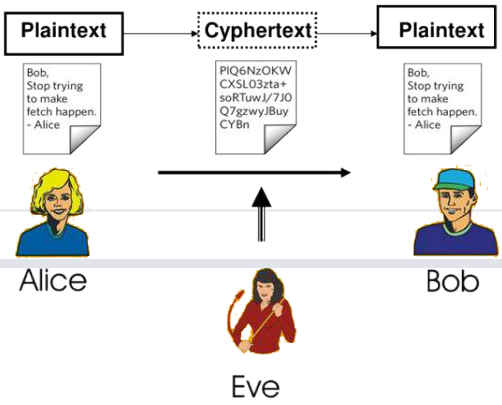
第12讲:信息安全和密码学



1.对称加密方法

对称加密,顾名思义,是指加密和解密使用同一个密钥的加密方法.这种方法的数学形式通常比较直接,但具体的算法可以有很多种变体,比如DES、AES、3DES等.以下是对称加密和解密过程的一般数学形式和一个简单算例:

1.1.一般数学形式

加密过程:设K是加密和解密所用的密钥,M是原始明文信息,C是加密后的密文,加密过程可以表示为:

$$C = E_K(M)$$

这里的E表示加密算法,它将密钥K和明文M作为输入,输出密文C.

解密过程:使用同样的密钥K对密文C进行解密,得到原始明文:

$$M = D_K(C)$$

这里的D表示解密算法,它将密钥K和密文C作为输入,输出原文M.在对称加密中,D是E的逆运算.

1.2.算例:使用仿射加密算法

假设我们使用一个简单的仿射加密算法进行示例,其数学形式如下:

设$K = (k_1, k_2)$是一对密钥,其中k_1和k_2是正整数.

加密函数:

$$C = (k_1 \cdot M + k_2) \mod n$$

这里n是模数,通常选择一个比字符集大的数,例如字符集的大小.

解密函数:

为了解密,我们需要找到k_1的乘法逆元k_1^{-1},满足$k_1 \cdot k_1^{-1} \equiv 1 \mod n$.

$$M = (k_1^{-1} \cdot (C - k_2)) \mod n$$

具体例子:

假设我们的字符集大小为 26(英文字母),$n = 26$,并选择$k_1 = 5$和$k_2 = 3$,那么$k_1^{-1} = 21$因为$5 \cdot 21 \equiv 1 \mod 26$.

为了简化例子,我们假设字符已经转换成了数字,比如 'A' = 0, 'B' = 1,以此类推,'Z' = 25.

如果我们加密字母 'B'(即明文$M = 1$):

$$C = (5 \times 1 + 3) \mod 26 = 8$$

加密后的结果是 'I',对应的数字是8.

为了解密 'I':

$$M = (21 \times (8 - 3)) \mod 26 = 21 \times 5 \mod 26 = 1$$

解密后我们得到1,也就是 'B'.

这个算例展示的是一个非常简化的对称加密和解密过程.但在实际应用中,使用的对称加密算法(例如AES)会更复杂,涉及多轮的替换和置换,以及多维数组和位操作,这是为了确保加密安全.在诸如AES这样的现代算法中,密钥K的生成和应用会更加复杂,以确保高强度的安全性.

2.Diffie-Hellman 密钥交换算法

实际上,Diffie-Hellman算法并不直接涉及加密和解密过程,而是一种密钥交换协议.它允许双方在完全不安全的通信渠道上产生一个共享的安全密钥,这个密钥可以用于后续的加密通信,但Diffie-Hellman协议本身并没有定义如何使用这个密钥来加密数据.接下来详细解释Diffie-Hellman密钥交换的过程:

2.1.密钥交换步骤

1. 双方约定一个大质数p和基数g,其中g是p的一个原根.这两个数是公开的.
2. Alice选择一个大的随机数a,它作为Alice的私钥,但保密不公开.
3. Alice计算$A = g^a \mod p$并将结果A发送给Bob.A称为Alice的公开值.
4. Bob选择一个大的随机数b,它作为Bob的私钥,但保密不公开.
5. Bob计算$B = g^b \mod p$并将结果B发送给Alice.B称为Bob的公开值.
6. Alice接收到B,利用自己的私钥a,计算$s = B^a \mod p$.这个s就是双方共享的密钥.
7. Bob接收到A,利用自己的私钥b,计算$s = A^b \mod p$.这个s也是双方共享的密钥.

由于$A^b = (g^a)^b = g^{ab}$和$B^a = (g^b)^a = g^{ab}$,因此无论是Alice还是Bob,他们计算出的s都是相同的.即$s = g^{ab} \mod p$.

2.2.示例

假设公开的质数$p = 23$,基数$g = 5$:

1. Alice选择私钥$a = 6$,计算$A = 5^6 \mod 23 = 8$并将$A = 8$发送给Bob.
2. Bob选择私钥$b = 15$,计算$B = 5^{15} \mod 23 = 19$并将$B = 19$发送给Alice.
3. Alice收到$B = 19$,计算共享密钥$s = B^a \mod p = 19^6 \mod 23 = 2$.
4. Bob收到$A = 8$,计算共享密钥$s = A^b \mod p = 8^{15} \mod 23 = 2$.

最终,Alice和Bob得到的共享密钥s都是2.这个密钥可以用于后续的通信加密,但Diffie-Hellman算法本身不定义具体如何加密数据.

需要注意的是,尽管Diffie-Hellman密钥交换算法可以安全地共享密钥,但它没有解决中间人攻击的问题.如果有攻击者可以拦截并修改通信,他们可以对两方各自执行密钥交换步骤,从而与两方各自建立共享密钥,而通信双方并不知情.

3.RSA 加密算法

RSA算法是一种非常著名的非对称加密算法,它依赖于大整数的因数分解问题,以下将详细解释RSA算法的加密和解密过程:

3.1.密钥生成

1. 随机选择两个大的质数p和q,且一般它们的位数相近,以确保安全性.
2. 计算p和q的乘积$n = p \cdot q$,n的长度(比特数)决定了密钥的长度.
3. 计算n的欧拉函数$\phi(n)$,对于质数p和q,$\phi(n) = (p - 1) \cdot (q - 1)$.
4. 选择一个整数e,作为公钥指数,它满足$1 < e < \phi(n)$且e与$\phi(n)$互质.常用的值有3或65537.
5. 计算e关于$\phi(n)$的模逆元d,即$d \cdot e \equiv 1 \mod \phi(n)$.
6. 生成的公钥为一个对(n, e),私钥为d.

3.2.加密过程

假设明文为M,M是一个数字,且$0 \leq M < n$.

使用接收方的公钥(n, e)将明文M加密为密文C:

$$C = M^e \mod n$$

加密后,将密文C发送给接收方.

3.3.解密过程

接收方已经得到了密文C,并拥有私钥d.

使用私钥d来将密文C解密回明文M:

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

3.4.实例

假设小明选择了$p = 61$和$q = 53$,则:

1.$n = p \cdot q = 61 \cdot 53 = 3233$;

2.$\phi(n) = (p-1) \cdot (q-1) = 60 \cdot 52 = 3120$;

3.选择$e = 17$(它与 3120 互质);

4.计算d:$d = 2753$因为$17 \cdot 2753 \equiv 1 \mod 3120$;

小明的公钥是$(n = 3233, e = 17)$,私钥是$d = 2753$.如果要加密消息$M = 65$:

$$C = 65^{17} \mod 3233 = 2790$$

接收方收到密文$C = 2790$后,使用私钥$d = 2753$解密:

$$M = 2790^{2753} \mod 3233 = 65$$

就这样,RSA实现了安全的消息加密和解密.避免数学符号歧义的关键是在于明确地表达每个步骤和每个符号所代表的含义,同时保证过程中的数学运算正确无误.