区块链与数字货币

浙江大学 杨小虎

2024年9月14日

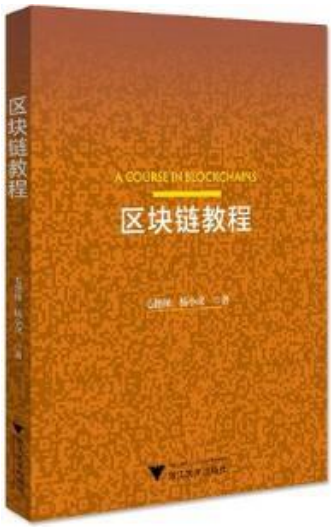
教学安排

- 数字货币和区块链技术原理
- 比特币区块链技术分析
- 以太坊技术原理与架构
- 智能合约与DApp开发
- HyperLedger Fabric技术原理与架构
- 数字货币和区块链发展趋势

课程考核

- 课程平时作业与课堂表现：40分
- 期末考试：60分

项目	占总成绩比例	备注
作业1	10%	区块链技术原理
作业2	20%	以太坊应用开发
作业3	10%	Fabric
期末考试	60%	开卷，可带一张A4纸



毛德操、杨小虎著，

《区块链教程》，

浙江大学出版社，2021.

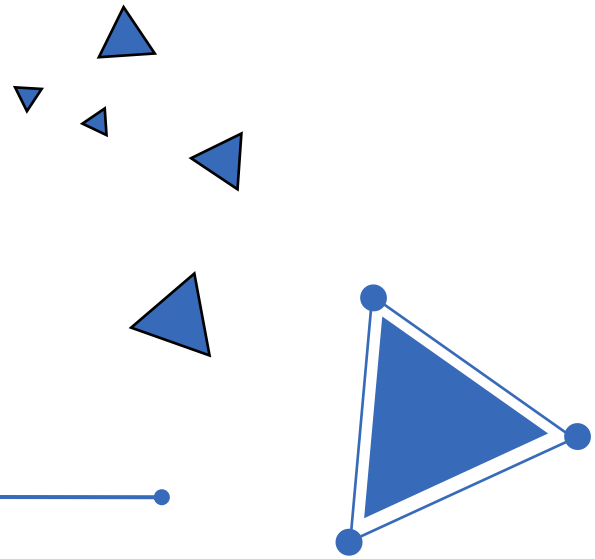
毛德操著，

《区块链技术》，

浙江大学出版社，2019.

参考资料：比特币、以太坊、HyperLedger等网上资料

01 基础知识



基础知识

1. 货币
2. Hash算法
3. 加密算法

货币知识

- 货币的价值
  - 使用价值
  - 交换价值
- 货币的形式
  - 贵金属：金、银、铜
  - 纸币
  - 数字货币

交子，宋代出现，世界上最早的纸币

货币知识

- 纸币的本质：信用
  - 金本位
  - 国家信用
  - 布雷顿森林体系
    - 美元与黄金锚定，每一美元的“含金量”为0.888671克黄金
    - 其他国家的法定货币与美元挂钩，规定与美元间的汇率
    - 实行可调整的固定汇率。各国货币对美元的汇率只可在法定汇率±1%的范围内波动。
    - 1944年启动，1971年结束

货币知识

- 纸币的全球化
  - 国际货币基金组织特别提款权
    - 特别提款权的价值最初确定为相当于0.888671克纯金，当时也相当于 1 美元。在布雷顿森林体系解体后，特别提款权价值被重新定义为一篮子货币。
    - 执董会每五年或在必要时提前检查特别提款权篮子，以确保特别提款权能反映各组成货币在世界贸易和金融体系中的相对重要性。
    - 特别提款权的价值：根据伦敦时间中午左右观察到的即期汇率，每日确定特别提款权的美元价值，并在基金组织网站上公布。

货币	2015年检查中确定的权重	货币单位的固定数量，为期五年，自2016年10月1日起
美元	41.73	0.58252
欧元	30.93	0.38671
人民币	8.33	1.0174
日元	8.09	11.900
英镑	10.92	0.085946

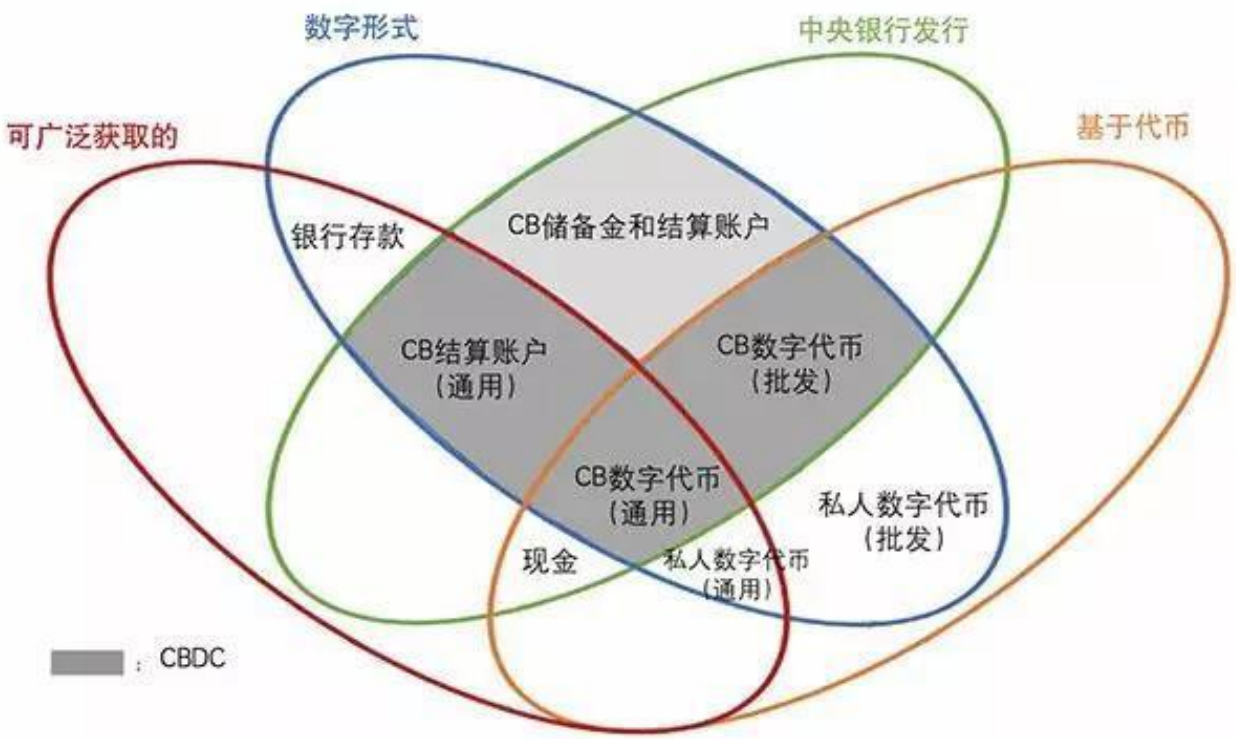
货币知识

- 纸币的全球化
  - 美元、欧元、英镑、日元……
  - 国际货币基金组织特别提款权
    - 特别提款权是国际货币基金组织为了补充国际储备资产的不足，创设的一种新的国际储备资产和记账单位，是IMF按照成员国缴存的基金份额分配给各成员国的，分配后就成为该成员国的资产，并与黄金和外汇一起构成该成员国的国际储备资产。
    - 当成员国发生国际收支逆差时，可将特别提款权划给另一个成员国，换取可兑换货币来密闭逆差，特别提款权还可用来偿还基金组织的贷款，但它不能兑换黄金，不能直接用于国际贸易或非贸易的支付，所以，特别提款权只是成员国在国际货币基金组织的特别提款权账户的一种账面资产。

货币理论

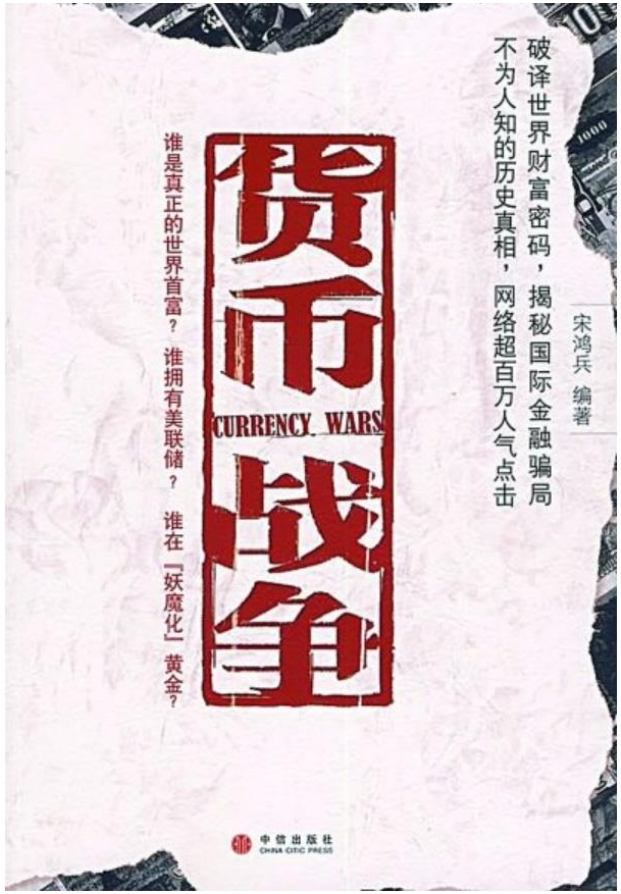
- 马克思政治经济学
  - 商品的使用价值与交换价值
- 凯恩斯经济学
  - 市场这只“看不见的手”有时候会失灵，需要政府干预
- 哈耶克经济学
  - 自由市场、自由经营、自由竞争、自动调节、自动均衡
  - 货币非国有化
- 现代货币理论MMT
  - 货币的本质就是欠条，即IOU（I Owe You）

货币之花



注：CB代表中央银行，CBDC代表中央银行数字货币。

货币战争？


破译世界财富密码，揭秘国际金融骗局
不为人知的历史真相，网络超百万人气点击
宋鸿兵 编著
货币战争
CURRENCY WARS
谁是真正的世界首富？谁拥有美联储？谁在「妖魔化」黄金？
中信出版社
CHINA CITIC PRESS

SWIFT

- SWIFT全称是环球银行间金融电讯协会（Society for Worldwide Interbank Financial Telecommunications），总部设在比利时首都布鲁塞尔，主要为金融机构提供跨境信息传送服务。
- **简单来说，它就是一个服务于金融业的全球通信网络。**

SWIFT

- 2022年2月26日，美国白宫发表声明说，为应对俄罗斯在乌克兰境内采取军事行动，美国与欧盟委员会、德国、法国、英国、意大利、加拿大领导人决定将部分俄罗斯银行排除在SWIFT系统之外。

- 据俄罗斯RT报道，金砖国家将建立类似 SWIFT的金融系统——名为BRICS Pay的数字支付平台，是由中、俄、印度、南非和巴西共同开发，旨在推动新兴市场国家在全球事务中发挥更大作用，减少发达国家的霸权影响。据悉，该系统可能在今年10月份推出，该系统允许在不使用美元的情况下进行单边结算。

Hash算法

- 把任意长度的输入通过hash算法变换成固定长度的输出，该输出就是hash值。
  - 输入域（空间）大于输出域（空间）。
  - 两个不同的输入可能会产生两个相同的输出，称为碰撞。
  - 从输出无法推导出输入。

最简单的hash算法

x mod y = z（整数取模运算）

x，y，z均为整数，其中y是质数，y不能为0

例：

y=7

9mod7=2；10mod7=3；50mod7=1，100mod7=2……

寻找一个足够大的质数y，可以实现长度为y的哈希表

Hash算法

- Hash实际上是一种思想，包含很多算法。
- 应用：
  - 快速定位（数据库中散列表）
  - 错误校验
  - 唯一性验证
- 常见算法：MD4、MD5、SHA1、SHA2

hash算法的安全性

- 逆向困难：难以从输出推导出输入。
  - MD、SHA算法都不可逆。
- 抗碰撞：很难找到两个不同的输入，可以产生相同的输出。
  - MD4、MD5已经被王小云教授攻破。
  - 王小云的研究成果是给定消息 M1，能够计算获取 M2，使得 M2 产生的hash值与 M1 产生的hash值相同。

安全哈希算法SHA256

- SHA-2，Secure Hash Algorithm 2的缩写，一种安全hash算法标准，由美国国家安全局National Security Agency研发。
- SHA256是SHA-2下细分出的一种算法，对于任意长度的输入数据，SHA256都会产生一个256bit长的哈希值，通常用一个长度为64的十六进制字符串来表示
  - 例如：A7FCFC6B5269BDCCE571798D618EA219A68B96CB87A0E21080C2E758D23E4CE9
- SHA256的安全性
  - 输出为256位，输出空间是2^{256}，是“亿亿亿亿亿亿亿亿”。
  - 目前尚未有破解方法。

安全哈希算法SHA256

```
Note: All variables are unsigned 32 bits and wrap modulo 232 when calculating


Initialize variables
(first 32 bits of the fractional parts of the square roots of the first 8 primes 2..19):
h0 := 0x6a09e667
h1 := 0xbb67ae85
h2 := 0x3c6ef372
h3 := 0xa54ff53a
h4 := 0x510e527f
h5 := 0x9b05688c
h6 := 0x1f83d9ab
h7 := 0x5be0cd19


Initialize table of round constants
(first 32 bits of the fractional parts of the cube roots of the first 64 primes 2..311):
k[0..63] :=
   0x428a2f98, 0x71374491, 0xb5c0fbcf, 0xe9b5dba5, 0x3956c25b, 0x59f111f1, 0x923f82a4, 0xab1c5ed5,
   0xd807aa98, 0x12835b01, 0x243185be, 0x550c7dc3, 0x72be5d74, 0x80deb1fe, 0x9bdc06a7, 0xc19bf174,
   0xe49b69c1, 0xefbe4786, 0x0fc19dc6, 0x240ca1cc, 0x2de92c6f, 0x4a7484aa, 0x5cb0a9dc, 0x76f988da,
   0x983e5152, 0xa831c66d, 0xb00327c8, 0xbf597fc7, 0xc6e00bf3, 0xd5a79147, 0x06ca6351, 0x14292967,
   0x27b70a85, 0x2e1b2138, 0x4d2c6dfc, 0x53380d13, 0x650a7354, 0x766a0abb, 0x81c2c92e, 0x92722c85,
   0xa2bfe8a1, 0xa81a664b, 0xc24b8b70, 0xc76c51a3, 0xd192e819, 0xd6990624, 0xf40e3585, 0x106aa070,
   0x19a4c116, 0x1e376c08, 0x2748774c, 0x34b0bcb5, 0x391c0cb3, 0x4ed8aa4a, 0x5b9cca4f, 0x682e6ff3,
   0x748f82ee, 0x78a5636f, 0x84c87814, 0x8cc70208, 0x90befffa, 0xa4506ceb, 0xbef9a3f7, 0xc67178f2


Pre-processing:
append the bit '1' to the message
append k bits '0', where k is the minimum number >= 0 such that the resulting message
    length (in bits) is congruent to 448(mod 512)
append length of message (before pre-processing), in bits, as 64-bit big-endian integer


Process the message in successive 512-bit chunks:
break message into 512-bit chunks
for each chunk
    break chunk into sixteen 32-bit big-endian words w[0..15]
```

```
    Extend the sixteen 32-bit words into sixty-four 32-bit words:
    for i from 16 to 63
        s0 := (w[i-15] rightrotate 7) xor (w[i-15] rightrotate 18) xor(w[i-15] rightshift 3)
        s1 := (w[i-2] rightrotate 17) xor (w[i-2] rightrotate 19) xor(w[i-2] rightshift 10)
        w[i] := w[i-16] + s0 + w[i-7] + s1

    Initialize hash value for this chunk:
    a := h0
    b := h1
    c := h2
    d := h3
    e := h4
    f := h5
    g := h6
    h := h7

    Main loop:
    for i from 0 to 63
        s0 := (a rightrotate 2) xor (a rightrotate 13) xor(a rightrotate 22)
        maj := (a and b) xor (a and c) xor(b and c)
        t2 := s0 + maj
        s1 := (e rightrotate 6) xor (e rightrotate 11) xor(e rightrotate 25)
        ch := (e and f) xor ((not e) and g)
        t1 := h + s1 + ch + k[i] + w[i]
        h := g
        g := f
        f := e
        e := d + t1
        d := c
        c := b
        b := a
        a := t1 + t2

    Add this chunk's hash to result so far:
    h0 := h0 + a
    h1 := h1 + b
    h2 := h2 + c
    h3 := h3 + d
    h4 := h4 + e
    h5 := h5 + f
    h6 := h6 + g
    h7 := h7 + h

Produce the final hash value (big-endian):
digest = hash = h0 append h1 append h2 append h3 append h4 append h5 append h6 append h7
```

https://blog.csdn.net/u011583927/article/details/80905740

作业1

- 给定字符串“Blockchain@ZhejiangUniversity”，后面添加nonce，提交一个SHA256的代码实现，实验说明SHA256找到一个前30、31、32位（二进制）为0的数时的nonce值和计算时间。
- 提交格式：word文件
- 文件名：姓名+学号+作业1.docx
- 提交邮箱：22221263@zju.edu.cn
- 提交截止日期：2024年9月23日8点

加密算法

最简单的加密算法：凯撒密码

- 明文字母表：ABCDEFGHIJKLMNOPQRSTUVWXYZ
- 密文字母表：DEFGHIJKLMNOPQRSTUVWXYZABC

- 明文：THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
- 密文：WKH TXLFN EURZQ IRA MXPSV RYHU WKH ODCB GRJ

加密算法

M是明文， C是密文

E是加密函数，D是解密函数，k是密钥

$$E_k(M)=C$$

$$D_k(C)=M$$

$$D_k(E_k(M))=M$$

加密算法

M是明文， C是密文
E是加密函数，D是解密函数，k1, k2是密钥

$$E_{k1}(M)=C$$

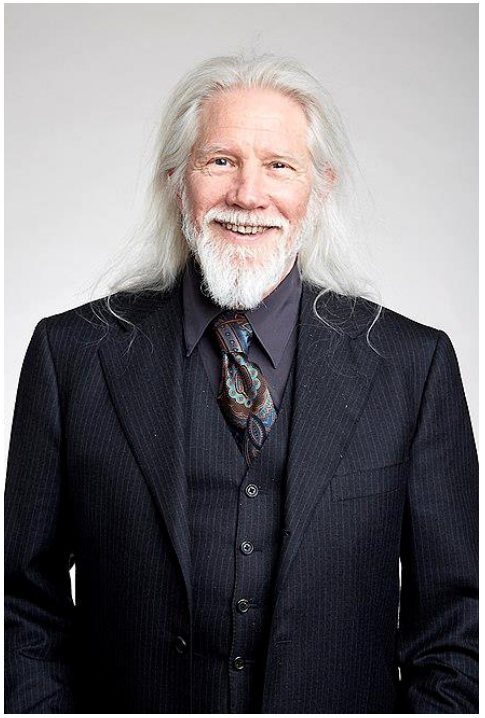
$$D_{k2}(C)=M$$

$$D_{k2}(E_{k1}(M))=M$$

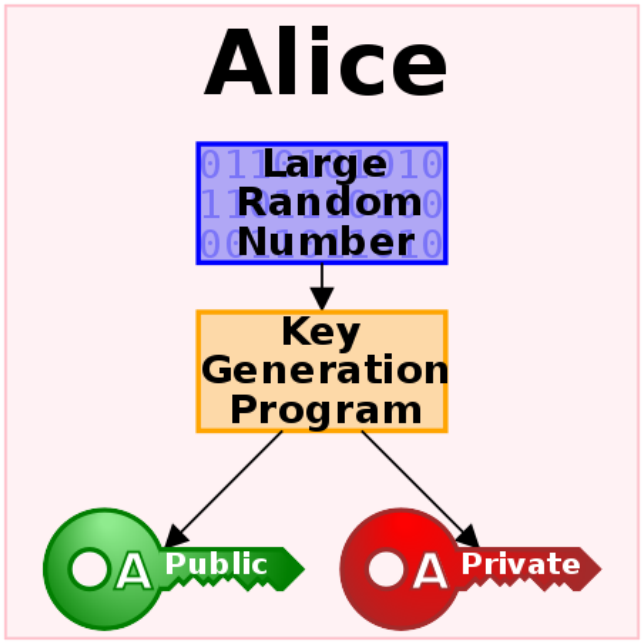
K1 = k2,　　对称加密算法
K1 <> k2　　非对称加密算法

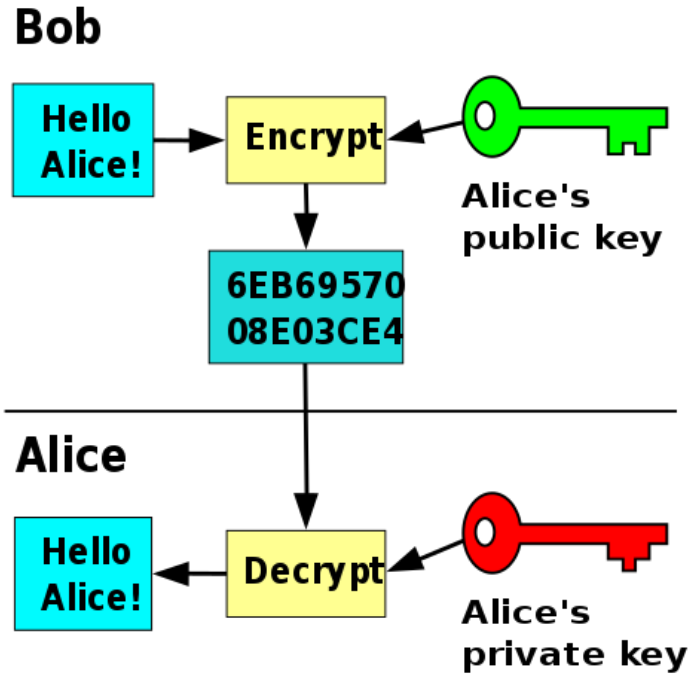
非对称加密算法

- 1970年，英国学者James H. Ellis提出设想
- 1976年，美国学者Whitfield Diffie and Martin Hellman首次提出一种可实现的非对称加密算法——exponentiation in a finite field

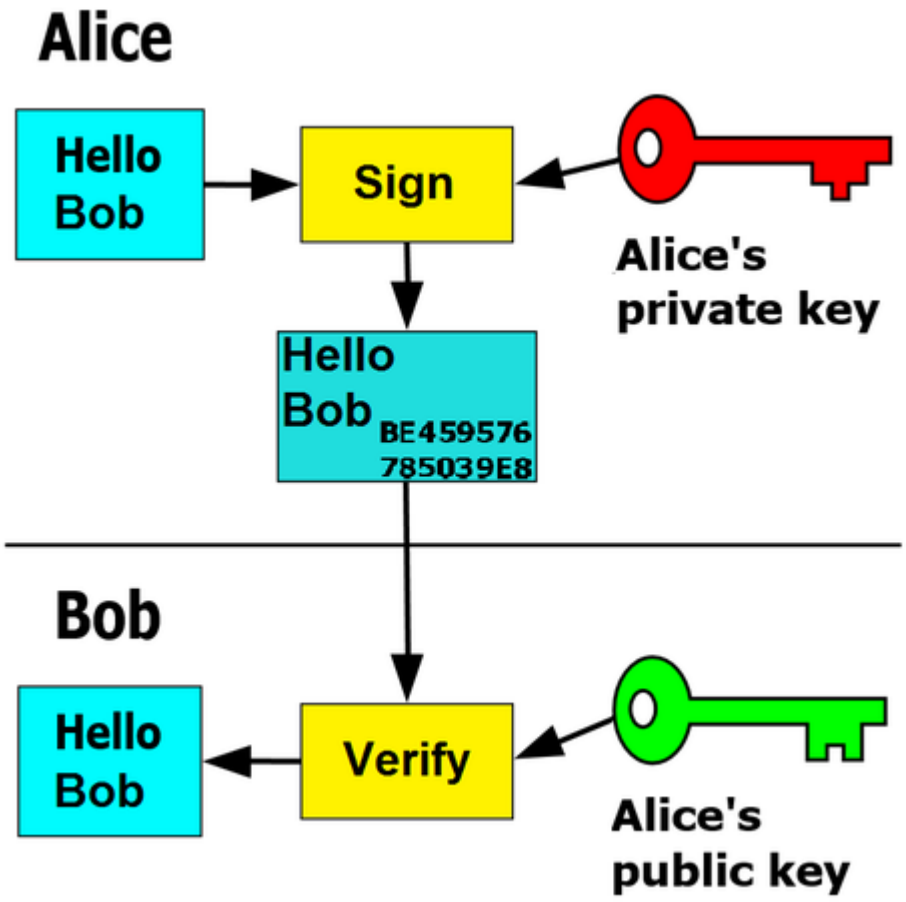


非对称加密算法


Alice
Large Random Number
Key Generation Program
Public
Private


Bob
Hello Alice!
Encrypt
Alice's public key
6EB69570 08E03CE4
Alice
Hello Alice!
Decrypt
Alice's private key

数字签名


Alice
Hello
Bob
Sign
Alice's
private key
Hello
Bob
BE459576
785039E8
Bob
Hello
Bob
Verify
Alice's
public key

非对称加密算法

- 基于大素数分解难题：
  - RSA
- 基于离散对数难解问题：
  - 椭圆曲线加密
  - Diffie-Hellman 加密

谢谢！