合用于攻击密码计算系统的不同部分，对传统的基于计算安全的密码体制提出了挑战。

作业

1. 给定消息“000……000”（512位二进制数），通过编程计算其MD5值。请给出主要计算步骤和中间结果（16进制）。

2. 习题4（2）。有人说“所有的散列函数都存在产生碰撞的问题，很不安全”，你认为正确与否，为什么？