第 5 章

双(公)钥密码体制

电子课件

密码学可以用来保护在公开信道上的信息传输。人们在使用 QQ、微信、电子邮件及网页与朋友进行数据交互,或访问某些特殊信息时,离不开密码学,确保信息没有被第三方窃听和攻击,或在传输的过程中没有被篡改,直观的方式是采用对称密码体制。但是理论和实践证实该体制存在以下缺点。

1. 密钥管理低效

采用对称密码体制来保护用户之间的通信,每个用户需要与其余的 $N-1$ 个用户共享密钥,整个系统需要管理 $N(N-1)/2$ 个密钥,当用户非常大时,这种密码体制是非常低效的。

2. 密钥分发与安全通信矛盾

使用对称密码体制进行通信,通信双方需要提前共享一个秘密信息,这与安全通信在某种意义上来说存在着一个矛盾。安全通信的目的是为了传输秘密信息,而该体制在执行安全通信的前提则需要共享一个秘密信息,这不正好是一个矛盾吗?尽管密钥分发可以采用面对面交换信息、物理接近、可信"快递",或者利用第三方来传递这个信息,但毕竟存在密钥分发这一特殊的缺陷。

3. 不支持"开放系统"

对称密码体制进行通信,通信双方是建立在彼此信任的基础上的。如果两个没有预先建立关系的用户需要建立安全通信时,他们是没有办法共享一个密钥的。

传统的对称密码体制无法解决上述问题,限制了它的应用范围。

随着密码学技术的发展,"公钥密码"这一概念诞生。其基本思想如图 5.1 所示。

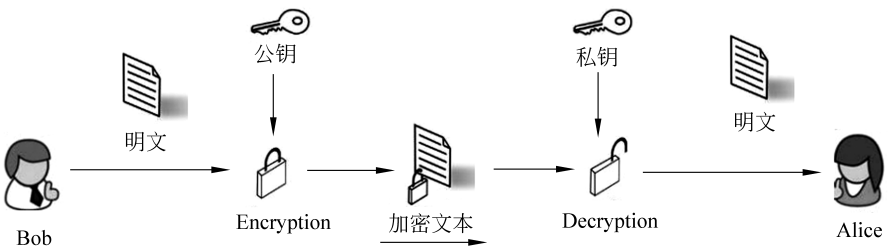

图 5.1　公钥密码体制

每个用户生成一个密钥对:一个是公钥 PK,另一个是与之对应的私钥 SK。公钥将在系统内被公开,私钥由用户本人安全保管。私钥由用户本人使用,而公钥则由系统中其他用户使用。由于通信双方使用的不是同一个密钥,所以公钥密码体制也被称为非对称密码体制。

图 5.1 中,Bob 利用 Alice 的公钥 PK 加密信息,在信道上传输的是加密的信息,Alice

收到这个加密的信息后，用自己的私钥 SK 解密这个信息，得到明文。在这样的一个通信环境里，他们不需要共享一个相同的秘密才能够执行加解密，非常好地解决了对称密码体制中固有的 3 个不同的缺点。

公钥密码体制中，公钥能够采用公开（认证的）信道进行传输；密钥管理得到了极大的简化。在用户为 N 个用户的系统中，每个用户只需安全保管自己的私钥和 $N-1$ 个其他用户的公钥。整个系统仅仅需要维护 N 个公钥；开放系统，即使是没有预先建立关系的用户也能通过对方的公钥建立。

“公钥密码”体制解决了对称密码体制的几个非常大的问题，是密码学发展史上的一个重大革命，扩展密码学的应用范畴。若用公钥作为加密密钥，以用户私钥作为解密密钥，则可实现多对一的关系，即多个用户加密的消息，而只能被一个用户解密；若用用户私钥作为加密密钥，而以公开密钥作为解密密钥，就可实现由一个用户加密的消息，被多个用户解密，即一对多的关系。前者可用于保密通信，后者可用于数字签名。这就说明“公钥密码”体制为解决计算机信息网络中的安全提供了新的理论和技术基础。在电子商务、电子政务以及人们的日常生活中，已经离不开公钥密码体制的应用。掌握“公钥密码”体制工作流程、工作原理以及相关的知识非常重要，本章将讲述双钥体制的基本原理和 RSA、ElGamal、椭圆曲线等密码算法。

5.1　公钥密码体制的数学基础[15]

在公钥密码体制中，应用最广泛、影响最深远、同时也是第一个可行的公钥加密方案，即 RSA 加密方案。随着密码学的发展，还涌现出很多公钥加密算法，如 ElGamal、椭圆曲线等密码算法。公钥密码体制涉及很多数学知识，所以在正式学习公钥加密方案前，需要对相关的数学机制进行必要的学习。本节对完全剩余系、简化剩余系、欧拉定理和单向函数等展开学习。这些数学概念一般比较抽象，本小节在讲述过程中，不详细从纯数学的角度去定义、去分析，而是用具体的例子来讲明其概念，在后续讲解 RSA 等算法时，明白其机理，搞定每种算法的理论基础即可。

1. 剩余类

在学习完全剩余系前，首先要掌握的一个基本概念，即剩余类。

什么是剩余类？大家知道不同的数有不同的划分方式，所以才有了剩余类（residue class）的说法。剩余类是对整数的划分，是关于 m 的剩余类。

设 $m\in Z^+$，$C_r=\{qm+r\mid q\in Z\}$，$r=0,1,2,\cdots,m-1$，Z 是所有的正整数。

这个公式是什么意思呢？

先看个实例：求模 5 的剩余类（即 $m=5$）。由上式可以得出：

$C_0=\{5q+0\mid q\in Z\}$　若 $q=0,-1,1,2,\cdots$，则可得到 $C_0=0,-5,5,10,\cdots$；

$C_1=\{5q+1\mid q\in Z\}$　若 $q=-1,0,1,2,\cdots$，则可得到 $C_1=-4,1,6,11,\cdots$；

$C_2=\{5q+2\mid q\in Z\}$　若 $q=0,1,2,3,\cdots$，则可得到 $C_2=2,7,12,17,\cdots$；

$C_3=\{5q+3\mid q\in Z\}$　若 $q=0,1,2,3,\cdots$，则可得到 $C_3=3,8,13,18,\cdots$；

$C_4=\{5q+4\mid q\in Z\}$　若 $q=0,1,2,3,\cdots$，则可得到 $C_4=4,9,14,19,\cdots$。

这里 Z 是整数，包括正整数和负整数。$q\in Z$，所以 q 可以取正整数，也可以取负整数。

由此就可以设置公式 C_0,C_1,C_2,C_3,C_4中不同的 q 值,计算出相应的输出值(每个公式的后边就是计算的结果)。这些计算出来的结果,如 0,−5,5,10;−4,1,6,11;2,7,12,17 等是剩余类包含的数字,而不是剩余类,只有这 5 个 C_0,C_1,C_2,C_3,C_4才称为模 5 的剩余类。从定义上理解,剩余类是一个集合,和这些具体数值没有关系。

2. 完全剩余系

理解了剩余类,再学习完全剩余系相对要容易一些。完全剩余系就是从模数 m 的剩余类 C_0,C_1,C_2,…,C_{m-1}中各取一个数,为模数 m 的一组完全剩余系。具体来讲就是令 $m=5$,从 C_0,C_1,C_2,C_3,C_4中各取一个数,组成的集合就是它的完全剩余系。例如,从 C_0中取 0,从 C_1 中取 1,从 C_2 中取 2,从 C_3 中取 3,从 C_4中取 4,那么 0,1,2,3,4 就是模 5 的一组完全剩余系。如果从 C_0中取 −5,从 C_1 中取 1,从 C_2 中取 2,从 C_3 中取 3,从 C_4中取 4,则 −5,1,2,3,4 是模 5 的另一组完全剩余系。若从 C_0中取 10,从 C_1 中取 1,从 C_2 中取 7,从 C_3 中取 3,从 C_4中取 4,那么这一组也是它的完全剩余系。

3. 非负最小完全剩余系

由完全剩余系中,可以定义出一个新的概念,即非负最小完全剩余系。

从字面的意思看,上述的一组完全剩余系中 −5,1,2,3,4 就不能称为非负最小完全剩余系,因为它里面有 −5。那么最小是什么意思呢? 示例如下。

$m=5$,其最小完全剩余系为 0,1,2,3,4。

$m=9$,其最小完全剩余系为 0,1,2,3,4,5,6,7,8。

$m=7$,其最小完全剩余系为 0,1,2,3,4,5,6。

这就是说在整数模 m 的所有剩余类中,$m=5$,$m=9$,$m=7$ 是不同的剩余类,每个剩余类都会组合成多个完全剩余系,其中 $q=0$ 时所组成的完全剩余系即为非负最小完全剩余系。

4. 简化剩余系

简化剩余系(或缩系)可以这样来定义:从每个与 m 互质的剩余类中各取一个数,得到的几个数称为简化剩余系。简化剩余系的个数用 $\varphi(m)$表示。这句话的含义是,$m=8$,0,1,2,3,4,5,6,7 是它的一个完全剩余系,它与 8 互质的数是 1,3,5,7,即称为模 8 的简化剩余系。这个简化剩余系的个数用 $\varphi(8)=4$ 表示。0,1,2,3,4,5,6,7 是模 8 的非负最小剩余系。如果 m 依然等于 8,那么 8,9,2,11,4,5,6,7 也是模 8 的完全剩余系,它与 8 互质的数是 9,11,5,7,记作 $\varphi(8)=4$,同样称为模 8 的简化剩余系,也可以写好多个。当然它们都是这 4 个,而且它们对应的内容如下所示。

1,	3,	5,	7
9,	11,	5,	7
↓	↓	↓	↓
$8q+1$	$8q+3$	$8q+5$	$8q+7$

从中可以发现它们对应的都是 $8q+1$,$8q+3$,$8q+5$,$8q+7$,这样就可以得到一个性质。

5. 如果一个剩余类中有一个数与 m 互质,那么该剩余类中的所有数都与 m 互质。用公式来表达就是

若$(a,m)=1$,则$(a+km,m)=1$。

这里要说明一点：这个小括号表示的是最大公因数，如(8,10)＝2,(6,9)＝3 等。如果是中括号，即[a,b]表示的是 a 和 b 的最小公倍数，如[8,10]＝40 等。用实例来进一步说明。

$(3,8)=1$　　$(3,8)=1$

$(3+8k,8)=1$　　$k=1$　　$(11,8)=1$

$(3+8k,8)=1$　　$k=2$　　$(19,8)=1$

$(3+8k,8)=1$　　$k=3$　　$(27,8)=1$

$(3+8k,8)=1$　　$k=4$　　$(35,8)=1$

$(3+8k,8)=1$　　$k=5$　　$(43,8)=1$

$(3+8k,8)=1$　　$k=6$　　$(51,8)=1$

$(3+8k,8)=1$　　$k=7$　　$(59,8)=1$

利用完全剩余系的这样一个性质，可以将一个完全剩余系转化成为另一个完全剩余系。

6. 欧拉定理

欧拉定理是：对于 $m>1$,$a\in Z_m$，若 $\gcd(a,m)=1$，则 $a^{\varphi(m)}\equiv 1 \pmod m$，这里的 $\varphi(m)$ 是非负最小完全剩余系的个数。

示例如下。

$a=3$,$m=5$，它们互质，即 $\gcd(a,m)=\gcd(3,5)=1$。

$\varphi(5)=?$,5 是素数，非负最小完全剩余系 0,1,2,3,4，其中 0 不能取，1,2,3,4 都与 5 是互素的，说明 $\varphi(5)=4$，即 $3^4\equiv 1 \pmod 5$。可以验证一下，$3^4=3^2\times 3^2=9\times 9=81$，mod 5 后就等于 1。这就证明了欧拉定理是正确的。

7. 费码小定理

费码小定理(Fermat 小定理)是：设 P 为素数，则对于每个整数 a，都有 $a^P\equiv a \pmod P$。

示例如下。

设 $P=3$,$a=2$，则 $2^3\equiv 2 \pmod 3$，验证结果是正确的。

再设 $P=5$,$a=4$，则 $4^5\equiv 4 \pmod 5$，验证结果也是正确的。

8. 单向函数

一个可逆函数 f：$A\to B$，若它满足

(1) 对所有 $x\in A$，易于计算 $f(x)$。

(2) 反过来，同样对“几乎所有 $x\in A$”，已知 $f(x)$来求 x 是极为困难，甚至可以说几乎是不可能的，此时称 f 是一个单向函数。

5.2　RSA 加密算法

1976 年，Diffie 和 Hellman 提出了“公钥密码”这一革命性的概念。两年后，于 1978 年由麻省理工学院 Ron Rivest，Adi Shamir 和 Leonard Adleman 一起提出 RSA 加密算法，该算法是截至目前应用最广泛、影响最深远的公钥加密算法，并受到广泛关注。对于公钥密码体制来说，其安全性主要不是算法，而是取决于构造公钥算法所依赖的数学问题，要求加密函数具有单向性。RSA 算法是基于大整数难分解这一数学困难问题提出的，在它的制造过程中，利用到完全剩余系、简化剩余系、欧拉定理及单向函数等相关的数学知识。本节主要

介绍 RSA 的工作流程以及它的工作原理。

5.2.1 RSA 公钥加密体制原理

RSA 公钥算法实际上包括密钥生成、加密、解密 3 个具体的算法。

回顾本章的开始介绍了 Alice 和 Bob 的通信过程，实际上就是经历了这 3 个过程。首先由 Alice 生成公钥和私钥，私钥由自己保管，公钥在网络中任意的发布。在加密过程中，Bob 会用 Alice 的公钥加密明文生成密文，而 Alice 会利用自己的私钥解密，从而恢复出明文。

1. 密钥生成[16]

(1) 选择两个大素数 p，q(如每个为 1024 位)。

(2) 计算 $n = pq$，$z = (p-1)(q-1)$。z 意味着什么？z 等于 n 的欧拉函数，它等于与 n 互质的数的数目。z 就等于 $\varphi(n)$。

(3) 随机选取 e (其中 $e < n$)，e 与 z 没有公因数(e，z“互为质数”)。

(4) 选取 d 使得 $ed-1$ 能够被 z 完全整除(换言之，$ed \bmod z = 1$)。

(5) 公钥是(n，e)，私钥是(n，d)。

如果给定 z 和 e，能否算出 d，其实利用数论中非常基础的一个扩展欧几里得算法，就能够很容易地利用 e 和 z 把 d 计算出来。对于扩展欧几里得算法，读者要明白。

到此为止，Alice 将会把 n 和 e 作为自己的公钥在系统中发布；同时 Alice 把 n 和 d 作为自己的私钥秘密地保存。

2. 加密和解密算法

Bob 收到用户 Alice 的公钥(n，e)后，他能够将自己想要与 Alice 共享的敏感信息 m 利用下面的公式计算得到密文。

如上所述给出(n，e)和(n，d)。

加密：由 $c = m^e \bmod n$ 将明文 m 转变为密文 c(即 m^e 除以 n 所得的余数)。

注意，$m < n$(如果需要，则分块)，这个 c 可以通过网络任意传输。

解密：$m = c^d \bmod n$(即 c^d 除以 n 所得的余数)。

$$\text{核心思想：} m = (\underbrace{m^e \bmod n}_{c})^d \bmod n$$

这里的核心是 Alice 收到密文后，可以利用自己秘密保存的私钥(n，d)恢复出明文 m。这是为什么呢？在充分理解 RSA 解密步骤的过程中，首先要理解欧拉定理。

由欧拉定理得出

当 $\gcd(a,n)=1$ 时，$a^{\varphi(n)} \bmod n = 1$ 是成立的，下面解释一下 RSA 中为什么解密也是成立的。

上述在密钥生成过程中，Alice 计算：

(1) $n = pq$。

(2) $\varphi(n) = (p-1)(q-1)$，即意味着 z 就是 $\varphi(n)$。

(3) Alice 通过某种特殊的形式，选择整数 e 和 d，使得 e 和 d 在模 $\varphi(n)$ 的情况下，互为逆元。也就是说，e 乘 d 在模 $\varphi(n)$ 的情况下，与 1 是同余的。由此很容易发现，若用公式表示就是(4)。

（4）$e \cdot d = 1 + k \cdot \varphi(n)(k > 0, k \in Z$，它是任意一个整数，所以可理解为它是没有任何意义的）。

接下来看 RSA 的解密为什么成立。大家知道在整个解密过程中，使用密文，然后使用私钥来执行解密步骤。

于是有：$C^d = (m^e)^d = m^{1+k\varphi(n)} = m^1 \cdot (m^{\varphi(n)})^k$。根据欧拉定理可知，当 m 与 n 互素时，$m^{\varphi(n)}$ 在 mod n 的情况下其实是等于 1 的，因此就很容易能够在 mod n 的情况下把明文 m 给恢复出来。上式就等于

$$m^1 \cdot (1)^k = m^1 = m \bmod n$$

这就是为什么 RSA 算法能够非常巧妙地利用欧拉定理，允许 Alice 和 Bob 在 RSA 加密算法的保护下，确保信息 m 只有 Alice 和 Bob 能够共同分享。

RSA 算法的安全性可归结到大整数难分解的问题上，那么大整数难分解和 RSA 算法有什么关系？大家回顾一下在计算欧拉函数的过程中，有这样一个结论：当给定一个任意正整数 n 时，若要计算 $\varphi(n)$，如果能知道 n 的素数分解，就很容易计算出 n 的欧拉函数；如果不知道 n 的素数分解，则难以计算出 n 的欧拉函数。在密钥生成过程中，Alice 自己选取了两个大素数 p 和 q，由此生成了 $n = pq$，$z = (p-1)(q-1)$。由于 Alice 只公布了 n 和 e，任何一个用户拿到 n 后，如果 p 和 q 足够大，谁都很难执行有效的素数分解，由此不知道 p 和 q，也就没有办法把 $(p-1)(q-1)$ 计算出来。从而可以发现任何用户都无法从 n 和 e 中把 z 反推出来，所以在不知道 z 的情况下，任何用户都没有办法从 e 中把 d 反推出来，也就没有办法计算解密步骤。也就是说，RSA 的安全性是可以归结到大整数难分解这样一个困难问题上。为了加深对 RSA 算法的理解，用一个实例来说明 RSA 算法的全过程。

5.2.2　RSA 算法实例

1. 给定 RSA 算法中的参数：$p = 47$，$q = 71$，$e = 79$，明文 $m = 9$，求公钥和私钥，并实现对明文的加密和解密过程。

解：

（1）已知两个不同的大素数 p 和 q，求出 $n = p \times q = 47 \times 71 = 3337$。

（2）计算欧拉函数值 $\varphi(n) = (p-1) \times (q-1) = (47-1) \times (71-1) = 3220$，并且给出 $e = 79$，满足与 $(p-1) \times (q-1)$ 互素的条件。

（3）求 d。满足 $(e \cdot d) \bmod \varphi(n) = 1$，即 $(e \cdot d) \div 3220 = k \cdots\cdots 1$（其中 k 为商）。可以转换为 $e \cdot d = 3220 \cdot k + 1$，其中 $e = 79$。

下面利用扩展的欧几里得算法求解密密钥 d[1]

① $79 \cdot d = 3220 \cdot k + 1$

② $3220 \bmod 79 = 60 \rightarrow 79 \cdot d = 60 \cdot k + 1$

③ $79 \bmod 60 = 19 \rightarrow 19 \cdot d = 60 \cdot k + 1$

④ $60 \bmod 19 = 3 \rightarrow 19 \cdot d = 3 \cdot k + 1$

⑤ $19 \bmod 3 = 1 \rightarrow 1 \cdot d = 3 \cdot k + 1$

令 $k = 0$，解出 $d = 1$。

将 $d = 1$ 代入第④步，解得 $k = 6$。

将 $k = 6$ 代入第③步，解得 $d = 19$。

将 $d=19$ 代入第②步，解得 $k=25$。

将 $k=25$ 代入第①步，解得 $d=1019$。

所以，公开密钥 $PK=(n,e)=(3337,79)$。

私密密钥 $SK=(n,d)=(3337,1019)$。

(4) 加密过程：$c_i=(m_i)^e \bmod n=9^{79} \bmod 3337=1605$。

(5) 解密过程：$m_i=(c_i)^d \bmod n=1605^{1019} \bmod 3337=9$。

这个过程，是完全利用扩展的欧几里得算法一步一步计算出来的。实际上求解密密钥的过程就是求乘法逆元的过程。下面给出一个求乘法逆元的方法。求出私钥，完成信息的加密和解密。

2. 若 $p=13$，$q=17$，明文 $m=20$。

1) 密钥产生

计算：$n=pq=221$，$\varphi(n)=(p-1)(q-1)=192$。取 $e=7$，满足 $1<e<\varphi(n)$，且 $\gcd(\varphi(n),e)=1$。

确定满足 $de=1 \bmod 192$ 且小于 192 的 d，即求 e 的逆。

求乘法逆元的方法如下。

Q	X_1	X_2	X_3	Y_1	Y_2	Y_3	
—	(1,	0,	192)	(0,	1,	7)	
27	0,	1,	7	1,	−27,	3	(T_1,T_2,T_3)
2	1,	−27,	3	−2,	55	1	(T_1,T_2,T_3)

1. 只要 $Y_3=1$ 时，则 Y_2 就是 e 的乘法逆元素。

2. Q 的值等于 X_3/Y_3 取整。

3. $T_1=X_1-QY_1$。

4. $T_2=X_2-QY_2$。

5. $T_3=X_3-QY_3$。

6. 如果 Y_2 是负数，则将这个负数 $+\varphi(n)$。

可以验证一下：因为 $55\times7=385=2\times192+1$，所以 $d=55$ 是正确的。

公开密钥为 $PK=\{221, 7\}$。

秘密密钥为 $SK=\{221, 55\}$。

2) 加密和解密

加密如下。

密文 $c=m^e \bmod n=20^7 \bmod 221=45$。

解密如下。

明文 $m=c^d \bmod n=45^{55} \bmod 221=20$。

5.3 ElGamal 密码体制

ElGamal 密码体制是一种基于离散对数问题的公钥密码体制，既能用于数据加密，也能用于数字签名，在讲解该体制前，同样先学习几个数学基础知识。

5.3.1　离散数学——群和循环群

在介绍群之前，先讲几个数学概念。

- 非空集合：在集合论里，至少含有一个元素的集合，叫作非空集合。
- 二元运算：是由两个元素形成第3个元素的一种规则。

在非空集合 S 上的二元运算中，会产生如下3个概念。

- 幺元：任取一个 x 属于非空集合 S，若在非空集合 S 中存在一个元素 e，$e \star x = x$ 且 $x \star e = x$ 就表示 e 是 $<S, *>$ 的单位元，也就是幺元。
- 零元：任取一个 x 属于非空集合 S，若在非空集合 S 中存在一个元素 o，$o * x = 0$ 且 $x \star o = 0$ 就表示 o 是 $<S, *>$ 的零元。
- 逆元：任取一个 b 属于非空集合 S，若在非空集合 S 中存在一个元素 a，$a \star b = e$ 且 $b \star a = e$，就表示 a 是 b 的逆元，也可以说 b 是 a 的逆元。

幺元，就是具有不变性，实数集合 **R** 上的加法运算中0就是零元，实数集合 **R** 上的乘法运算中，1就是幺元；若 $ab = ba = 1$，则 a 与 b 互为逆元。实数乘法运算中，互为倒数的两个数互称逆元。例如，2和1/2互为逆元，1和其本身互为逆元；乘法运算中，零元就是对任意元 x，都有 $xa = ax = a$，则 a 为零，因此0即为零元。

1. 群

群是一种抽象的代数结构，一个群 $(G, \cdot)$ 是在集合 G 上赋予了一个二元运算 $\cdot$ 的结构，该运算满足以下要求。

(1) 封闭性(closure)：$\forall x, y \in G, x \cdot y \in G$，即任意 G 中元素 x，y 满足 $x \cdot y$ 仍是 G 中元素。

(2) 结合性(associativity)：$\forall x, y, z \in G, x \cdot (y \cdot z) = (x \cdot y) \cdot z$。

(3) 单位元(identity element)存在性：$\exists e \in G, \forall x \in G, e \cdot x = x \cdot e = x$。

(4) 逆元(inverse element)存在性：$\forall x \in G, \exists y \in G, x \cdot y = y \cdot x = e$，通常会把这样的 y 称作 x 的逆元，并记为 x^{-1}。

上述表达式说明如下。

$\exists$：数学上 $\exists$ 是一种存在量词。$\exists$ 存在量词 $\exists x$：$P(x)$ 表示存在至少一个 x 使得 $P(x)$ 为真。

$\forall$：$\forall$ 即全称量化符号，是一种数学符号，用以代表全称量词。在汉语中，该符号读作任意。

满足上述结构就是一个群。严格来说，这样的一个群应该表示为 $(G, \cdot)$，而 G 表示的是没有赋予运算的集合。但是为了方便讨论，通常也会直接用定义群的这个集合来称呼这个群，比如简单地把上述定义的群叫作群 G。

集合的元素数量被称为集合的基数或势，而群 G 的元素数量也可以称为群的阶(order)记作 $|G|$。

2. 群的性质

(1) 群中无零元。

(2) $\langle G, * \rangle$ 是群，$\forall a, b \in G$，必存在唯一的 $x \in G$，使得 $a * x = b$。

(3) $\langle G, * \rangle$ 是群，$\forall a, b, c \in G$，若 $a * b = a * c$ 或 $b * a = c * a$，则 $b = c$(消去律)。

(4) 在群$\langle G, * \rangle$中,只有幺元 e 是等幂元。

(5) 在有限群$\langle G, * \rangle$中,每个元素都具有有限阶,且阶数至多是$|G|$。

(6) 在群$\langle G, * \rangle$的运算表中的每一行或每一列都是 G 的元素的一个置换。

3. 循环群[18]

循环群是具有一定特殊属性的群,循环群能够很好地与 ElGamal 公钥加密体制相结合,产生一些特殊的属性。

(1) 定义:$\langle G, * \rangle$是一个群,如果存在 $a \in G$,使得 G 群中任意元素都由 a 的幂组成,就称$\langle G, * \rangle$为循环群;元素 a 称为它的生成元。若 $o(a) = \infty$,则 G 称为无限循环群;若 $o(a) = n$,n 是某个正整数,则 G 称为有限循环群。

例如以下几种情况。

① 整数加法群 Z 是循环群,其生成元为 1 或 -1。

② 模整数 m 的剩余类加群 Z_m 是循环群,其生成元为[1]。

③ 模整数 m 的简化剩余类乘群 Z_m^* 是循环群。

(2) 群中的离散对数问题。

在循环群中有这样一个问题,叫作群中的离散对数问题。首先来看什么是群中的离散对数问题。

定义:设 $G = \langle a \rangle$是循环群。群 G 中的离散对数问题是指:给定 G 中一个元素 h,找到正整数 k,使得 $h = a^k$,就把 k 称为 h 相对于生成元的离散对数,记作

$$k = \log_a h$$

实践中可发现,不管是在整数加法群$(Z, +)$,还是 Z_m 模 m 剩余类组成的加法群中,都非常容易解决离散对数问题。那有无法很好地解决离散对数问题的情况吗?实际上有一个大家都比较熟悉的群,在这个群中是不太好解决离散对数问题的。

目前数学界的研究成果可以发现,目前针对 Z_m^* 这样的一个乘法群中的离散对数问题实际上是困难的。

例如,Z_m^* 是模 m 简化剩余系组成的乘法群,g 为 Z_m^* 的一个生成元。离散对数问题为:给定 $h \in Z_m^*$,求解 x,使得

$$g^x \equiv h \pmod{m}$$

在有限时间之内是不可能的。只有 m 的取值足够长,就能够保证在 Z_m^* 构成的乘法群中,离散对数问题是难解的。

再如,Z_m^* 实际上很容易表示为循环群。如当 $m = 7$ 时,$\{1,2,3,4,5,6\}$ 关于模 7 乘法构成循环群,很容易发现 3 或 5 是该群的生成元,$3^0 \equiv 1 \pmod 7$,$3^1 \equiv 3 \pmod 7$,$3^2 \equiv 2 \pmod 7$,$3^3 \equiv 6 \pmod 7$,$3^4 \equiv 4 \pmod 7$,$3^5 \equiv 5 \pmod 7$,$3^6 \equiv 1 \pmod 7$。由此可以发现,在模 7 的简化剩余系中,所有的元素都可以用 3 的次方模 7 构成。由此可知,Z_m^* 根据模 m 的求法很容易形成一个循环群。由于 Z_m^* 构成的这个循环群中,离散对数问题是困难的,由此基于离散对数问题,能够像基于大数难分解一样,可以构造出一个公钥密码体制的。

5.3.2 ElGamal 公钥加密体制

在学习本节内容前,大家已经学习了群以及循环群这两个非常重要的概念,基于群和循

环群中离散对数问题，ElGamal 公钥加密体制能够被很好地理解。尽管 1978 年 RSA 公钥加密机制问世后，给密码体制带来了重大的变革，但是有大量的学者和专家尝试采用不同的数学困难问题去构造更加高效、更加优雅的公钥加密体制，ElGamal 加密机制就是在这样的背景下诞生的。1985 年，由 ElGamal 这位著名的密码学专家利用循环群中离散对数问题构造了 ElGamal 公钥加密体制，一经提出就受到了业界的广泛关注。

1. ElGamal 公钥加密体制的工作原理[19]

在 RSA 公钥加密体制中，Alice 和 Bob 想利用 Internet 进行安全通信要经过 3 个算法，ElGamal 公钥加密体制同样要经过 3 个算法。首先，由 Alice 利用密钥生成算法，生成自己的公钥和私钥。随后，Alice 会将自己的公钥在系统中进行分发。而 Bob 想要给 Alice 发送一个特定信息时，将会利用 Alice 的公钥和明文实行加密算法，从而形成密文。只有 Alice 本人可以利用由她本人独立保存的私钥才能对 Bob 形成的密文进行解密，从而恢复出明文。以下是 3 个具体的算法[20]。

（1）密钥生成。

✧ P：首先 Alice 会生成的一个较大素数 P，然后利用准备的整数 P 能够构造一个 Z_P^*，即构造一个模 P 的简化剩余系。在 Z_P^* 中找到一个生成元 g。

✧ g：Z_P^* 中的生成元。随后 Alice 会从 Z_{p-1} 中任意选取随机数 α。

✧ $\alpha \in Z_{p-1}$，计算 $\beta = g^{\alpha} \bmod p$：在这里 g^{α} 是在模 p 前提下计算的。大家会很容易发现 β 实际上就是 Z_P^* 这个循环群中的一个元素。回想循环群中 g^{α} 实际上可以理解为 α 个 g 相乘，因此 β 可以理解为 α 个 g 相乘得到的元素，由于群中所有二元运算都要满足封闭性，显而易见 β 也是循环群中的一个元素。

✧ p，g，β 为公钥，作为自己的公钥，Alice 将其公开。

✧ α 为私钥：Alice 秘密地保存 α，作为自己的私钥。

（2）加密：Bob 想要为 Alice 生成一个对应的密文信息，Bob 就需要在 Z_{p-1} 的集合中任意选取一个随机数 k，即

✧ 随机生成一个秘密数 k，$k \in Z_{p-1}$。然后根据明文 x，遵循如下步骤形成密文。

✧ $E(x,k) = (r,s)$，其中

$$r = g^k \bmod p$$

$$s = x\beta^k \bmod p$$

r 和 s 作为密文在网络上进行传输，任何用户拿到密文后，都无法从 r 与 s 中把 x 给恢复出来。而只有 Alice 本人可以利用自己的私钥 α 能从 r 与 s 中把 x 恢复出来。

（3）解密：Alice 收到 r 和 s 后，做 $s(r^{\alpha})^{-1} \bmod p$ 的运算，从中可以发现，r 和 s 实际上是公共传输的密文信息，而 α 则是 Alice 的私钥，因此只有 Alice 利用自己的私钥才能执行这个算法。S 实际上等同于 $xg^{\alpha k}$，$(r^{\alpha})^{-1}$ 可理解成 $g^{-\alpha k}$，它们可以约掉，由此恢复出 x。

$$D(r,s) = s(r^{\alpha})^{-1} \bmod p = xg^{\alpha k}g^{-\alpha k} \bmod p = x$$

ElGamal 公钥加密机制有史以来，实际上也受到广泛关注。

2. ElGamal 与 RSA 的区别

尽管 RSA 公钥算法得到了大量的部署，但是和 ElGamal 加密算法相比较而言，两种加密算法有如下不同，如表 5.1 所示。