一种改进的冗余序列算法在椭圆曲线密码体制中的实现*1)

郝　林
(云南大学信息学院　昆明 650091)

李　彤　柳　青
(云南大学软件学院　昆明 650091)

摘　要

为了提高椭圆曲线上点的数乘的运算效率，本文提出了椭圆曲线离散对数 (ECDLP) 上一种改进的快速冗余算法. 算法就文献 [5] 提出的任一大的正整数的二进制冗余序列，给出了新的消除了序列转换中不必要的步骤的构建方法，从而使得大数倍乘中加运算得以大大减少. 分析表明，新算法的效率较基本算法有明显提高.

关键词: 椭圆曲线，冗余算法，密码体制

A BETTER REDUNDANT BINARY ALGORITHM FOR THE ELLIPTIC CURVES CRYPTOSYSTEM

Hao Lin
(*School of Information Science and Engineering, Yunnan University, Kunming* 650091)

Li Tong　Liu Qing
(*School of Software, Yunnan University, Kunming* 650091)

Abstract

For the operational efficiency on the numerical multiplication of points on elliptic curves to be heighten, in this paper, a better fast redundant binary algorithm for the elliptic curves discrete logarithm problem (ECDLP) is presented. A new binary redundant representation which is necessity and suitability to transformation for a very large integer is defined. This algorithm has obtained the speed improvement as compared with the essential one which was presented in the paper[5] by decreasing the addition computing steps.

Key words: Elliptic curve, redundant binary algorithm, cryptosystem.

* 2003 年 9 月 5 日收到.

1) 国家 973 项目 (No.G1998030420)，云南省自然科学基金项目 (2002F0010M)，云南省信息网络开发技术专项计划项目 (No.2001I710) 联合资助.

§1. 引　言

Neal Koblitz[1] 和 Victor Miller[2] 利用椭圆曲线上点形成的 Abelian 加法群, 构造了椭圆曲线上的离散对数问题 (ECDLP), 并将 ECDLP 用于公钥密码体制的构建. 椭圆曲线密码体制问世十多年来, 由于它提供了一个具有更短密钥长度且运算速度相对较快的公钥密码系统, 而受到人们广泛关注 [3]. 实验表明, 在椭圆曲线加密算法中采用 160bits 的密钥可与 1024bits 密钥的 RSA 算法的安全性相当 [4], 且随着模数的增大, 它们之间加密强度的差距相应增大. 这使得在椭圆曲线密码体制中可以使用长度小得多的密钥和分组长度, 提供诸如实现数据加密, 密钥交换, 数字签名 [3] 等密码方案的有力工具. 由于椭圆曲线离散对数问题算法速度远低于对称密钥的算法速度, 在一定程度上影响了它的使用, 所以如何改善、提高 ECDLP 算法的运算效率, 成为受到关注的问题.

设 $K = GF(2^k)$ 为一个有限域, P 为该域 K 上椭圆曲线 E 上的一 m 阶点, 给定的任一大正整数 S 的二进制序列, 即 $S = a_{t-1}2^{t-1} + a_{t-2}2^{t-2} + \cdots + a_1 2^1 + a_0 2^0, a_{t-1} \neq 0$.

记 $B(s) = a_{t-1}a_{t-2}\cdots a_1 a_0$, 其中 $a_j \in \{0,1\}, j = 0,\cdots,t-1$. 由椭圆曲线 E 上点的数乘定义, SP 写成:

$$\begin{aligned} SP &= (a_{t-1}2^{t-1} + a_{t-2}2^{t-2} + \cdots + a_1 2 + a_0)P \\ &= a_{t-1}2^{t-1}P + a_{t-2}2^{t-2}P + \cdots + a_1 2P + a_0 P. \end{aligned} \tag{1.1}$$

由式 (1.1) 明显可知 SP 运算效率的提高将取决于 $B(s)$ 长度的缩短, 或者 $B(s)$ 中非零元个数的减少. 二进制冗余序列是在 $B(s)$ 中引入一个新元素 (-1)(方便起见, 以下将 (-1) 记为 $\bar{1}$) 构成新序列 (记为 $B'(s)$), 使得 $B'(s)$ 中非零元相应的减少, 从而降低了运算量. 在研究过程中, 我们注意到, 对一给定大数 S 的二进制序列 $B(s)$, 由 [5] 可等价得到一冗余序列 $B'(s)$. 若将此序列直接用于计算 SP, 从实际降低运算量的效果而言, 该算法所得的 $B'(s)$ 中某些子序列的构建是不必要的. 将 $B(s)$ 中 11 这样的子序列转换为 $B'(s)$ 中的 $10\bar{1}$ 后, 非但未能使得非零元素减少, 反而有可能将原对应序列的幂次增高了一次, 运算因此增加. 对此在文献 [5] 中虽然也给出了一个修正的方案, 但未能从根本上解决问题.

例. 设有一个二进制序列 $B(s) = 10110011$, 由文献 [5] 有修正后的 $B'(s) = 110\bar{1}010\bar{1}$.

此例中, 若将 $B(s)$ 与 $B'(s)$ 用于计算 SP, 则需要大致相同的计算量, 表明类似的转换是多余的. 本文为解决这一问题, 给出了一个新的算法来构建起冗余序列, 并进行了算法效率讨论.

§2. 相关背景

定义 1. 设 S 为任一正整数, $B(s)$ 为 S 的二进制序列. 若 S 同时能有其它序列 $B'(s)$ 的表示, 则称 $B(s)$ 与 $B'(s)$ 是等价的, 记为 $B(s) \cong B'(s)$.

定义 2. 设 S 为任一正整数, 则有

$$S = b_{t-1}2^{t-1} + b_{t-2}2^{t-2} + \cdots + b_1 2 + b_0, \tag{2.1}$$

其中 $b_i \in \{\bar{1},0,1\}, i = 0,1,\cdots t-1, b_{t-1} \neq 0$. 称式 (2.1) 为 S 的二进制冗余序列，记为 $B'(s) = b_{t-1}b_{t-2}\cdots b_1b_0$.

引理. 设 n 为一正整数，则对任意正整数 h，有 $2^{n+h} + 2^{n+(h-1)} + \cdots + 2^{n+1} + 2^n = 2^{n+(h+1)} - 2^n$.

由引理可知，若 S 的二进制序列 $B(s)$ 中有任一子列 $B_i(s) = \overbrace{11\cdots\cdots1}^{h个1}$，则必有一对应的冗余子列 $B'_i(s) = 1\overbrace{00\cdots\cdots0}^{h-1个0}\bar{1}$，使得 $B_i(s) \cong B'_i(s)$.

定义 3. 设 $B'(s) = b_{t-1}b_{t-2}\cdots b_1b_0$ 为 S 的二进制冗余序列，若该序列中除去开头的 $b_{t-1}b_{t-2}b_{t-3}$ 三个元素有可能全是 1 外，其它任意彼此相邻的三个元素 $b_{i+1}b_ib_{i-1}(i = 1.\cdots,t-5)$ 中至少有一个为 0，则称 $B'(s)$ 为 S 的二进制冗余正则序列，简称为 S 的正则序列，记为 $\overline{B}(s)$.

§3. 冗余算法及讨论

设 R` 表示当前序列的实际状态，其真即为 1，否则为 0. $B(s) = a_{t-1}a_{t-2}\cdots a_1a_0$ 为 S 的二进制序列. 令

$$B(s)_t = \begin{cases} a_{t+3}a_{t+2}a_{t+1}\cdots a_1a_0, & \text{当 } t \text{ 为偶数时，} a_{t+3} = a_{t+2} = a_{t+1} = a_t = 0 \\ a_{t+2}a_{t+1}a_t\cdots a_1a_0, & \text{当 } t \text{ 为奇数时，} a_{t+2} = a_{t+1} = a_t = 0 \end{cases}$$

为输入.

算法一. (其中设 t 为偶数. 若 t 为奇数，则在算法一的循环语句中将 t 换成 t−1 即可)

Begin

R=0

For i=0 to t step 2 do[case ($a_{i+3}a_{i+2}a_{i+1}a_i$ and R) of]

If R'=R then

If $a_i = 0$ then

If $a_{i+1} = 0$ then begin $b_{i+1} = 0; b_i = 1$; R=0 end

Else if $a_{i+2} = 0$ then begin $b_{i+1} = 1; b_i = 1$;R=0 end

Else begin $b_{i+1} = 0; b_i = \bar{1}$; R=1 end

Else if $a_{i+1} = 1$ then begin $b_{i+1} = 0; b_i = 0$;R=1 end

Else if $a_{i+2} = 0$ then begin $b_{i+1} = 1; b_i = 0$; R=0 end

Else if $a_{i+3} = 0$ then begin $b_{i+1} = 1; b_i = 0$;R=0 end

Else begin $b_{i+1} = \bar{1}; b_i = 0$;R=1 end

Else if $a_i = 1$ and $a_{i+1} = 1$ and $a_{i+2} = 1$ then begin $b_{i+1} = 0; b_i = \bar{1}$;R=1 end

Else if $a_i = 0$ and $a_{i+1} = 1$ and $a_{i+2} = 1$ and $a_{i+3} = 1$ then begin $b_{i+1} = \bar{1}; b_i = 0$;R=1 end

Else begin $b_{i+1} = a_{i+1}; b_i = a_i$;R=0 end

If $b_t = 1\, and\, b_{t-1} = 0\, and\, b_{t-2} = 0\, and\, b_{t-3} = \overline{1}$ then $B'(s)_t = 111b_{t-4}b_{t-5}\cdots b_1b_0$

Else $B'(s)_{t+1} = b_tb_{t-1}\cdots b_1b_0$

End

定理. 设 S 为任一正整数, 对任一整数 $t \geq 0$ 则由算法一给出的序列 $B'(s)_t \cong \bar{B}(s)_t$.

证. 设 S 的二进制序列为 $B(s)_t = a_{t-1}a_{t-2}\cdots a_1a_0$, 序列长度为 t. 经由算法一得到的冗余序列为 $B'(s)_{t+1} = b_tb_{t-1}b_{t-2}\cdots b_1b_0$, 其中 b_t 为 1 或为 0.

对序列长度 t, 用数学归纳法证明. 当 t=2 时, 结论显然成立. 设 $t = k$ 时, 有 $B'(s)_k \cong \bar{B}(s)_k$.

$$\text{当 } t = k+2 \text{ 时,}\quad B(s)_{k+2} = \begin{cases} 00B(s)_k, \\ 01B(s)_k, \\ 10B(s)_k, \\ 11B(s)_k, \end{cases}$$

分下列两种情形讨论:

a. 若 $B(s)_k = B'(s)_k$, 则由归纳假设有输入

$$B'(s)_{k+2} = \begin{cases} 000000b_{k-1}b_{k-2}\cdots b_1b_0 \\ 000001b_{k-1}b_{k-2}\cdots b_1b_0 \\ 000010b_{k-1}b_{k-2}\cdots b_1b_0 \\ 000011b_{k-1}b_{k-2}\cdots b_1b_0 \end{cases} = \begin{cases} 000000b_{k-1}b_{k-2}\cdots b_1b_0, \\ \begin{cases} 00000110b_{k-3}b_{k-4}\cdots b_1b_0, \\ 000001110b_{k-4}b_{k-5}\cdots b_1b_0, \\ 0000011110b_{k-5}b_{k-6}\cdots b_1b_0, \end{cases} \\ 000010b_{k-1}b_{k-2}\cdots b_1b_0, \\ \begin{cases} 0000101110b_{k-5}b_{k-6}\cdots b_1b_0, \\ 00001110b_{k-3}b_{k-4}\cdots b_1b_0, \\ 000011110b_{k-4}b_{k-5}\cdots b_1b_0, \\ 0000111110b_{k-5}b_{k-6}\cdots b_1b_0. \end{cases} \end{cases}$$

经运算, 得输出

$$B'(s)_{k+2} = \begin{cases} b_{k-1}b_{k-2}\cdots b_1b_0, \\ \begin{cases} 110b_{k-3}b_{k-4}\cdots b_1b_0, \\ 1110b_{k-4}b_{k-5}\cdots b_1b_0, \\ 1000\overline{1}0b_{k-5}b_{k-6}\cdots b_1b_0, \end{cases} \\ 10b_{k-1}b_{k-2}\cdots b_1b_0, \\ \begin{cases} 1100\overline{1}0b_{k-5}b_{k-6}\cdots b_1b_0, \\ 1110b_{k-3}b_{k-4}\cdots b_1b_0, \\ 1000\overline{1}0b_{k-4}b_{k-5}\cdots b_1b_0, \\ 10000\overline{1}0b_{k-5}b_{k-6}\cdots b_1b_0. \end{cases} \end{cases}$$

由引理，有 $B'(s)_{k+2} \cong \bar{B}(s)_{k+2}$.

b. 若 $B(s)_k = B'(s)_k - 1$，此时因为在算法一变换中形成的加法进位，新序列的长度比原二进制序列长度多 1. $B'(s)_k$ 中增添一非零元素 b_k=1，由引理知 b_{k-1}=0，即有输入

$$B'(s)_{k+2} = \begin{cases} 00000010b_{k-2}b_{k-3}\cdots b_1b_0 \\ 00000110b_{k-2}b_{k-3}\cdots b_1b_0 \\ 00001010b_{k-2}b_{k-3}\cdots b_1b_0 \\ 00001110b_{k-2}b_{k-3}\cdots b_1b_0 \end{cases} = \begin{cases} 10b_{k-2}b_{k-3}\cdots b_1b_0, \\ 110b_{k-2}b_{k-3}\cdots b_1b_0. \\ 1010b_{k-2}b_{k-3}\cdots b_1b_0. \\ 1110b_{k-2}b_{k-3}\cdots b_1b_0. \end{cases}$$

此时有 $B'(s)_{k+2} \cong \overline{B}(s)_{k+2}$. 所以对任意的 t，有 $B'(s)_t \cong \bar{B}(s)_t$. 证毕.

算法一演示了由正整数 S 的二进制序列转换成等价的正则序列的全过程.

例. 设 S 为一正整数，其二进制序列为 $B(s)$=1101110101111001. 求与 $B(s)$ 等价的正则序列 $B'(s)$.

解. 由算法一，作输入 00001101110101111001. 初始状态为 R=0.

i=0. $a_3a_2a_1a_0 = 1001$, R′ ≠ R, 有 $b_1b_0 = 01$ 及 R=0.

i=2. $a_5a_4a_3a_2 = 1110$, R′ ≠ R, 有 $b_3b_2 = \bar{1}0$ 及 R=1.

i=4. $a_7a_6a_5a_4 = 0111$, R′ =R, 有 $b_5b_4 = 00$ 及 R=1,

i=6. $a_9a_8a_7a_6 = 0101$, R′ =R, 有 $b_7b_6 = 10$ 及 R=0,

i=8. $a_{11}a_{10}a_9a_8 = 1101$, R′ ≠ R, 有 $b_9b_8 = 01$ 及 R=0,

i=10. $a_{13}a_{12}a_{11}a_{10} = 0111$, R′ ≠ R, 有 $b_{11}b_{10} = 0\bar{1}$ 及 R=1,

i=12. $a_{15}a_{14}a_{13}a_{12} = 1101$, R′ =R, 有 $b_{13}b_{12} = \bar{1}0$ 及 R=1.

i=14. $a_{17}a_{16}a_{15}a_{14} = 0011$, R′ =R, 有 $b_{15}b_{14} = 00$ 及 R=1,

i=16. $a_{19}a_{18}a_{17}a_{16} = 0000$, R′ =R, 有 $b_{17}b_{16} = 01$ 及 R=0.

所以有 $B'(s) = 100\bar{1}00\bar{1}011000\bar{1}001 = 11100\bar{1}011000\bar{1}001$.

设 K=$GF(2^k)$ 为有限域. 为讨论方便，取域 K 上的椭圆曲线为 E: $y^2 + xy = x^3 + a_4x^2 + a_6$. 其中 $x, y, a_i \in$ K, (i = 4,6), $a_6 \neq 0$. 定义 [6]E 上的加法运算如下：

设 $P, Q \in E, P, Q \neq O_k$($O_k$ 为无穷远点). $P = (x_1, y_1), Q = (x_2, y_2)$. 则 P 的逆元 $\overline{P} = (x_1, y_1 + x_1)$，且 $P + \overline{P} = O_k$.

若 $Q \neq \overline{P}, Q \neq P$，则 $P + Q = (x_3, y_3)$，其中

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a_4, \quad y_3 = \lambda(x_1 + x_3) + x_3 + y_1, \quad \lambda = \frac{y_2 + y_1}{x_2 + x_1}.$$

若 $Q = P = (x_1, y_1), P \neq O_k$，则 $P + Q = 2P = (x_3, y_3)$，其中

$$x_3 = \lambda^2 + \lambda + a_4, \quad y_3 = x_1^2 + (\lambda + 1)x_3, \quad \lambda = x_1 + \frac{y_1}{x_1}.$$

特别地，对 $\forall P \in E$，若 $Q = O_k$，则 $P + Q = P$. 对实数 0，有 $0P = O_k$. 设 S 为任一正整数，则 [6]$SP = \overbrace{P + P + \cdots + P}^{S\text{个}P}, (-S)P = S\bar{P}$.

以下由算法二和算法三构成求 SP 的冗余算法.

设 $B'(s)=b_{t-1}b_{t-2}\cdots b_1b_0$ 是由算法一得到的 S 的正则序列, 则式 (1.1) 分为两步计算. 首先, 因为 $b_{t-1}\neq 0$, 故 $2^{t-1}P$ 是必须计算的. 所以先对 $P,2P,\cdots,2^{t-1}P$ 依次作倍点运算.

算法二.

```
Begin
   i=0;
   R[0]=P;
   for i=1 to t-1 do R[i]=2R[i-1]
End
```

然后, 据正则序列的定义, 至多作 $\left\lfloor\frac{2(t-1)}{3}\right\rfloor$ 次点的加法运算.

算法三.

```
Begin
T= O_k;
   for i=t-1 to 0 by -1 do
   if b[i]≠0 then T=T+b[i]R[i]
End
```

§4. 结　论

从上述讨论中我们知道, 由算法一可将任一二进制序列转换为与之等价的正则序列. 所谓的正则序列就是一个除去开头可能有的连续三个 1 外, 序列中其它部分不再出现三个 (或以上) 连续 1 的二进制冗余序列. 假定有二进制序列 $B(s)=a_{t-1}a_{t-2}\cdots a_1a_0$, 其中由三个 (或以上) 连续 1 形成的子列共有 j 个, 设为 $x_1,x_2,\cdots,x_{\rm j}$, 其长度分别为 $|x_1|,|x_2|\cdots,|x_{\rm j}|$, 即有 $|x_i|\geq 3,(i=1,\cdots,j)$. 由引理, 算法一所得到的正则序列相对应的 j 个子列中, 第一个子列减少了 $|x_1|-2$ 个 1, 第二个子列减少了 $|x_2|-2$ 个 $1,\cdots$, 第 j 个子列减少了 $|x_{\rm j}|-2$ 个 1. 再加上由二进制加法进位所产生的由三个 (或以上) 连续 1 形成的子列, 不妨设这样的子列在转换中减少了 h 个 1. 如此最后得到的正则序列中 1 的个数共减少了

$$(|x_1|-2)+(|x_2|-2)+\cdots+(|x_{\rm j}|-2)+h=(|x_1|+|x_2|+\cdots+|x_{\rm j}|+h-2j)\geq 3j+h-2j=j+h.$$

由式 (1.1) 计算 SP 的过程中, 就至少减少了 $(j+h)$ 次点的加法运算. 每次这样的运算又需要域 K 上的元素的 3 次乘法, 9 次加法和 1 次求逆. 这样当 j 和 h 较大时, 以上给出的算法所提高的效率是十分可观的.

参 考 文 献

[1] N.Kobliz, Elliptic curve cryptosystem[J], Mathematics of Computation, 48:177 (1987) 203-209.

[2] V.S. Miller. Use of Elliptic Curve in Cryptography[C]. Advances in Cryptology—CRYPTO'85 Proceeding, Springer-Verlag, 1986, 417-426

[3] Bruce Schneier. Applied Cryptography——Protocols, Algorithm and Source Code in C[M]. John Wiley & Sons,InC.1996. 中译本: 吴世忠等译. 应用密码学 —— 协议, 算法与源程序 [M]. 机械工业出版社, 北京, 2000.1.

[4] Certicom. http://www.certicom.com

[5] Shi Ronghua. A Redundannt Binary Algorithm for RSA[J]. J.of Comput. Sci. & Technol. 11:4 (1996) 416-420.

[6] Joseph H.Silverman. The Arithmetic of Elliptic Curves[M]. Springer-Verlag. New York. Inc. 1986