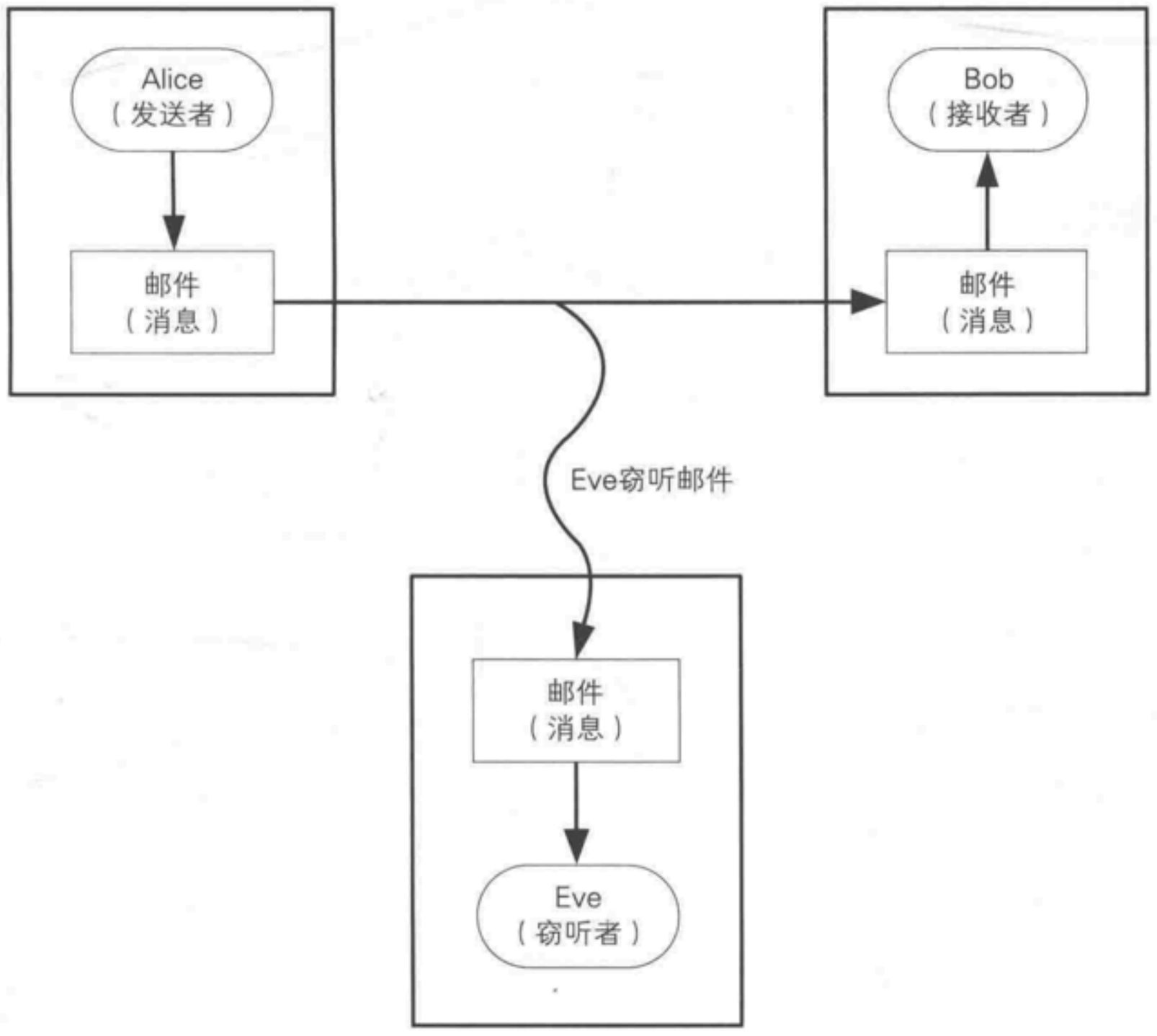
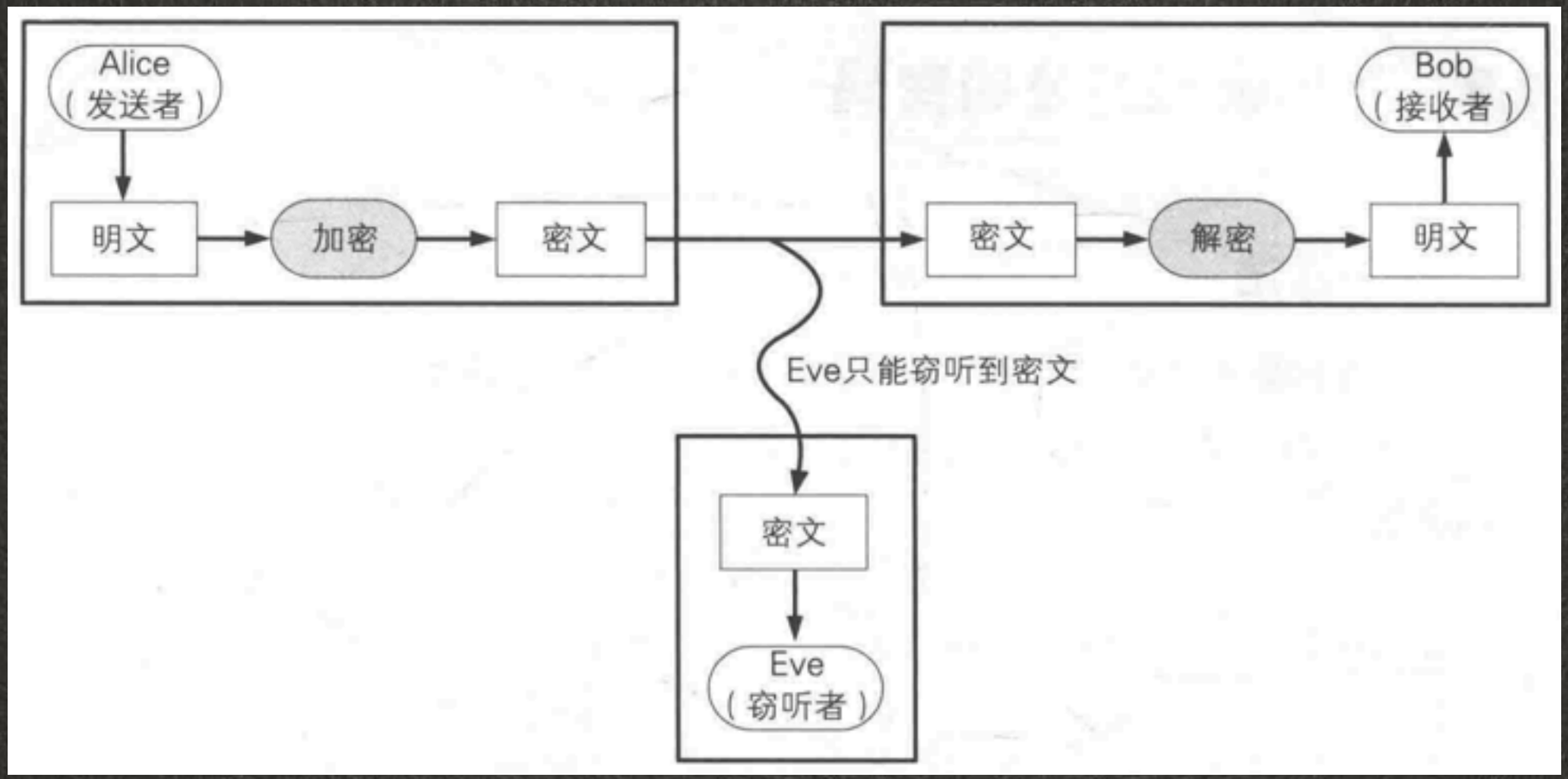


消息在发送的过程中可能被窃听

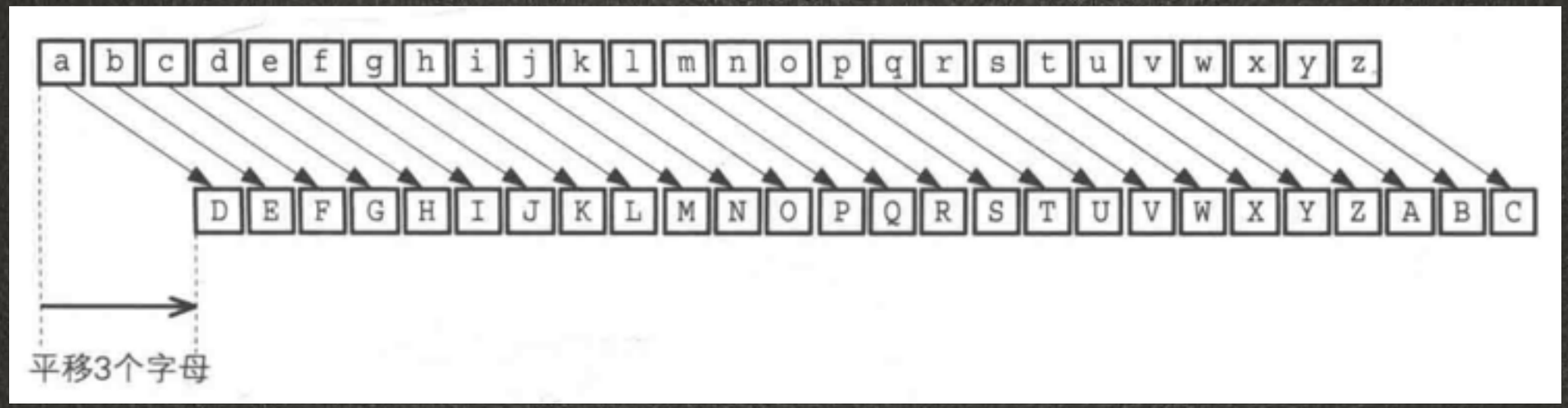
5201314 —2→ 7423536 —2→ 5201314

加密 解密



加密后窃听者只能窃听到密文

凯撒密码

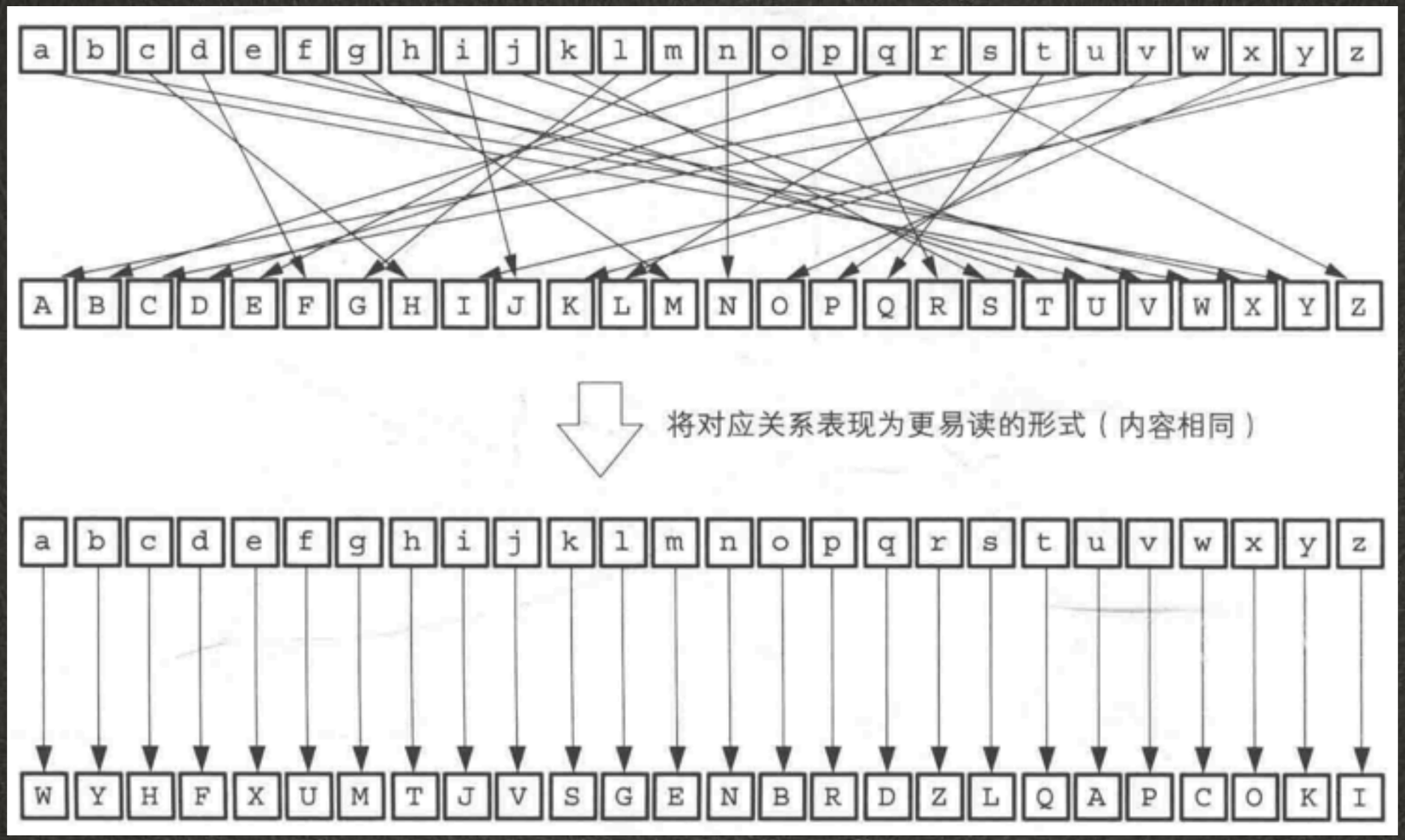


将字母平移一定数量达到加密效果

凯撒密码的破译

- 暴力破解

简单替换密码



将一个字符替换为另一个字符

简单替换密码的破译

频率分析

```
MEYLGVIWAMEYOPINYZGWYEGMZRUUYPZAIXILGVSIZZMPGKKDWOMEPGROEIWGPCEIPAMDKKEYCIUYMGIF
RWCEGLOPINYZHRZMPDNYWDWOGWITDWYSEDCEEIAFYYWMPIDWYAGTYPIKGLMXFPIWCEHRZMMEYMEDWOMG
QRYWCEUXMEDPZMQRGMEEYAPISDWOFICJILYSNICYZEYMGGJIPRWIWAIHRUNIWAHRZMUDZZYAMEYFRWCE
MRPWDWOPGRWAIOIDWSDMEIGWYMSGMEPYYEYHRUNYARNFRMSDMEWGOPYIMYPZRCCYZZIOIDWIWAIOIDWE
YMPDYAILMYPMEYMYUNMDWOUGPZYKFRMIMKIZMEIAMGODTYDMRNIWASIKJYAISIXSDMEEDZWGZYDWMEYI
DPZIXDWODIUZRPYMEYXIPYZGRPDMDZYIZXMGAYZNDZYSEIMXGRCIWWGMOYM
```

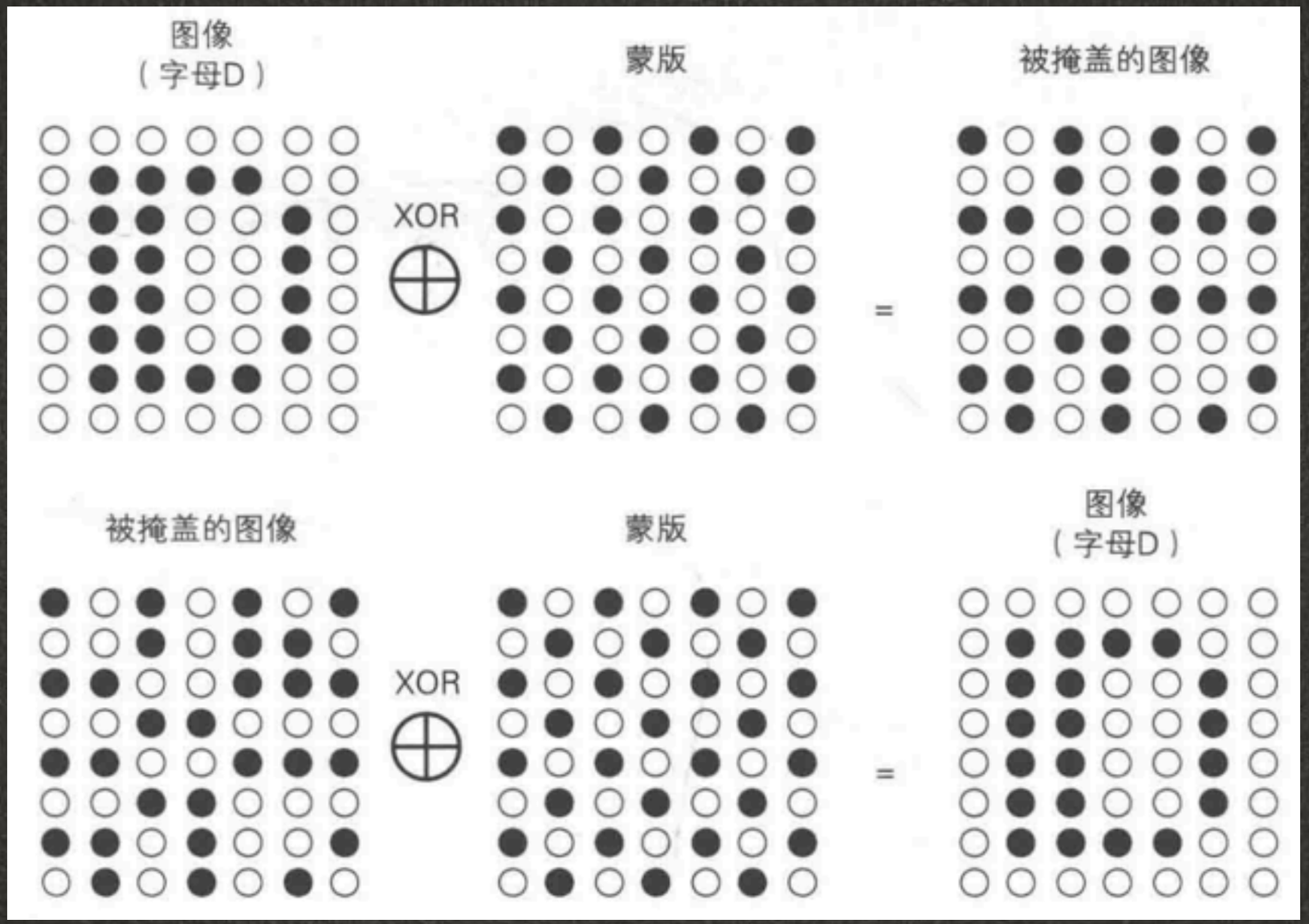
字母	个数	字母	个数	字母	个数	字母	个数	字母	个数
I	47 个	G	27 个	C	12 个	F	7 个	V	2 个
Y	47 个	Z	27 个	S	11 个	L	6 个	B	0 个
M	45 个	P	26 个	N	10 个	H	5 个		
W	35 个	R	22 个	U	10 个	J	3 个		
E	33 个	A	17 个	K	8 个	T	3 个		
D	30 个	O	16 个	X	8 个	Q	2 个		

异或运算

$$
\begin{aligned}
0 \oplus 0 &= 0 \\
0 \oplus 1 &= 1 \\
1 \oplus 0 &= 1 \\
1 \oplus 1 &= 0
\end{aligned}
$$

$$
\begin{array}{rcccccccccc}
 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & \cdots & A \\
\oplus & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & \cdots & B \\
\hline
 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & \cdots & A \oplus B
\end{array}
$$

$$
\begin{array}{rcccccccccc}
 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & \cdots & A \oplus B \\
\oplus & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & \cdots & B \\
\hline
 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & \cdots & A
\end{array}
$$


图像
（字母D）
XOR
蒙版
=
被掩盖的图像
被掩盖的图像
XOR
蒙版
=
图像
（字母D）

一次性密码本

	01101101 01101001 01100100 01101110 01101001 01100111 01101000 01110100	明文“midnight”
⊕	01101011 11111010 01001000 11011000 01100101 11010101 10101111 00011100	密钥
	00000110 10010011 00101100 10110110 00001100 10110010 11000111 01101000	密文

	00000110 10010011 00101100 10110110 00001100 10110010 11000111 01101000	密文
⊕	01101011 11111010 01001000 11011000 01100101 11010101 10101111 00011100	密钥
	01101101 01101001 01100100 01101110 01101001 01100111 01101000 01110100	解密后得到明文 midnight

一次性密码本

- 缺点
  1. 密钥如何发送
  2. 密钥长度问题
  3. 密钥无法复用

对称密码

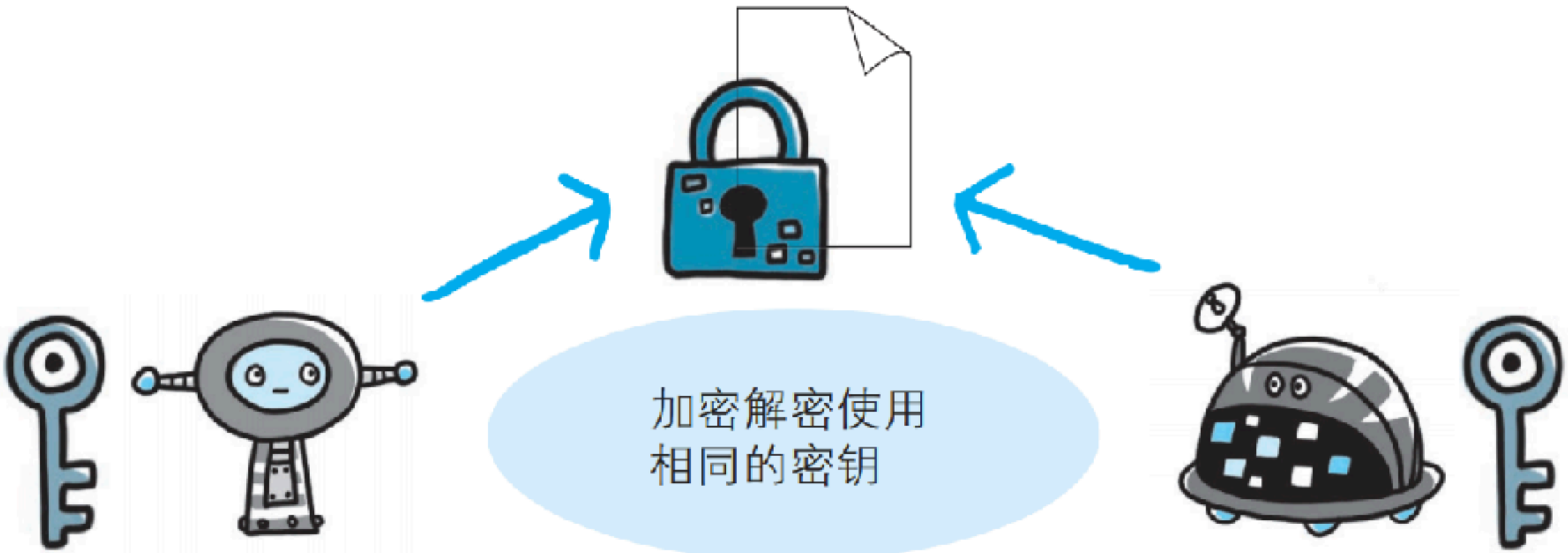
- DES
- AES

DES 算法的安全性

- 1997 DES-challenge I 96天破译
- 1998 DES-challenge II-1 41天破译
- 1998 DES-challenge II-2 56小时破译
- 1998 DES-challenge III 22小时破译

三重DES

- 加密-解密-加密


加密解密使用
相同的密钥
客户端
服务器
呵呵呵
攻击者
只要拿到密钥，
任何人都能破解
密码

对称密码的问题

- 密钥如何配送

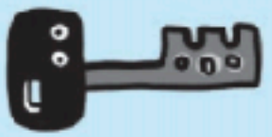
    1. 事先共享
    2. 密钥分配中心
    3. Diffie-Hellman：共享一部分信息生成密码
    4. 非对称密码（公钥密码）

非对称密码

1. Bob 生成包含 公钥/私钥 的密钥对
2. 公钥发给 Alice，私钥由 Bob 保存
3. Alice 用 Bob 的公钥进行加密，发送给 Bob
4. Bob 用私钥解密

公开密钥可
转交给任何人

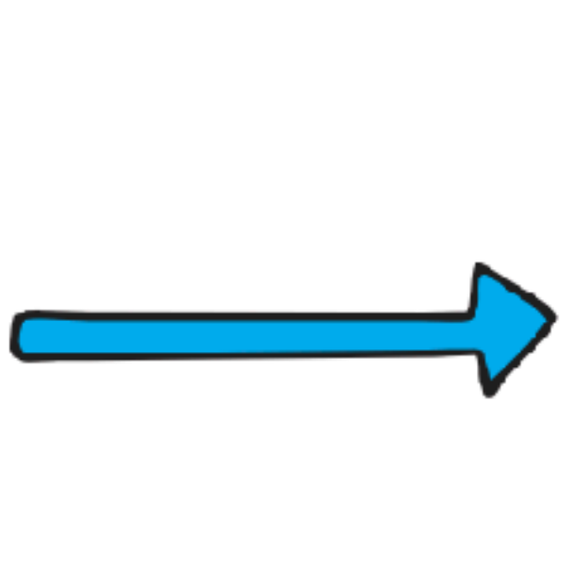
公开密钥



私有密钥

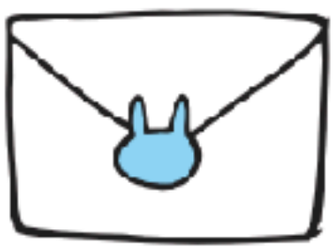
公开密钥

公开密钥和私有密钥
是配对的一套密钥



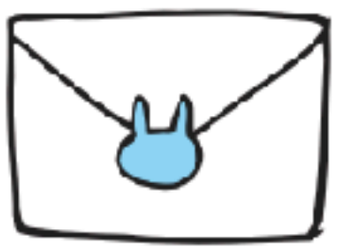
公开密钥

私有密钥



使用公开密钥
进行加密

使用私有密钥
进行解密



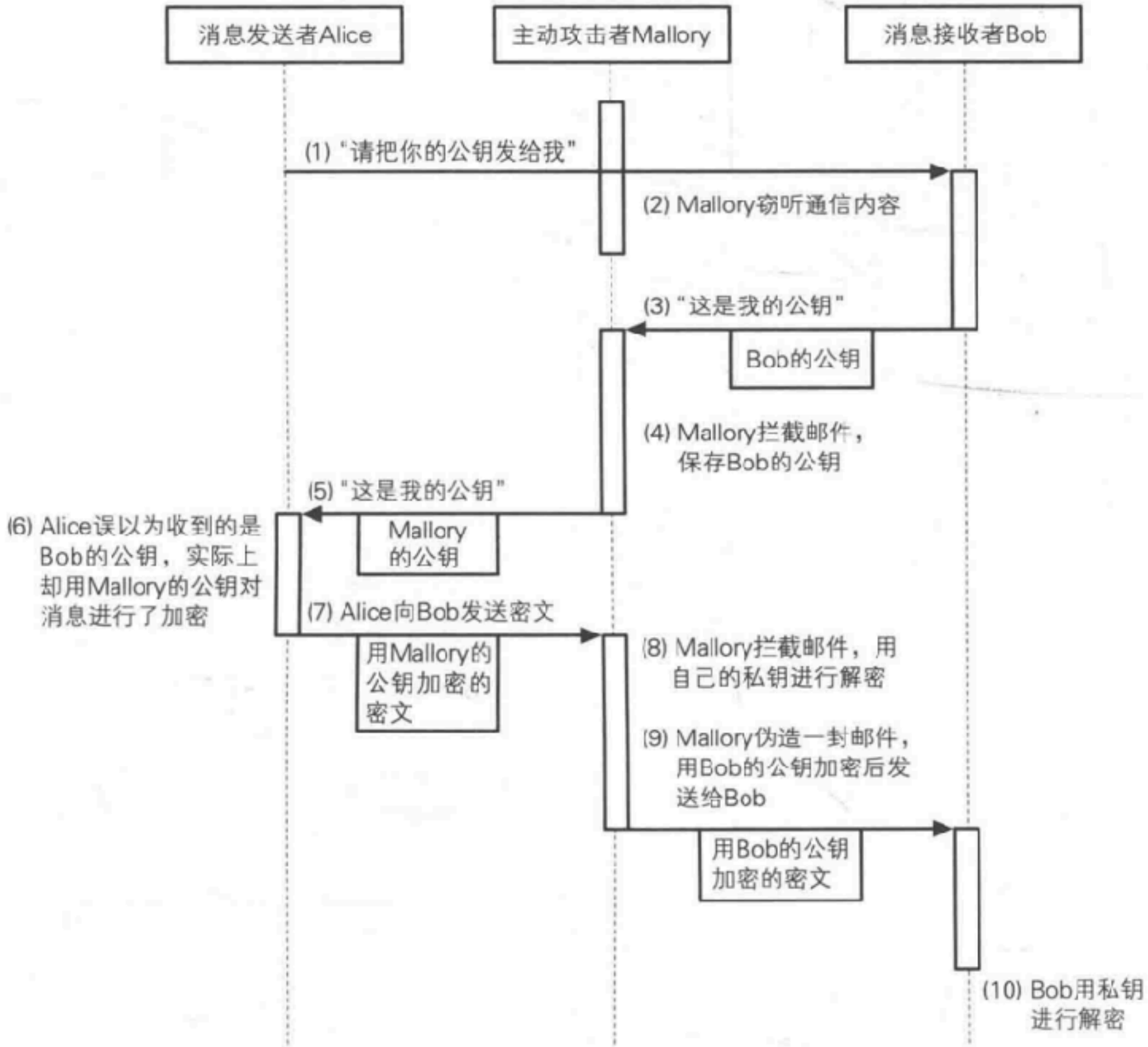
RSA

- 准备两个很大的质数，p 和 q
- 求 N，N = p * q
- 求 L，L = p - 1，q - 1 的最小公倍数
- 求 E，1 < E < L，E 和 L 互质
- 求 D，1 < D < L，E * D Mod L = 1

- 密文 = 明文^E mod N
- 明文 = 密文^D mod N

- 非对称密码的缺点：效率低，解决方案：混合密码

中间人攻击


消息发送者Alice
主动攻击者Mallory
消息接收者Bob
(1) "请把你的公钥发给我"
(2) Mallory窃听通信内容
(3) "这是我的公钥"
Bob的公钥
(4) Mallory拦截邮件，保存Bob的公钥
(5) "这是我的公钥"
Mallory的公钥
(6) Alice误以为收到的是Bob的公钥，实际上却用Mallory的公钥对消息进行了加密
(7) Alice向Bob发送密文
用Mallory的公钥加密的密文
(8) Mallory拦截邮件，用自己的私钥进行解密
(9) Mallory伪造一封邮件，用Bob的公钥加密后发送给Bob
用Bob的公钥加密的密文
(10) Bob用私钥进行解密

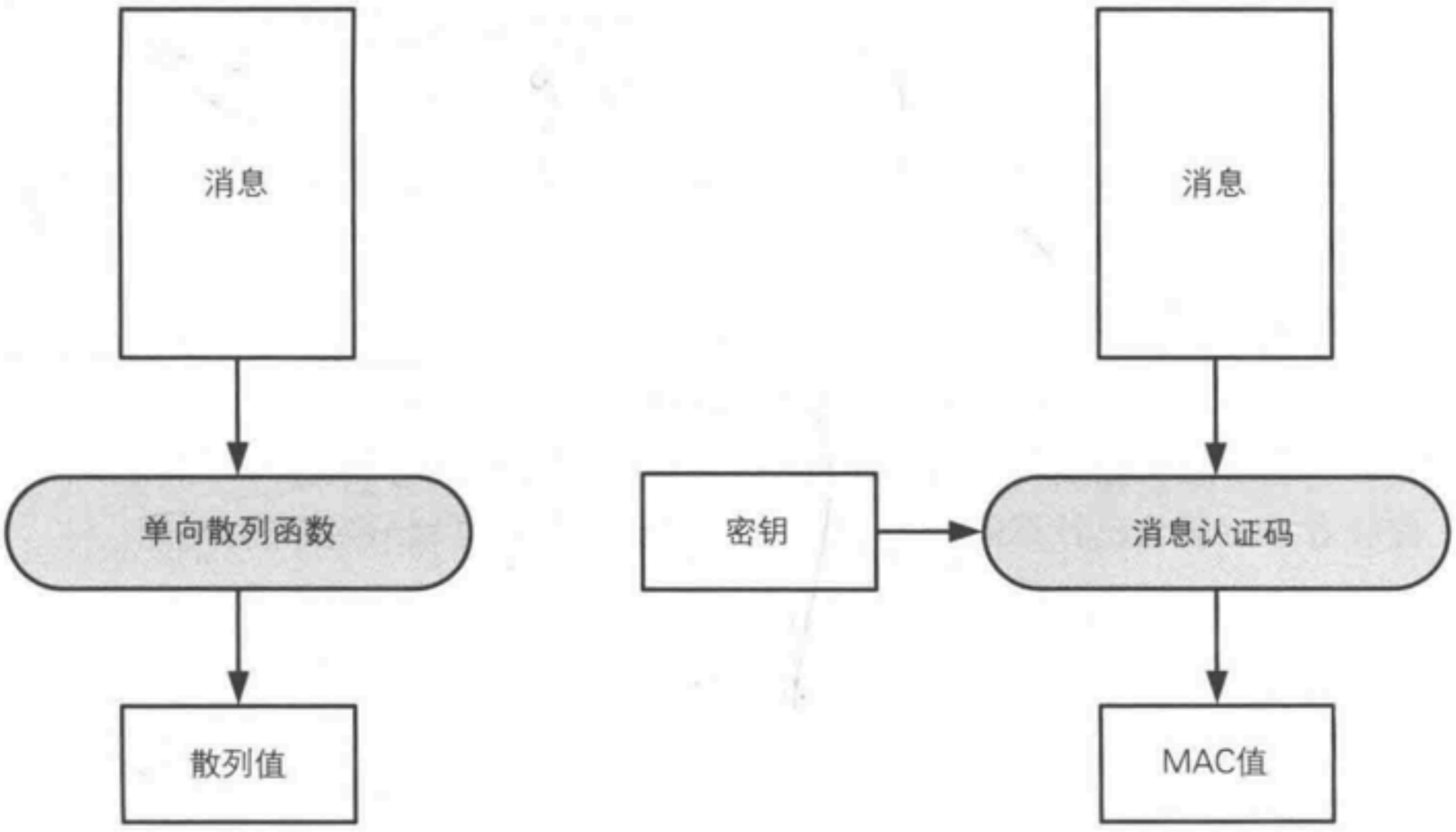
- 密码解决的问题：机密性
- 密码不能解决的问题：完整性，认证，不可否认性

散列函数

- 根据任意长度的内容计算出固定长度的散列值
- 能够快速计算出散列值
- 内容不同散列值不同
- 单向性

- MD4（抗碰撞性已被攻破）
- MD5（抗碰撞性已被攻破）
- SHA-1（抗碰撞性已被攻破）
- SHA-2: SHA-256、SHA-384、SHA-512
- SHA-3
- RIPEMD-160

消息认证码



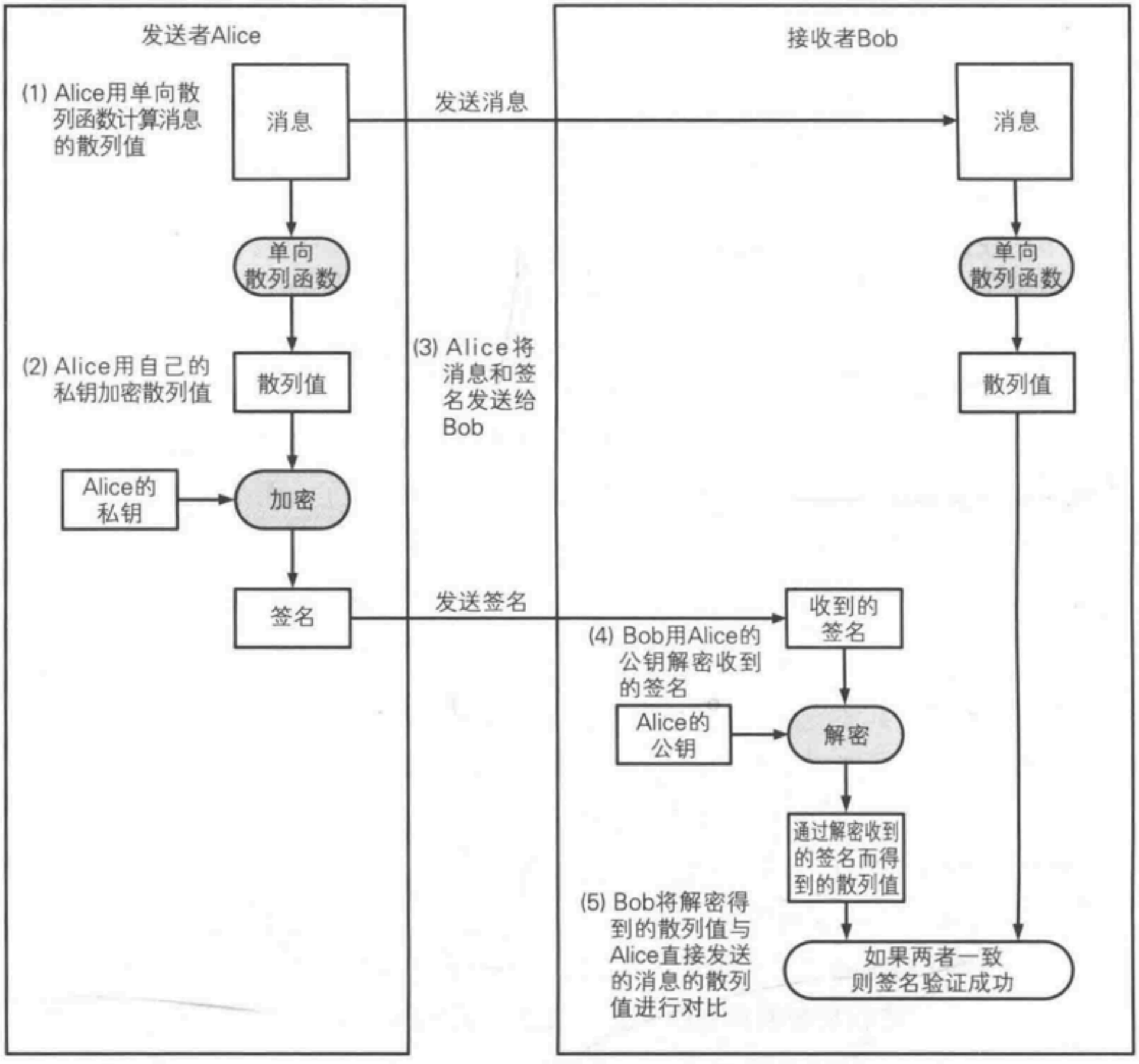
重放攻击

- 解决方式

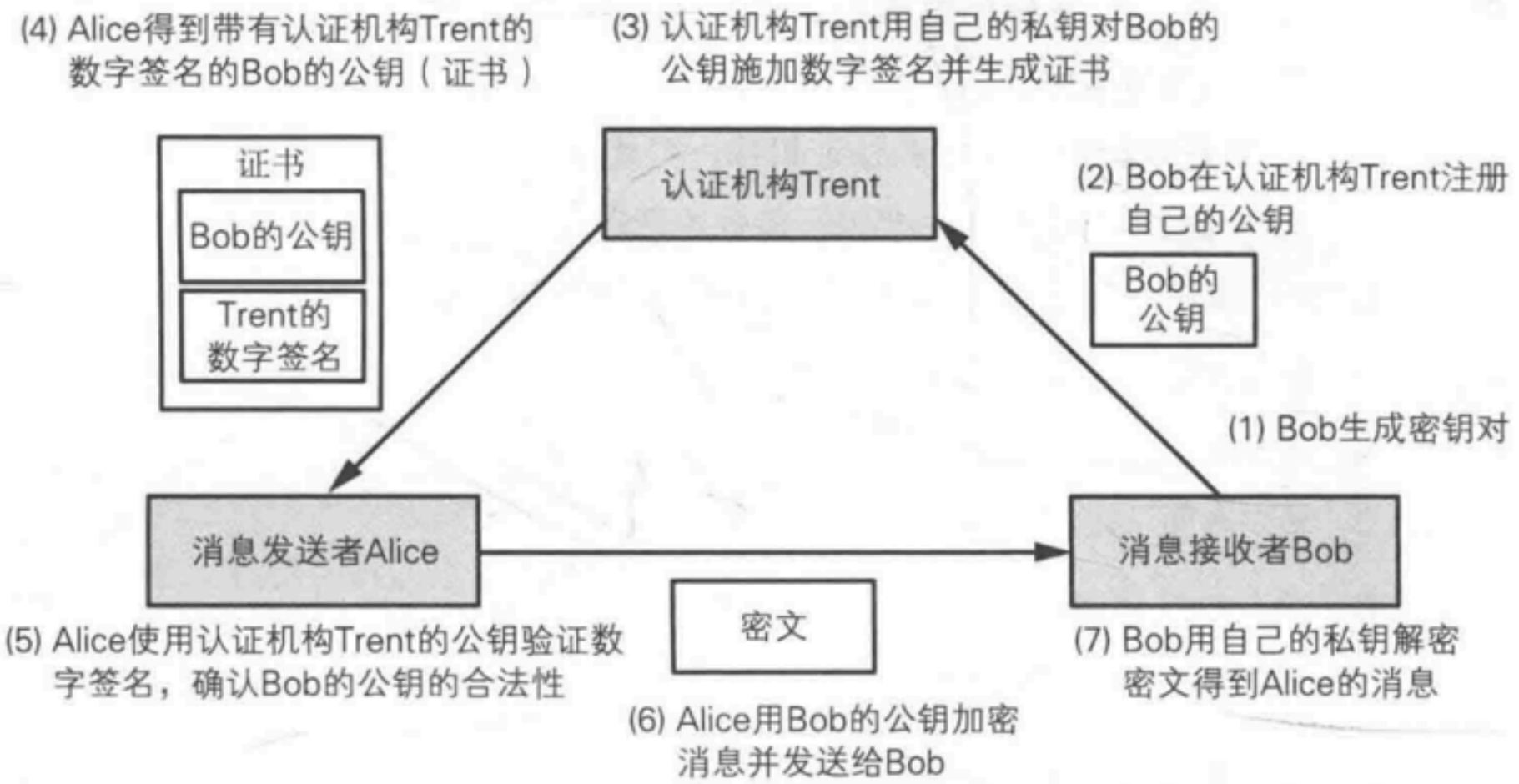
    1. 序号
    2. 时间戳
    3. 随机数

- 消息认证码解决的问题：完整性
- 无法解决的问题：不可否认性

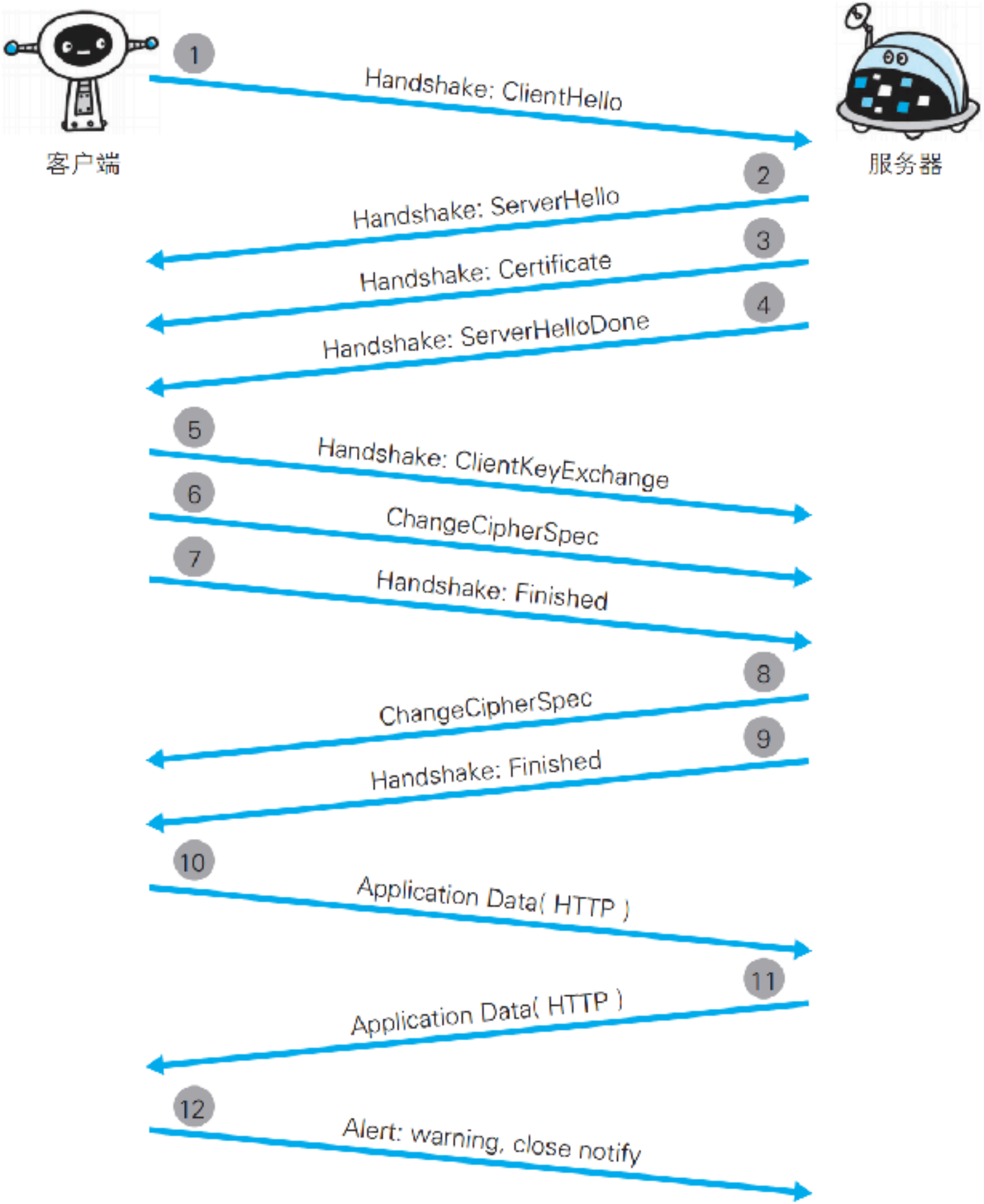
数字签名


发送者Alice
接收者Bob
(1) Alice用单向散列函数计算消息的散列值
消息
发送消息
消息
单向散列函数
单向散列函数
(2) Alice用自己的私钥加密散列值
散列值
(3) Alice将消息和签名发送给Bob
散列值
Alice的私钥
加密
签名
发送签名
收到的签名
(4) Bob用Alice的公钥解密收到的签名
Alice的公钥
解密
通过解密收到的签名而得到的散列值
(5) Bob将解密得到的散列值与Alice直接发送的消息的散列值进行对比
如果两者一致则签名验证成功

证书


(4) Alice得到带有认证机构Trent的数字签名的Bob的公钥（证书）
(3) 认证机构Trent用自己的私钥对Bob的公钥施加数字签名并生成证书
证书
Bob的公钥
Trent的数字签名
认证机构Trent
(2) Bob在认证机构Trent注册自己的公钥
Bob的公钥
(1) Bob生成密钥对
消息发送者Alice
消息接收者Bob
密文
(5) Alice使用认证机构Trent的公钥验证数字签名，确认Bob的公钥的合法性
(6) Alice用Bob的公钥加密消息并发送给Bob
(7) Bob用自己的私钥解密密文得到Alice的消息

HTTPS


客户端
服务器
1
Handshake: ClientHello
2
Handshake: ServerHello
3
Handshake: Certificate
4
Handshake: ServerHelloDone
5
Handshake: ClientKeyExchange
6
ChangeCipherSpec
7
Handshake: Finished
8
ChangeCipherSpec
9
Handshake: Finished
10
Application Data(HTTP)
11
Application Data(HTTP)
12
Alert: warning, close notify

加密在比特币中的应用

- 对交易账单计算散列值
- 对散列值用私钥签名
- 矿工使用公钥验证，追加到区块中