Journal of Computer Applications
计算机应用,2012,32(11):3121-3124
ISSN 1001-9081
CODEN JYIIDU
2012-11-01
http://www.joca.cn

文章编号:1001-9081(2012)11-3121-04
doi:10.3724/SP.J.1087.2012.03121

整数的带符号二进制表示数的快速计算

李　忠[1,2*],彭代渊[2]
(1.宜宾学院 计算机与信息工程学院,四川 宜宾 644000;　2.西南交通大学 信息科学与技术学院,成都 610031)
(*通信作者电子邮箱 lz8056859@163.com)

摘　要:整数的带符号数字表示广泛应用于计算机算术、密码学、数字信号处理等领域。一个长度为 n 比特的整数有多种带符号二进制表示。对整数的带符号二进制表示数的性质进行研究,给出了两个改进的非递归算法,所得算法能快速计算给定整数的给定长度的带符号二进制表示数,且降低了空间消耗。

关键词:整数;带符号二进制表示;表示数;递归算法;非递归算法
中图分类号:TP301.6　　**文献标志码**:A

Fast calculation of number of binary signed digit representations of an integer

LI Zhong[1,2*], PENG Dai-yuan[1]
(1. *School of Computer and Information Engineering, Yibin University, Yibin Sichuan* 644000, *China*;
2. *School of Information Science and Technology, Southwest Jiaotong University, Chengdu Sichuan* 610031, *China*)

Abstract: Binary Signed Digit (BSD) representation of an integer is widely used in computer arithmetic, cryptography and digital signal processing. An integer of length n bits can have several BSD representations. In this paper, the authors studied the properties of the number of BSD representation of an integer, and presented two improved non-recursion algorithms. They can rapidly calculate the exact number of BSD representations of an integer of a certain length, and the storage requirements get reduced.

Key words: integer; binary signed digit representation; number of representation; recursion algorithm; non-recursion algorithm

0　引言

文献[1-9]对整数的带符号数字表示及其在椭圆曲线密码体制(Elliptic Curve Cryptosystem,ECC)等领域的应用进行了研究。一个长度为 n 比特的整数有多种带符号二进制(Binary Signed Digit, BSD)表示,表示数依赖于整数的值及表示的长度。整数的非相邻形式(Non-Adjacent Form,NAF)表示[10]、互反形式(Mutual Opposite Form,MOF)表示[3]等均为符号二进制(Binary Signed Digit,BSD)表示,其中 NAF 表示是最具代表性的 BSD 表示,对椭圆曲线密码(Elliptic Curve Cryptography,ECC)的标量乘法具有十重要的意义。文献[1]对整数的 BSD 表示数进行了研究,给出计算整数的给定长度的 BSD 表示数计算算法。本文进一步研究整数的 BSD 表示数的性质,给出两个改进的算法,所得算法能快速计算给定整数的给定长度的 BSD 表示数,且降低了空间消耗。

1　整数的 BSD 表示数的基本性质

对于正整数 k、n,称 $k=k_{n-1}2^{n-1}+\cdots+k_1 2^1+k_0 2^0$(其中 $k_i\in\{-1,0,1\}$,$0\leqslant i\leqslant n$)为 k 的长度为 n 的 BSD 表示,并用 $\lambda(k,n)$ 表示整数 $k\in[0,2^n-1]$ 的长度为 n 的不同 BSD 表示数。文献[1]对 $\lambda(k,n)$ 进行了研究,获得了以下结论。

定理 1　对于整数的 BSD 表示有:

1) $\lambda(0,n)=1$;

2) $\lambda(1,n)=n$;

3) $\lambda(2^i,n)=n-i$。

定理 2　对于整数 k 有:

1) 若 $2^{n-1}\leqslant k\leqslant 2^n-1$,则 $\lambda(k,n)=\lambda(k-2^{n-1},n-1)$;

2) 若 k 是偶数,则 $\lambda(k,n)=\lambda(k/2,n-1)$;

3) 若 k 是奇数,则 $\lambda(k,n)=\lambda((k-1)/2,n-1)+\lambda((k+1)/2,n-1)$。

2　整数的 BSD 表示数的计算

根据定理 1、定理 2 得到如算法 1 所示的计算整数的 BSD 表示数的递归算法。

算法 1　基于递归方法的 $\lambda(k,n)$ 计算。

输入:$k\in[0,2^n-1]$,n;

输出:$\lambda(k,n)$。

if $k=0$ then $C=1$;

else if $k=1$ then $C=n$;

else if $k\geqslant 2^{n-1}$ then $C=\lambda(k-2^{n-1},n-1)$;

else if k is even then $C=\lambda(k/2,n-1)$;

else $C=\lambda\left(\frac{k-1}{2},n-1\right)+\lambda\left(\frac{k+1}{2},n-1\right)$;

return C.

利用算法 1 计算 $\lambda(k,n)$ 存在大量的重复计算,对于 ECC 应用的最小有限域 $GF(2^{163})$ 中的整数 $k=(k_{162}\cdots k_1 k_0)_2$,$k_{162}\neq 0$,计算 $\lambda(k,164)$ 的递归深度达 164,一般的个人数字助理(Personal Digital Assistant,PDA)、无线传感网(Wireless Sensor Network,WSN)设备根本无法运行(将产生空间溢出)。

为减少重复计算,文献[1]给出如算法 2 所示的计算整

收稿日期:2012-04-17;**修回日期**:2012-06-15。　**基金项目**:四川省教育厅重点科研资助项目(07ZA145)。
作者简介:李忠(1963-),男,四川安岳人,副教授,博士,主要研究方向:密码学、信息安全;　彭代渊(1955-),男,四川资阳人,教授,博士生导师,博士,主要研究方向:密码学、信息安全、编码理论。

数的 BSD 表示数的优化算法。

算法 2　优化的整数的 BSD 表示数的计算。

输入: $k \in [0, 2^n - 1], n$;

输出: $\lambda(k, n)$。

if $k = 0$ then $C = 1$;

else if $k = 1$ then $C = n$;

else if $k \geqslant 2^{n-1}$ then $C = \lambda(k - 2^{n-1}, n - 1)$;

else if k is even then $C = \lambda(k/2, n - 1)$;

else

if $k \equiv 1 \bmod 4$ then $C = \lambda 2((k-1)/2, (k+1)/2, 1, 1, n-1)$; //利用算法 3

else $C = \lambda 2((k+1)/2, (k-1)/2, 1, 1, n-1)$; //利用算法 3

return C.

算法 3　计算 $\lambda(k, n)$ 的辅助算法。

输入: k_e, k_o, w_e, w_e, n;

输出: $c = \lambda 2(k_e, k_o, w_e, w_o, n)$。

if $k_o = 1$ and $k_e = 2$ then $c = n * w_o + (n-1) * w_e$;

else

if $k_e \equiv 0 \bmod 4$ then

if $k_o \equiv 1 \bmod 4$ then $c = \lambda 2(k_e/2, k_e/2 + 1, w_o + w_e, w_o, n-1)$;

else $c = \lambda 2(k_e/2, k_e/2 - 1, w_o + w_e, w_o, n-1)$;

else

if $k_o \equiv 1 \bmod 4$ then $c = \lambda 2(k_e/2 - 1, k_e/2, w_o, w_o + w_e, n-1)$;

else $c = \lambda 2(k_e/2 + 1, k_e/2, w_o, w_o + w_e, n-1)$;

return c.

利用算法 2 计算 315 的长度为 12 的 BSD 表示数的执行过程如图 1 所示。

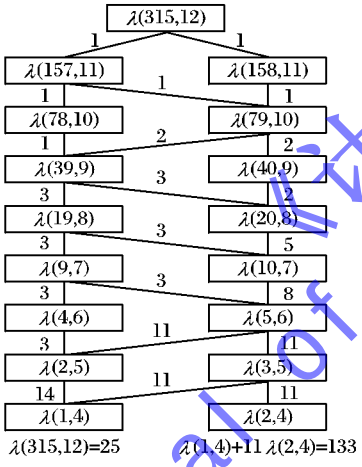

图 1　算法 2 求解 $\lambda(315, 12)$ 的计算过程

3　整数的 BSD 表示数的快速计算

算法 2 的效率明显高于算法 1，但算法 2 仍然使用了递归，且一次调用只处理整数的一个比特位，算法的时间、空间消耗仍然较大。事实上有：

定理 3　对于正整数 $n, k = (k_{n-1} \cdots k_1 k_0)_2 \in [0, 2^n - 1]$，有：

1) 若 $k_{s-1} \cdots k_1 k_0 = 0 \cdots 00$，令 $k' = (k_{n-1} \cdots k_{s+1} k_s)_2$，则 $\lambda(k, n) = \lambda(k', n - s)$；

2) 若 $k_{s-1} \cdots k_1 k_0 = 1 \cdots 11$，令 $k' = (k_{n-1} \cdots k_{s+1} k_s)_2$，则 $\lambda(k, n) = \lambda(k', n - s) + s\lambda(k' + 1, n - s)$。

证明

1) 反复利用定理 2 的 2) 可得结果；

2) 对 s 用数学归纳法进行证明。

① 当 $s = 1$，即 $k_0 = 1$ 时，有

$k' = (k_{n-1} k_{n-2} \cdots k_1)_2 = (k - 1)/2$

$k' + 1 = (k_{n-1} k_{n-2} \cdots k_1)_2 + 1 = \dfrac{k-1}{2} + 1 = \dfrac{k+1}{2}$

由定理 2 的 3) 有

$$\lambda(k, n) = \lambda\left(\frac{k-1}{2}, n-1\right) + \lambda\left(\frac{k+1}{2}, n-1\right) = \lambda(k', n-s) + s\lambda(k'+1, n-s)$$

所以，当 $s = 1$ 时，2) 成立。

② 假设当 $s = i - 1$，即 $k_{i-2} \cdots k_1 k_0 = 1 \cdots 11$ 时，2) 成立，即 $\lambda(k, n) = \lambda(k', n - i + 1) + (i-1)\lambda(k' + 1, n - i + 1)$，此时，$k' = (k_{n-1} \cdots k_i k_{i-1})_2$。

③ 当 $s = i$，即 $k_{i-1} k_{i-2} \cdots k_1 k_0 = 1 \cdots 11$ 时，令 $k = (k_{n-1} \cdots k_i k_{i-1})_2, k' = (k_{n-1} \cdots k_{i+1} k_i)_2$

由 ② 的假设有

$$\lambda(k, n) = \lambda(k, n - i + 1) + (i-1)\lambda(k + 1, n - i + 1) \tag{1}$$

由 $k = (k_{n-1} \cdots k_i k_{i-1})_2$，且 $k_{i-1} = 1$，有

$$\frac{k-1}{2} = (k_{n-1} \cdots k_{i+1} k_i)_2 = k', \frac{k-1}{2} = \frac{k-1}{2} + 1 = k' + 1$$

由定理 2 的 3) 有

$$\lambda(k, n - i + 1) = \lambda\left(\frac{k-1}{2}, n - i + 1 - 1\right) + \lambda\left(\frac{k-1}{2}, n - i + 1 - 1\right) = \lambda(k', n - i) + \lambda(k' + 1, n - i)$$

即

$$\lambda(k, n - i + 1) = \lambda(k', n - i) + \lambda(k' + 1, n - i) \tag{2}$$

由 $k = (k_{n-1} \cdots k_i k_{i-1})_2$ 及 $k_{i-1} = 1$ 知 $k + 1$ 为偶数，由定理 2 的 2) 有

$$\lambda(k + 1, n - i + 1) = \lambda\left(\frac{k-1}{2}, n - i + 1 - 1\right) = \lambda(k' + 1, n - i)$$

即

$$\lambda(k + 1, n - i + 1) = \lambda(k' + 1, n - i) \tag{3}$$

将式(2)、(3) 代入式(1) 得

$$\lambda(k, n) = \lambda(k, n - i + 1) + (i-1)\lambda(k + 1, n - i + 1) = \lambda(k', n - i) + \lambda(k' + 1, n - i) + (i-1)\lambda(k' + 1, n - i) = \lambda(k', n - i) + i\lambda(k' + 1, n - i)$$

所以，当 $s = i$ 时，2) 成立。

由数学归纳法原理可知，2) 成立。

定理 4　对于正整数 $n, k = (0 \cdots 01 \underbrace{1 \cdots 1}_{s_m} \underbrace{0 \cdots 0}_{s_{m-1}} \cdots \underbrace{1 \cdots 1}_{s_3} \underbrace{0 \cdots 0}_{s_2} \underbrace{1 \cdots 1}_{s_1})_2 \in [0, 2^n - 1]$，则

$$\lambda(k, n) = u_m(n - S) + v_m(n - S - 1)$$

其中，$S = \sum_{i=1}^{m} s_i$，且

$$u_i = \begin{cases} 1, & i = 1 \\ u_{i-1} + v_{i-1} s_i, & i \text{ 为偶数}, 1 < i \leqslant m \\ u_{i-1}, & i \text{ 为奇数}, 1 < i \leqslant m \end{cases}$$

$$v_i = \begin{cases} s_1, & i = 1 \\ v_{i-1}, & i \text{ 为偶数}, 1 < i \leqslant m \\ v_{i-1} + u_{i-1} \times s_i, & i \text{ 为奇数}, 1 < i \leqslant m \end{cases}$$

借助于定理 3 的 2)、定理 3，利用数学归纳法很容易证明定理 4 的结论。

由定理 4 可得如算法 4 所示的计算 $\lambda(k, n)$ 的非递归算法。

算法 4　计算 $\lambda(k, n)$ 的非递归算法。

输入: $k = (k_{l-1} \ldots k_1 k_0)_2, n$，其中 $k_{l-1} = 1, n \geqslant l$;

输出:$\lambda(k,n)$。
if $k=0$ then $c=1$;
else if $k=1$ then $c=n$;
$w_1=1, w_2=0, i=0, j=0$;
while $i<l-1$ and $k_i=0$ do
  $n=n-1, i=i+1$;
while $i<l-1$ and $k_i=1$ do
  $w_2=w_2+1,\ n=n-1, i=i+1$;
while $i<l-1$ do
  $j=0$;
  while $i<l-1$ and $k_i=0$ do
    $j=j+1, i=i+1$;
  $w_1=w_1+w_2*j, n=n-j$;
  $j=0$;
  while $i<l-1$ and $k_i=1$ do
    $j=j+1, i=i+1, w_2=w_2+w_1*j, n=n-j$;
$c=w_1*n+w_2*(n-1)$
return c;

利用算法 4 计算 315 的长度为 12 的 BSD 表示数的执行过程如图 2 所示。

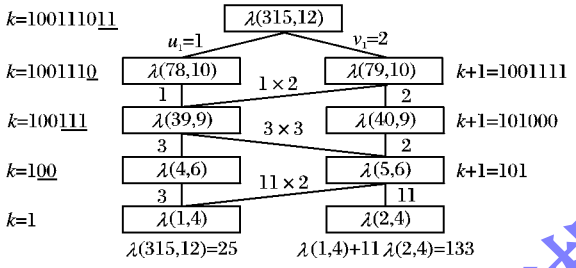

图 2　算法 4 计算 $\lambda(315,12)$ 的计算过程

由算法 4 及图 2 可知,算法 4 消除了递归,只须扫描一趟标量的基本二进制表示便可计算出 $\lambda(k,n)$,在循环迭代中一次处理整数的若干比特位,与算法 1 相比其效率有显著提高。

对于 NIST 推荐的 P-192 的基域 $GF(2^{192}-2^{64}-1)$ 的随机元素:

$k=$
$(8D11AE7A30E00345FF299DC52D3305C3103487D6565414E5)_{16}\in GF(2^{192}-2^{64}-1)$

$\lambda(k,193)=631692706283294261997761720714417$。而利用算法 1、算法 2 计算 $\lambda(k,193)$ 将产生空间溢出,得不到应有的结果。

定理 5　对于正整数 $n, k=(k_{n-1}\cdots k_1k_0)_2\in[0,2^n-1]$,有

1) 若 $k_{n-1}\cdots k_{s+1}k_s=0\cdots00, k_{s-1}\cdots k_1k_0=1\cdots11$,则 $\lambda(k,n)=s(n-s)+1$。

2) 若 $k=(k_{n-1}\cdots k_{s+t+1}k_{s+t}\underbrace{0\cdots00}_{t}\underbrace{1\cdots11}_{s})_2$,令 $k'=(k_{n-1}\cdots k_{s+t+1}k_{s+t})_2$,有

$$\lambda(k,n)=(s\times t+1)\lambda(k',n-s-t)+s\lambda(k'+1,n-s-t) \tag{4}$$

证明　1) 若 $k_{n-1}\cdots k_{s+1}k_s=0\cdots00, k_{s-1}\cdots k_1k_0=1\cdots11$,由定理 3 的 2) 有

$$\lambda(k,n)=\lambda(0,n-s)+s\lambda(1,n-s)=1+s(n-s)=s(n-s)+1$$

2) 令 $k=(k_{n-1}\cdots k_{s+t}k_{s+t-1}\cdots k_{s+1}k_s)_2$,由定理 3 有

$$\lambda(k,n)=\lambda(k,n-s)+s\lambda(k+1,n-s)=\lambda(k',n-s-t)+s\lambda(k+1,n-s)$$

于是,只需证明

$$\lambda(k',n-s-t)+s\lambda(k+1,n-s)=(s\times t+1)\lambda(k',n-s-t)+s\lambda(k'+1,n-s-t)$$

即

$$\lambda(k+1,n-s)=t\lambda(k',n-s-t)+\lambda(k'+1,n-s-t) \tag{5}$$

下面对 t 用数学归纳法证明式(5)的正确性。

① 当 $t=1$,即 $k_s=0$ 时,$k'=(k_{n-1}\cdots k_{s+2}k_{s+1})_2$,且 $k+1=(k_{n-1}k_{n-2}\cdots k_{s+1}1)_2$ 为奇数。

由定理 2 的 3) 有

$$\lambda(k+1,n-s)=\lambda(\frac{(k+1)-1}{2},n-s-1)+\lambda(\frac{(k+1)+1}{2},n-s-1),n-s-1)+\lambda(k'+1,n-s-1)$$

所以,当 $t=1$ 时,式(5)成立。

② 假设当 $t=i-1$,即 $k_{s+i-2}\cdots k_{s+1}k_s=0\cdots00$ 时,式(5)成立,即

$$k'=(k_{n-1}\cdots k_{s+i}k_{s+i-1})_2, k=(k_{n-1}\cdots k_{s+i-1}k_{s+i-2}\cdots k_{s+1}k_s)_2=(k_{n-1}\cdots k_{s+i-1}0\cdots00)_2$$

有

$$\lambda(k+1,n-s)=(i-1)\lambda(k',n-s-i+1)+\lambda(k'+1,n-s-i+1)$$

③ 当 $t=i$,即 $k_{s+i-1}\cdots k_{s+1}k_s=0\cdots00$ 时,有

$$k'=(k_{n-1}\cdots k_{s+i+1}k_{s+i})_2$$
$$k=(k_{n-1}\cdots k_{s+i}k_{s+i-1}k_{s+i-2}\cdots k_{s+1}k_s)_2=(k_{n-1}\cdots k_{s+i}00\cdots00)_2$$

令 $\bar{k}=(k_{n-1}\cdots k_{s+i}k_{s+i-1})_2$,此时 $k_{s+i-1}=0$。

由②的假设有

$$\lambda(\bar{k}+1,n-s)=(i-1)\lambda(\bar{k},n-s-i+1)+\lambda(\bar{k}+1,n-s-i+1) \tag{6}$$

由 $k_{s+i-1}=0$ 知,$\bar{k}$ 为偶数,$\bar{k}+1$ 为奇数,由定理 2 有:

$$\lambda(\bar{k},n-s-i+1)=\lambda(\frac{\bar{k}}{2},n-s-i+1-1)=\lambda(k',n-s-i) \tag{7}$$

$$\lambda(\bar{k}+1,n-s-i+1)=\lambda(\frac{(\bar{k}+1)-1}{2},n-s-i)+\lambda(\frac{(\bar{k}+1)+1}{2},n-s-i)=\lambda(k',n-s-i)+\lambda(k'+1,n-s-i) \tag{8}$$

将式(7)、式(8)代入式(6)有

$$\lambda(k+1,n-s)=(i-1)\lambda(k',n-s-i)+[\lambda(k',n-s-i)+\lambda(k'+1,n-s-i)]=i\lambda(k',n-s-i)+\lambda(k'+1,n-s-i)$$

所以,当 $t=i$ 时,式(5)成立。

由数学归纳法原理知,式(5)成立,进而式(4)成立

定理 6　对于正整数 $n, k\in[0,2^n-1]$,若 $k=(\underbrace{0\cdots0}_{t_m}\underbrace{1\cdots1}_{s_m}\cdots\underbrace{0\cdots0}_{t_2}\underbrace{1\cdots1}_{s_2}\underbrace{0\cdots0}_{t_1}\underbrace{1\cdots1}_{s_1})_2$,且 $\sum_{i=1}^{m}(s_i+t_i)=n$,则

$$\lambda(k,n)=u_{m-1}(s_mt_m+1)+v_{m-1}t_m$$

其中

$$u_i=\begin{cases}1, & i=0\\ u_{i-1}(s_it_i+1)+v_{i-1}t_i, & 0<i<m\end{cases}$$

$$v_i=\begin{cases}0, & i=1\\ v_{i-1}+u_{i-1}s_i, & 0<i<m\end{cases}$$

借助于定理 3、定理 1 的 3),利用数学归纳法很容易证明

定理 6 的结论。

由定理 6 可得如算法 5 所示的计算 $\lambda(k,n)$ 的快速算法。

算法 5　计算 $\lambda(k,n)$ 的快速算法。

输入: $n, k=(k_{n-1}\ldots k_1k_0)_2\in[0,2^n-1]$;
输出: $\lambda(k,n)$。
if $k\geqslant 2^{n-1}$ then $k_{n-1}=0, n=n-1$;
else if $k=0$ then $c=1$;
else if $k=1$ then $c=n$;
else
  find the smallest i shch that $k_i=1$;　　//寻找首非 0 元
  $n=n-i, len=n, u=1, v=0$;　　//跳过最右边的'0'
  while $i<len$ do
    $s=1, t=1$;
    while $i<len$ and $k_i=1$ do $s=s+1$;
    while $i<len$ and $k_i=0$ do $t=t+1$;
    if $i<len$ then $u=u(st+1)+tv, v=us+v$;
  $c=u(st+1)+vt$;
return c;

利用算法 5 计算 315 的长度为 12 的 BSD 表示数的执行过程如图 3 所示。

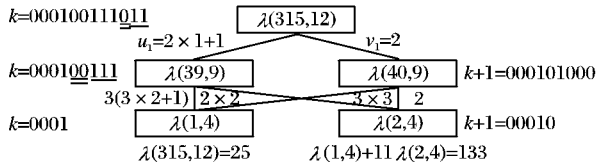

图 3　算法 5 计算 $\lambda(315,12)$ 的计算过程

4　整数的所有 BSD 表示的生成

由前述定理可知，对于任意整数 $k\in[0,2^n-1]$，可按以下方式构造其长度为 n 的所有 BSD 表示：

1）若 $k=0$，则只有一种表示：$(\underbrace{0\cdots00}_{n})_2$。

2）若 $k=1$，则有 n 种表示：$(\underbrace{0\cdots001}_{n-t-1}\underbrace{\bar{1}\cdots\bar{1}}_{t})_2, 0\leqslant t<n$。

3）若 $k=2^i$，则有 $n-i$ 种表示：$(\underbrace{0\cdots001}_{n-i-t-1}\underbrace{\bar{1}\cdots\bar{1}}_{t}\underbrace{0\cdots0}_{i})_2, 0\leqslant t<n-i$。

4）若 $2^{n-1}\leqslant k\leqslant 2^n-1$，令 $d=k-2^{n-1}$，则有 $\lambda(d,n-1)$ 种表示：在 d 的 $n-1$ 位表示前加"1"即可。

5）若 k 是偶数，则有 $\lambda(k/2,n-1)$ 种表示：在 $k/2$ 的 $n-1$ 位表示后加"0"即可。

6）若 k 是奇数，则有 $\lambda(k-1,n)+\lambda(k+1,n)$ 种表示：将 $k-1$ 的 n 位表示的最后一位"0"换成"1"，将 $k+1$ 的 n 位表示的最后一位"0"换成"$\bar{1}$"即可。其中，$k-1$ 的 n 位表示在(5)中已求出，只须求出 $k+1$ 的 n 位表示即可。

5　结语

本文对给定整数的带符号二进制表示数的性质进行了深入研究，给出了两个改进的算法，由图 1 ~ 图 3 可知，所得算法能快速计算给定整数的给定长度的带符号二进制表示数；同时，给出了构造 $k\in[0,2^n-1]$ 的长度为 n 的所有 BSD 表示的方法。

参考文献：

[1] EBEID N, HASAN M A. On binary signed digit representations of integers[J]. Design Code Cryptogr, 2007, 42(1): 43 - 65.

[2] JOYE M, YEN S M. Optimal left-to-right binary signed-digit recoding [J]. IEEE Transactions on Computers, 2000, 49(7): 740 - 748.

[3] AVANZI R M. A note on the signed sliding window integer recoding and a left-to-right analogue[C]// Proceedings of the Selected Areas in Cryptography, LNCS 3357. Berlin: Springer, 2004: 130 - 143.

[4] OKEYA K, SCHMIDT-SAMOA K, SPAHN C, *et al*. Signed binary representations revisited[C]// CRYPTO 2004: Advances in Cryptology, LNCS 3152. Berlin: Springer, 2004: 123 - 139.

[5] KHABBAZIAN M, GULLIVER T A, BHARGAVA V K. A new minimal average weight representation for left-to-right point multiplication methods[J]. IEEE Transactions on Computer, 2005, 54(11): 1454 - 1459.

[6] BALASUBRAMANIAM P, KARTHIKEYAN E. Elliptic curve scalar multiplication algorithm using complementary recoding[J]. Applied Mathematics and Computation, 2007, 190(1): 51 - 56.

[7] WANG BANGJU, ZHANG HUAN-GUO, WANG ZHANGYI, *et al*. Speeding up scalar multiplication using a new signed binary representation for integers[C]// MCAM'07: Proceedings of the 2007 international conference on Multimedia content analysis and mining, LNCS 4577. Berlin: Springer, 2007: 277 - 285.

[8] QIN BAODONG, LI MING, KONG FANYU, *et al*. New left-to-right minimal weight signed-digit radix-r representation[J]. Computers and Electrical Engineering, 2009, 35(1): 150 - 158.

[9] WU TING, ZHANG MIN, DU HUANQIANG, *et al*. On optimal binary signed digit representations of integers[J]. Applied Mathematics, 2010, 25(3): 331 - 340.

[10] HANKERSON D, MENEZES A, VANSTONE S. Guide to elliptic curve cryptography[M]. Berlin: Springer-Verlag, 2004.

(上接第 3114 页)

[11] RAMACHANDRAN I, DAS A K, ROY S. Analysis of the contention access period of IEEE 802.15.4 MAC[J]. ACM Transactions on Sensor Networks, 2007, 3(1): 70 - 77.

[12] PARK T R, KIM T H, CHOI J Y, *et al*. Throughput and energy consumption analysis of IEEE 802.15.4 slotted CSMA/CA [J]. Electronics Letters, 2005, 41(18): 1017 - 1019.

[13] ZHAI H Q, CHEN X, FANG Y G. Improving transport layer performance in multihop Ad Hoc networks by exploiting MAC layer information[J]. IEEE Transactions on Wireless Communication, 2007, 6(5): 1692 - 1701.

[14] NG P C, LIEW S C. Throughput analysis of IEEE 802.11 multihop Ad Hoc networks[J]. IEEE Transactions on Networking, 2007, 15(2): 309 - 322.

[15] LI XIANG-YANG. Multicast capacity of wireless Ad Hoc networks [J]. IEEE/ACM Transactions on Networking, 2009, 17(3): 950 - 962.

[16] 龙图景，孙政顺，李春文，等. 一种新的网络业务流的多重分形小波模型[J]. 计算机学报，2004，27(8)：1074 - 1082.

[17] ROECKER J A. A class of near optimal JPDA algorithm[J]. IEEE Transactions on Aerospace and Electronic Systems, 1994, 30(2): 504 - 510.

[18] CHALLA S, EVANS R J, WANG X, *et al*. A fixed lag smoothing solution to out-of-sequence information fusion problems [J]. Communications in Information and Systems, 2002, 2(4): 325 - 348.

[19] HE X, YENER A. Cooperation with an untrusted relay: A secrecy perspective[J]. IEEE Transactions on Information Theory, 2010, 56(8): 3807 - 3827.