新型有效的秘密共享方案

石润华[1,2]，黄刘生[2,3]，杨威[2,3]，仲红[1]

(1. 安徽大学 计算机科学与技术学院，安徽 合肥 230039；

2. 中国科学技术大学 计算机科学与技术系国家高性能计算中心，安徽 合肥 230026；3. 中国科学技术大学 苏州研究院，江苏 苏州 215123)

摘 要：提出了一种新的秘密共享方案。该方案分两层实现：上层，基于 Stern-Brocot 树把一个大的秘密拆分为 *t* 个小整数（子秘密）；底层，借鉴一维元胞自动机模型中的进化方法，把上层的 *t* 个子秘密作为初始状态，动态生成各参与者的共享。特别地，该方案能够动态扩展参与者，动态调整门限值，动态更新秘密和共享。另外，还具有计算简单，各参与者共享份额短的优点。分析结果表明，该方案安全、有效。

关键词：秘密共享；门限；动态；Stern-Brocot 树

中图分类号：TP309 **文献标识码**：A **文章编号**：1000-436X(2012)01-0010-07

Novel and effective secret sharing scheme

SHI Run-hua[1,2], HUANG Liu-sheng[2,3], YANG Wei[2,3], ZHONG Hong[1]

(1. School of Computer Science and Technology, Anhui University, Hefei 230039, China;

2. NHPCC, Depart. of CS. & Tech., USTC, Hefei 230026, China; 3.Suzhou Institute for Advanced Study, USTC, Suzhou 215123, China)

Abstract: A novel secret sharing scheme was proposed. This scheme consisted of two layer protocols: in the first layer, a larger secret was split into *t* smaller integers (sub-secrets) based on the Stern-Brocot tree; in the lower layer, *t* sub-secrets obtained from the first layer were regarded as *t* initial states in one-dimensional cellular automaton model, and then from the *t* initial states it could dynamic create all participants' shares according to the simple fixed rule. This scheme could dynamic add new member, adjust the threshold value and renew the secret and the shares. Besides, there were still other advantages that the costs of the computation were very low and the size of the shares was very small. The results of analysis show that it was secure and very efficient.

Key words: secret sharing; threshold; dynamic; Stern-Brocot tree

1 引言

秘密共享在现实生活中有着非常重要的应用。例如，假定有一个银行主管 Alice 想将某个秘密传给远方的代理 Bob 和 Charlie，从而能够共同执行某个秘密任务。但是，Alice 并不完全信任 Bob 或 Charlie，且不知道谁是不诚实的一方。然而，Alice 知道如果他们 2 人合作来完成这个任务，诚实的一方将会阻止不诚实的一方破坏该任务。根据秘密共享方法，Alice 可以将秘密分割成 2 部分，分别秘密发送给 Bob 和 Charlie。使得 Bob 和 Charlie 任何一方都不可能得到秘密，但是 2 人合作能够恢复出原

收稿日期：2010-01-16；**修回日期**：2010-12-20

基金项目：国家自然科学基金资助项目（61173187, 61173188）；安徽省自然科学基金资助项目（11040606M141）；安徽高校省级重点自然科学研究基金资助项目（KJ2010A009）；安徽大学 211 工程基金资助项目

Foundation Items: The National Natural Science Foundation of China (61173187, 61173188); The Natural Science Foundation of Anhui Province (11040606M141); Research Program of Anhui Province Education Department (KJ2010A009); Thr 211 Project of Anhui University

始秘密。

秘密共享方案，最早是由 Blakley[1]和 Shamir[2]各自独立提出的。(t,n)门限方案是它的最初形式。在(t,n)门限方案中，秘密持有者把秘密拆分成n份（称为共享或影子），分别交由n个参与者秘密保存。任意t个或多于t个参与者堆积他们的共享就可以恢复秘密，而任意少于t个参与者却不能。接着，Ito 等[3]和 Benaloh 等[4]提出了更一般意义上的秘密共享方案。在现代密码学中，这些方案有着非常重要的应用，诸如在密钥分发、存取控制、安全多方计算、电子商务等方面。所以它一经提出就备受关注。

如果对于任意非授权的子组不仅不能恢复秘密，而且不能得到任何有关秘密的信息，这样的方案称之为完备（perfect）的秘密共享方案。另外，定义信息率为共享秘密的位长与参与者共享的比特数之比，即：$R_{\mathrm{p}}=\dfrac{\text{size of the shared secret}}{\text{size of the participant's share}}$

一般地，信息率小于 1，而理想情况是信息率$R_p=1$。因此称信息率等于 1 的秘密共享方案为理想（ideal）的秘密共享方案。

先回顾一下，最广泛应用的 Shamir 秘密共享方案：密钥分发者（dealer）持有一个秘密$S\in \mathrm{GF}(q)$（其中q是一个大素数，GF(q) 是有限 Galois 域）。为了防止秘密泄露、丢失，要在n个参与者$P_1,P_2,\cdots,P_n$中分享。并假定每个参与者有一个唯一的标识$x_i\in \mathrm{GF}(q)$。当恢复秘密时，需要至少t个参与者协作完成。

秘密分发：由秘密分发者（dealer）执行。

①Dealer 随机生成$t-1$次多项式：

$$f(x)=S+a_1x+a_2x^2+\cdots+a_{t-1}x^{t-1}\bmod q \qquad (1)$$

其中，S是将要分享的秘密（$S\in \mathrm{GF}(q)$），$a_1,a_2,\cdots,a_{t-1}\in \mathrm{GF}(q)$随机生成。②Dealer 计算$n$个共享：$s_i=f(x_i)\bmod q$，$i=1,2,\cdots,n$；并通过秘密信道把共享$s_i$分发给参与者$P_i$（$i=1,2,\cdots,n$）。③Dealer 销毁秘密$S$。

秘密重构：假定秘密分发者（dealer）作为密钥重构者。Dealer 收集至少t个参与者秘密发送来的共享$s_{i_1},s_{i_2},\cdots,s_{i_t}$（其中$i_j\in\{1,2,\cdots,n\}$）后，计算秘密多项式：

$$f(x)=\sum_{j=1}^{t}s_{i_j}\left(\prod_{\substack{1\leqslant k\leqslant t\\ k\neq j}}\frac{x-x_{i_j}}{x_{i_k}-x_{i_j}}\right)\bmod q \qquad (2)$$

从而重构出秘密$S=f(0)\bmod q$。

以上 Shamir 共享方案是一种完备、理想的秘密共享方案，但在秘密分发和重构过程中需要多次进行大整数的模幂、模乘运算。而在有些情况下，例如，在移动 ad hoc 网络[5]中的各移动节点或各种智能卡芯片[6]，其计算和存储能力相对较低，甚至通信带宽又受限的条件下，对于大整数的模幂、模乘运算，负担过重，势必影响其更广泛的应用。又例如，不仅文本数据可以作为秘密进行分享，图像也可以作为秘密进行分享[7]。而一幅大的图像数据较多，若采用 Shamir 方案，其开销也很庞大。所以有必要探寻其他更高效的秘密共享方案。基于这样的动机，本文提出并设计了一种新型高效、灵活的秘密共享方案。分析结果表明，该方案安全、有效，特别适宜于各种无线传感器网络或智能卡设备，以及类似于图像，数据较多的秘密进行分享。

2　基于 Stern-Brocot 树的秘密共享方案

2.1　Stern-Brocot 树

这一节简要介绍 Stern-Brocot 树及其基本属性[8]。若从 2 个分数 0/1、1/0 开始，在 2 个邻近的分数m_1/m_2和m_1'/m_2'之间插入$(m_1+m_1')/(m_2+m_2')$，那么就可得到 0/1、1/1、1/0。重复这个过程，得到 0/1、1/2、1/1、2/1、1/0，然后有 0/1、1/3、1/2、2/3、1/1、3/2、2/1、3/1、1/0。整个数组能视为无限二叉树结构，它的顶部级如图 1 所示。每个分数是m_1+m_1'/m_2+m_2'，其中m_1/m_2是左边的上面的最接近的祖先，m_1'/m_2'是右边的上面的最接近的祖先。

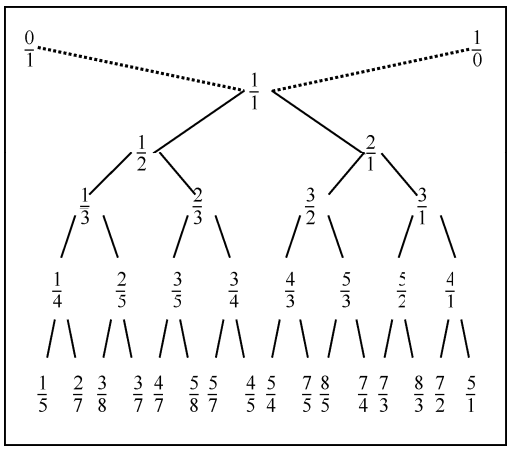


图 1　Stern-Brocot 树

事实上能把 Stern-Brocot 树视为代表有理数的一个数系，因为每个正的，简约分数恰好出现一次。

当从树的根转到一个特殊分数时，用字母 L 和 R 表示下到左分支或右分支，于是 L 和 R 的一个串唯一确定树中的一个位置。例如 *RLRL* 意味着从 1/1 下到右边的 2/1，然后下到左边的 3/2，然后下到右边的 5/3，然后下到左边的 8/5。能把 *RLRL* 考虑为 8/5 的一种表示法。

2.2 方案描述

从 Stern-Brocot 树的基本属性可以看出，每一个正分数以此方式表示为唯一的一个 L 和 R 的串。反过来，任意一个 L 和 R 的串表示一个正分数。已知正整数 m_1 和 m_2（$m_1 \perp m_2$），在 Stern-Brocot 树上，通过二分搜索方法，可以得到分数 m_1/m_2 的 *LR* 串，见算法 1。

算法 1 LR representation of the fraction m_1/m_2

while $m_1 \neq m_2$ do
  if $m_1 < m_2$ then
    {output(L); $m_2 \leftarrow m_2 - m_1$}
  else
    {output(R); $m_1 \leftarrow m_1 - m_2$}.

进而，若字符 L 表示数字“1”，而字符 R 表示数字“0”，则 m_1/m_2 的 *LR* 串能唯一地对应一个二进制数，具体见算法 2。

算法 2 Transforming (m_1, m_2) into S

Input: (m_1, m_2)
Output: S
$j \leftarrow 0$;
while $m_1 \neq m_2$ do
{ if $m_1 < m_2$ then
    { $S[j] \leftarrow 1$; $m_2 \leftarrow m_2 - m_1$ }
  else
    { $S[j] \leftarrow 0$; $m_1 \leftarrow m_1 - m_2$ }
  j++;
}
Return (S)

反过来，任意 0、1 串 S 对应一对互素的正整数 m_1, m_2，见算法 3。

算法 3 Splitting S into (m_1, m_2)

Input: $S[N]$; // where $S[i] \in \{0,1\}$
Output: (m_1, m_2)
$m_{1\_left} \leftarrow 0$;
$m_{2\_left} \leftarrow 1$;
$m_{1\_right} \leftarrow 1$;
$m_{2\_right} \leftarrow 0$;
$m_1 \leftarrow 1$;
$m_2 \leftarrow 1$;
for i from $N-1$ down to 0 do
{
  if $S[i]=1$ then
    { $m_{1\_right} \leftarrow m_1$; $m_{2\_right} \leftarrow m_2$;}
  else
    { $m_{1\_left} \leftarrow m_1$; $m_{2\_left} \leftarrow m_2$;}
    $m_1 \leftarrow m_{1\_left} + m_{1\_right}$;
    $m_2 \leftarrow m_{2\_left} + m_{2\_right}$
}
Return (m_1, m_2)

根据算法 3，可以把一个秘密 S 拆分为 2 个互素的正整数：m_1、m_2；反过来，根据算法 2，若已知整数 m_1、m_2 的值，则能计算出秘密 S 的值。这里，整数 m_1、m_2 的比特数远小于秘密 S 的比特数（参见 2.3 节定理 1、定理 2）。同样，继续调用算法 3，m_1、m_2 各自可以拆分为更小的整数 (m_{11}, m_{12})，(m_{21}, m_{22})。重复这个过程，能得到序列 m_{111}、m_{112} m_{121}、m_{122}、m_{211}、m_{212}、m_{221}、m_{222}（如图 2 所示）。这样，一个大的秘密 S 就被拆分成若干个较小的整数。反过来，重复调用算法 2，依照相反的计算次序，可以由这些小整数计算出原始的秘密 S。

例如，

$$S \Rightarrow \begin{cases} m_1 \Rightarrow \begin{cases} m_{11} \Rightarrow \begin{cases} m_{111} \\ m_{112} \end{cases} \\ m_{12} \Rightarrow \begin{cases} m_{121} \\ m_{122} \end{cases} \end{cases} \\ m_2 \Rightarrow \begin{cases} m_{21} \Rightarrow \begin{cases} m_{211} \\ m_{212} \end{cases} \\ m_{22} \Rightarrow \begin{cases} m_{221} \\ m_{222} \end{cases} \end{cases} \end{cases}$$

$S = B2759B5D$；$m_1 = DFE41$；$m_2 = 17D5DF$
$m_{11} = 15A$；$m_{12} = 3E9$；$m_{21} = 67E$；$m_{22} = C01$
$m_{111} = 31$；$m_{112} = 50$；$m_{121} = C$；$m_{122} = 43$
$m_{211} = 1C$；$m_{212} = 45$；$m_{221} = 13$；$m_{222} = 28$

图 2 原始秘密拆分成若干个小整数

显然，以上共享方案并不是门限方案，为此，做以下改进。整个秘密共享分 2 层实现：上层基于 Stern-Brocot 树把一个原始秘密拆分成若干个小整

数；底层再采用门限技术把这些小整数在所有参与者中分享。

这里，借鉴一维元胞自动机模型中进化的思想，把“从秘密生成所有参与者共享的过程”看作是一个进化过程，每个共享看作是一个状态，并且该状态仅与以前的连续 t 个状态有关，这样做的好处是容易扩展新成员和动态更新各共享。先定义如下序列：

$$m_1, m_2, m_3, \cdots, m_t, s_1, s_2, \cdots, s_n \tag{3}$$

其中，$m_1, m_2, \cdots, m_t$ 是由原始秘密 S 多次执行算法 3 而生成的一组小整数（上层分享）。例如，图 2 所示，这里假定 $t = 2^k$。而 s_i（$1 \leqslant i \leqslant n$）均由它前面连续的 t 项相加模 p 得来，p 是大于但又最接近 2^q 的素数，其中 $q = \lfloor 0.694^k N \rfloor$（参见 2.3 节定理 2）。即有：

$$\begin{cases} s_1 = (m_1 + m_2 + m_3 + \cdots + m_t) \bmod p \\ s_2 = (m_2 + m_3 + \cdots + m_t + s_1) \bmod p \\ \quad\vdots \qquad\qquad\qquad \vdots \\ s_n = (s_{n-t} + s_{n-t+1} + \cdots + s_{n-1}) \bmod p \end{cases} \tag{4}$$

这样，由 $m_1, m_2, \cdots, m_t$ 就可以唯一确定式(3)序列，而且 n 可以不固定任意扩展。显然，这个过程可逆。即在序列(3)中，任意 t 个连续的 s_i 经过若干次模减运算，最终可以反方向计算得到 $m_1, m_2, \cdots, m_t$。

得到 $m_1, m_2, \cdots, m_t$ 后，依照相反的计算顺序，多次执行算法 2，即可计算出原始秘密 S。这样，若把 s_i $(1 \leqslant i \leqslant n)$作为共享分发给用户 P_i，那么 t 个“连续”的用户联合起来就可以恢复出秘密 S。

2.3 方案分析

定理 1　在第 N 级 Stern-Brocot 树上，最大的分子或分母等于 F_{N+1}，其中 F_{N+1} 为第 $N+1$ 个斐波那契（Fibonacci）数。

证明　从图 1 容易看出，若 $S = 10101010\cdots$，则 S 所对应的简约分数 m_1 / m_2 中的分母 $m_2 = F_{N+1}$，而且是在 Stern-Brocot 树的 N 级中分母最大的数。其中，N 为 S 的比特数。相对应地，若 $S = 01010101\cdots$，则 S 所对应的简约分数 m_1 / m_2 中的分子 $m_1 = F_{N+1}$，而且是在 Stern-Brocot 树的 N 级中分子最大的数。

定理 2　若 l 表示正整数 m_1，m_2 中较大者的比特数，其中 m_1，m_2 是“具有 N bit 的秘密 S 按照算法 3 拆分而成”的正整数。则 $l \leqslant \lfloor 0.694N \rfloor$ 成立。

证明　不妨假定 S 位串中最左边数字为“1”，则 $m_1 < m_2$。而根据定理 1，有 $m_2 \leqslant F_{N+1}$。既然，

$$F_N = \frac{1}{\sqrt{5}}(\varphi^N - \hat{\varphi}^N) \tag{5}$$

其中，$\varphi = \dfrac{1+\sqrt{5}}{2}$，$\hat{\varphi} = \dfrac{1-\sqrt{5}}{2}$。所以，

$$F_{N+1} = \frac{1}{\sqrt{5}}(\varphi^{N+1} - \hat{\varphi}^{N+1}) \tag{6}$$

$$F_{N+1} \approx \frac{\varphi^{N+1}}{\sqrt{5}} \quad (\text{当 } N \gg 0, \left|\hat{\varphi}^N / \sqrt{5}\right| \approx 0) \tag{7}$$

$$\begin{aligned} \mathrm{lb} F_{N+1} &= (N+1)\mathrm{lb}\varphi - \frac{1}{2}\mathrm{lb}5 \\ &= 0.694N - 0.4667 \end{aligned} \tag{8}$$

显然，$l \leqslant \lfloor 0.694N \rfloor$。

定理 3　根据算法 2，仅仅知道小整数 m_1 或 m_2，猜出原始秘密 S 的最大概率为 $\dfrac{1}{4 \times \varphi(m_{\max})}$，当秘密 S 很大时，这在计算上是不可行的，即该方案是计算上安全的。其中 $m_{\max}$ 为 m_1 和 m_2 中较大者，而 $\varphi(\cdot)$ 为欧拉函数。

证明　根据算法 2，计算原始秘密 S，必须知道 m_1 和 m_2 的值，且还要知道它们的顺序（是分子还是分母）。假定参与方分别为甲方和乙方，他们所拥有的子秘密分别为 m_1、m_2，且假定 $m_1 < m_2$。显然，$H(S \mid m_2) > H(S \mid m_1)$，即乙方由 m_2 所获得的有关秘密 S 的信息量要比甲方所获得的多，其中 $H(\cdot)$ 为香农熵（Shannon entropy）。下面重点来讨论“乙方根据子秘密 m_2 穷猜出原始秘密 S 的概率”。首先，乙方有 $\dfrac{1}{2}$ 的概率猜出其所保留的子秘密是分子还是分母；其次，再有 $\dfrac{1}{2}$ 的概率正确猜出 $m_1 < m_2$；然后根据 m_1 和 m_2 互素的信息，能正确猜出 m_1 的值的概率为 $\dfrac{1}{\varphi(m_2)}$，其中 $\varphi(\cdot)$ 为欧拉函数，即 $\varphi(m_2)$ 表示小于 m_2 且与 m_2 互素的正整数的个数。因而，她猜出原始秘密 S 的总的概率为 $\dfrac{1}{2} \times \dfrac{1}{2} \times \dfrac{1}{\varphi(m_2)} = \dfrac{1}{4 \times \varphi(m_2)}$。通常秘密 S 是一个很大的数，因此 $\varphi(m_2)$ 也是一个相对较大的数。显然，要想正确猜出 m_1 并最终计算出 S，其概率很小，这在计算上是不可行的，即该方案是计算

上安全的。

另外，当原始秘密进行多次迭代，拆分为多个小整数 $m_1,m_2,\cdots,m_t$ 时，相互间的计算顺序是保密的。因此，单凭某个 m_i，正确猜出原始秘密的概率更低。也即在实际应用中，正确猜出原始秘密是不可行的。

定理 4 在式(3)序列中，t 个连续的共享可以正确计算出 $m_1,m_2,\cdots,m_t$，从而能够构造出原始秘密 S，但小于 t 个共享则不能。

证明 实际上，重构秘密的主要原理是：基于 t 个线性方程求解 t 个未知变量，当系数矩阵的行列式不等于零时有唯一解。不妨假定 $s_i,s_{i+1},\cdots,s_{i+t-1}$ 是已知的 t 个“连续”的共享，并且式(3)序列为：$m_1,m_2,m_3,\cdots,m_t$，$s_1,s_2,\cdots,s_{i-t},s_{i-t+1},\cdots,s_{i-1},s_i,s_{i+1},\cdots,s_{i+t-1},\cdots,s_n$。根据式(4)，$s_i,s_{i+1},\cdots,s_{i+t-1}$ 可由 $s_{i-t},s_{i-t+1},\cdots,s_{i-1}$ 线性表示，且有

$$\begin{bmatrix} s_i \\ s_{i+1} \\ \vdots \\ s_{i+t-1} \end{bmatrix} = \boldsymbol{A} \begin{bmatrix} s_{i-t} \\ s_{i-t+1} \\ \vdots \\ s_{i-1} \end{bmatrix} \tag{9}$$

不难发现：

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \cdots \\ 1 & 2 & 2 & 2 & 2 & \cdots \\ 2 & 3 & 4 & 4 & 4 & \cdots \\ 4 & 6 & 7 & 8 & 8 & \cdots \\ 8 & 12 & 14 & 15 & 16 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \end{bmatrix} \tag{10}$$

设 $\boldsymbol{A}_i$ 表示矩阵 $\boldsymbol{A}$ 的第 i 行向量，例 $\boldsymbol{A}_2 = [1 \quad 2 \quad 2 \quad 2 \quad 2 \quad \cdots]$。则有：

$$\begin{aligned} &2\boldsymbol{A}_1 - \boldsymbol{A}_2 = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad \cdots] \\ &2\boldsymbol{A}_2 - \boldsymbol{A}_3 = [0 \quad 1 \quad 0 \quad 0 \quad 0 \quad \cdots] \\ &\qquad\qquad \vdots \\ &2\boldsymbol{A}_{t-1} - \boldsymbol{A}_t = [0 \quad 0 \quad \cdots \quad 0 \quad 1 \quad 0] \\ &\boldsymbol{A}_t - \boldsymbol{A}_1 - \boldsymbol{A}_2 - \cdots - \boldsymbol{A}_{t-1} = [0 \quad 0 \quad 0 \quad \cdots \quad 0 \quad 1] \end{aligned} \tag{11}$$

故 $\boldsymbol{A}$ 经过若干次矩阵的初等变换后可得到单位矩阵 $\boldsymbol{I}$，也即矩阵 $\boldsymbol{A}$ 可逆。因而，在方程组(9)中，已知 $s_i,s_{i+1},\cdots,s_{i+t-1}$ 反过来可唯一求解出序列 $s_{i-t},s_{i-t+1},\cdots,s_{i-1}$。以此类推，反方向可以计算出 $m_1,m_2,\cdots,m_t$。实际上，由 t 个连续的共享计算原始的 t 个子秘密时，只需反方向进行若干次模减运算，并不需要真的求解线性方程组。再根据由 S 到 $m_1,m_2,\cdots,m_t$ 的计算顺序，反方向调用算法 2，迭代几次即可计算得到 S。

另一方面，由式(9)可知，少于 t 个共享，只能得到少于 t 个“含有 t 个未知变量的”线性方程。而少于 t 个方程求解 t 个变量，有无数解。从而，不能得到正确的 $m_1,m_2,\cdots,m_t$，当然也就得不到最终的秘密 S。

在上层分享过程中：对于算法 1、算法 2 和算法 3，其主要计算均为小整数的加法（或减法），分别需要 N 次加法（或减法）运算。故它们的时间复杂性均为 $O(N)$，其中 N 为 S 的比特数。特别地，该方案并不需要其他复杂的计算，因而效率高（而在 Shamir 秘密共享方案中，需要很多次大整数的模幂、模乘运算）。另外，仅仅知道单个小整数 m_i，计算出原始秘密 S 在计算上不可行。因而，基于 Stern-Brocot 树，对秘密进行分割，虽然不是一种完备的方案，但实现简单，效率高，特别具有共享份额短的优点。

而在底层分享过程中：生成每个参与者的共享需要 t 次加法运算；由 t 个共享计算 t 个初始小整数，主要运算为矩阵的求逆，也即线性方程组的求解。而底层的所有这些计算均是在有限域 GF(p) 中进行，p 是比原始秘密 S 小得多的小整数（在上层，经过算法 3 的 k 次迭代后，$l \leqslant \lfloor 0.694^k N \rfloor$。类似多米诺效应（Domino effect），$k$ 越大，上层拆分后的子秘密 m_i 将越小，从而 p 也越小）。相对于大整数来说，小整数的运算效率更高。

由以上分析可知，基于 Stern-Brocot 树，对秘密进行分割，然后借鉴自然进化的思想，分别生成各参与者的共享，整个方案实现简单，具有效率高、共享份额短的优点。

3 方案改进

3.1 门限值的改变

实际上在式(3)序列中，多数情形并不需严格要求“连续”的 t 个共享来重构秘密。例如，$t=3$ 时，若共享集合为 $\{s_1,s_2,s_4\}$，则由式(4)可得：

$$\begin{cases} s_1 = m_1 + m_2 + m_3 \\ s_2 = m_1 + 2m_2 + 2m_3 \\ s_4 = 4m_1 + 6m_2 + 7m_3 \end{cases} \Rightarrow$$

$$\begin{cases} m_1 = 2s_1 - s_2 \\ m_2 = s_1 + 3s_2 - s_4 \\ m_3 = -2s_1 - 2s_2 + s_4 \end{cases} \tag{12}$$

显然$\{s_1, s_3, s_5\}$、$\{s_1, s_3, s_6\}$等也可以恢复m_1、m_2、m_3。但$\{s_1, s_4, s_5\}$不能恢复m_1、m_2、m_3。因为$2s_4 - s_5 = s_1$，这样由m_1、m_2、m_3线性表示s_1, s_4, s_5的系数矩阵不可逆，从而由$\{s_1, s_4, s_5\}$不能计算出m_1、m_2、m_3。特别地，在式(3)序列中恒有$2s_i - s_{i+1} = s_{i-t}$成立。

为了推广以上特殊门限方案为更一般的门限方案，做如下处理：随机生成r个小整数：$x_1, x_2, \cdots, x_r$，其中$x_i \in Z_p^*$（$1 \leqslant i \leqslant r$），加上$m_1, m_2, \cdots, m_t$，一起生成新的序列：

$$x_1, x_2, \cdots, x_r, m_1, m_2, m_3, \cdots, m_t, s_1, s_2, \cdots, s_n \tag{13}$$

其中，每个s_i等于它前面连续的$t+r$项之和。即，

$$\begin{aligned} s_1 &= (x_1 + x_2 + \cdots + x_r + m_1 + m_2 + \cdots + m_t) \bmod p \\ s_2 &= (x_2 + \cdots + x_r + m_1 + m_2 + \cdots + m_t + s_1) \bmod p \end{aligned} \tag{14}$$

显然，这样做增大了门限值。但在重构$m_1, m_2, \cdots, m_t$时，若公开$x_1, x_2, \cdots, x_r$，这样实际门限值还是t。经过这样处理后，t个并不一定连续的共享中，同时出现s_i、s_{i+1}、$s_{i-(t+r)}$的概率将会大大降低($2s_i - s_{i+1} = s_{i-(t+r)}$，相当于少一个共享)，从而能够正确恢复$m_1, m_2, \cdots, m_t$，并正确计算出原始秘密$S$。

例如，上面$t=3$时，$\{s_1, s_4, s_5\}$不能恢复m_1、m_2、m_3。若随机生成$x_1, x_2 \in Z_q^*$，重新生成序列：$x_1, x_2, m_1, m_2, m_3, s_1, s_2, \cdots, s_n$。则有（公开$x_1, x_2$）：

$$\begin{cases} s_1 = x_1 + x_2 + m_1 + m_2 + m_3 \\ s_4 = 4x_1 + 6x_2 + 7m_1 + 8m_2 + 8m_3 \\ s_5 = 8x_1 + 12x_2 + 14m_1 + 15m_2 + 16m_3 \end{cases} \Rightarrow$$

$$\begin{cases} m_1 = -4x_1 - 2x_2 + 8s_1 - s_4 \\ m_2 = 2s_4 - s_5 \\ m_3 = 3x_1 + x_2 - 7s_1 - s_4 + s_5 \end{cases} \tag{15}$$

另外，这样做的好处是：根据实际安全需要，可以在区间$[t, t+r]$动态改变门限参数，当门限值需要提高时，公开x_i的个数减少，反之增加。

3.2　秘密和共享的更新

实际上，在式(3)序列中，每个s_i均可以由$m_1, m_2, \cdots, m_t$线性表示，即有：

$$\begin{cases} s_1 = (a_{11}m_1 + a_{12}m_2 + a_{13}m_3 + \cdots + a_{1t}m_t) \bmod p \\ s_2 = (a_{21}m_1 + a_{22}m_2 + a_{23}m_3 + \cdots + a_{2t}m_t) \bmod p \\ \vdots \qquad\qquad \vdots \\ s_n = (a_{n1}m_1 + a_{n2}m_2 + a_{n3}m_3 + \cdots + a_{nt}m_t) \bmod p \end{cases} \tag{16}$$

$$\Rightarrow$$

$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_{n-1} \\ s_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1(t-1)} & a_{1t} \\ a_{21} & a_{22} & \cdots & a_{2(t-1)} & a_{2t} \\ a_{31} & a_{32} & \cdots & a_{3(t-1)} & a_{3t} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{(n-1)(t-1)} & a_{(n-1)t} \\ a_{n1} & a_{n2} & \cdots & a_{n(t-1)} & a_{nt} \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_{t-1} \\ m_t \end{bmatrix} \tag{17}$$

例如，$t=5$时，

$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 2 \\ 2 & 3 & 4 & 4 & 4 \\ 4 & 6 & 7 & 8 & 8 \\ 8 & 12 & 14 & 15 & 16 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \end{bmatrix} \tag{18}$$

当秘密S更新为S^*时，$m_1, m_2, \cdots, m_t$也将相应更新为$m_1^*, m_2^*, \cdots, m_t^*$。设$\Delta m_i = m_i^* - m_i$，则有：

$$\begin{bmatrix} s_1^* \\ s_2^* \\ s_3^* \\ \vdots \\ s_{n-1}^* \\ s_n^* \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1(t-1)} & a_{1t} \\ a_{21} & a_{22} & \cdots & a_{2(t-1)} & a_{2t} \\ a_{31} & a_{32} & \cdots & a_{3(t-1)} & a_{3t} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{(n-1)(t-1)} & a_{(n-1)t} \\ a_{n1} & a_{n2} & \cdots & a_{n(t-1)} & a_{nt} \end{bmatrix} \begin{bmatrix} m_1^* \\ m_2^* \\ \vdots \\ m_{t-1}^* \\ m_t^* \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1(t-1)} & a_{1t} \\ a_{21} & a_{22} & \cdots & a_{2(t-1)} & a_{2t} \\ a_{31} & a_{32} & \cdots & a_{3(t-1)} & a_{3t} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{(n-1)(t-1)} & a_{(n-1)t} \\ a_{n1} & a_{n2} & \cdots & a_{n(t-1)} & a_{nt} \end{bmatrix} \begin{bmatrix} m_1 + \Delta m_1 \\ m_2 + \Delta m_2 \\ \vdots \\ m_{t-1} + \Delta m_{t-1} \\ m_t + \Delta m_t \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1(t-1)} & a_{1t} \\ a_{21} & a_{22} & \cdots & a_{2(t-1)} & a_{2t} \\ a_{31} & a_{32} & \cdots & a_{3(t-1)} & a_{3t} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{(n-1)(t-1)} & a_{(n-1)t} \\ a_{n1} & a_{n2} & \cdots & a_{n(t-1)} & a_{nt} \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_{t-1} \\ m_t \end{bmatrix} +$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1(t-1)} & a_{1t} \\ a_{21} & a_{22} & \cdots & a_{2(t-1)} & a_{2t} \\ a_{31} & a_{32} & \cdots & a_{3(t-1)} & a_{3t} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{(n-1)(t-1)} & a_{(n-1)t} \\ a_{n1} & a_{n2} & \cdots & a_{n(t-1)} & a_{nt} \end{bmatrix} \begin{bmatrix} \Delta m_1 \\ \Delta m_2 \\ \vdots \\ \Delta m_{t-1} \\ \Delta m_t \end{bmatrix}$$

$$= \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_{n-1} \\ s_n \end{bmatrix} + \begin{bmatrix} \Delta s_1 \\ \Delta s_2 \\ \Delta s_3 \\ \vdots \\ \Delta s_{n-1} \\ \Delta s_n \end{bmatrix} \tag{19}$$

即有 $s_i^* = s_i + \Delta s_i$（$1 \leqslant i \leqslant n$）。按照式(3)序列的生成策略，秘密分发者重新生成序列：$\Delta m_1, \Delta m_2, \Delta m_3, \cdots, \Delta m_t, \Delta s_1 \Delta s_2, \cdots, \Delta s_n$，其中 Δs_i 等于它前面连续的 t 项相加模 p；然后秘密发送 Δs_i 至参与者 P_i（$1 \leqslant i \leqslant n$）；参与者 P_i 更新共享为 $s_i^* = s_i + \Delta s_i$，同时销毁旧共享 s_i。

4 结束语

本文提出了一种新的高效秘密共享方案。该方案分两层实现：上层，基于 Stern-Brocot 树把一个秘密分割成一组小整数；底层，借鉴自然进化的思想，把上层生成的小整数进化成各参与者的共享，秘密发送给各参与者。需要重构秘密时，t 个参与者合作，通过求解线性方程组（或是若干次模减运算），能够计算出 t 个子秘密，从而再根据 Stern-Brocot 树的性质，最终计算出原始的秘密；而任意小于 t 个参与者合作不能得到原始秘密。

方案最大的优点是高效、灵活，容易实现，参与者共享份额短。整个方案只需若干次加、减（或模加、模减）运算，其中多是小整数的运算，并不需要其他复杂运算。而且，该方案能动态改变门限值、能动态扩展参与者、能动态更新秘密和共享。另外，若加上认证信道或是密码技术，还可以改进为防欺诈的、可验证的秘密共享方案。

实际上，Stern-Brocot 树不仅仅能够用来进行秘密的传递、分享，还能够用来作为信息隐藏，图形图像压缩、加解密等其他编码的工具，甚至能够用来求解欧几里德算法的渐进上界等。这些均是下一步深入研究的方向。

参考文献：

[1] BLAKLEY G. Safeguarding cryptographic keys[A]. Proc of the 1979 AFIPS National Computer Conference[C]. AFIPS Press, 1979, 48: 313-317.

[2] SHAMIR. How to share a secret[J]. Communications of the ACM, 1979, 22 (11): 612-613.

[3] ITO M, SAITO A, NISHIZCKI T. Secret sharing scheme realizing general access structure[A]. Proc of IEEE Global Telecommunication Conference[C]. 1987. 99-102.

[4] BENALOH J C, LEICHTER J. Generalized secret sharing and monotone functions[A]. Advances in Cryptology- CRYPTO'88[C]. LNCS 403, 1990.27-35.

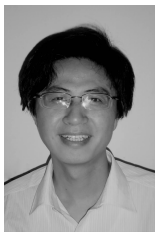
[5] VANDER M J, DAWOUD D, MCDONALE S. A survey on peer-to-peer key management for mobile ad hoc networks[J]. ACM Comput Surv, 2007, 39(1):1-45.

[6] WOLFGANG R, WOLFGANG E. Smart Card Handbook, 3rd Edition[M]. New York: Wiley Press, 2004.

[7] SHI R, ZHONG H, HUANG L S, *et al*. A (*t*, *n*) secret sharing scheme for image encryption[A]. Proceedings of 2008 Congress on Image and Signal Processing[C]. IEEE Computer Society, 2008. 3-6.

[8] GRAHAM R L, KNUTH D E, PATASHNIK O. Concrete Mathematics[M]. New York: Addison- Wesley Publishing Company, 1990.

作者简介：



石润华（1974-），男，安徽安庆人，安徽大学副教授、硕士生导师，主要研究方向为信息安全、量子信息处理。

黄刘生（1957-），男，安徽安庆人，中国科学技术大学教授、博士生导师，主要研究方向为无线传感器网络、信息安全、分布式计算等。

杨威（1978-），男，安徽六安人，中国科学技术大学博士、博士后，主要研究方向为信息安全、量子信息与计算。

仲红（1965-），女，安徽固镇人，博士，安徽大学教授，主要研究方向为信息安全、高性能计算。