第5章 公钥密码

前面几章主要讨论了对称密码,本章将讨论非对称密码,即公开密钥密码(一般简称公钥密码)。


公钥密码

5.1 公钥密码的基本概念

利用传统密码进行保密通信,通信的双方必须首先预约持有相同的密钥才能进行。而私人和商业之间想通过通信工具洽谈生意又要保持商业秘密,有时很难做到事先预约密钥。另外,对于大型计算机网络,设 n 个用户,任意两个用户之间都可能进行通信,共有 $C_n^2=n(n-1)/2$ 种不同的通信方式,当 n 较大时这一数目是很大的。从安全角度考虑,密钥应当经常更换。在网络上产生、存储、分配、管理如此大量的密钥,其复杂性和危险性都是很大的。

因此,密钥管理上的困难是传统密码应用的主要障碍。这种困难在计算机网络环境下更显得突出。另外,传统密码不易实现数字签名,也限制了它的应用范围。

为此,人们希望能设计一种新的密码,从根本上克服传统密码在密钥管理上的困难,而且容易实现数字签名,从而适合计算机网络环境的应用,适合各种需要数字签名的应用。

1976 年,美国斯坦福大学的博士生 W.Diffie 和他的导师 M.Hellman 教授发表了“密码学新方向”的论文,第一次提出公钥密码的概念,从此开创了一个密码新时代。

公钥密码的基本思想是将传统密码的密钥 K 一分为二,分为加密钥 K_e 和解密钥 K_d。用加密钥 K_e 控制加密,用解密钥 K_d 控制解密,而且通过计算复杂性确保在计算上由加密钥 K_e 不能推出解密钥 K_d。这样,即使是将 K_e 公开,也不会暴露 K_d,也不会损害密码的安全。于是便可将 K_e 公开,而只对 K_d 保密。由于 K_e 是公开的,只有 K_d 是保密的,因此从根本上克服了传统密码在密钥分配上的困难。

根据公钥密码的基本思想,可知一个公钥密码应当满足以下 3 个条件。

① 解密算法 D 与加密算法 E 互逆,即对于所有明文 M,都有

$$D(E(M,K_e),K_d)=M \tag{5-1}$$

② 在计算上不能由 K_e 求出 K_d。

③ 算法 E 和 D 都是高效的。

条件①是构成密码的基本条件，是传统密码和公钥密码都必须具备的条件。

条件②是公钥密码的安全条件，是公钥密码的安全基础。而且这一条件是最难满足的。由于数学水平的限制，目前尚不能从数学上证明一个公钥密码完全满足这一条件，而只能证明它不满足这一条件。这就是困难的根本原因。

条件③是公钥密码的实用条件。因为只有算法 E 和 D 都是高效的，密码才能实际应用。否则，可能只有理论意义，而不能实际应用。

满足以上 3 个条件，便可构成一个公钥密码，这个密码可以确保数据的保密性。

进而，如果满足第四个条件，则能够确保数据的真实性。

④ 对于所有明文 M，都有

$$E(D(M,K_d),K_e)=M \tag{5-2}$$

如果满足条件①、②、④，同样可构成一个公钥密码，这个密码可以确保数据的真实性。

如果同时满足以上四个条件，则这个公钥密码可以同时确保数据的保密性和真实性。此时，对于所有的明文 M，都有

$$D(E(M,K_e),K_d)=E(D(M,K_d),K_e)=M \tag{5-3}$$

公钥密码从根本上克服了传统密码在密钥分配上的困难，利用公钥密码进行保密通信需要成立一个密钥管理中心(Key Management Center，KMC)，每个用户都将自己的姓名、地址和公开的加密钥等信息在 KMC 登记注册，将公钥记入共享的公钥数据库(Public Key Database，PKDB)。KMC 负责密钥的管理，并且对用户是可信赖的。这样，用户利用公钥密码进行保密通信就像查电话号码簿打电话一样方便，再无通信双方预约密钥之苦，因此特别适合计算机网络应用。加上公钥密码实现数字签名容易，所以特别受到欢迎。

5.2 RSA 密码

1978 年，美国麻省理工学院的三名密码学者 R.L.Rivest、A.Shamir 和 L.Adleman 提出一种基于大合数因子分解困难性的公钥密码，简称为 RSA 密码。RSA 密码被誉为是一种风格幽雅的公钥密码。由于 RSA 密码既可用于加密，又可用于数字签名，通俗易懂，因此 RSA 密码已成为目前应用较广泛的公钥密码之一。许多国际标准化组织，如 ISO、ITU、SWIFT 和 TCG 等都已接受 RSA 作为标准。Internet 网络的 Email 保密系统 GPG 以及国际 VISA 和 MASTER 组织的电子商务协议(SET 协议)中都将 RSA 密码作为传送会话密钥和数字签名的标准。

我国学者在公钥密码的设计、分析和应用等方面都做出了许多卓越贡献。例如，我国商用公钥密码 SM2 已成为 ISO 和 TCG 的标准。

5.2.1 RSA 加解密算法

RSA 密码的密钥生成、加密及解密过程如下。

① 随机选择两个大素数 p 和 q,并且保密;

② 计算 $n=pq$,将 n 公开;

③ 计算 $\varphi(n)=(p-1)(q-1)$,对 $\varphi(n)$保密;

④ 随机选取一个正整数 e,$1<e<\varphi(n)$且$(e,\varphi(n))=1$,将 e 公开;

⑤ 根据 $ed=1 \bmod \varphi(n)$,求出 d,并对 d 保密;

⑥ 加密运算:

$$C=M^e \bmod n \tag{5-4}$$

⑦ 解密运算:

$$M=C^d \bmod n \tag{5-5}$$

由以上算法可知,RSA 密码的公钥 $K_e=<n,e>$,而保密的私钥 $K_d=<p,q,d,\varphi(n)>$。

说明:上述算法中的 $\varphi(n)$称为欧拉(Euler)函数,意指比 n 小的正整数中与 n 互素的数的个数。例如,$\varphi(6)=2$,因为在 1,2,3,4,5 中与 6 互素的数只有 1 和 5 两个数。若 p 和 q 为素数,且 $n=pq$,则 $\varphi(n)=(p-1)(q-1)$。

例 5-1 令 $p=47$,$q=71$,$n=47\times71=3337$,$\varphi(n)=\varphi(3337)=46\times70=3220$。选取 $e=79$,计算 $d=e^{-1} \bmod 3220=1019 \bmod 3220$。公开 $e=79$ 和 $n=3337$,保密 $p=47$,$q=71$,$d=1019$ 和 $\varphi(n)=3220$。

设明文 $M=688\ 232\ 687\ 966\ 668\ 3$,进行分组,$M_1=688$,$M_2=232$,$M_3=687$,$M_4=966$,$M_5=668$,$M_6=003$。$M_1$的密文 $C_1=688^{79} \bmod 3337=1570$,继续进行类似计算,可得最终密文 $C=1570\ 2756\ 2091\ 2276\ 2423\ 158$。

若解密,则计算 $M_1=1570^{1019} \bmod 3337=688$,类似地,可解密还原出其他明文。

现在对 RSA 算法给出一些证明和说明。

(1) 加解密算法的可逆性。

要证明加解密算法的可逆性,根据式(5-1)即证明:

$$M=C^d=(M^e)^d=M^{ed} \bmod n$$

因为 $ed=1 \bmod \varphi(n)$,这说明 $ed=t\varphi(n)+1$,其中 t 为某整数。所以

$$M^{ed}=M^{t\varphi(n)+1} \bmod n$$

因此要证明 $M^{ed}=M \bmod n$,只需证明

$$M^{t\varphi(n)+1}=M \bmod n$$

在$(M,n)=1$ 的情况下,根据数论知识,有

$$M^{t\varphi(n)}=1 \bmod n$$

于是有

$$M^{t\varphi(n)+1}=M \bmod n$$

在$(M,n)\neq1$ 的情况下,分两种情况 $M\in\{1,2,3,\cdots,n-1\}$和 $M=0$:

① $M\in\{1,2,3,\cdots,n-1\}$。

因为 $n=pq$,p 和 q 为素数,$M\in\{1,2,3,\cdots,n-1\}$,且$(M,n)\neq1$,这说明 M 必含 p 或 q 之一为其因子,而且不能同时包含两者,否则将有 $M\geqslant n$,与 $M\in\{1,2,3,\cdots,n-1\}$矛盾。不妨设 $M=ap$,其中 a 为某正整数。

又因 q 为素数，且 M 不包含 q，故有 $(M,q)=1$，于是有

$$M^{\varphi(q)}=1 \bmod q$$

进一步，有

$$M^{t(p-1)\varphi(q)}=1 \bmod q$$

因为 q 是素数，$\varphi(q)=(q-1)$，所以 $t(p-1)\varphi(q)=t(p-1)(q-1)=t\varphi(n)$，所以有

$$M^{t\varphi(n)}=1 \bmod q$$

于是

$$M^{t\varphi(n)}=bq+1,\text{其中 } b \text{ 为某整数}$$

两边同乘 M，得

$$M^{t\varphi(n)+1}=bqM+M$$

因为 $M=ap$，故

$$M^{t\varphi(n)+1}=bqap+M=abn+M$$

取模 n 得

$$M^{\varphi(n)+1}=M \bmod n$$

② $M=0$。

当 $M=0$ 时，直接验证，可知命题成立。

(2) 加密和解密运算的可交换性。

根据式(5-4)和式(5-5)，有

$$D(E(M))=(M^e)^d=M^{ed}=(M^d)^e=E(D(M)) \bmod n$$

所以，根据式(5-3)可知，RSA 密码可同时确保数据的保密性和数据的真实性(完整性)。

(3) 加解密算法的有效性。

根据式(5-4)和式(5-5)可知，RSA 密码的加解密运算的核心是模幂运算。目前已有多种有效的模幂运算算法，因此 RSA 密码算法是有效的。但是，为了安全，RSA 密码的参数必须选用很大的数，这时 RSA 密码算法的运算还是比较困难的。因此，提高 RSA 密码的加解密效率仍是一个值得研究的问题。

(4) 在计算上由公钥不能求出私钥。

这一问题在 5.2.2 节分析论证。

5.2.2　RSA 密码的安全性

小合数的因子分解是容易的，然而大合数的因子分解却十分困难。关于大合数 N 的因子分解的时间复杂度下限，目前尚没有一般的结果，迄今为止的各种因子分解算法提示人们这一时间下限将不低于 $O(\mathrm{EXP}(\ln N\ln(\ln N)^{1/2})$。根据这一结论，只要合数 N 足够大，进行因子分解是相当困难的。

密码分析者攻击 RSA 密码的一种可能的途径是截获密文 C，从 C 中求出明文 M。他知道

$$M \equiv C^d \bmod n$$

因为 n 是公开的，要从 C 中求出明文 M，必须先求出 d，而 d 是保密的。但他知道：

$$ed \equiv 1 \bmod \varphi(n)$$

e 是公开的,要从中求出 d,必须先求出 $\varphi(n)$,而 $\varphi(n)$是保密的。但他又知道:

$$\varphi(n)=(p-1)(q-1)$$

要从中求出 $\varphi(n)$,必须先求出 p 和 q,而 p 和 q 是保密的。但他也知道:

$$n=pq$$

要从 n 求出 p 和 q,只有对 n 进行因子分解。而当 n 足够大时,这是很困难的。

由此可见,只要能对 n 进行因子分解,便可攻破 RSA 密码。由此可以得出:破译 RSA 密码的困难性小于或等于对 n 进行因子分解的困难性。目前尚不能证明两者是否能确切相等,因为不能确知除对 n 进行因子分解的方法外,是否还有别的更简捷的破译方法。

应用 RSA 密码应密切关注世界大数因子分解的进展。虽然大数的因子分解是十分困难的,但是随着科学技术的发展,计算机的计算能力不断提高,人们对大数因子分解的能力在不断提高,而且分解所需的成本在不断下降,因此对 RSA 密码的威胁在不断上升。

1994 年 4 月 2 日,由 40 多个国家的 600 多位科学家参加,通过 Internet 网,历时 9 个月,成功地分解了十进制 129 位的大合数,破译了 Rivest 等悬赏 100 美元的 RSA-129。1996 年 4 月 10 日又破译了 RSA-130。更令人惊喜的是,1999 年 2 月,美国、荷兰、英国、法国和澳大利亚的数学家和计算机专家,通过 Internet 网,历时 1 个月,成功地分解了十进制 140 位的大合数,破译了 RSA-140。2007 年 5 月,人们成功分解了一个十进制 313 位(1038 位二进制位)的大合数 $2^{1039}-1$。这个大整数是一个梅森数。这是迄今为止被分解的最大整数。2009 年 12 月,人们又成功破译了 RSA-232。2019 年 12 月,法国和美国科学家宣布,成功地分解了十进制 240 位的大合数,从而破译了 RSA-240。该团队在 2020 年 3 月又宣布,成功分解了十进制 250 位的大合数,从而破译了 RSA-250。

其中,RSA-240 的大整数分解具体如下:

$$
\begin{aligned}
&124620366781718784065835044608106590434820374651678805754818788883289666801\\
&188210855036039570272508747509864768438458621054865537970253930571891217684\\
&318286362846948405301614416430468066875699415246993185704183030512549594371\\
&372159029236099\\
=\ &\\
&509435952285839914555051023580843714132648382024111473186660296521821206469\\
&746700620316443478873837606252372049619334517\\
\times\ &\\
&244624208838318150567813139024002896653802092578931401452041221336558477095\\
&178155258218897735030590669041302045908071447
\end{aligned}
$$

根据大整数因子分解能力的进展,今天要应用 RSA 密码,应当采用足够大的整数 n。人们普遍认为,对于一般应用,n 至少应取 1024 位;对于重要应用,n 最好取 2048 位。作者的研究小组在不同的项目中分别用软硬件方式研制开发出 1024 位和 2048 位的 RSA 密码系统。

大合数因子分解算法的研究是当前数论和密码学的一个十分活跃的领域。应用 RSA 密码应密切关注世界因子分解的进展。目前大合数因子分解的主要算法有

Pomerance 的二次筛法、Lenstra 的椭圆曲线分解算法、Pollard 的数域筛法及其各种改进、广义数域筛法、格数域筛法和针对特殊数的特殊数域筛法。要了解这些内容，请查阅有关文献。表 5-1 给出了采用广义数域筛法进行因子分解所需的计算机资源。表 5-2 给出了近年来因子分解问题的进展情况。

表 5-1　因子分解所需的计算机资源

合数/位	所需 MIPS 年	合数/位	所需 MIPS 年
116	4×10^2	768	2×10^7
129	5×10^3	1024	3×10^{11}
512	3×10^4	2048	3×10^{20}

表 5-2　因子分解问题的进展情况

十进制位数	二进制位数(近似值)	完 成 日 期	MIPS 年	算　　法
100	332	1991 年 4 月	7	二次筛法
110	365	1992 年 4 月	75	二次筛法
120	398	1993 年 6 月	830	二次筛法
129	428	1994 年 4 月	5000	二次筛法
130	431	1996 年 4 月	1000	广义数域筛法
140	465	1999 年 2 月	2000	广义数域筛法
155	512	1999 年 8 月	8000	广义数域筛法
160	530	2003 年 4 月		格数域筛法
174	576	2003 年 12 月		格数域筛法
200	663	2005 年 5 月		格数域筛法
232	768	2009 年 12 月		
313	1038	2007 年 5 月		特殊数域筛法
240	795	2019 年 12 月		
250	829	2020 年 3 月		

由上可知，RSA 密码的安全性除了与上述攻击密切相关外，还与其参数(p，q，e，d)的选取有密切关系。只要合理地选取参数，并且正确使用，在目前 RSA 密码仍是安全的。这就是目前 RSA 密码仍然广泛使用的重要原因。

最后指出，除了要继续关注用电子计算机进行大整数的因子分解之外，还必须关注用量子计算机进行的大整数因子分解。因为利用 Shor 算法可以在多项式时间内求解整数分解问题。理论分析表明，利用 Shor 算法 2048 量子位的量子计算机可以攻破 1024 位的 RSA 密码。但是，目前多数量子计算机都是专用型计算机，尚不能执行 Shor 算法。只有少数量子计算机能够执行 Shor 算法，可惜量子位数太少，只能分解小整数，不能分解大整

数,尚不能对 RSA 密码构成实际威胁。

请读者注意:除 Shor 算法可以分解大整数外,又出现了其他类型的大整数分解方法和不用进行因子分解而直接对 RSA 进行唯密文攻击的方法。文献[26,27]把演化密码扩展到量子计算机领域,不采用 Shor 算法和 Grover 算法,而采用量子模拟退火算法进行大整数因子分解,成功分解了大整数 $1245407=1109\times1123$,创造了 2019 年的世界最好量子分解纪录,目前已经能够分解小于 2^{80} 的大整数。文献[64,85]给出了两种唯密文攻击 RSA 的多项式复杂性的量子算法。两个算法都不需要因子分解,直接由 RSA 的密文求解明文。两个算法的成功概率都高于 Shor 算法,而且也没有 Shor 算法要求密文阶为偶数的限制。应当指出,这些研究是有意义的,但是目前的研究还是初步的,还需要进一步提高。

5.3 ElGamal 密码

ElGamal 密码是除 RSA 密码外最有代表性的公钥密码之一,它的安全性建立在离散对数问题的困难性之上,是一种公认安全的公钥密码。目前,ElGamal 密码已经得到广泛应用。

5.3.1 离散对数问题

设 p 为素数,若存在一个正整数 α,使得 $\alpha^1,\alpha^2,\alpha^3,\cdots,\alpha^{p-1}$,关于模 p 互不同余,则称 α 为模 p 的本原元。显而易见,若 α 为模 p 的本原元,则对于 $y\in\{1,2,3,\cdots,p-1\}$,一定存在一个正整数 x,使得 $y=\alpha^x \bmod p$。

于是有如下的运算:

设 p 为素数,α 为模 p 的本原元,α 的幂乘运算为

$$y=\alpha^x \bmod p,\quad 1\leqslant x\leqslant p-1 \tag{5-6}$$

则称 x 为以 α 为底的模 p 的对数。求解对数 x 的运算为

$$x=\log_\alpha p,\quad 1\leqslant y\leqslant p-1 \tag{5-7}$$

由于上述运算是定义在模 p 有限域 GF(p)上的,所以称为离散对数运算。

例 5-2 取 $p=13$,则 $\alpha=2$ 是模 p 的本原元。理由如下:

$\alpha^1=2,\alpha^2=4,\alpha^3=8,\alpha^4=3,\alpha^5=6,\alpha^6=12,\alpha^7=11,\alpha^8=9,\alpha^9=5,\alpha^{10}=10,\alpha^{11}=7,\alpha^{12}=1$

$y=\alpha^x \bmod p$	1	2	3	4	5	6	7	8	9	10	11	12
$x=\log_\alpha p \bmod p$	12	1	4	2	9	5	11	3	8	10	7	6

从 x 计算 y 是容易的,至多需要 $2\times\log_2 p$ 次乘法运算。可是从 y 计算 x 就困难得多,目前已知最快的求解离散对数算法的时间复杂度为

$$O(\exp((\ln p)^{\frac{1}{3}}\ln(\ln p))^{\frac{2}{3}})$$

可见,只要 p 足够大,求解离散对数问题是相当困难的。这便是著名的离散对数问

题。而且，离散对数问题具有较好的单向性。

由于离散对数问题具有较好的单向性，所以离散对数问题在公钥密码学中得到广泛应用。除 ElGamal 密码外，著名的 Diffie-Hellman 密钥交换协议和美国数字签名标准算法 DSA、俄罗斯数字签名标准 GOST 等都是建立在离散对数问题之上的。

5.3.2　ElGamal 密码概述

ElGamal 改进了 Diffie 和 Hellman 的基于离散对数的密钥交换协议，提出了基于离散对数的公钥密码和数字签名体制。

随机选择一个大素数 p，且要求 $p-1$ 有大素数因子。再选择一个模 p 的本原元 α。将 p 和 α 公开。

1. 密钥生成

用户随机选择一个整数 d 作为自己的秘密的解密钥，$1<d<p-1$，计算 $y=\alpha^d \bmod p$，把 y 作为自己的公开的加密钥。

由公开密钥 y 计算秘密密钥 d，必须求解离散对数，而这是极其困难的。

2. 加密过程

将明文消息 $M(0\leqslant M\leqslant p-1)$加密成密文的过程如下。

① 随机选取一个整数 k，$1<k<p-1$。

② 计算 $U=y^k \bmod p$　　(5-8)

$$C_1=\alpha^k \bmod p \tag{5-9}$$

$$C_2=UM \bmod p \tag{5-10}$$

③ 取(C_1,C_2)作为密文。

3. 解密过程

对密文(C_1,C_2)解密的过程如下。

① 计算 $V=C_1^d \bmod p$；　　(5-11)

② 计算 $M=C_2V^{-1} \bmod p$。　　(5-12)

解密的可还原性证明如下：

因为

$$\begin{aligned}C_2V^{-1} \bmod p &=(UM)V^{-1} \bmod p\\ &=UM(C_1^d)^{-1} \bmod p\\ &=UM((\alpha^k)^d)^{-1} \bmod p\\ &=UM((\alpha^d)^k)^{-1} \bmod p\\ &=UM((y)^k)^{-1} \bmod p\\ &=UM(U)^{-1} \bmod p\\ &=M \bmod p\end{aligned}$$

故解密可还原。

例 5-3　设 $p=2579$，取 $\alpha=2$，秘密钥 $d=765$，计算公开密钥 $y=2^{765} \bmod 2579=$

949。再取明文 $M=1299$,随机数 $k=853$,则 $C_1=2^{853} \bmod 2579=435$,$C_2=1299\times 949^{853} \bmod 2579=2396$,所以密文为 $(C_1, C_2)=(435, 2396)$。解密时计算 $M=2396\times(435^{765})^{-1} \bmod 2579=1299$,从而还原出明文。

4. 安全性

由于 ElGamal 密码的安全性建立在 GF(p)上离散对数的困难性之上,而目前尚无求解 GF(p)上离散对数的有效算法,所以在 p 足够大时 ElGamal 密码是安全的。为了安全,p 应为 150 位以上的十进制数,而且 $p-1$ 应有大素因子。因为 p 为大素数,$p-1$ 为偶数,所以 $p-1$ 一定有因子 2。我们希望除了因子 2 外,其余因子为大素数因子。理想情况是 p 为强素数,$p-1=2q$,其中 q 为大素数。

此外,为了安全,加密使用的 k 必须是一次性的。这是因为如果使用的 k 不是一次性的,时间长了就可能被攻击者获得。又因为 y 是公开密钥,攻击者自然知道。于是攻击者就可以根据式(5-8)计算出 U,进而利用 Euclid 算法求出 U^{-1}。又因为攻击者可以获得密文 C_2,于是可根据式(5-10)通过计算 $U^{-1}C_2$ 得到明文 M。另外,设用同一个 k 加密两个不同的明文 M 和 M',相应的密文为 (C_1, C_2) 和 (C_1', C_2')。因为 $C_2/C_2'=M/M'$,如果攻击者知道 M,则很容易求出 M'。

注意:理论上解密钥 d 的选择范围为 $1<d<p-1$,但是 d 太小或太大都不好。因为攻击者在用穷举方法猜测 d 时,一般会首先试验太小或太大的 d。同理,随机数 k 也不要太小或太大。随机数 k 的选择还要保证按式(5-8)计算的 $U \bmod p \neq 1$。如果 $U \bmod p=1$,则根据式(5-10)可知,$C_2=M$,从而暴露明文 M。

提醒读者注意,Shor 算法不仅可以在多项式时间内求解整数分解问题,而且能够在多项式时间内求解离散对数问题。因此,在量子计算环境下,Shor 算法不仅是 RSA 密码的主要威胁之一,也是 ElGamal 密码的主要威胁之一。

5.4 椭圆曲线密码

人们对椭圆曲线的研究已有 100 多年的历史,而椭圆曲线密码学(Elliptical Curve Cryptograpy,ECC)是 Koblitz 和 Miller 于 20 世纪 80 年代提出的。ElGamal 密码是建立在有限域 GF(p)之上的,其中 p 是一个大素数,这是因为有限域 GF(p)的乘法群中的离散对数问题是难解的。受此启发,在其他任何离散对数问题难解的群中,同样可以构成 ElGamal 密码。于是,人们开始寻找其他离散问题难解的群。研究发现,有限域上的椭圆曲线上的一些点构成交换群,而且离散对数问题是难解的。于是可在此群上定义 ElGamal 密码,并称为椭圆曲线密码。椭圆曲线密码的密钥短、签名短,软件实现规模小、硬件实现电路省电。人们普遍认为,160 位长的椭圆曲线密码的安全性相当于 1024 位的 RSA 密码,而且运算速度也较快。正因为如此,一些国际标准化组织已把椭圆曲线密码作为新的信息安全标准,如 IEEE P1363/D4、ANSI F9.62、ANSI F9.63 等标准,分别规范了椭圆曲线密码在 Internet 协议安全、电子商务、Web 服务器、空间通信、移动通信、智能卡等方面的应用。

5.4.1　椭圆曲线

椭圆曲线并不是椭圆，之所以称为椭圆曲线，是因为它们与计算椭圆周长的方程相似。椭圆曲线可以定义在不同的有限域上，本书仅介绍定义在素域 GF(p)上的椭圆曲线。

定义 5-1　设 p 是大于 3 的素数，且 $4a^3+27b^2 \neq 0 \bmod p$，称曲线

$$y^2 = x^3 + ax + b, a, b \in \mathrm{GF}(p) \tag{5-13}$$

为 GF(p)上的椭圆曲线。

由式(5-13)的椭圆曲线可得到一个同余方程：

$$y^2 - (x^3 + ax + b) = 0 \bmod p \tag{5-14}$$

式(5-14)的解为一个二元组(x, y)，其中 $x, y \in \mathrm{GF}(p)$，将此二元组描画到椭圆曲线上便为一个点，并称其为解点。

为了利用解点构成交换加群，需要引进一个 0 元素作为单位元，并定义如下的加法运算。

① 引进一个无穷点 $O(\infty, \infty)$，简记为 O，作为 0 元素。

$$O(\infty, \infty) + O(\infty, \infty) = 0 + 0 = 0 \tag{5-15}$$

并定义对于所有的解点 $P(x, y)$，有

$$P(x, y) + O = O + P(x, y) = P(x, y) \tag{5-16}$$

② 设 $P(x_1, y_1)$和 $Q(x_2, y_2)$是解点，如果 $x_1 = x_2$ 且 $y_1 = -y_2$，则

$$P(x_1, y_1) + Q(x_2, y_2) = 0 \tag{5-17}$$

这说明任何解点 $R(x, y)$的逆都是 $R(x, -y)$。

③ 设 $P(x_1, y_1)$和 $Q(x_2, y_2)$是解点，如果 $P \neq \pm Q$，则

$$P(x_1, y_1) + Q(x_2, y_2) = R(x_3, y_3)$$

其中

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2 \\ y_3 = \lambda(x_1 - x_3) - y_1 \\ \lambda = \dfrac{(y_2 - y_1)}{(x_2 - x_1)} \end{cases} \tag{5-18}$$

④ 当 $P(x_1, y_1) = Q(x_2, y_2)$时，有

$$P(x_1, y_1) + Q(x_2, y_2) = 2P(x_1, y_1) = R(x_3, y_3)$$

其中

$$\begin{cases} x_3 = \lambda^2 - 2x_1 \\ y_3 = \lambda(x_1 - x_3) - y_1 \\ \lambda = \dfrac{3x_1^2 + a}{2y_1} \end{cases} \tag{5-19}$$

作集合 $E = \{$全体解点，无穷点 $O\}$。

可以验证，如上定义的集合 E 和加法运算构成加法交换群。

椭圆曲线及其解点的加法运算的几何意义如图 5-1 所示。