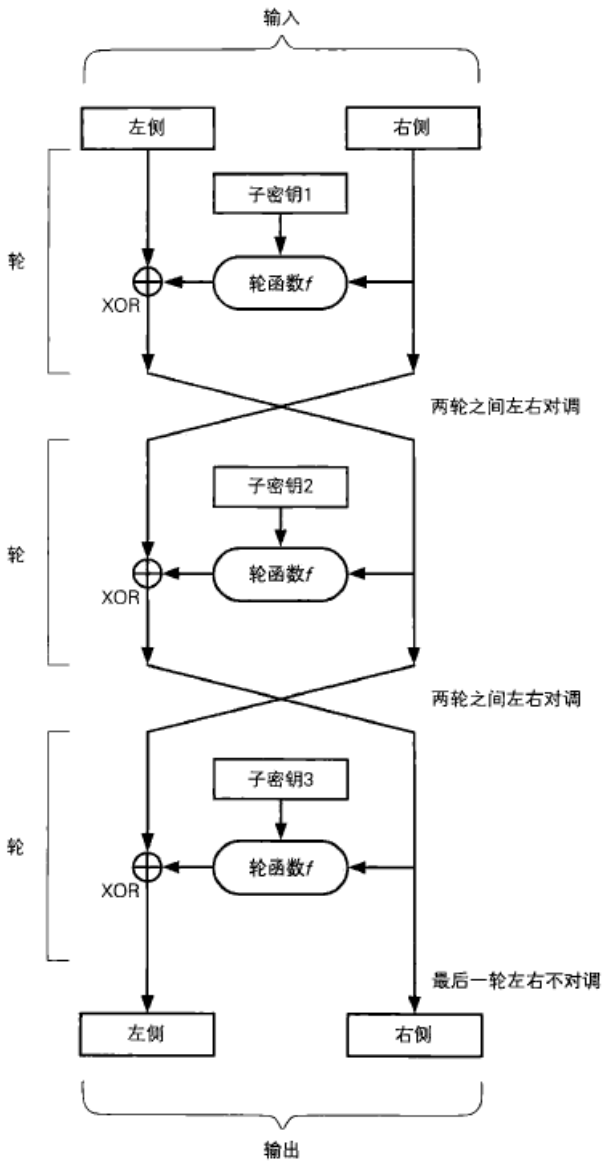


图 3-4　Feistel 网络的加密（3 轮）

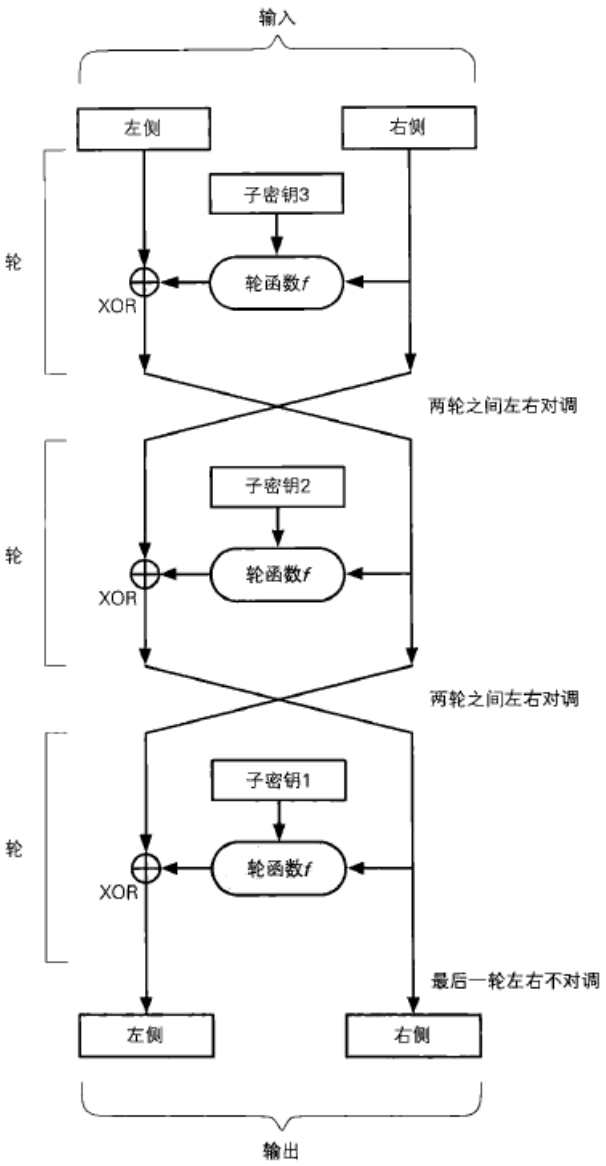


图 3-6　Feistel 网络的解密（3 轮）

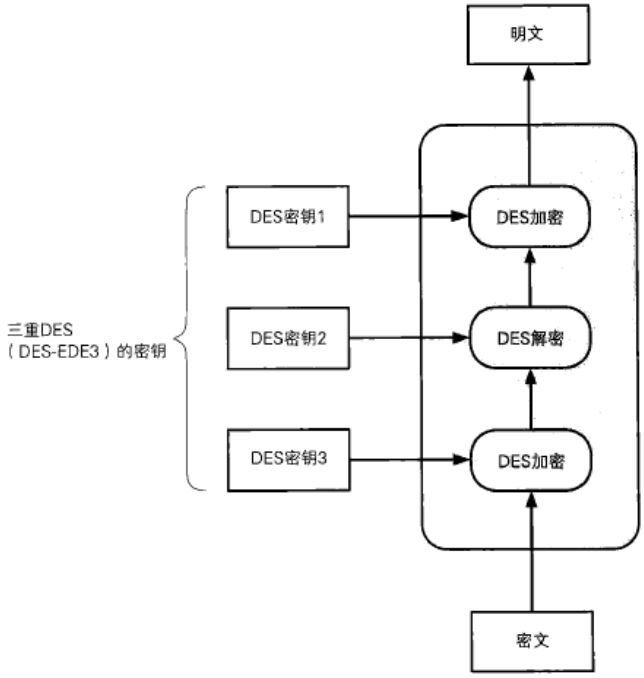


图 3-7　三重 DES 的加密

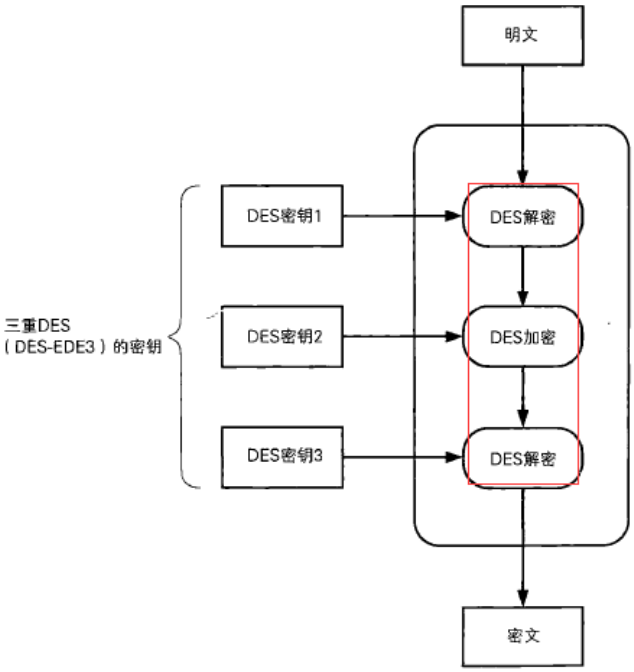


图 3-10　三重 DES（DES-EDE3）的解密

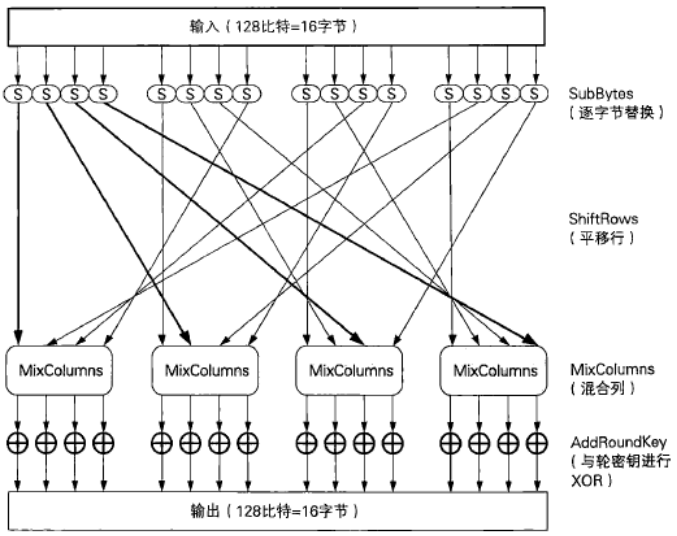


图 3-11　Rijndael 加密中的一轮

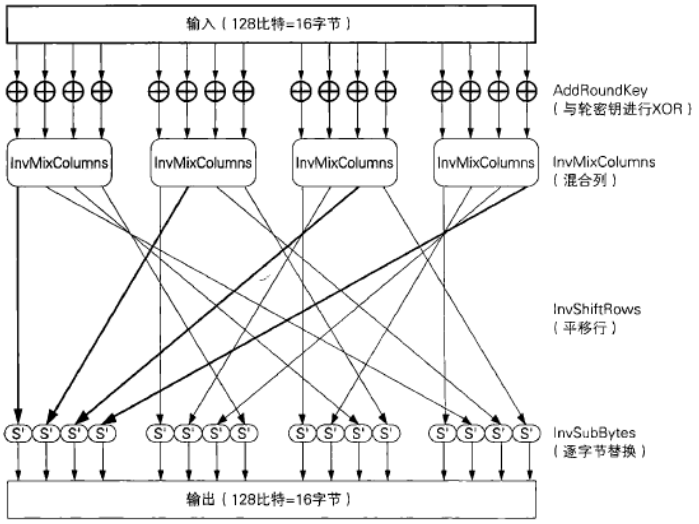
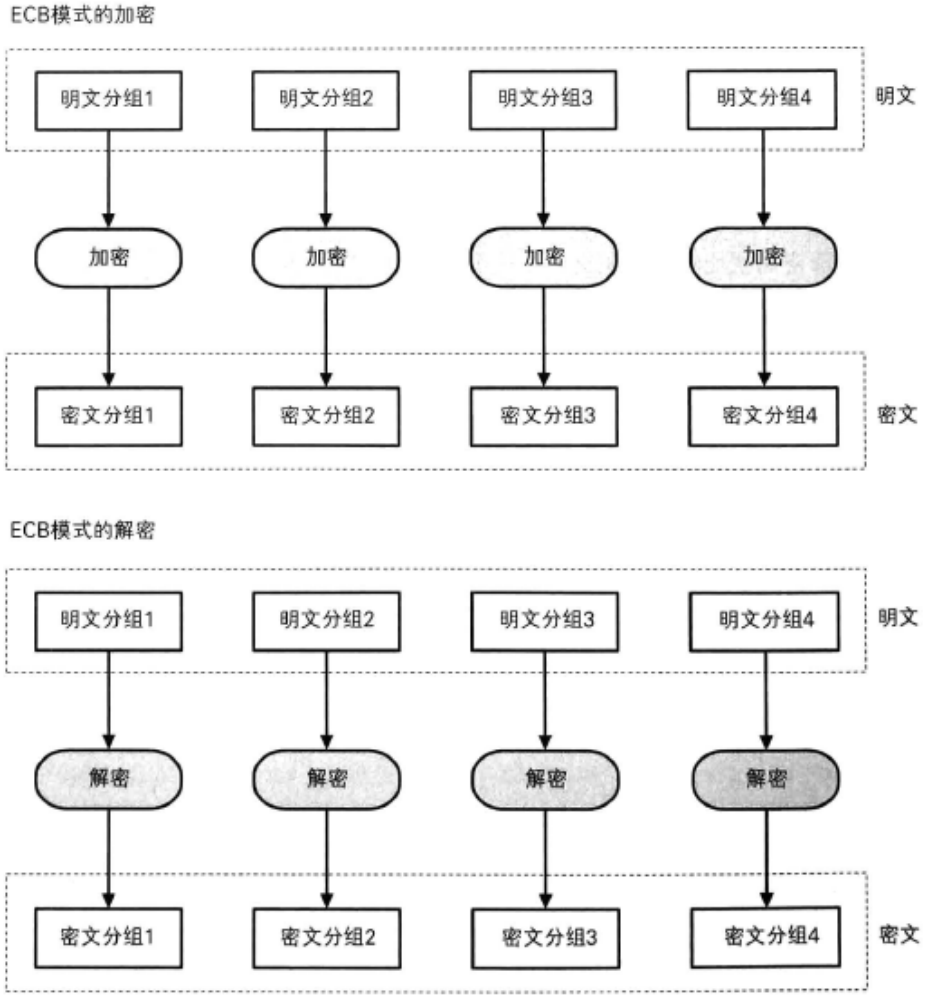
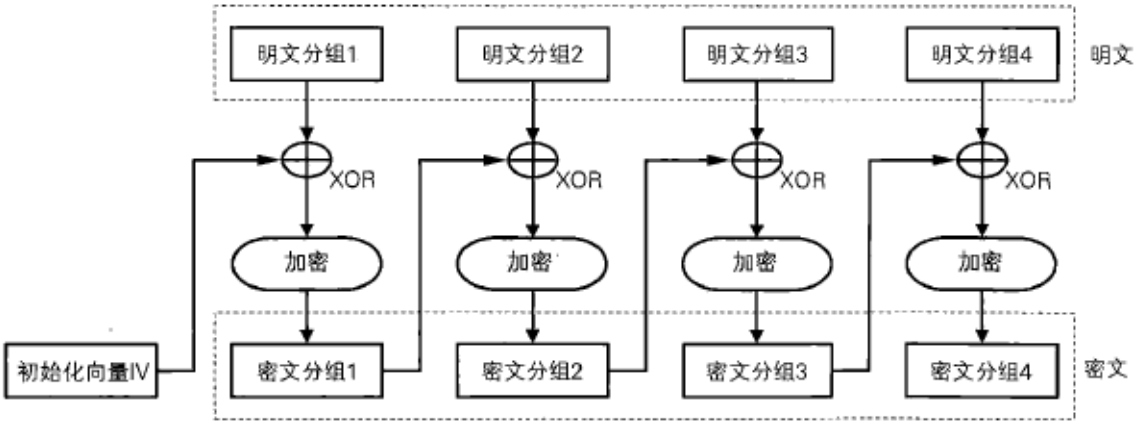


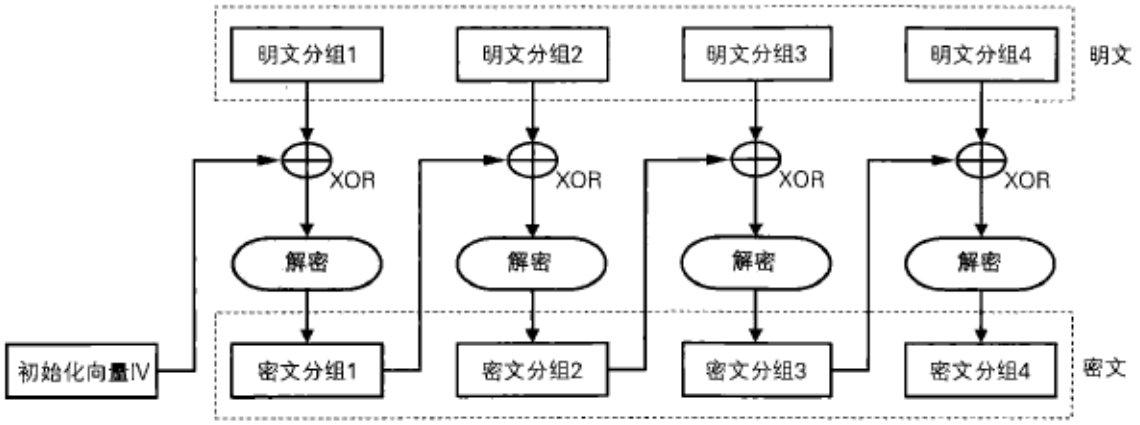
图 3-12　Rijndael 解密中的一轮



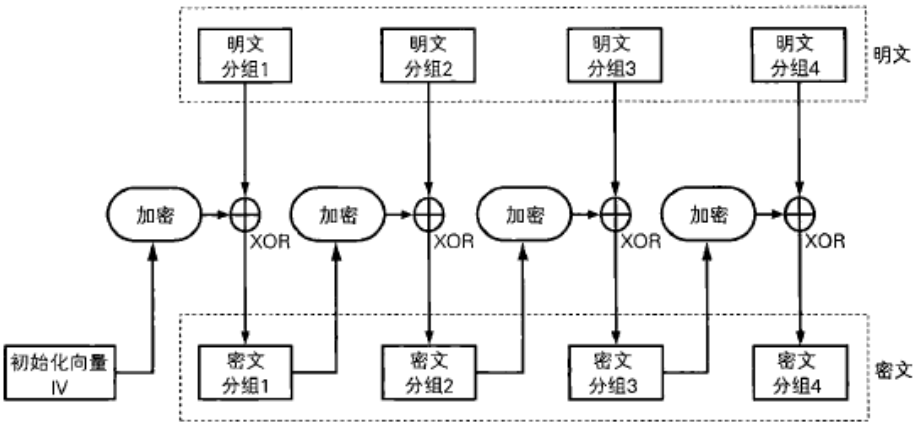
CBC模式的加密



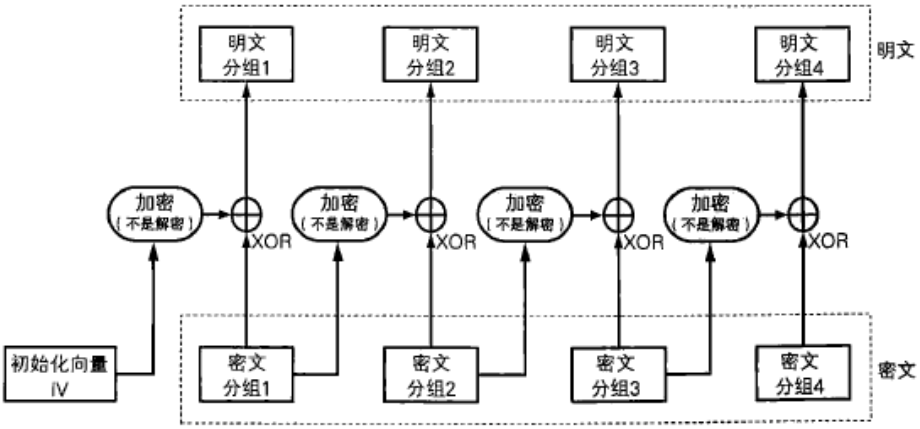
CBC模式的解密



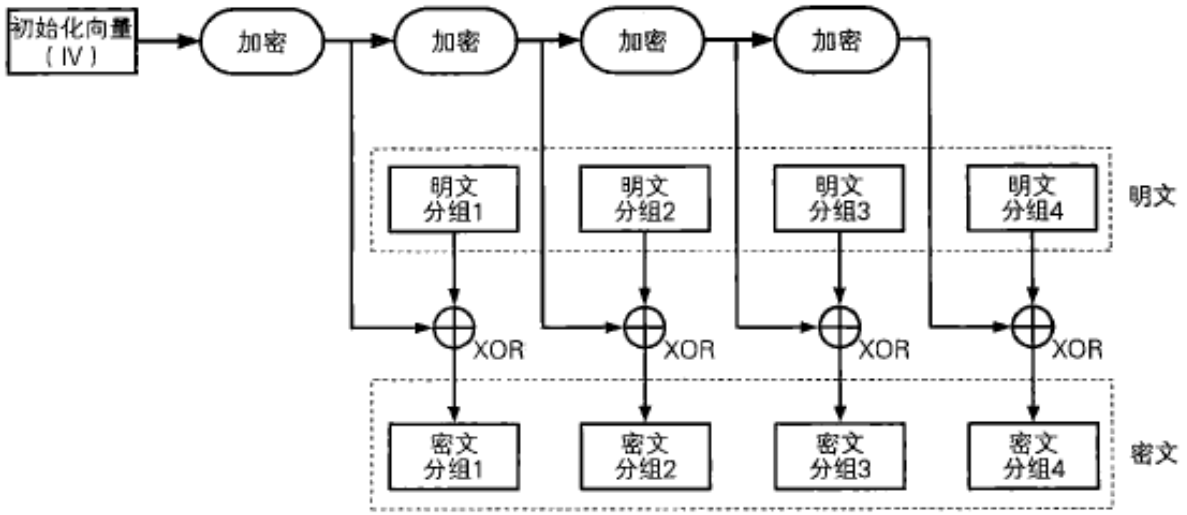
CFB模式的加密



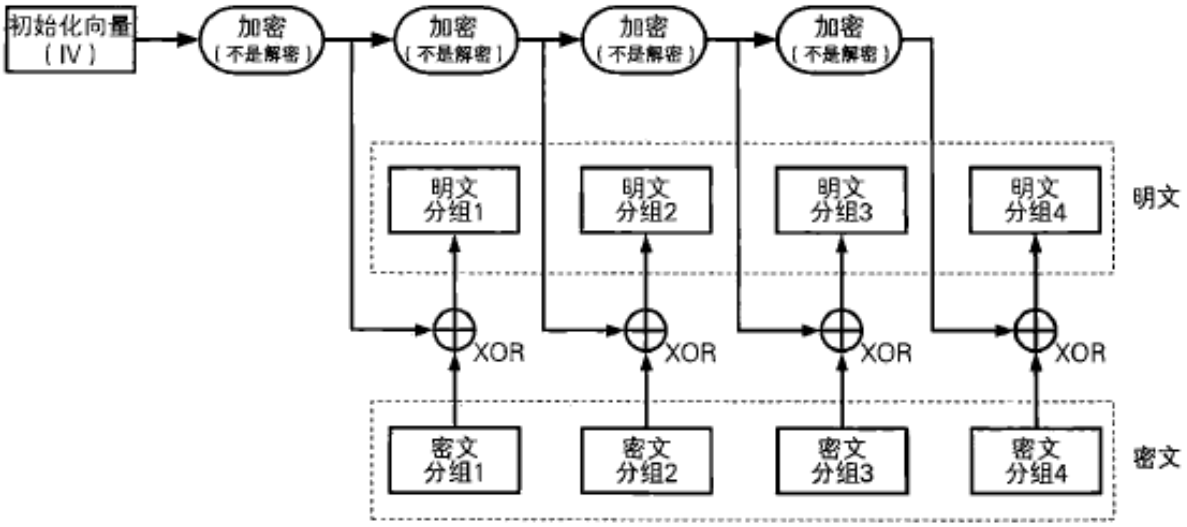
CFB模式的解密



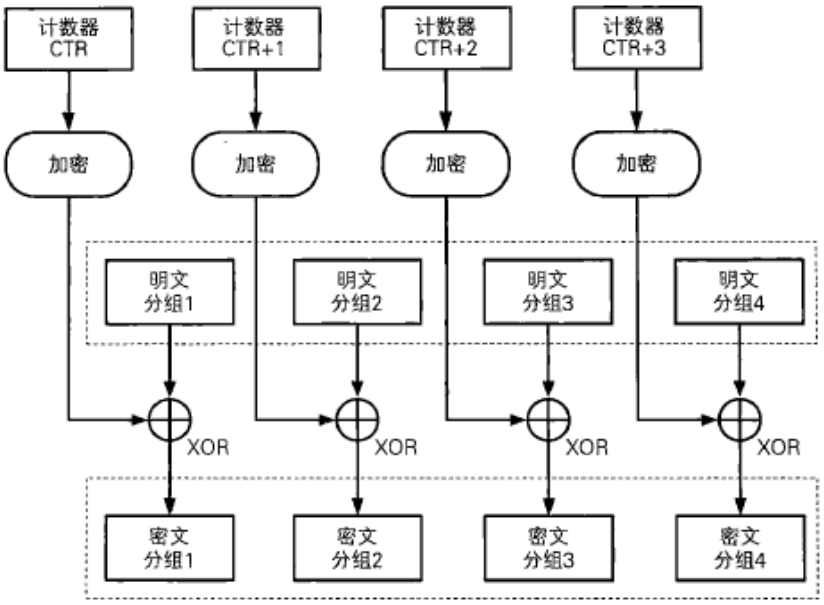
OFB模式的加密



OFB模式的解密



CTR模式的加密



CTR模式的解密

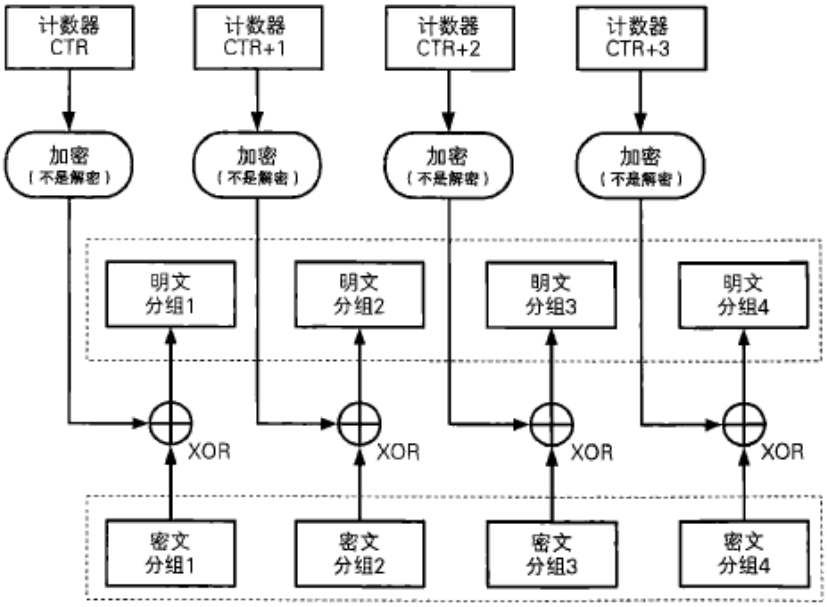


表 4-1　分组密码模式比较表

模式	名称	优点	缺点	备注
ECB 模式	Electronic CodeBook 电子密码本模式	• 简单 • 快速 • 支持并行计算（加密、解密）	• 明文中的重复排列会反映在密文中 • 通过删除、替换密文分组可以对明文进行操作 • 对包含某些比特错误的密文进行解密时，对应的分组会出错 • 不能抵御重放攻击	不应使用
CBC 模式	Cipher Block Chaining 密文分组链接模式	• 明文的重复排列不会反映在密文中 • 支持并行计算（仅解密） • 能够解密任意密文分组	• 对包含某些错误比特的密文进行解密时，第一个分组的全部比特以及后一个分组的相应比特会出错 • 加密不支持并行计算	推荐使用
CFB 模式	Cipher-FeedBack 密文反馈模式	• 不需要填充（padding） • 支持并行计算（仅解密） • 能够解密任意密文分组	• 加密不支持并行计算 • 对包含某些错误比特的密文进行解密时，第一个分组的全部比特以及后一个分组的相应比特会出错 • 不能抵御重放攻击	• 现在已不使用 • 推荐用 CTR 模式代替
OFB 模式	Output-FeedBack 输出反馈模式	• 不需要填充（padding） • 可事先进行加密、解密的准备 • 加密、解密使用相同结构 • 对包含某些错误比特的密文进行解密时，只有明文中相对应的比特会出错	• 不支持并行计算 • 主动攻击者反转密文分组中的某些比特时，明文分组中相对应的比特也会被反转	推荐用 CTR 模式代替
CTR 模式	CounTeR 计数器模式	• 不需要填充（padding） • 可事先进行加密、解密的准备 • 加密、解密使用相同结构 • 对包含某些错误比特的密文进行解密时，只有明文中相对应的比特会出错 • 支持并行计算（加密、解密）	主动攻击者反转密文分组中的某些比特时，明文分组中相对应的比特也会被反转	推荐使用

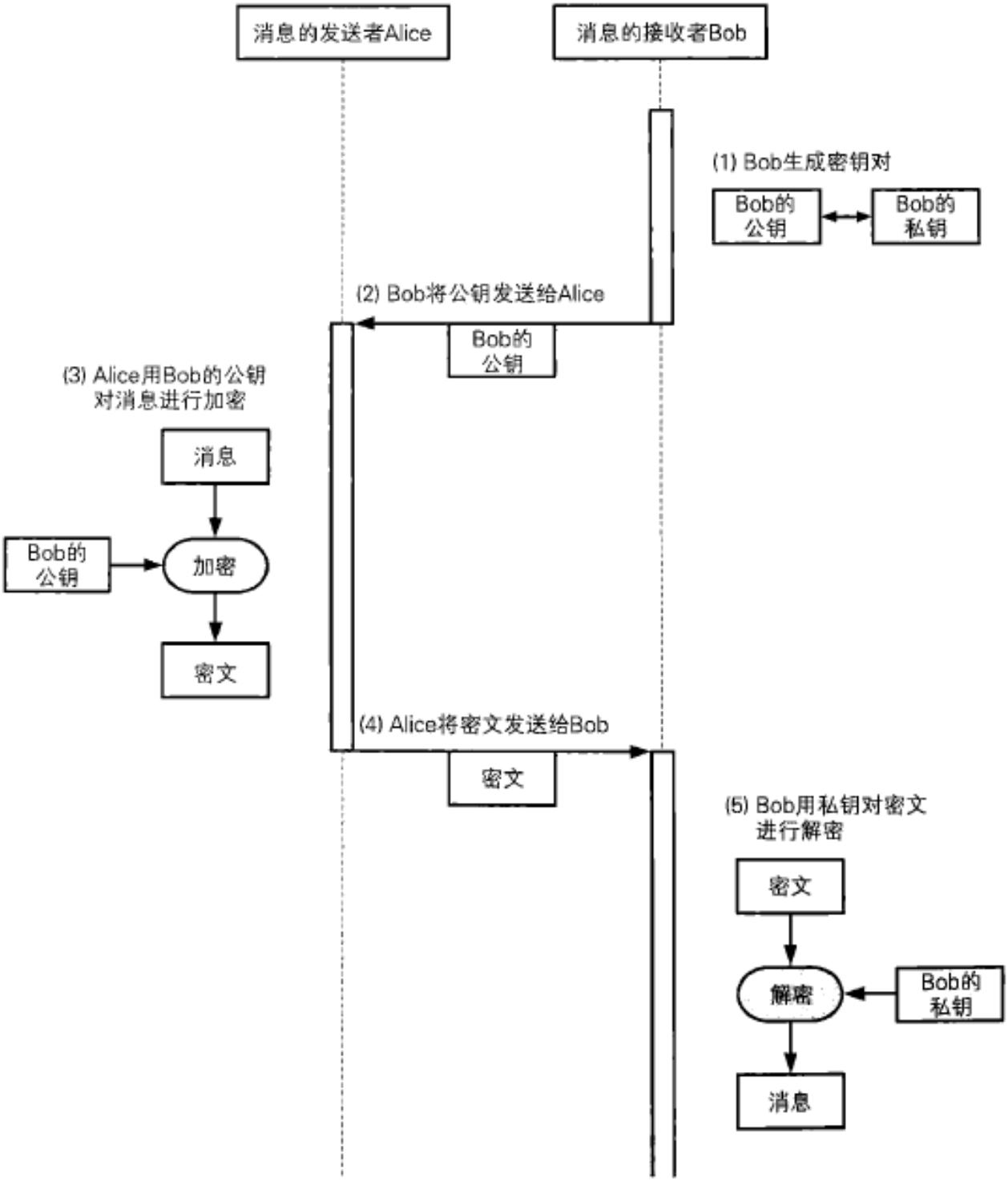


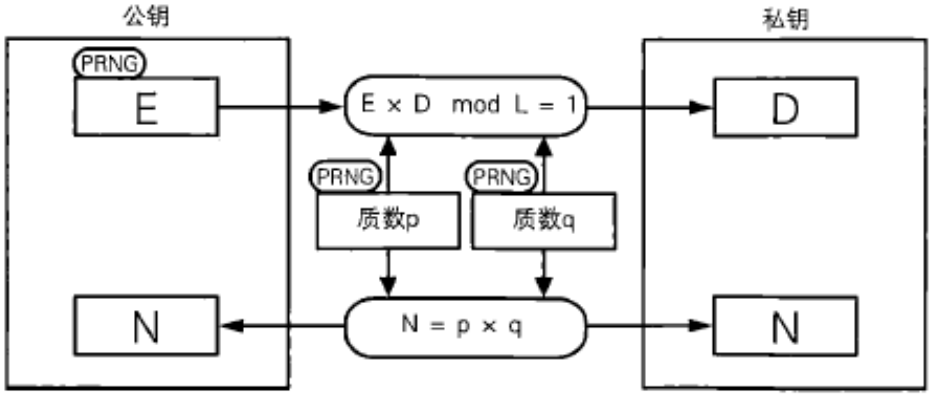
图 5-2　使用公钥密码，Alice 向 Bob 发送消息

表 5-2　RSA 的加密和解密

密钥对	公钥	数 E 和数 N
	私钥	数 D 和数 N
加密		密文 = 明文 E mod N　　（明文的 E 次方除以 N 的余数）
解密		明文 = 密文 D mod N　　（密文的 D 次方除以 N 的余数）

表 5-3　RSA 中密钥对的生成

(1) 求 N	(3) 求 E1 < E < L
用伪随机数生成器求 p 和 q，p 和 q 都是质数 N = p × q	gcd(E, L) = 1；E 和 L 的最大公约数为 1（E 和 L 互质）
(2) 求 L	(4) 求 D
L = lcm(p - 1, q - 1)；L 是 p - 1 和 q - 1 的最小公倍数	1 < D < L E × D mod L = 1



PRNG = 伪随机数生成器

L = lcm（p − 1, q − 1）

gcd（E, L）= 1

1 < E < L

1 < D < L

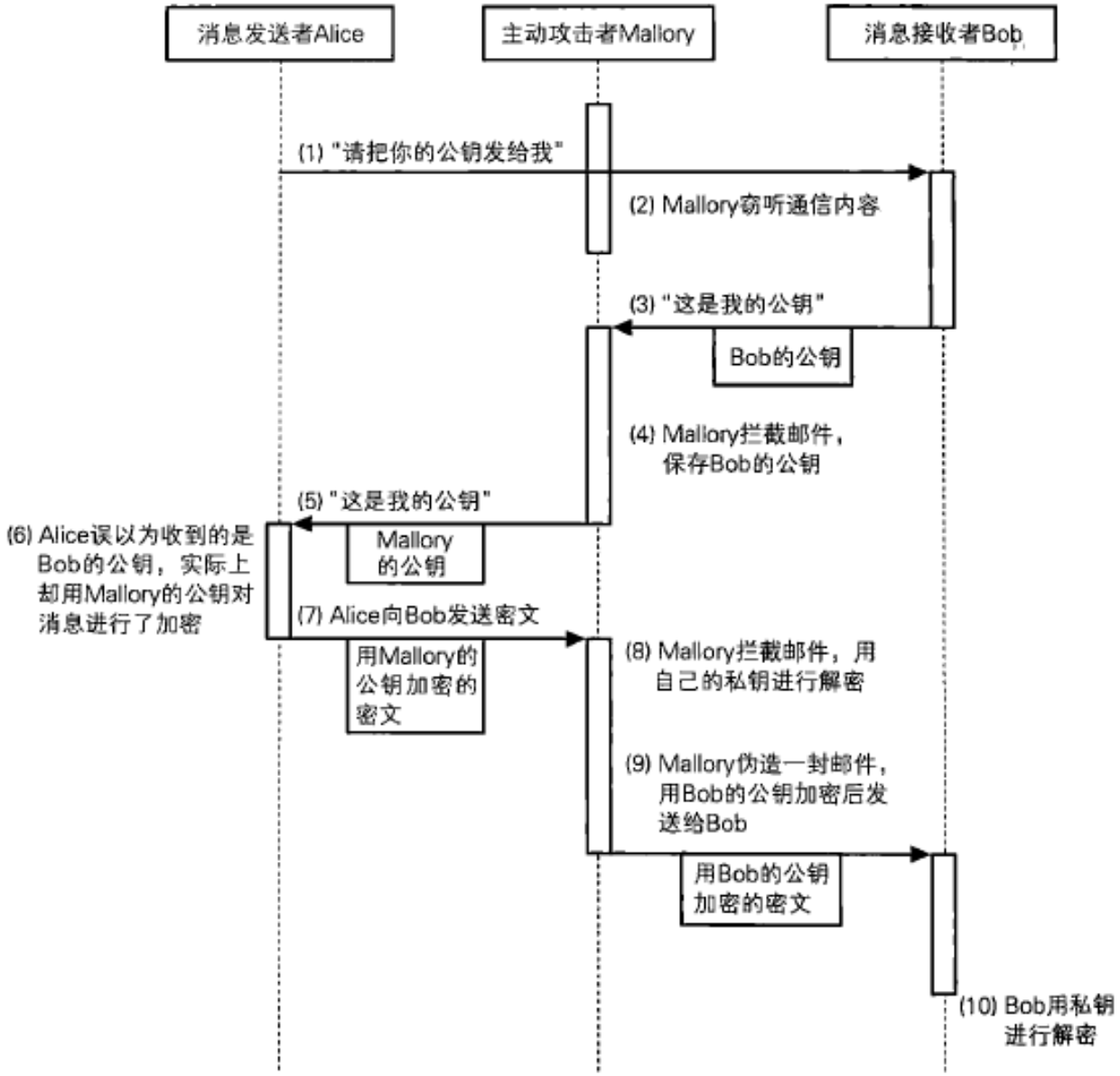


图 5-6　Mallory 进行中间人攻击

表 5-4　具备同等抵御暴力破解强度的密钥长度比较

对称密码的密钥长度	公钥密码的密钥长度
128 比特	2304 比特
112 比特	1792 比特
80 比特	768 比特
64 比特	512 比特
56 比特	384 比特

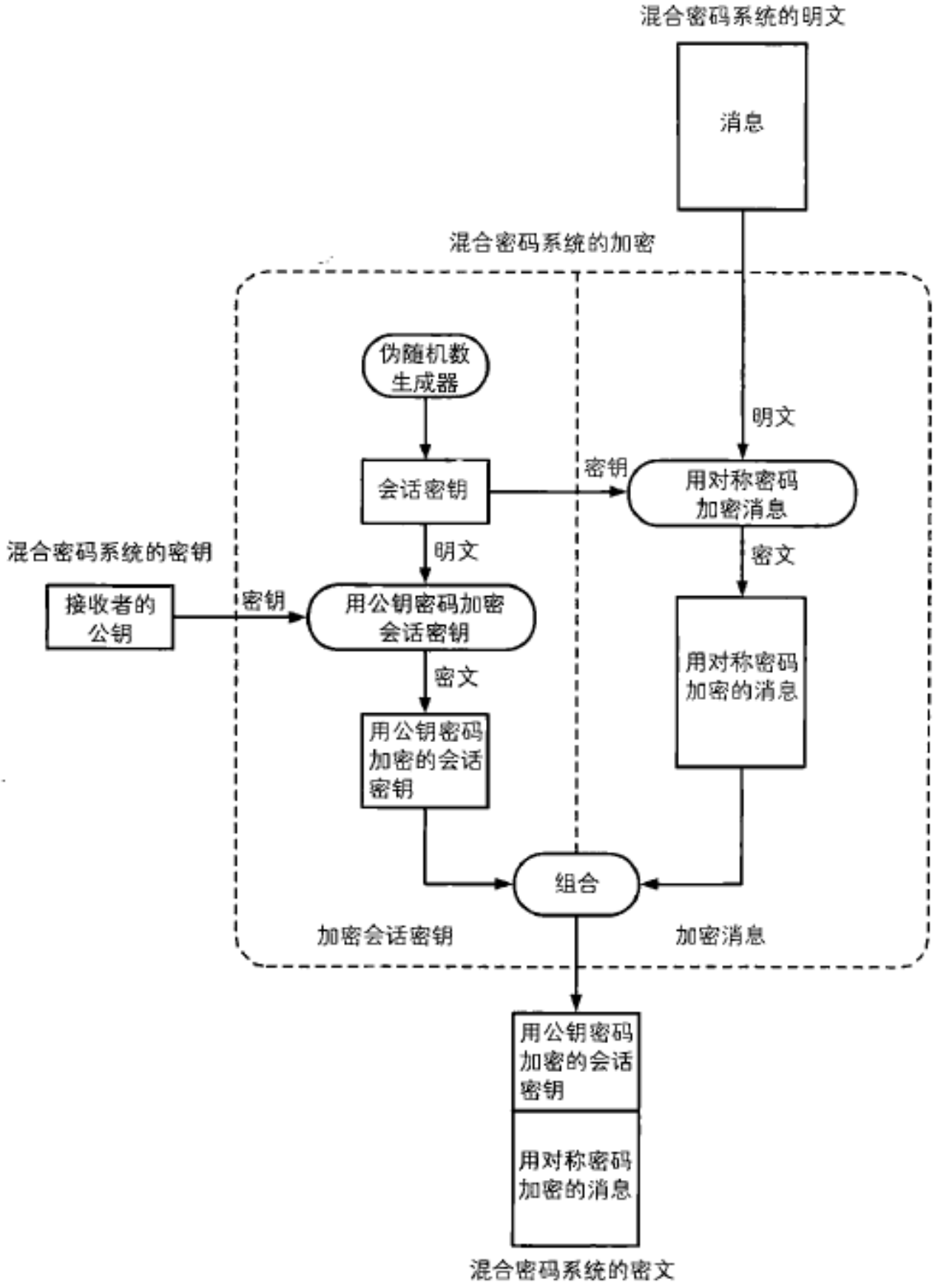


图 6-2　混合密码系统的加密

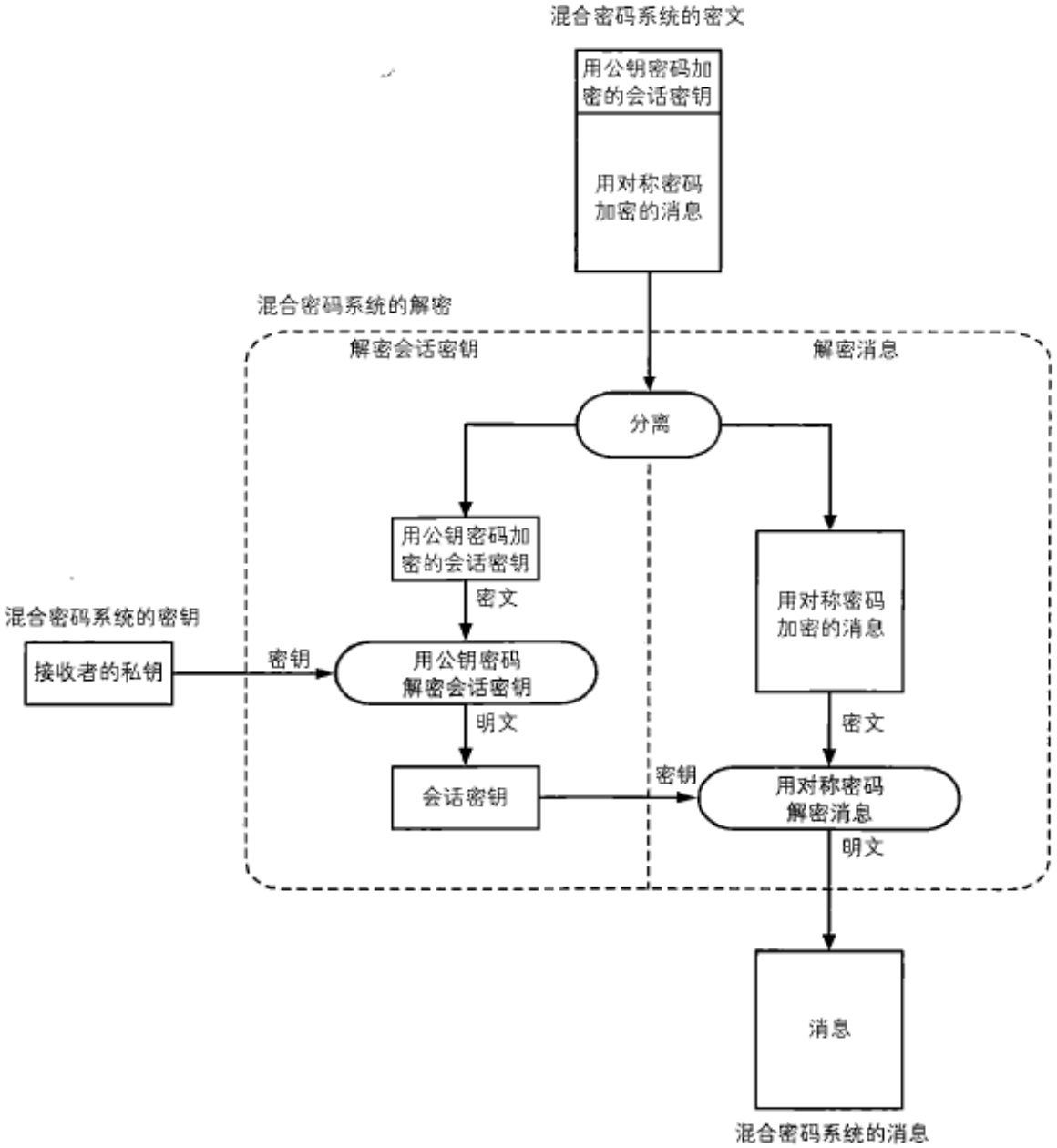


图 6-3　混合密码系统的解密

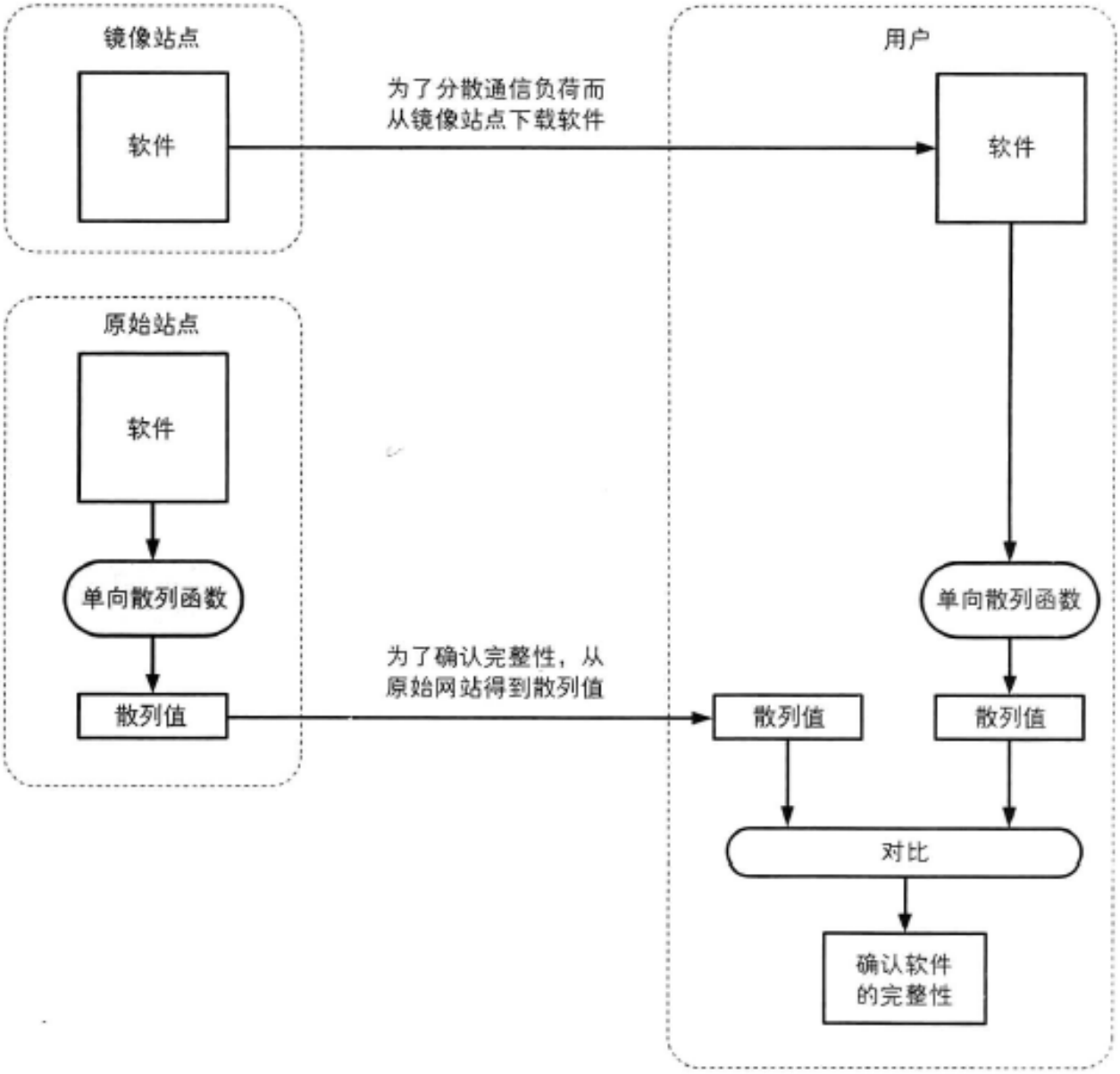


图 7-9　使用单向散列函数检测软件是否被篡改

7.4 单向散列函数的具体例子

下面我们来具体介绍几种单向散列函数。

7.4.1 MD4、MD5

MD4 是由 Rivest 于 1990 年设计的单向散列函数，能够产生 128 比特的散列值（RFC1186，修订版 RFC1320）。不过，随着 Dobbertin 提出寻找 MD4 散列碰撞的方法，因此现在它已经不安全了。

MD5 是由 Rivest 于 1991 年设计的单项散列函数，能够产生 128 比特的散列值（RFC1321）。

MD5 的强抗碰撞性已经被攻破，也就是说，现在已经能够产生具备相同散列值的两条不同的消息，因此它也已经不安全了。

MD4 和 MD5 中的 MD 是消息摘要（Message Digest）的缩写。

7.4.2 SHA-1、SHA-256、SHA-384、SHA-512

SHA-1 是由 NIST（National Institute of Standards and Technology，美国国家标准技术研究所）设计的一种能够产生 160 比特的散列值的单向散列函数。1993 年被作为美国联邦信息处理标准规格（FIPS PUB 180）发布的是 SHA，1995 年发布的修订版 FIPS PUB 180-1 称为 SHA-1。SHA-1 的消息长度存在上限，但这个值接近于 264 比特，是个非常巨大的数值，因此在实际应用中没有问题。关于 SHA-1 的具体算法我们会在后面介绍。

SHA-256、SHA-384 和 SHA512 都是由 NIST 设计的单向散列函数，它们的散列值长度分别为 256 比特、384 比特和 512 比特。这些单向散列函数合起来统称 SHA-2，它们的消息长度也存在上限（SHA-256 的上限接近于 2^{64} 比特，SHA-384 和 SHA-512 的上限接近于 2^{128} 比特）。这些单向散列函数是于 2002 年和 SHA-1 一起作为 FIPS PUB 180-2 发布的。

SHA-1 的强抗碰撞性已于 2005 年被攻破[1]，也就是说，现在已经能够产生具备相同散列值的两条不同的消息。不过，SHA-2 还尚未被攻破。

7.4.3 RIPEMD-160

RIPEMD-160 是于 1996 年由 Hans Dobbertin、Antoon Bosselaers 和 Bart Preneel 设计的一种

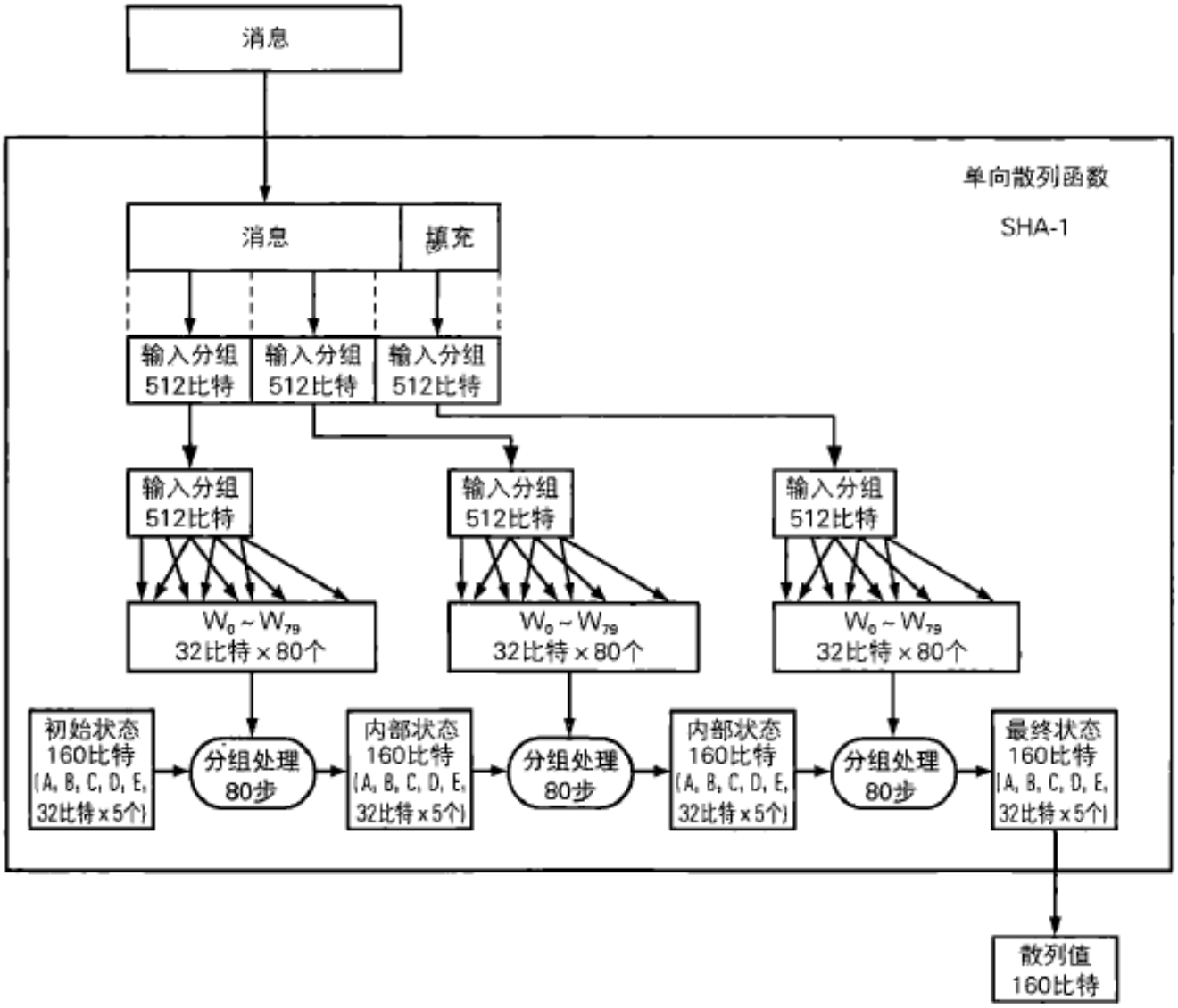


图 7-10　单向散列函数 SHA-1 的概要

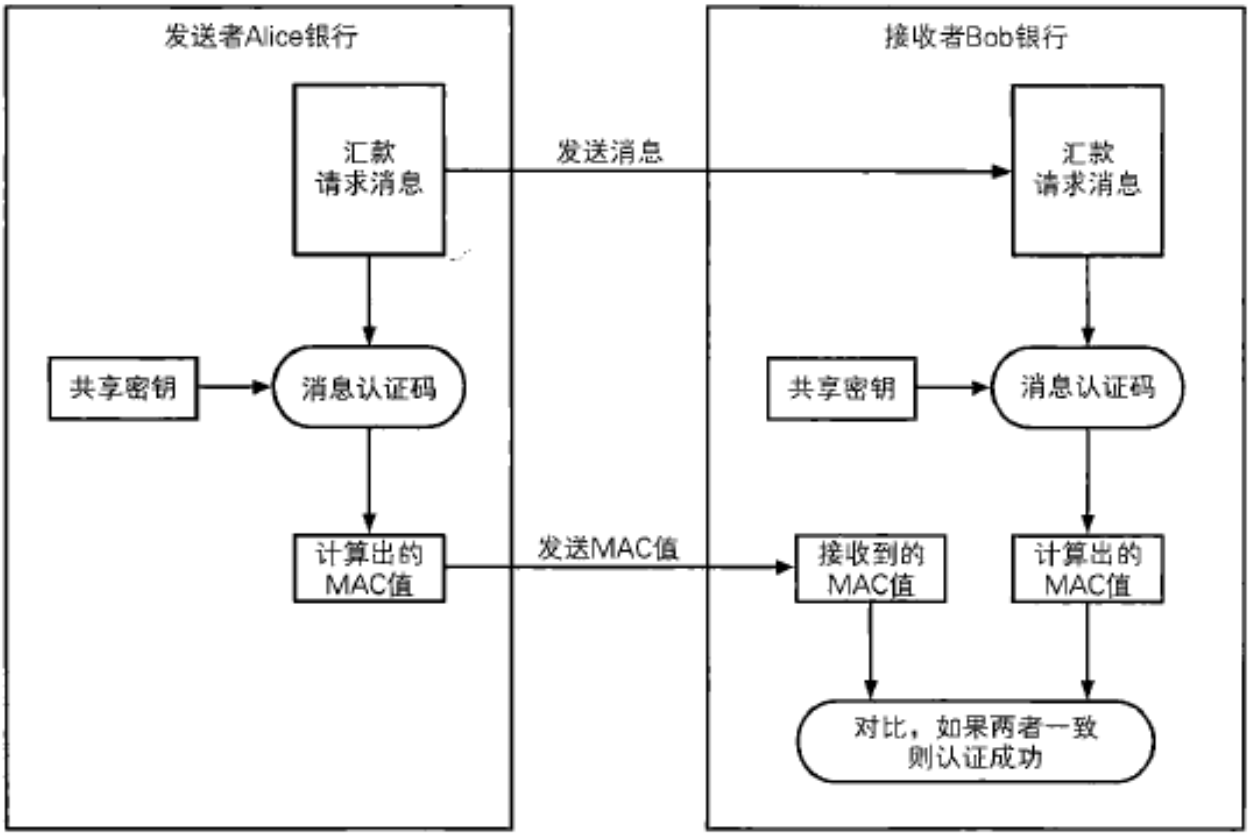


图 8-2　消息认证码的使用步骤

(1) 发送者 Alice 与接收者 Bob 事先共享密钥。

(2) 发送者 Alice 根据汇款请求消息计算 MAC 值（使用共享密钥）。

(3) 发送者 Alice 将汇款请求消息和 MAC 值两者发送给接收者 Bob。

(4) 接收者 Bob 根据接收到的汇款请求消息计算 MAC 值（使用共享密钥）。

(5) 接收者 Bob 将自己计算的 MAC 值与从 Alice 处收到的 MAC 值进行对比。

(6) 如果两个 MAC 值一致，则接收者 Bob 就可以断定汇款请求的确来自 Alice（认证成功）；如果不一致，则可以断定消息不是来自 Alice（认证失败）。

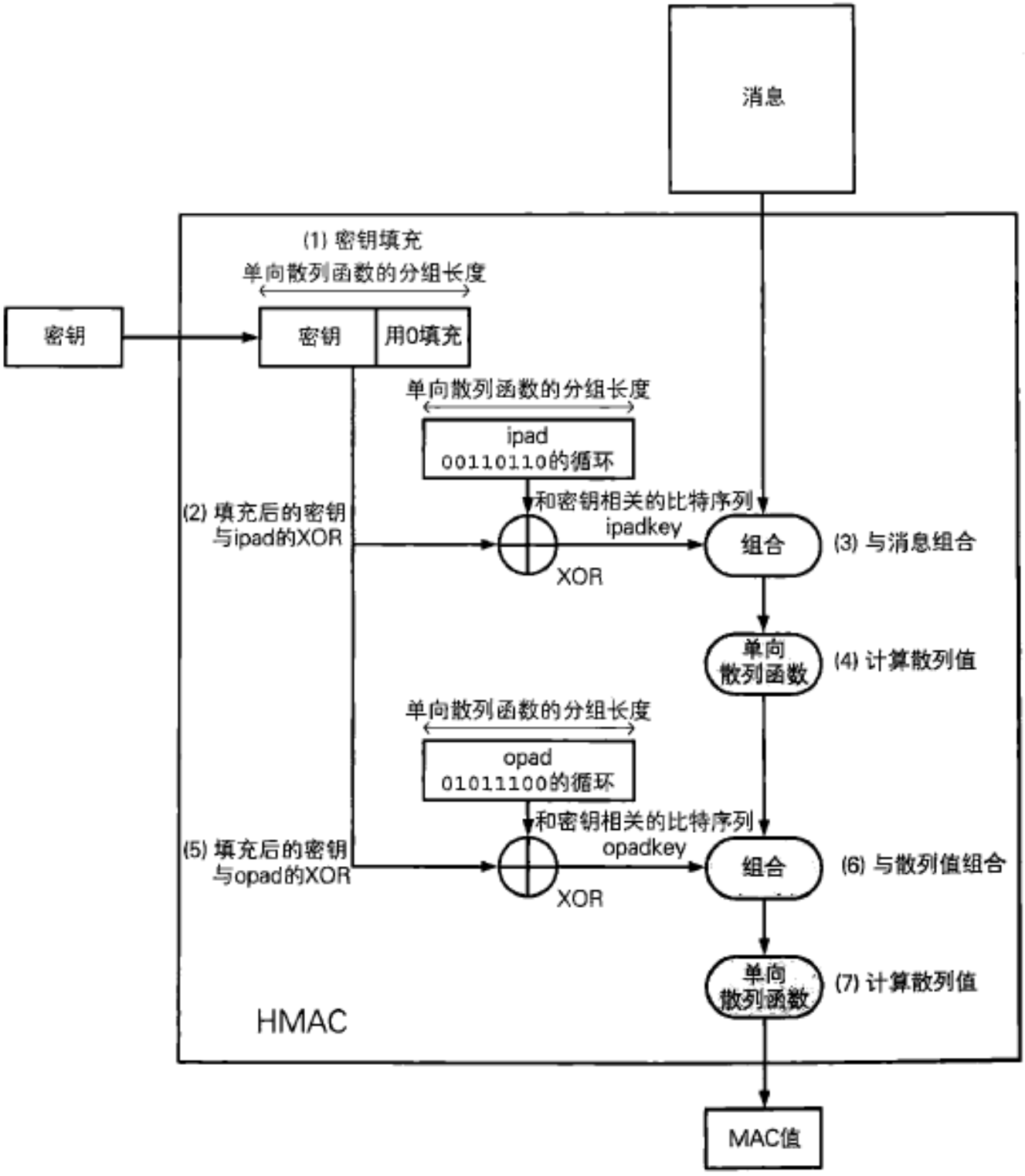


图 8-3　使用单向散列函数实现消息认证码的例子（HMAC）

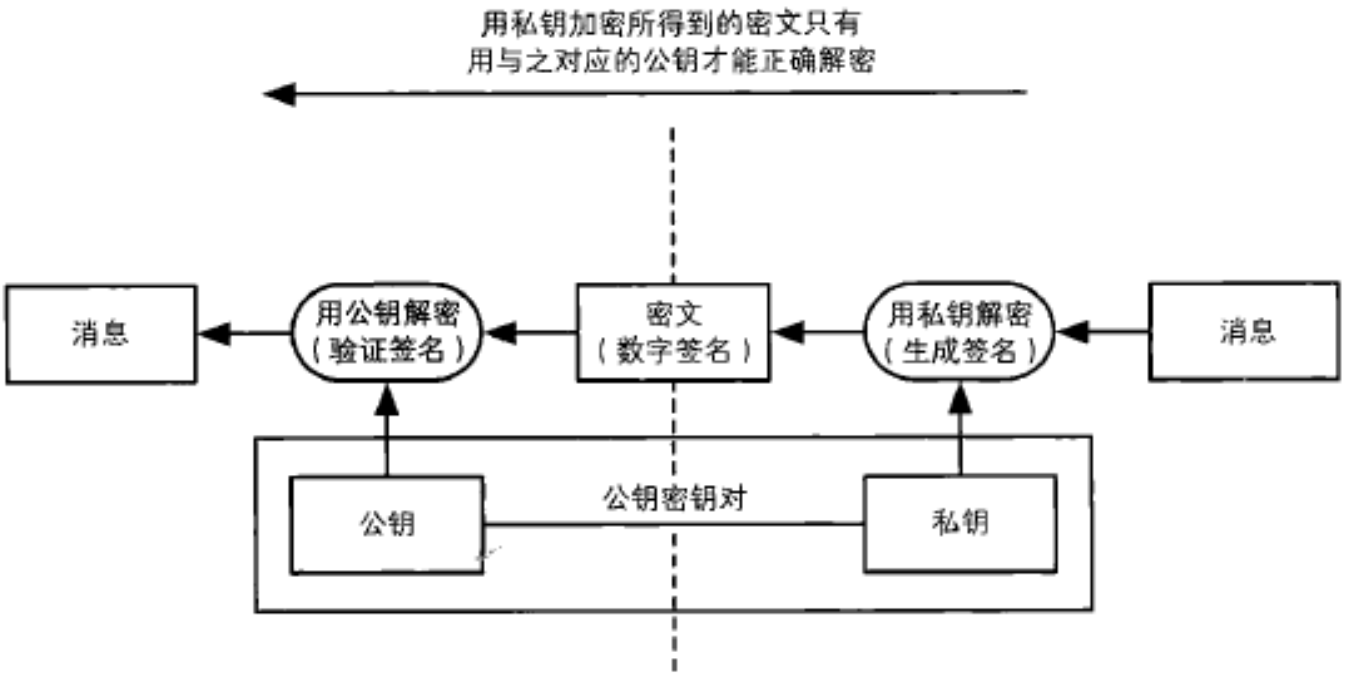


图 9-2　用私钥进行加密（数字签名）

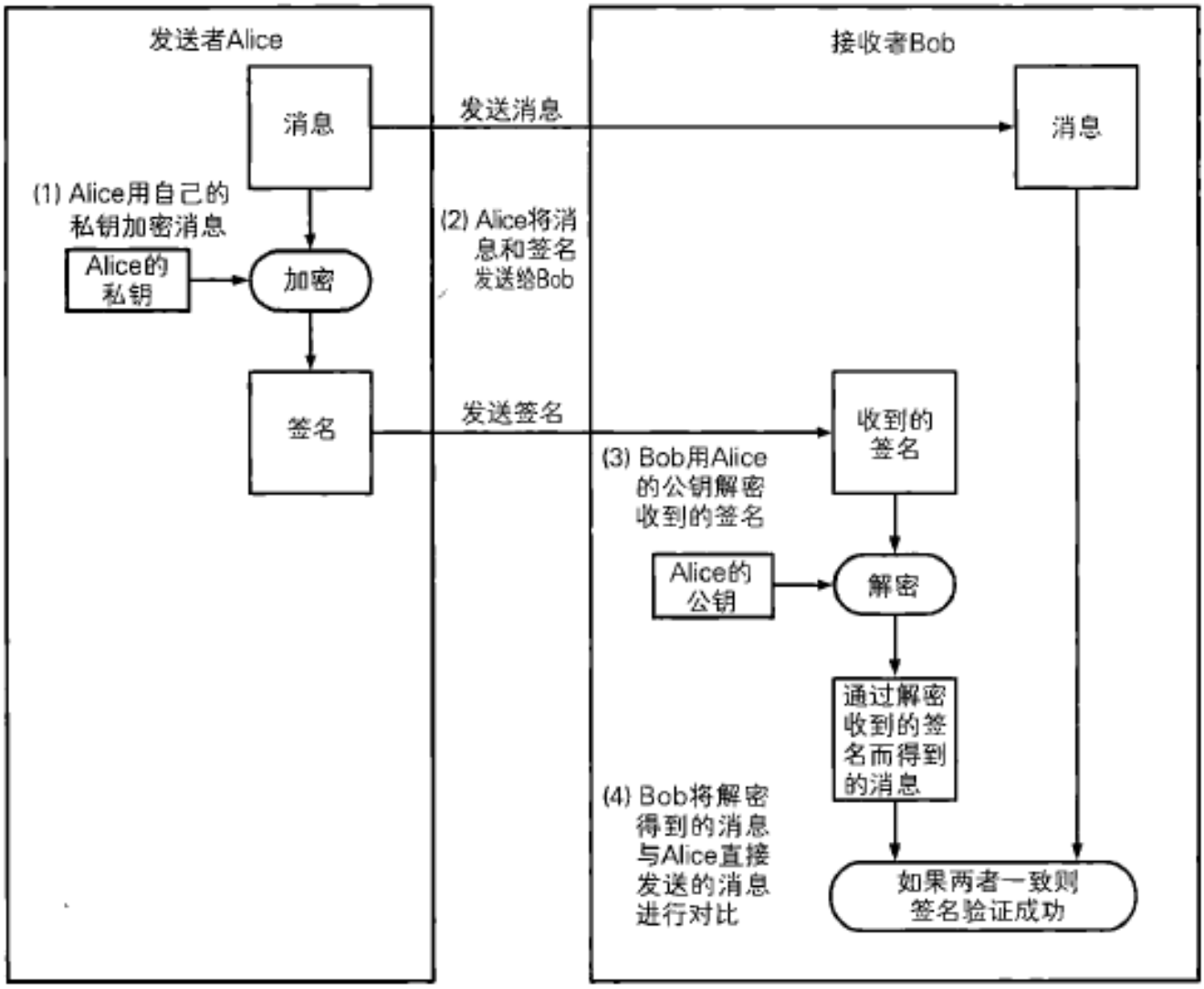


图 9-5　Alice 对消息签名，Bob 验证签名

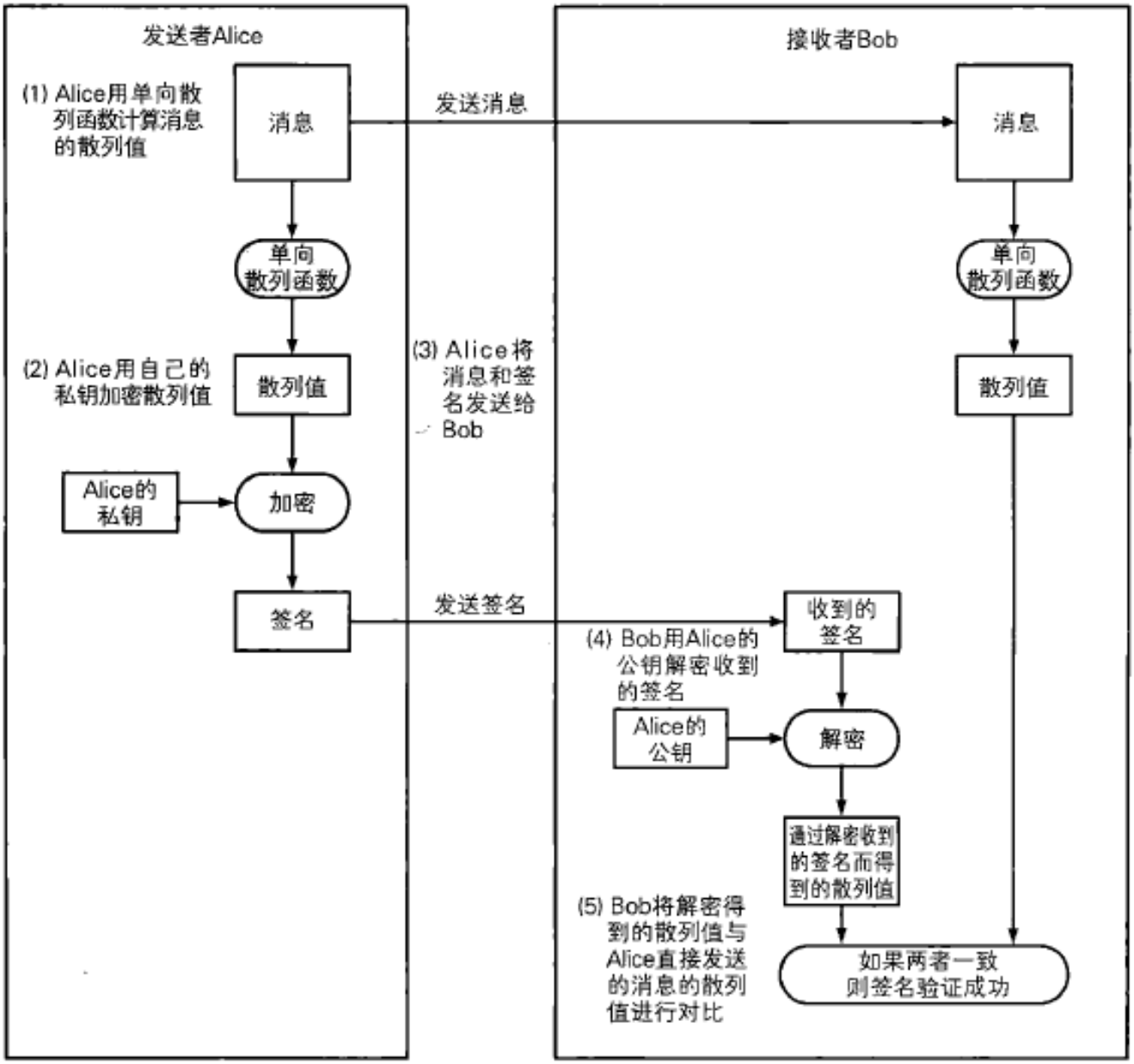


图 9-6　Alice 对消息的散列值签名，Bob 验证签名

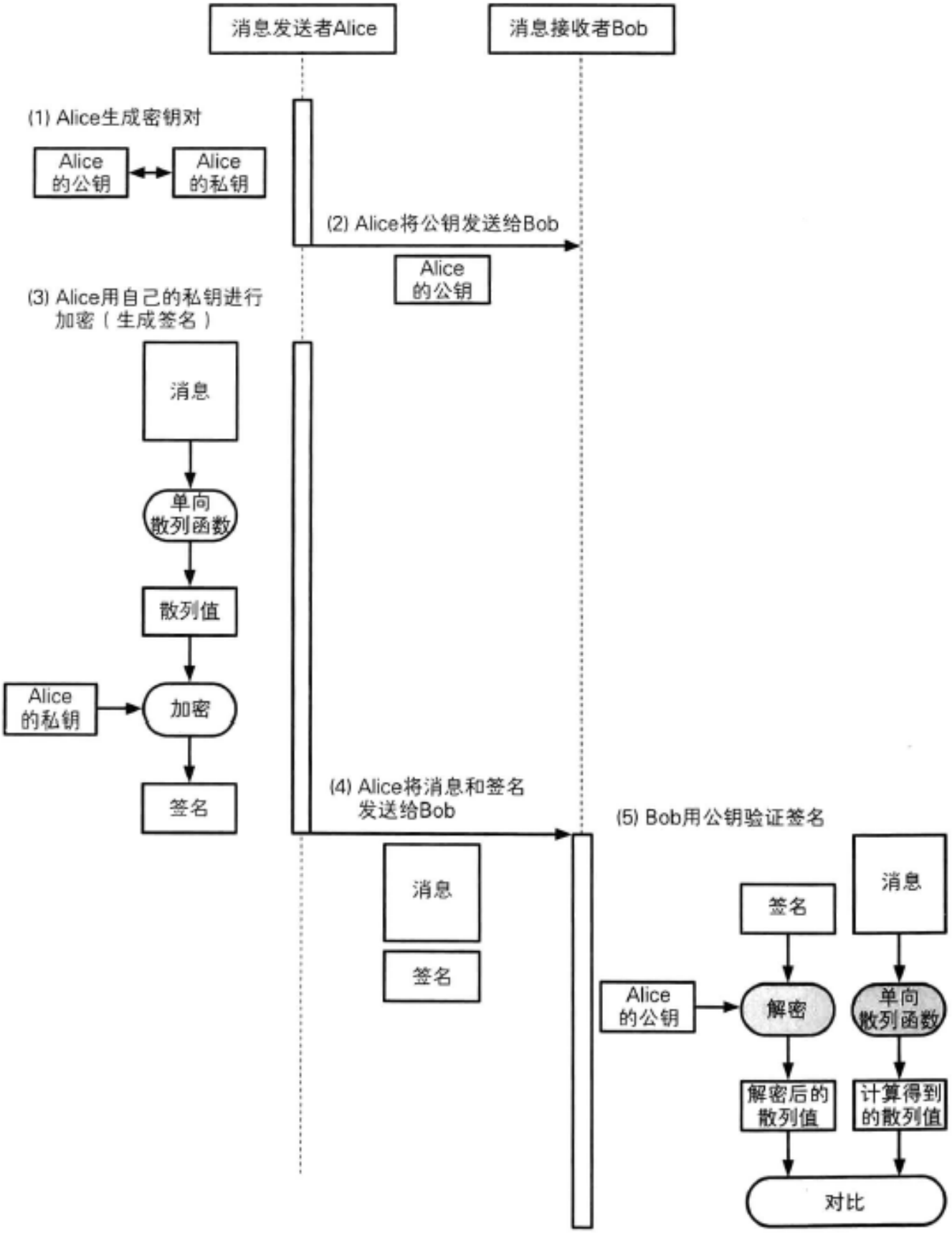


图 9-7　Alice 对消息的散列值签名，Bob 验证签名（按时间顺序）

表 9-2　RSA 的签名生成和验证

密钥对	**公钥**	数 E 和数 N
	私钥	数 D 和数 N
生成签名		签名 = 消息^{D} mod N（消息的 D 次方除以 N 的余数）
验证签名		由签名求得的消息 = 签名^{E} mod N（签名的 E 次方除以 N 的余数），将“由签名求得的消息”与“消息”进行对比

表 9-3　对称密码与公钥密码的对比，以及消息认证码与数字签名的对比

	对称密码	**公钥密码**
发送者	用共享密钥加密	用公钥加密
接收者	用共享密钥解密	用私钥解密
密钥配送问题	存在	不存在，但公钥需要另外认证
机密性	○	○

	消息认证码	**数字签名**
发送者	用共享密钥计算 MAC 值	用私钥生成签名
接收者	用共享密钥计算 MAC 值	用公钥验证签名
密钥配送问题	存在	不存在，但公钥需要另外认证
完整性	○	○
认证	○（仅限通信对象双方）	○（可适用于任何第三方）
防止否认	×	○

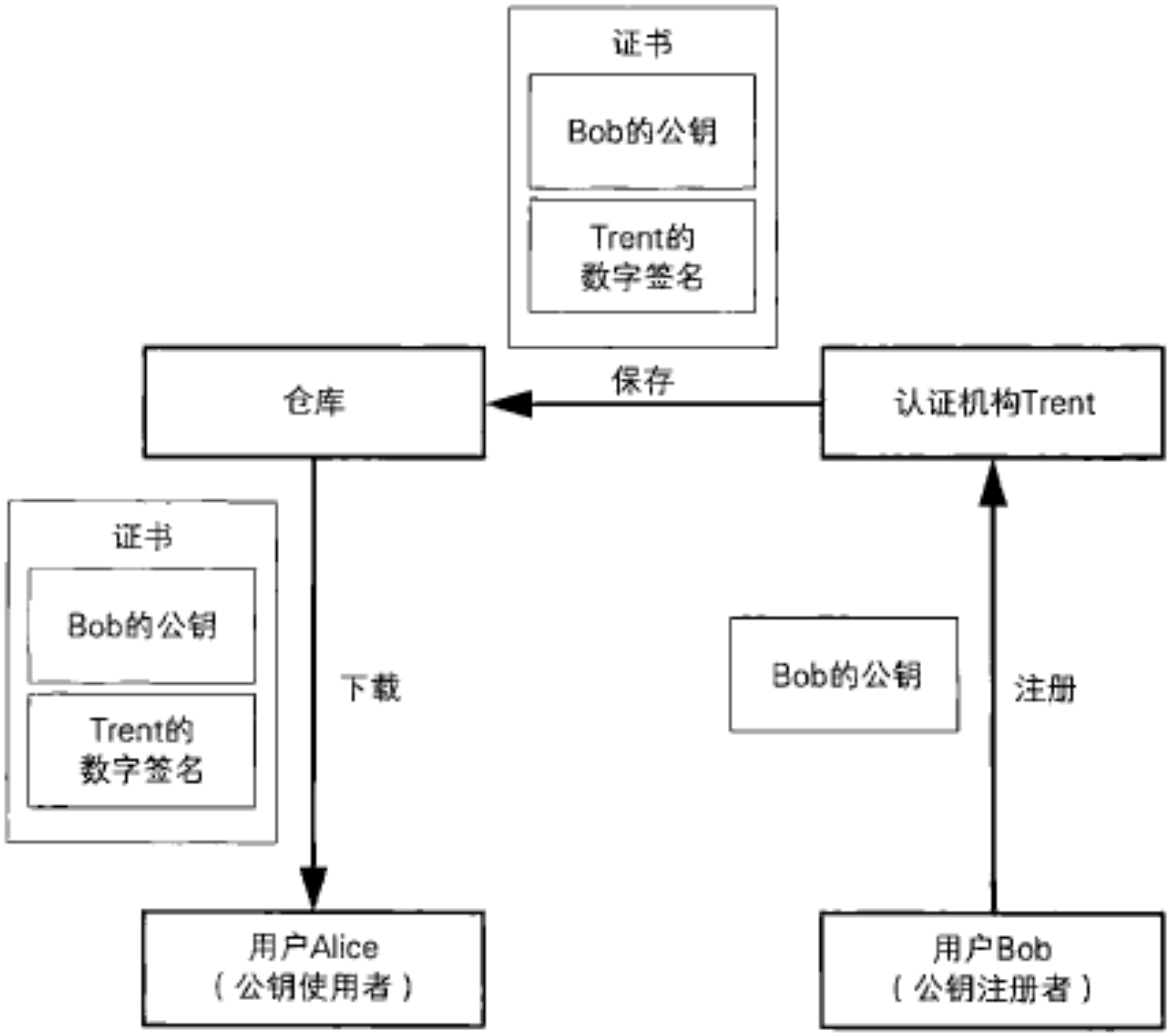


图 10-8　PKI 的组成要素

【注册公钥的用户所进行的操作】

- 生成密钥对（也可以由认证机构生成）
- 在认证机构注册公钥
- 向认证机构申请证书
- 根据需要申请作废已注册的公钥
- 解密接收到的密文
- 对消息进行数字签名

【使用已注册公钥的用户所进行的操作】

- 将消息加密后发送给接收者
- 验证数字签名

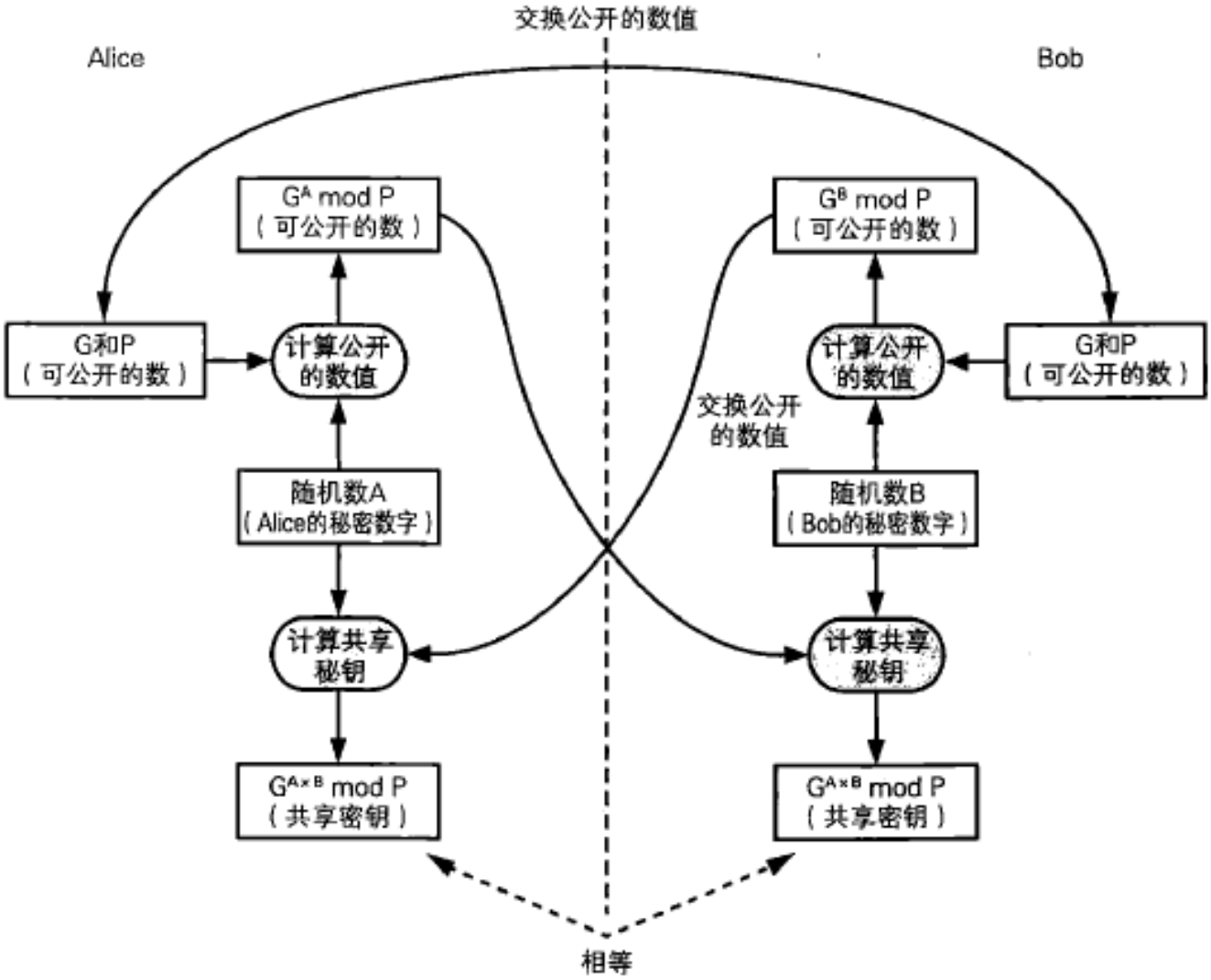


图 11-7　Diffie-Hellman 密钥交换

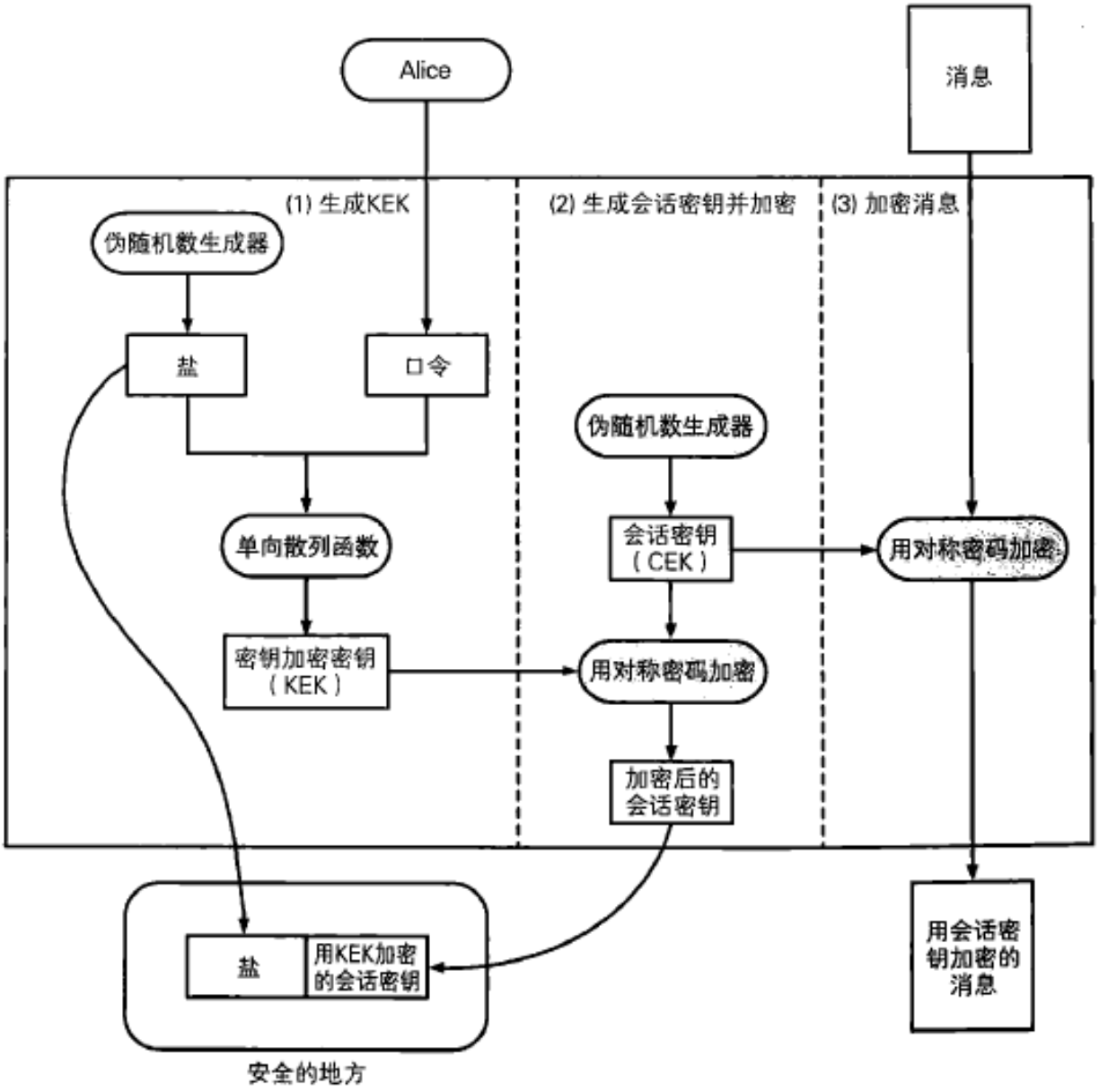

Alice
消息
(1) 生成KEK
(2) 生成会话密钥并加密
(3) 加密消息
伪随机数生成器
盐
口令
单向散列函数
密钥加密密钥
（KEK）
伪随机数生成器
会话密钥
（CEK）
用对称密码加密
用对称密码加密
加密后的
会话密钥
盐
用KEK加密
的会话密钥
安全的地方
用会话密
钥加密的
消息

图 11-8　PBE 加密

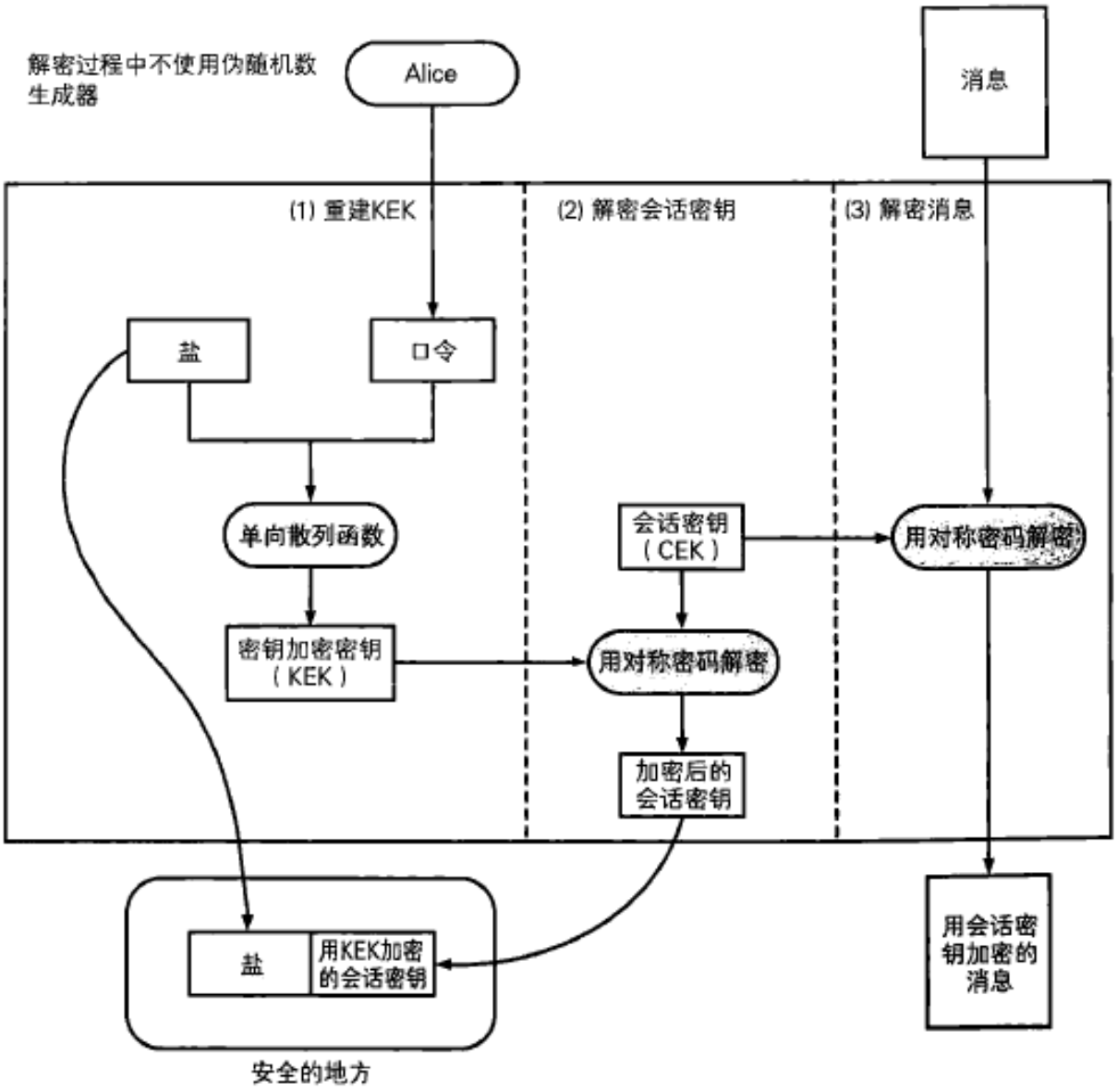

解密过程中不使用伪随机数生成器
Alice
消息
(1) 重建KEK
(2) 解密会话密钥
(3) 解密消息
盐
口令
单向散列函数
密钥加密密钥（KEK）
会话密钥（CEK）
用对称密码解密
用对称密码解密
加密后的会话密钥
盐
用KEK加密的会话密钥
安全的地方
用会话密钥加密的消息

图 11-9　PBE 解密

随机数作用：

生成密钥	用于对称密码和消息认证码
生成密钥对	用于公钥密钥和数字签名
生成初始化向量 IV	用于分组密码 CBC、CFB、OFB
生成 nonce	用于防御重放攻击以及分组密钥的 CTR 模式
生成盐值	用于基于口令的密码（PBE）

随机数性质：

随机性：不存在统计学偏差，完全杂乱的数列

不可预测性：不能从过去的数列推测出下一个出现的数

不可重现性：除非将数列本身保存下来，否则不能重现相同的数

表 12-1　随机数的分类

	随机性	不可预测性	不可重现性	
弱伪随机数	○	×	×	只具备随机性
强伪随机数	○	○	×	具备不可预测性
真随机数	○	○	○	具备不可重现性

↑ 不可用于密码技术

↓ 可用于密码技术

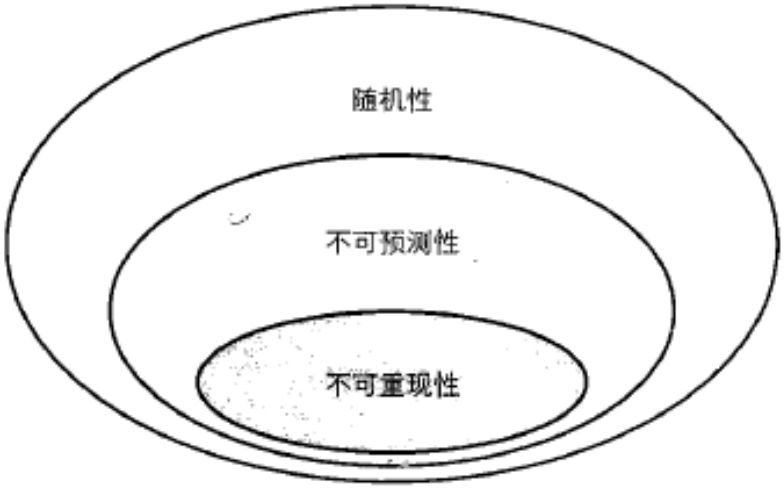


图 12-1　随机数的性质

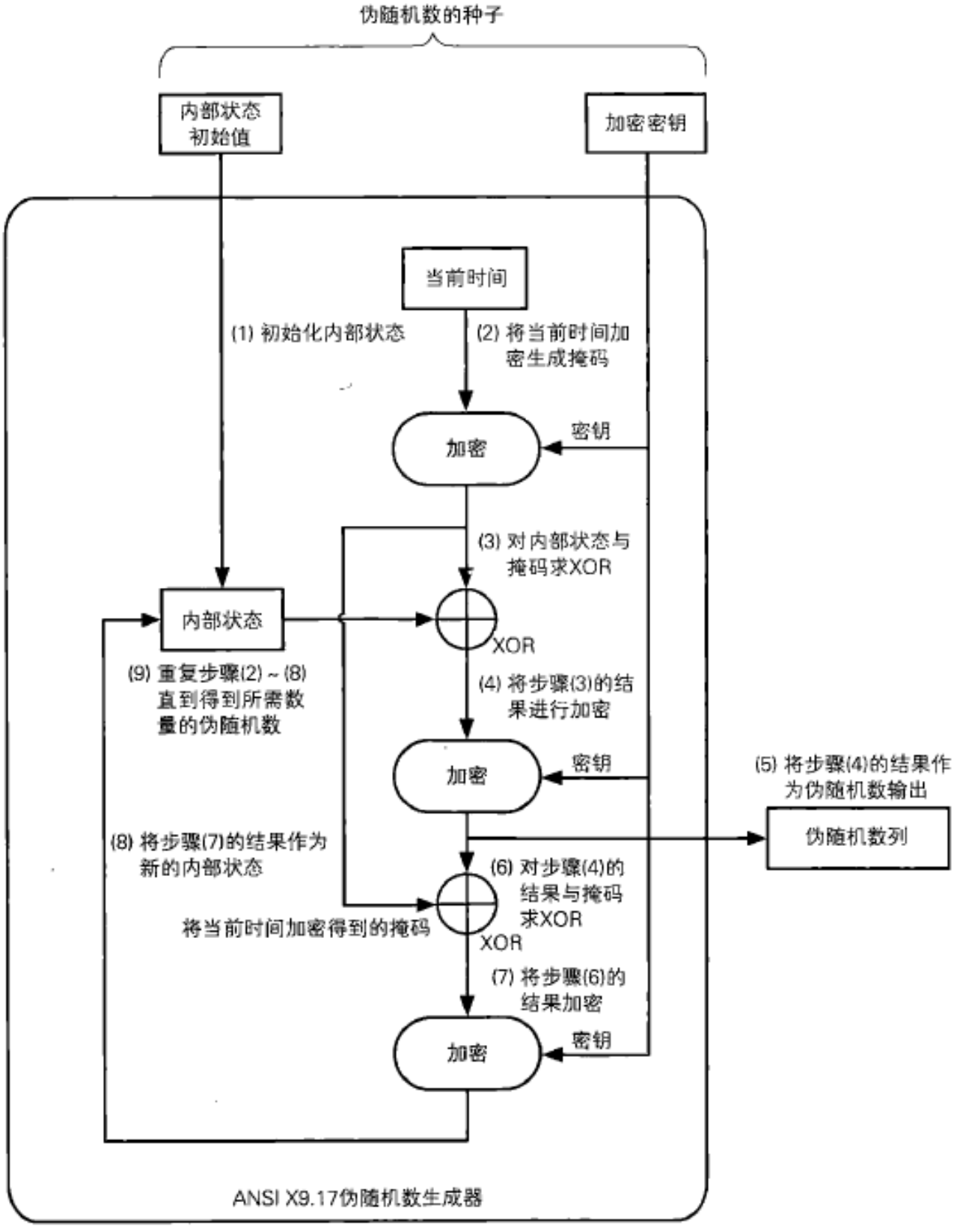


图 12-8　用 ANSI X9.17 方法实现伪随机数生成器

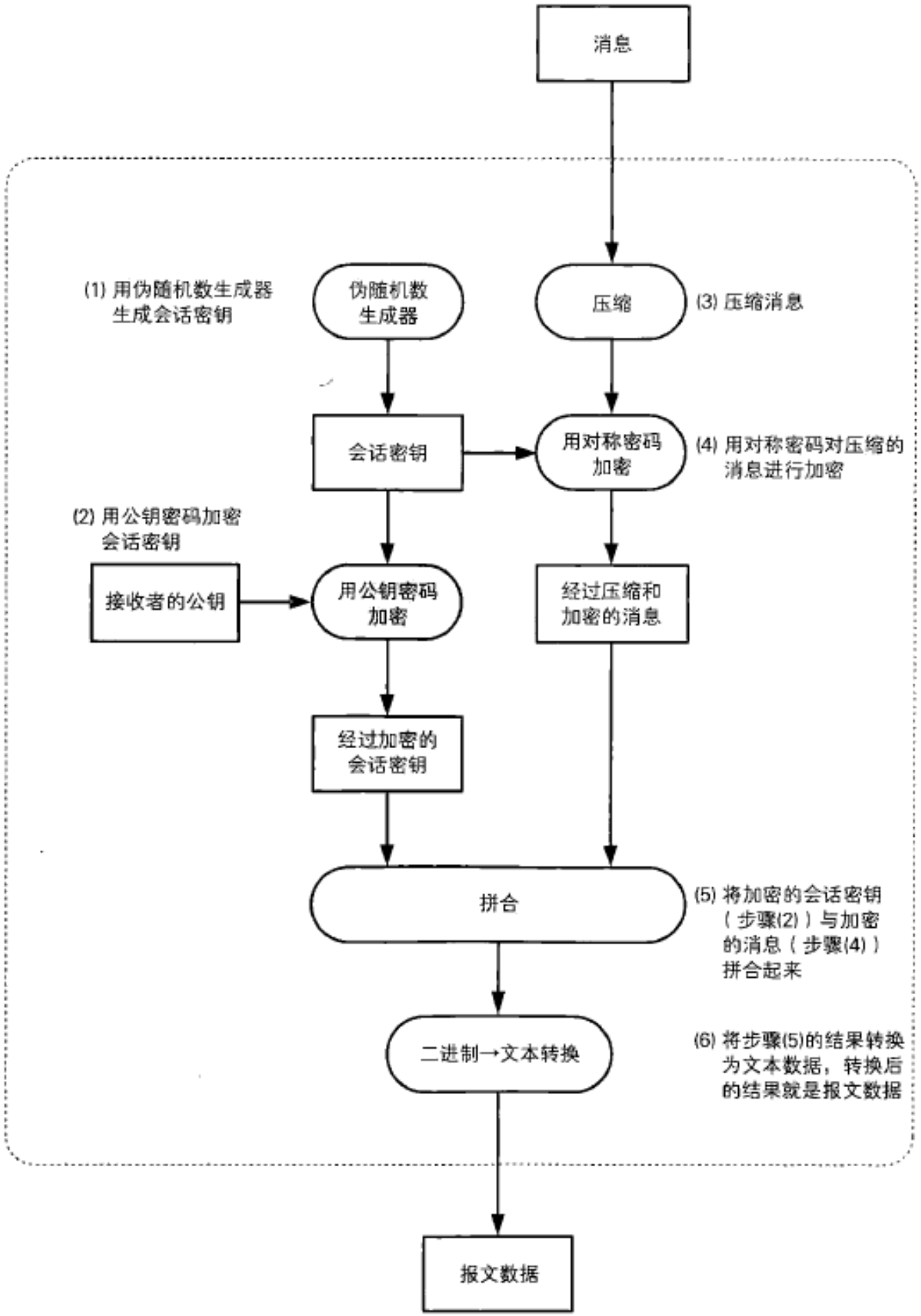

消息
(1) 用伪随机数生成器生成会话密钥
伪随机数生成器
压缩
(3) 压缩消息
会话密钥
用对称密码加密
(4) 用对称密码对压缩的消息进行加密
(2) 用公钥密码加密会话密钥
接收者的公钥
用公钥密码加密
经过压缩和加密的消息
经过加密的会话密钥
拼合
(5) 将加密的会话密钥（步骤(2)）与加密的消息（步骤(4)）拼合起来
二进制→文本转换
(6) 将步骤(5)的结果转换为文本数据，转换后的结果就是报文数据
报文数据

图 13-4　用 PGP 加密

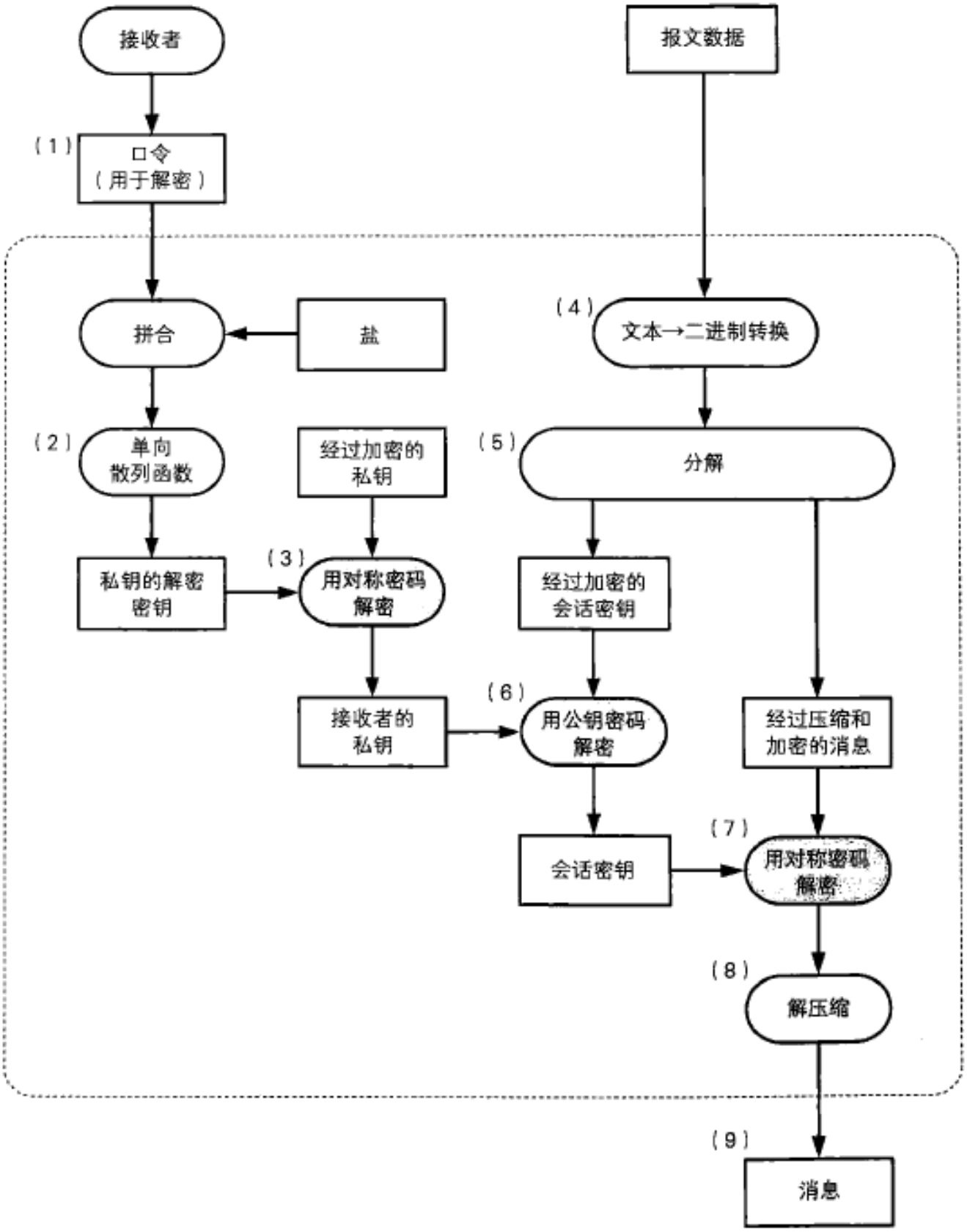


图 13-5　用 PGP 解密（各步骤的说明参见上文）

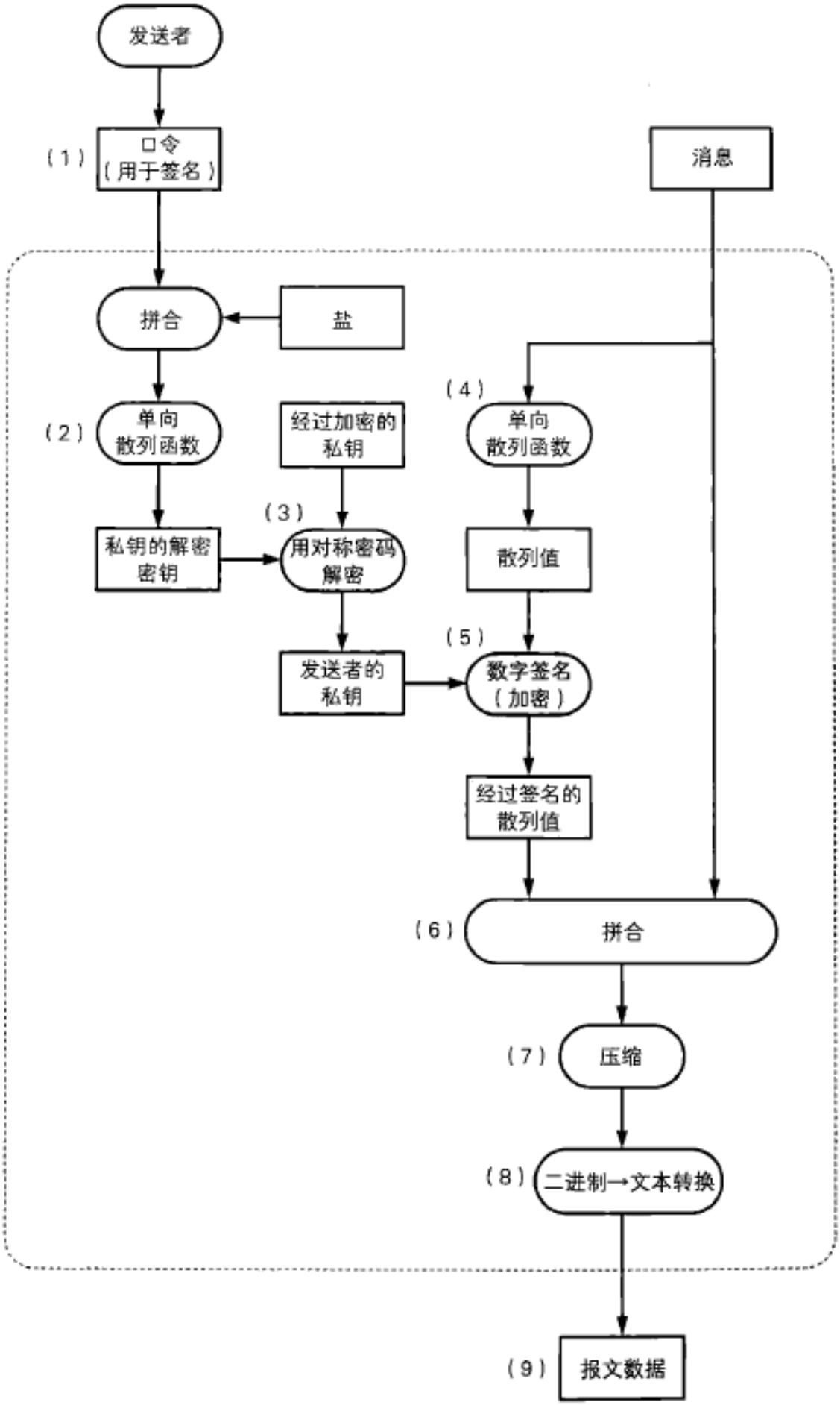


图 13-7　用 PGP 生成数字签名

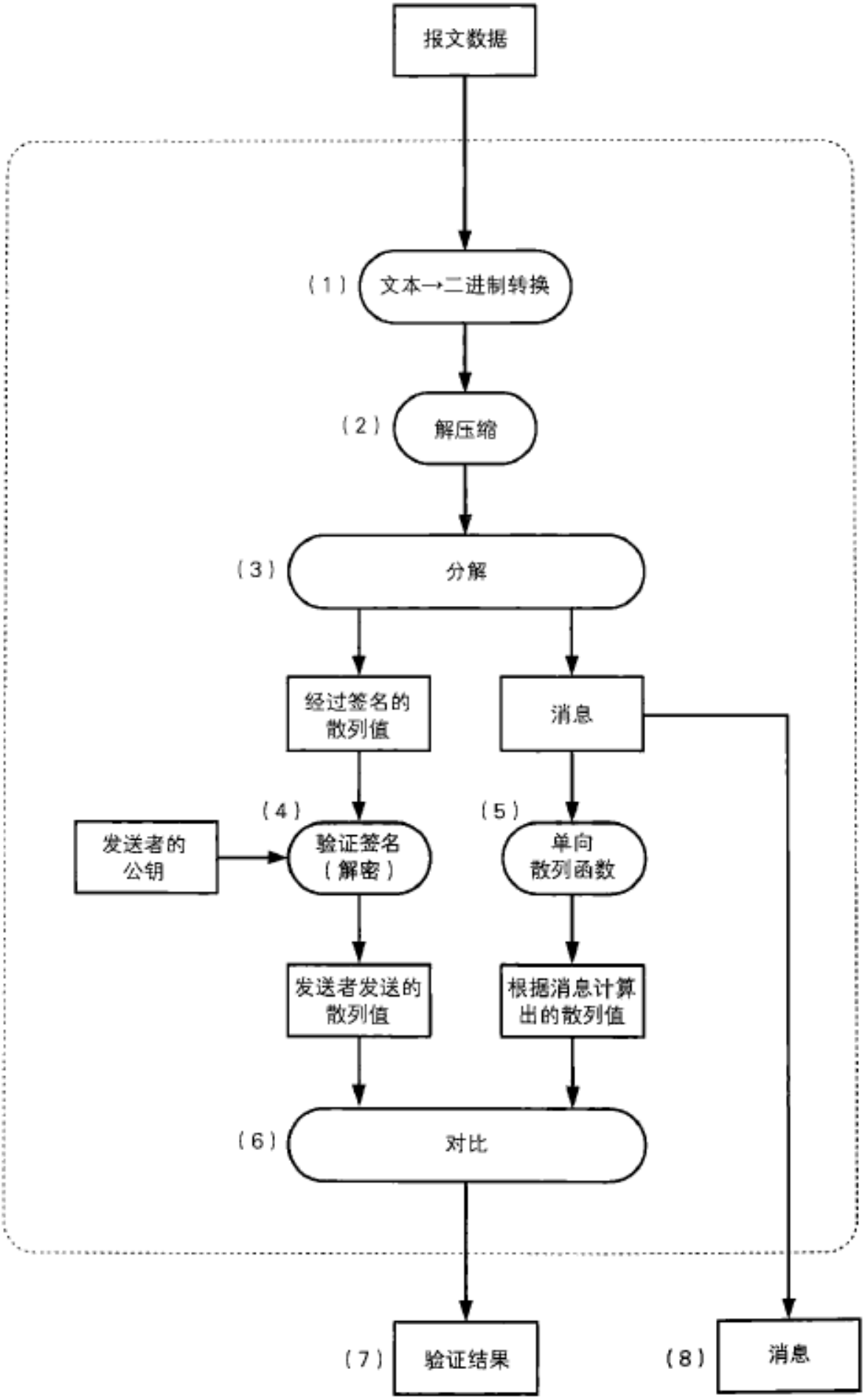


图 13-8　用 PGP 验证数字签名

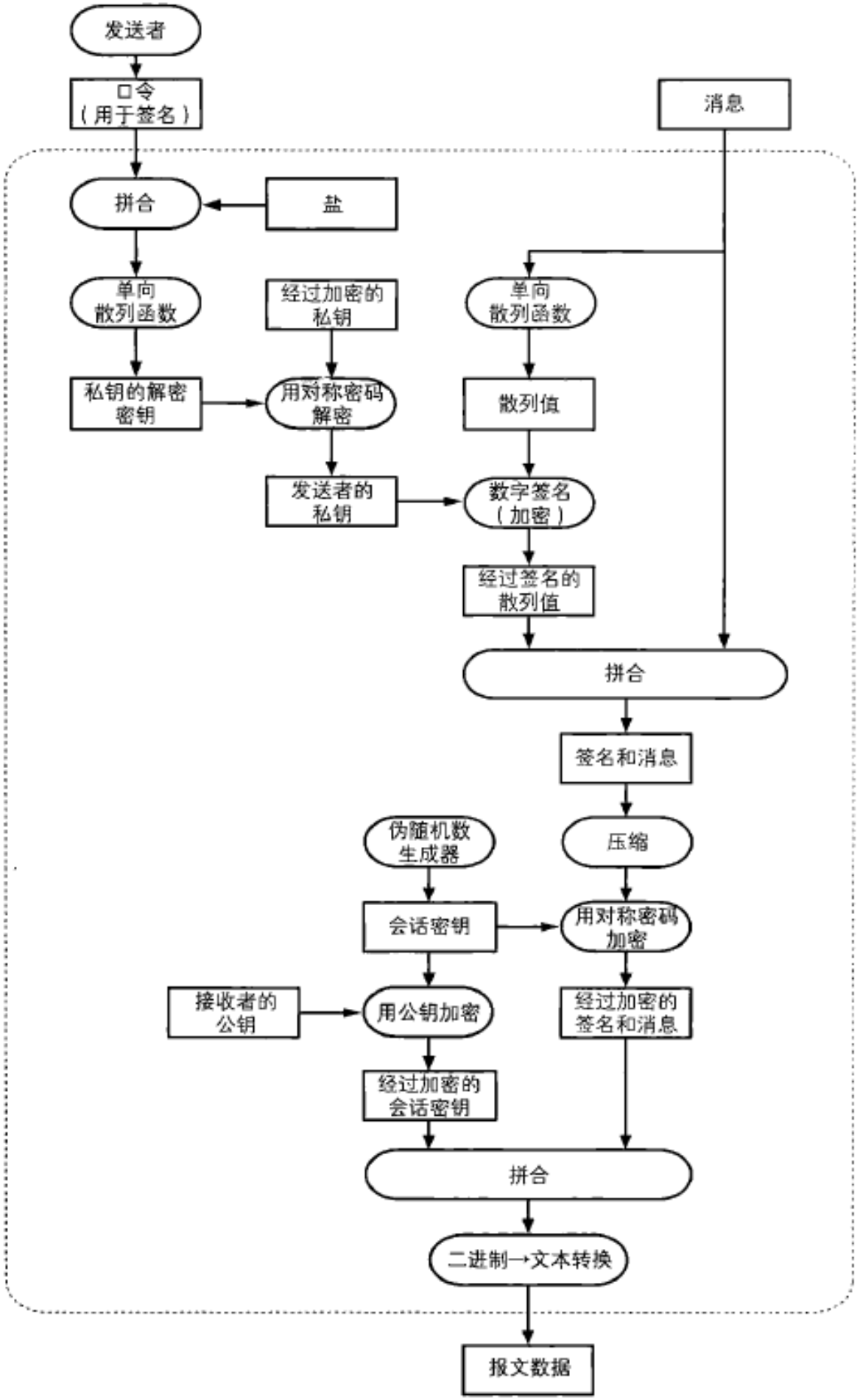


图 13-10　用 PGP 生成数字签名并加密

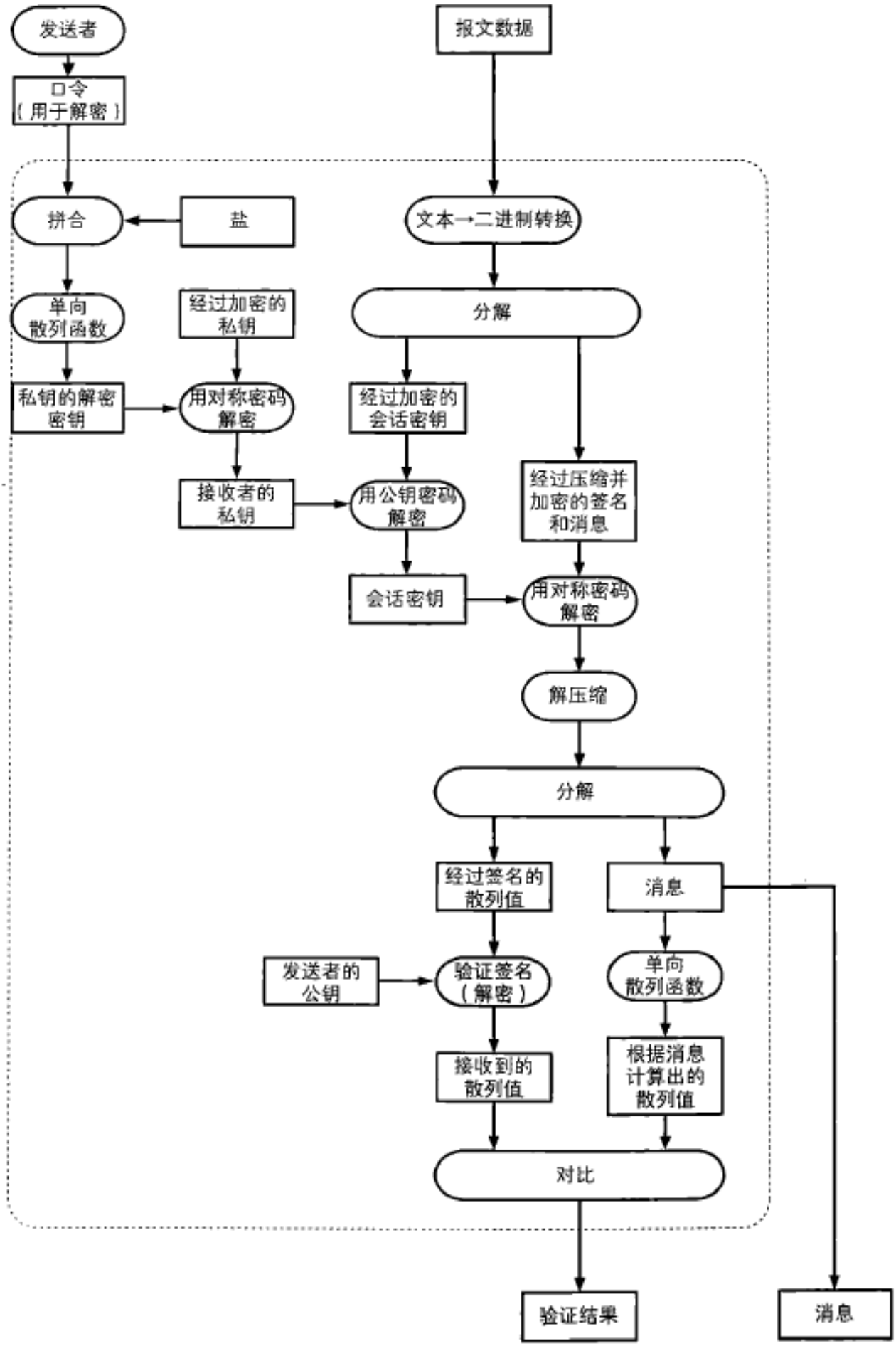


图 13-11　用 PGP 解密并验证数字签名

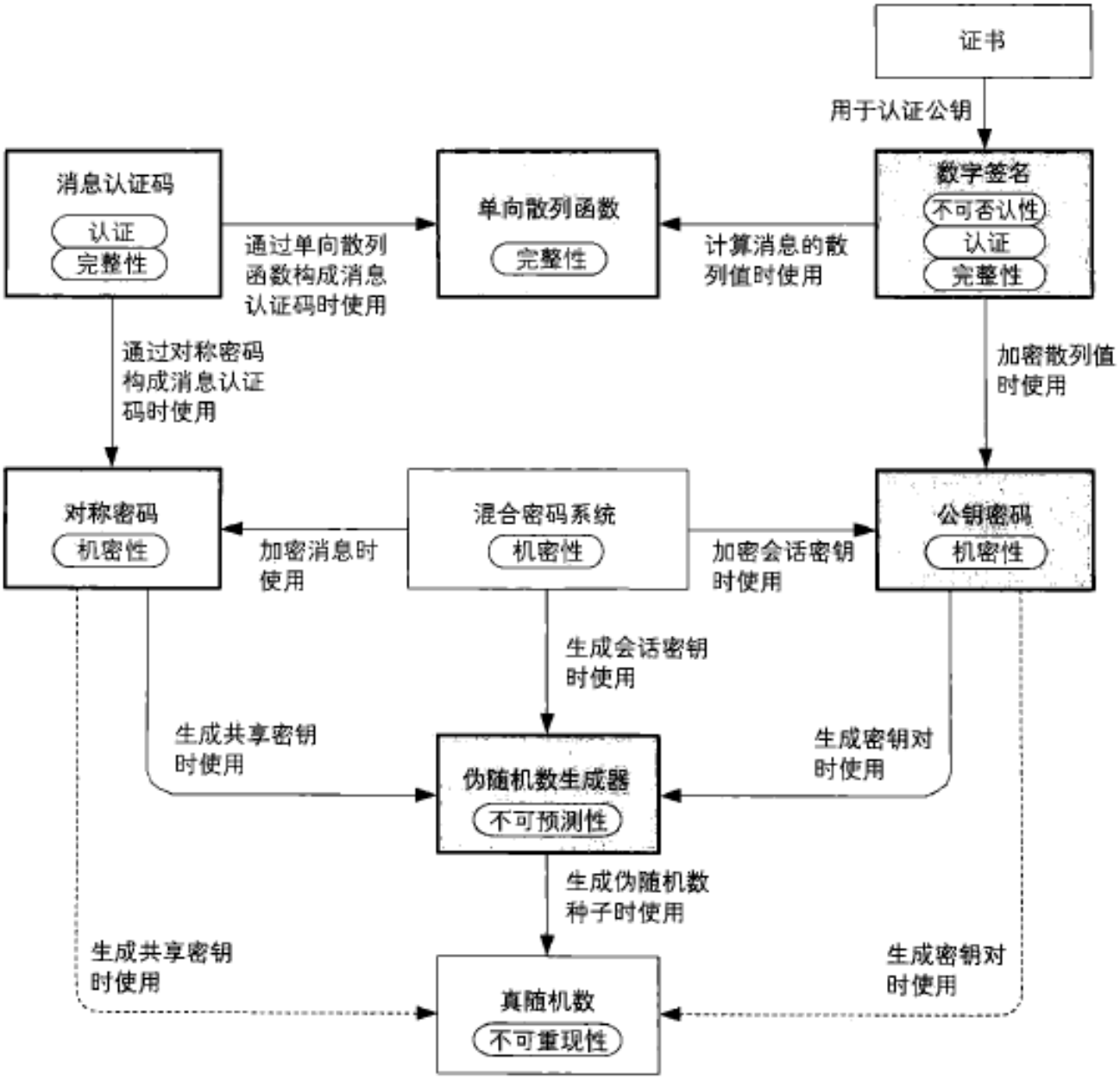


图 15-1　密码学家的工具箱

对称密码	DES	差分分析、线性分析	
	ECB 模式	交换分组密文	
	CBC 模式	1. 缺失某一密文分组导致之后的密文分组都受影响 2. 初始化向量比特反转 3. 填充式攻击 4. 攻击初始化 IV	
	CFB 模式	重放攻击	
公钥密码	RSA	1. 通过密文来求明文 2. 暴力破解求 D 3. 通过 E、N 求 D 4. 中间人攻击 5. 选择密文攻击	
	ElGamal		
	Rabin		

	椭圆曲线		
单项散列函数		暴力破解、冲突攻击	
消息认证码		重放攻击 密钥推测攻击	增加序号、时间戳等
数字签名		1. 中间人攻击 2. 攻击单项散列函数 3. 利用数字签名攻击公钥密码	
证书		攻击者伪装成认证机构攻击 钻 CRL 空子攻击（利用 CRL 发布时间差来攻击）	
伪随机数生成器		攻击种子 攻击随机数池	