一种简单阈值方案的优化*

苏中民 林行良 戴一奇

(清华大学计算机系 北京 100084)

摘要 秘密分存是一种安全有效的密钥管理技术,现已广泛应用于数据安全的各个方面.以往的(k,n)阈值方案,计算量较大.本文提出了用赋标号降低数据扩展的优化方法,给出了一种优化后的简单(k,n)阈值方案论证了其安全性并分析了其数据扩展.这种方案适用于图象秘密分存等秘密数据量较大的情况.

关键词 秘密分存,密钥管理,数据安全,(k,n)阈值方案,数据扩展.

密码学中研究的加密算法是公开的,其安全性主要靠密钥来保证.因此密钥的管理一直是密码学的一个重要研究课题.1979年Shamir[1]和Blakley[2]各自提出了密钥分散保存(简称分存)的方案,在此方案中,主密钥被分解成n个子密钥,分别由n个分存的参与者保管,要求其中的任意k个参与者合作可以恢复主密钥,而在只有不超过$k-1$个参与者的情况下,则不能获得主密钥的任何信息,这样的方案被称为(k,n)阈值方案.此后,各种各样的密钥分存方案被相继提出,其中典型的有:Asmuth-Bloom方法[3]、利用线性码构造的方法[4]、二次密钥分存方法[5]等.

然而目前对分存的研究已不仅局限于密钥管理.在'94年的欧密会上Shamir提出了二值图象分存的方法[6]:由一幅黑白二值图象产生n张幻灯片使其中的任意k张能叠合出原图,而任意$k-1$张则不能得到原图的任何信息,这样的方案也称为(k,n)分存.此方案的原理是把原二值图中的每个象素扩充为一个由多个子象素构成的阈值(黑与白)向量,并给定一个阈值,若幻灯片叠合后呈黑色的子象素的个数不小于给定的阈值,则认为原象素是黑象素,否则原象素为白象素.此分存方案可以用一个矩阵的集合来表示,它有如下优点:黑白象素的区分标准与人的主观感受一致,象素的合成比较直观;方案的生成不需复杂的计算,方案可以重复使用,而不影响其安全性,其代价是数据扩展(分存规模与原图象数据规模之比)比较大.

若在计算机上实现这种图象的分存方案,则数据的扩展会使实际应用难以接受,因此必须对现有方案进行改进,因为图象的数据量很大,所以改进的方案也必须保持Shamir二值图象分存算法简单的特点.综上,我们的改进目标应是在算法简单并保证安全性的前提下,

---

* 作者**苏中民**,1968年生,博士生,主要研究领域为图象加密.**林行良**,1936年生,教授,主要研究领域为计算理论.**戴一奇**,1946年生,副教授,主要研究领域为图论及加密算法.

本文通讯联系人:林行良,北京 100084,清华大学计算机系

本文1996-02-06收到修改稿

尽量减少数据的扩展. 方案改进的可能性在于在计算机内部所采用的象素分存方法可不拘泥于可视性的限制.

从上述的需求而言,目前已有的密钥分存算法绝大多数都显得不够简单,它们适用于一般的密钥分存,但对大量数据分存时计算量就显得过大.

1986 年,曹珍富等基于有限集合理论提出的二次密钥方案[7]可以说是一种算法比较简单的分存方法. 现简单介绍如下:设 A 是一有穷集,$A_i(i=1,2,\cdots,n)$ 是 A 的子集,若满足其中的任意 k 个子集之并与 A 相等,而任意 $k-1$ 个子集之并是 A 的真子集,且 $|A_1|=|A_2|=\cdots=|A_n|$($|A_i|$ 表示 A_i 的元素数),则 $\{A_i|i=1,2,\cdots,n\}$ 称为 A 的 (k,n) 均匀分拆,二次密钥方案的基本定理为:设 $A=\{D_1,D_2,\cdots,D_m\}$ 是密钥 D 使用任意一种 (m,m) 阈值方案得到的密钥集合(一次密钥),这里 $m\geqslant C(n,k-1)$,组合数 $C(n,k)$ 表示从 n 个数中取 k 个的不同的方案数. 再设 $A_1,A_2,\cdots,A_n$ 是对 A 使用 (k,n) 均匀分拆得到的 n 个子集,则 $\{A_1,A_2,\cdots,A_n\}$ 是一个 (k,n) 阈值方案(二次密钥). 例如:$D=11101011$,$D_1=10000110$,$D_2=01010111$,$D_3=00111010$,则 $D=D_1\oplus D_2\oplus D_3$,令 $A=\{D_1,D_2,D_3\}$,$A_1=\{D_1,D_2\}$,$A_2=\{D_1,D_3\}$,$A_3=\{D_2,D_3\}$,显然,$A_1\subset A$,$A_2\subset A$,$A_3\subset A$,而 $A_1\cup A_2=A_1\cup A_3=A_2\cup A_3=A$,即 A_1,A_2,A_3 构成一个(2,3)分存方案,当 $1<k<n$ 时,这样的方案还可以达到发现假用户和蓄意破坏的目的,但其数据扩展为密钥规模的 $C(n-1,k-1)$ 倍,且不便进行优化. 由于图的直观特性,对于图的分存来说,识别假冒用户和蓄意破坏的功能并不一定需要通过分存方案来体现. 基于这个思想,我们提出了另一种算法简单(复杂度与上述方案相当)、数据扩展低于上述方案的用于图象分存的方案,对于这个方案我们暂且称之为一种简单的阈值方案. 因为在计算机上无论是密钥、图象或其它信息,都体现为一个二值数字串,所以下面我们将不分是密钥分存或是图象分存或是其它信息分存,都统称为"秘密分存",被分存的信息称为"秘密",分存的结果体现为一组"数".

1　一种简单的阈值方案

设 F 是定长的二值数字串集,把秘密 $K\in F$ 按 $C(n,k)$ 种不同的方法拆成 F 中一组 k 个数的异或,按 n 中取 k 的不同组合方案把每一组的 k 个数分别送给不同的分存的参与者,使得任意 k 个参与者掌握的数据包含某一种拆分方案的所有 k 个数,从而可以恢复出秘密. 如:$K=11101011$,$s_1=11010101$,$s_2=00111110$,$s_3=01010011$,$s_4=10111000$,$s_5=10101101$,$s_6=01000110$,$K=s_1\oplus s_2=s_3\oplus s_4=s_5\oplus s_6$. 若把 s_1,s_3 分配给分存的第 1 个参与者 s_2,s_5 和 s_4,s_6 分别分配给第 2、第 3 个参与者,则可以构成一个(2,3)分存方案,可用矩阵表示为 $M=\begin{bmatrix}s_1 & s_3 & 0\\ s_2 & 0 & s_5\\ 0 & s_4 & s_6\end{bmatrix}$,显然这个矩阵中的各列表示 K 的一种分解方法,每行对应一个分存的参与者. 当把 M 中的所有 $s_i,i=1,\cdots,6$,代之以"1",则得矩阵 $\begin{bmatrix}1 & 1 & 0\\ 1 & 0 & 1\\ 0 & 1 & 1\end{bmatrix}$,显然这个矩阵中的各列代表了在 3 个对象中取 2 个的每种方案("1"表示取,"0"表示不取).

我们称这种矩阵为(2,3)组合方案矩阵. 按这种理解我们可以为所有的 $k,n;n\geqslant 1,k$

$\geqslant 0, n \geqslant k$ 定义(k,n)组合方案矩阵$A(k,n)$.

设:V^T表示向量V的转置,a^b表示所有分量均为符号a的b维向量.

$A(0,n)=(0^n)^T;(n\geqslant 1)$,

$A(n,n)=(1^n)^T;(n\geqslant 1)$,

$A(k,n)=\left[\begin{array}{c|c} 1^p & 0^q \\ \hline A(k-1,n-1) & A(k,n-1) \end{array}\right]$,其中$p=C(n-1,k-1)$,$q=C(n-1,k)$;$1\leqslant k<n$.

$A(k,n)$定义中的左下方的等于$A(k-1,n-1)$的子矩阵被称为左降阶子矩阵,而右下方的等于$A(k,n-1)$的子矩阵则称为右降阶子矩阵.

命题1. $A(k,n)$是一个$n\times C(n,k)$的布尔矩阵,其各列中1所在的行的序号组成的k元组包括了在n个数中取k个数的全部组合,其每行中1的个数为$C(n-1,k-1)$.

证明:$k=0$时,显然成立;$k=n$时,显然成立;$k=1,n=1$时,显然成立.

用数学归纳法对n进行归纳:

$n=2$时,验证$k=1$的情形,命题成立.

假设$n=m$时,对任意的$k<n$,命题成立;$n=m+1$时,对于任意k,$1\leqslant k<n$,据定义有

$$A(k,m+1)=\left[\begin{array}{c|c} 1^p & 0^q \\ \hline A(k-1,m) & A(k,m) \end{array}\right],p=C(m,k-1),q=C(m,k)$$

根据归纳假设,$A(k,m)$是一个$m\times C(m,k)$的布尔矩阵,且其每行中"1"的个数为$C(m-1,k-1)$;$A(k-1,m)$是一个$m\times C(m,k-1)$的布尔矩阵,且其每行中"1"的个数为$C(m-1,k-2)$,据公式$C(n,k)=C(n-1,k-1)+C(n-1,k)$,可得$C(m,k-1)+C(m,k)=C(m+1,k)$和$C(m-1,k-2)+C(m-1,k-1)=C(m,k-1)$,故$A(k,m+1)$是一个$m+1$行$C(m+1,k)$列的布尔矩阵,且其每一行中"1"的个数为$C(m,k-1)$.

由归纳假设$A(k,m)$的各列表达了在m个数中取k个数的所有组合,$A(k-1,m)$的各列表达了在m个数中取$k-1$个数的所有组合,$A(k,m+1)$中其左降阶子矩阵和其第1行的p个"1"表达了在$m+1$个数中取k个数但必包含第1个数的所有取法,而其右降阶子矩阵和其第1行的q个"0"表达了在$m+1$个数中取k个数但必不包含第1个数的所有取法,故其全部的列表达了在$m+1$个数中取k个数的全部组合. □

例:(4,6)组合方案矩阵为

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

令$A(k,n)=(a_{ij})$,$i=1,\cdots,n$,$j=1,\cdots,p$,$p=C(n,k)$的每一行对应一个分存的参与者,每一列对应一种秘密K的分解方法.$a_{ij}=1$表示第i个参与者掌握第j种分存方案的k个数中的一个,$a_{ij}=0$则表示不掌握相应方案中的任一个数.逐列对$A(k,n)$中的每个"1"元素赋值,使满足各列数的异或都等于秘密K的要求,则可以构成一个(k,n)阈值方案,我们

称这种赋值方法为对 $A(k,n)$ 的赋阈值法.

这样的方案很简单,但若不加以优化,则所需掌握的分存信息仍然很多.由命题 1,$A(k,n)$ 中每行的元素 1 的个数为 $C(n-1,k-1)$,即每个参与者掌握的分存信息规模是秘密规模的 $C(n-1,k-1)$ 倍,与文献[7]的方案等效.下面我们给出优化的赋阈值法以降低数据扩展.

2　赋标号方法及其有效性

我们考虑如下的例子:设 F 是定长的二值数字串集,$K\in F,s_1,s_2,s_3,s_4\in F,s_1\oplus s_2=K,s_3\oplus s_4=K$,容易验证:由对 $A(2,3)$ 赋阈值得到的矩阵 $M'=\begin{bmatrix}s_1 & s_1 & 0\\ s_2 & 0 & s_3\\ 0 & s_2 & s_4\end{bmatrix}$ 对应一个(2,3)阈值方案,此时对第 1 个参与者来说,只要掌握一个元素 s_1 即可,因而平均分存信息规模与秘密规模之比为 $(1+2+2)/3=5/3<2$,数据扩展有所降低.

为简单起见,我们可直接采用元素的下标(称为标号)来代替元素本身,则上述矩阵 M' 可用它的标号矩阵代替:$\begin{bmatrix}1 & 1 & 0\\ 2 & 0 & 3\\ 0 & 2 & 4\end{bmatrix}$,它可看作是对 $A(2,3)$ 中的"1"元素赋以标号得到的.

这样,标号矩阵中的标号与 F 中的元素直接对应,它代表一个数值,相同的标号表示相同的元素,不同的标号所代表的数则不必相同,赋标号只需对 $A(k,n)$ 中的"1"进行(可认为"0"的标号为 0).

优化方法就是要通过对矩阵 $A(k,n)$ 赋标号以满足如下要求:1)标号所对应的数值在矩阵中构成一个 (k,n) 阈值方案;2)各行平均使用的标号数少于 $C(n-1,k-1)$.

为后面讨论的方便,先给出如下 2 个定义:

定义 1. 标号矩阵称为是可满足约束的,如果对任意给定的秘密 K,都存在标号的对应数,使得同一列的各标号所对应的数经过运算后,可以得到秘密 K.

定义 2. 可满足约束的标号矩阵称为是满足安全性要求的,如果在满足约束前提下确定了标号的对应数后由任意 $k-1$ 行,不能确定秘密 K.

下面提出一般的赋标号算法 *mark*,算法以递归的形式给出.

算法中用 M 表示一个矩阵空间,$M(x,y)$ 表示 x 行,y 列的矩阵,$M(a\to b,c\to d)$ 表示 $M(x,y)$ 矩阵中从第 a 行到第 b 行,从第 c 列到第 d 列所界成的子矩阵空间.$A(k,n)$ 仍表示一个 (k,n) 组合方案矩阵.

主过程:

(1) *label*←1;

(2) 申请矩阵空间 $M(n,C(n,k))$;

(3) M=*mark*(k,n);

(4) 结束.

mark(k,n)

(1) 申请矩阵空间 $M(n,C(n,k))$;

(2) $M=A(k,n)$;

(3) 若 $k=1$,把 M 中所有"1"元替以标号 $label$,$label \leftarrow label+1$,以 M 中的值返回.

(4) 若 $k>1$,把 M 第1行中所有"1"元替以标号 $label$,$label \leftarrow label+1$;

(5) $M(2 \rightarrow n, 1 \rightarrow C(n-1,k-1)) = mark(k-1,n-1)$;

(6) 若 $k<n$,$M(2 \rightarrow n, C(n-1,k-1)+1 \rightarrow C(n,k)) = mark(k,n-1)$;

(7) 返回 M 中的值.

例如:当 $n=6$,$k=4$ 时,由赋标号算法得到的标号矩阵为

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 17 & 17 & 17 & 17 & 0 \\ 3 & 3 & 3 & 0 & 0 & 0 & 9 & 9 & 9 & 0 & 18 & 18 & 18 & 0 & 26 \\ 4 & 0 & 0 & 5 & 5 & 0 & 10 & 10 & 0 & 14 & 19 & 19 & 0 & 23 & 27 \\ 0 & 4 & 0 & 6 & 0 & 7 & 11 & 0 & 12 & 15 & 20 & 0 & 21 & 24 & 28 \\ 0 & 0 & 4 & 0 & 6 & 8 & 0 & 11 & 13 & 16 & 0 & 20 & 22 & 25 & 29 \end{bmatrix}$$

下面我们说明上述赋标号方法是满足我们提出的要求的.

根据赋标号算法容易得到下面的引理.

引理 1. 用赋标号方法 $mark$ 对 $A(k,n)$ 赋标号后,在其左降阶子标号矩阵中出现的标号与其在右降阶子标号矩阵中出现的标号不相交.

引理 2. 在任何一层调用赋标号函数 $mark$ 对 $A(k,n)$,$k>1$,赋标号后其左、右降阶子标号矩阵中的所有非0标号都与其第1行的标号不同.

定义 3. 2个行、列数相同的标号矩阵称为是等价的,若满足下列条件:(1)2个矩阵在相同位置上的标号或都是0或都不是0;(2)若 a 是一个矩阵中的一个非0标号,则在另一矩阵中有一个非0标号 b,a 和 b 分别出现在2个矩阵的相同位置.

显然,在不同层次上调用 $mark$ 函数对 $A(k,n)$ 赋标号的结果是相互等价的标号矩阵.

定理 1. 用赋标号方法 $mark$ 得到的标号矩阵是可以满足约束的.

证明:因为任意 K 都是一个二值数字串,可以分解成单个数字来讨论,所以只要证明对一位数字结论成立即可.

用数学归纳法对 n 进行归纳:

当 $k=1$ 时,$A(1,n)$ 的标号矩阵形式为 n 阶单位矩阵:标号1的对应数为 K;

当 $n=2$,$k=2$ 时,$A(2,2)$ 的标号矩阵为:$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$.

当 $K=0$ 时,标号1对应数为1,标号2对应数为1,或标号1对应数为0,标号2对应数为0;

当 $K=1$ 时,标号1对应数为1,标号2对应数为0,或标号1对应数为0,标号2对应数为1.

现对 n 进行归纳:

假设 $n=m$,$m \geqslant 2$ 时命题成立.

当 $n=m+1$,$k>1$ 时,$A(k,m+1)$ 的标号矩阵由左、右2个子矩阵组成,右子矩阵第1行都是"0",从第2行到第 n 行是 $A(k,m+1)$ 的右降阶标号子矩阵,它与 $A(k,m)$ 的标号矩阵等价,根据归纳假设它是可满足约束的.左子矩阵的第1行都是"1",从第2行到第 n 行是 $A(k,m+1)$ 的左降阶标号子矩阵,它与 $A(k-1,m)$ 的标号矩阵等价.根据归纳假设它是可

满足约束的,即每一列上标号所对应数的运算(我们采用⊕运算),都得到 K,如令标号 1 对应数 0;或令标号 1 对应数 1,并将左降阶标号子矩阵中每一列的第 1 个非 0 标号所对应的数取反,如果这样的对应方法不引起对同一个标号有不同对应要求的话,则左子矩阵的每一列标号对应数的运算结果仍为 K,即左子矩阵也是可满足约束的,实际上引理 1、2 保证了这种对应方法对于每个标号与数对应的一致性. 从而整个 $A(k,m+1)$ 的标号矩阵是可满足约束的. 故命题成立. □

定理 2. 用赋标号方法 *mark* 得到的标号矩阵是满足安全性要求的.

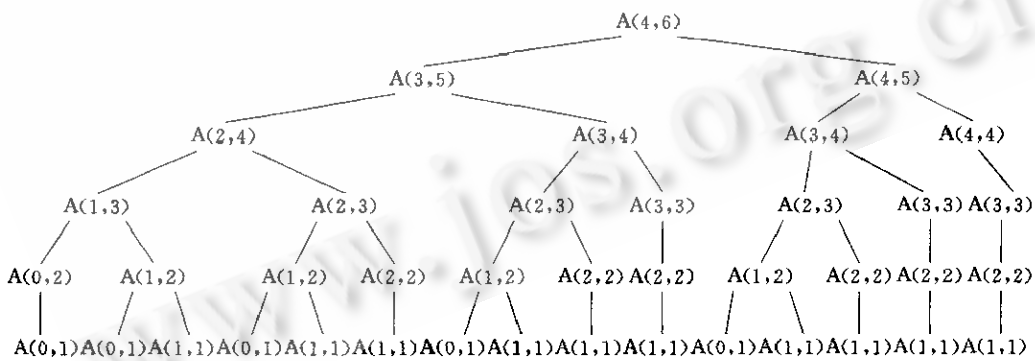
证明:根据定理 1 的证明可见,当 $k>1$ 时,每个非 0 标号与数的对应都有 2 种可能:0 或 1,因为采用的是⊕运算,所以在一列中对任意非 0 标号来说,所对应的数不同都会使整列运算的结果不同. 换句话说,在标号所对应的数已经确定的情况下,若取 $k-1$ 行,则对每列来说为确定 K 至少还缺一个数,不知道所缺的数就不能确定 K,而引理 1、2 又保证了在一列中所缺少的数不能通过其它列得到.

故标号矩阵满足安全性要求. □

3　对分存的平均规模的分析

我们按 $A(k,n)$ 的定义来构造一棵分解树:把 $A(k,n)$ 作为父结点,当 $0<k<n$ 时把它的左降阶子矩阵和右降阶子矩阵分别作为它的子结点;当 $k=0$ 时,把 $A(0,n-1)$ 作为它的子结点;当 $k=n$ 时,把 $A(n-1,n-1)$ 作为它的子结点. 按此原则可将 $A(k,n)$ 分解成一棵 n 层的树.

例如,对于 $A(4,6)$ 可分解为如下一棵树:



实际上,赋标号的顺序正是 $A(k,n)$ 分解树的前序遍历的顺序.

引理 3. $A(k,n)$ 分解树中第 i 层中的结点都具形式 $A(x,n-i+1)$,其中 $0\leqslant x\leqslant n-i+1$ 且 $k-i\leqslant x\leqslant k$.

证明:使用归纳法.

$A(k,n)$ 是第 1 层的唯一结点,满足引理. 假设 $A(k,n)$ 分解树的第 i 层中的结点满足引理. 根据 $A(k,n)$ 的定义,它的降价子矩阵必为 $A(k-1,n-1)$ 或 $A(k,n-1)$. 从而第 i 层中的结点 $A(x,n-i+1)$ 的降价子矩阵必为 $A(x-1,n-i)$ 或 $A(x,n-i)$,即 $A(x-1,n-(i+1)+1)$ 或 $A(x,n-(i+1)+1)$,根据归纳假设有:$0\leqslant x\leqslant n-i+1$ 且 $k-i\leqslant x\leqslant k$,因此 k

$-(i+1)\leqslant x, k-(i+1)\leqslant x-1, x-1\leqslant k$，若 $x=0$，则根据分解树构造原则，$A(x-1,n-i)$不存在，若 $x=n-i+1$，则根据分解树构造原则 $A(x,n-i)$不存在，这就是说第 $i+1$ 层中的结点也都满足引理. □

引理 4. 采用赋标号方法 *mark* 对 $A(k,n)$赋标号后，标号矩阵中第 i 行所采用的不同标号的个数为 $A(k,n)$分解树中第 i 层中 $A(x,n-i)$，$x>0$ 的个数.

证明：根据赋标号的算法，在对 $A(k,n)$分解树作前序遍历时，对应每一结点 $A(x,n-i+1)$，$x>0$ 的子矩阵的第 1 行上的所有"1"元素都被赋以同一个标号，因 $A(0,n-i+1)$中无"1"元素，故不赋标号. 由引理 1 和引理 2，本引理得证.

引理 5. $A(k,n)$分解树的第 $i+1$ 层中 $A(x,n-i)$的个数为 $C(i,i-k+x)$.

证明：对 i 进行归纳.

$i=0$ 时，$i+1$ 层即第 1 层，只有 $A(k,n)$一项，$k\leqslant n$，而 $C(0,0-k+k)=C(0,0)=1$，命题成立.

假设 $i=m$ 时命题成立，当 $i=m+1$ 时，即在 $A(k,n)$分解树的第 $m+2$ 层中，结点 $A(x,n-m-1)$只可能作为 $m+1$ 层上 $A(x,n-m)$或 $A(x+1,n-m)$的子结点出现. 由归纳假设，第 $m+1$ 层中 $A(x,n-m)$，$A(x+1,n-m)$ 的个数分别为 $C(m,m-k+x)$和 $C(m,m-k+x+1)$，所以第 $m+2$ 层中 $A(x,n-m+1)$的个数为 $C(m,m-k+x)+C(m,m-k+x+1)=C(m+1,m-k+x+1)=C(m+1,(m+1)-k+x)$，命题成立. □

根据引理 3～5，$A(k,n)$中第 $i+1$ 行所用的不同标号的个数为

$C(i,i-k+1)+C(i,i-k+2)+\cdots+C(i,i-k+(n-i))$；$C(i,i-k+(n-i))=C(i,n-k)$.

定理 3. 设秘密规模为 1，则用赋标号方法 *mark* 得到的阈值方案，其分存的平均规模为 $((n+1)/(k(n-k+1)))C(n-1,k-1)-(1/n)$.

证明：根据引理 4、5 可将由 *mark* 函数在 $A(k,n)$的每行中所赋的不同标号的个数逐行列出如下：

行号	$x<k-i$	$x>k$	
1	$C(0,-k+1)+C(0,-k+2)+\cdots+C(0,-1)$	$+C(0,0)$ $+C(0,1)+$	$\cdots$ $+C(0,n-k)$
2	$C(1,-k+2)+\cdots+C(1,-1)$	$+C(1,0)+C(1,1)$ $+C(1,2)+$	$\cdots$ $+C(1,n-k)$
3	$C(2,-k+3)+\cdots+C(2,-1)$	$+C(2,0)+C(2,1)+C(2,2)$ $+C(2,3)+\cdots$	$+C(2,n-k)$
		……	
k		$C(k-1,0)+C(k-1,1)+C(k-1,2)+\cdots$	$\cdots$ $+C(k-1,n-k)$
$k+1$		$C(k,1)+C(k,2)+\cdots\cdots$	$+C(k,n-k)$
$k+2$		$C(k+1,2)+\cdots$	$+C(k+1,n-k)$
		……	
$n-k+1$		$\cdots$ $+C(n-k,n-k-1)+C(n-k,n-k)$	
		……	
n			$C(n-1,n-k)$

我们可把这些运算项分成 3 部分，把左部的对应 $x<k-i$ 的情况的那些项作为一部分，把右上角对应 $x>k$ 的情况的那些项作为一部分，根据引理 3，这 2 部分中的项不应参加运算，实际上，这些项的值均为 0，所以只要算出中间部分的这些项的和即可. 即

$$\sum_{j=0}^{n-k}\sum_{i=0}^{k-1}C(i+j,j)=\sum_{j=0}^{n-k}C(k+j,j+1)=\sum_{j=0}^{n-k}C(k+j,k-1)=\sum_{j=0}^{n-k+1}C(k-1+j,k-1)-C$$

$(k-1,k-1)=C(n+1,k)-1;\frac{C(n+1,k)-1}{n}=\frac{n+1}{k(n-k+1)}C(n-1,k-1)-\frac{1}{n}$.

由命题1,在未优化的方案中,分存的规模为$C(n-1,k-1)$,当$k=1$或$k=n$时,优化前后分存规模均为1,下面不再讨论.设$g(k)=(n+1)/(k(n+1-k))$,$1<k<n$,$g(k)$大致反映了优化后与优化前分存规模之比.容易看出,$g(k)=g(n+1-k)$,当$k=n/2$或$k=(n+1)/2$时,$g(k)$取最小值(优化效果最佳),约为$4/n$;当$k=2$或$k=n-1$时,$g(k)$取最大值$(n+1)/(2(n-1))$(优化效果最差),约为1/2;当$2<k<n/2$或$(n+1)/2<k<n$时,优化效果介于前2种情况之间.

下面的表格对优化前后的分存规模进行了比较(括号里外分别是优化前、后的分存规模).

$k\backslash n$	3	4	5	6	7	8
2	5/3(2)	9/4(3)	14/5(4)	20/6(5)	27/7(6)	35/8(7)
3		9/4(3)	19/5(6)	34/6(10)	55/7(15)	83/8(21)
4			14/5(4)	34/6(10)	69/7(20)	125/8(35)
5				20/6(5)	55/7(15)	125/8(35)
6					27/7(6)	83/8(21)
7						35/8(7)

4 结　论

本文的主要目的是讨论采用赋标号方法对一种简单阈值方案进行优化的问题,我们提出了一种优化的方案,它能够降低数据扩展,但对一般的k和n,这个方案并不是最优的,请看下面的例子.

按*mark*方法对$A(3,4)$赋标号得到如下标号矩阵

$$\begin{bmatrix}1&1&1&0\\2&2&0&6\\3&0&4&7\\0&3&5&8\end{bmatrix},$$

此方案分存的平均规模为9/4,而我们可以设计另一种同样安全有效的标号矩阵

$$\begin{bmatrix}1&1&1&0\\2&2&0&2\\3&0&4&6\\0&3&5&7\end{bmatrix},$$

此方案的平均分存规模为2,由此可见,对赋标号方法还可进一步优化.这是有待继续研究的问题.

我们提出上述分存的目的是为用之于秘密图象的分存,实践证明经优化的方案已可基本满足实用要求.

参考文献

1 Shamir A. How to share a secret. Comm. ACM, 1979, **22**(**11**): 612～613.

2 Blakley G R. Safeguarding cryptographic keys. In: Proc. NCC, AFIPS Press, 1979, **48**: 313～317.

3 Asmuth C, Bloom J. A modular approach to key safeguarding. IEEE Trans. IT, 1983,29(2):208～210.
4 Karnin E D, Greene J W, Hellman M E. On secret sharing systems. IEEE Trans. IT, 1983,29(1):35～41.
5 曹珍富.关于密钥分享的二次密钥方案.密码学进展CHINACRYPT'92.北京:科学出版社,1992.267～274.
6 Noar M, Shamir A. Visual cryptography. In: Proc. EUROCRYPT'94, 1994.1～12.
7 刘锐,曹珍富.通信密钥分散管理的两个新方案.通信学报,1987,8(4):10～14.

THE IMPROVEMENT OF A SIMPLE THRESHOLD SCHEME

SU Zhongmin LIN Xingliang DAI Yiqi

(*Department of Computer Science and Technology Tsinghua University Beijing* 100084)

Abstract Secret sharing is an effective technique for key management, and has been widely used in many aspects of data security. The existing (k,n) threshold schemes for the secret sharing are relatively complex and the data expansion they cause is remarkable. In this paper, a label assignment method is suggested for reducing the data expansion of a simple threshold scheme. The effect of the improved scheme is obvious and its security is ensured too. Such scheme can be used in the secret sharing of the data on a large scale such as images.

Key words Secret sharing, key management, data security, (k,n)threshold scheme, data expansion.