基于离散对数的动态 (k,n)-门限方案 [1]

刘焕平 * ** 季振洲 * 胡铭曾 * 方滨兴 * 杨义先 ***

*(哈尔滨工业大学计算机科学与技术系 哈尔滨 150001)
**(哈尔滨师范大学计算机科学系, 哈尔滨 150080)
***(北京邮电大学信息安全中心 126 信箱, 北京 100876)

摘 要 该文给出了一个基于离散对数的动态 (k,n)-门限方案，它具有下述特点：(1) 每个成员的子密钥可无限制地多次使用；(2) 能够确认欺骗者；(3) 当某个成员的子密钥泄密时，系统只须为该成员重新分配子密钥而不必更改其它成员的子密钥；(4) 系统可以很方便地增加或删除一个成员；(5) 恢复系统密钥时，采用并行过程.

关键词 数据安全，密码学，秘密分存，离散对数
中图号 TN918.1

1 引言

在保密通信中，为了实现信息的安全保密，人们主要采用密钥加密信息，从而使不拥有密钥的非法用户无法窃获信息. 这使得信息的安全保密主要维系于密钥的安全，从而如何有效地管理密钥就成为密码学中十分重要的课题. 1979 年, Shamir[1] 和 Blakley[2] 独立地提出了密钥分散管理的概念，实现这一思想的机制称为 (k,n)-门限方案. 该方案是将一个密钥 (称为系统密钥) 分成 n 个部分 (称为 n 个子密钥或影子，分别交给 n 个人保管，使得对确定的整数 $k(k<n)$ 满足：(1) 在这 n 个人中，任意 $r(r\geq k)$ 个人协作利用它们的子密钥能够恢复出系统密钥；(2) 任意 $k-1$ 个人协作对恢复系统密钥没有任何帮助. 这种密钥分散管理的思想使密钥管理更加安全灵活. 目前这一思想除用于密钥管理外，在密码学的其它领域 (如组签名和组认证等方面) 也有诸多应用.

在 (k,n)-门限思想提出后，很多学者对其进行了研究，并提出了许多方案来实现它 [1-8]. 在早期提出的 (k,n)-门限方案 [1-3] 中大都存在下述几方面的不足：(1) 当要更新系统密钥 (比如原密钥已恢复或由于某种原因而需要更换原密钥) 时，系统必须为每个成员重新分配子密钥 (尽管这些子密钥可能还从没被用过)，即每个子密钥至多只能使用一次；(2) 当某个成员的子密钥泄密时，系统不能做到只为该成员重新分配子密钥而不影响其它成员的子密钥；(3) 当有新成员加入时，系统也必须重新为每个成员分配子密钥. 为了克服上述不足，人们又提出了许多能够重复使用子密钥的 (k,n)-门限方案 [4-6]，但是这些子密钥只能保存或恢复系统预先确定的一个密钥集合中的密钥，而要保存一个新的密钥 (确定密钥集合之外的密钥)，系统则必须更新每个成员的子密钥. 文献 [7,8] 在 $k=n$ 时，分别给出了一个可无限制地多次使用子密钥来恢复系统密钥的 (n,n)-门限方案，但在恢复系统密钥时所有成员必须按一个强制性序列 (即一个串行过程)$m_1,m_2,\cdots,m_n$ 来恢复系统密钥，这样在恢复密钥时势必要造成一个较大的时间开销.

针对上述不足，我们将在第 2 节中给出一个基于离散对数的动态 (k,n)-门限方案，它具有下述特点：(1) 系统在更新系统密钥时，无须更改每个成员的子密钥；(2) 当某个成员的子密

[1] 2000-09-01 收到，2000-12-14 定稿
黑龙江省科委资助

钥泄密时，系统只须为该成员重新分配子密钥而不必更改其它成员的子密钥；(3) 当有新成员加入时，系统只须为新成员分配一个子密钥，而其他成员不受任何影响；(4) 子密钥可无限制地多次使用；(5) 恢复系统密钥时，采用并行过程。

2 本文所提方案

设 Z_p 是一个有限域，其中 p 是一个大素数，$(Z_p;+)$ 和 $(Z_p;\bullet)$ 分别是其相应的加群和乘群，$P_1,P_2,\cdots,P_n$ 是系统中的 n 个成员。

2.1 系统初始化

(1) 系统随机地选取 n 个互不相同的元素 $s_1,s_2,\cdots,s_n$，并将 s_i 通过安全信道秘密地发送给 P_i，作为 P_i 的秘密子密钥，$i=1,2,\cdots,n$；

(2) 系统随机地选择 Z_p 的一个本原元 α 以及一个 $(k-1)$ 次多项式 $h(x)$，满足：$h(0)=K$ 为系统要保存的系统密钥。计算

$$y_i=h(\alpha^{s_i}),\quad i=1,2,\cdots,n$$

所涉及的运算均为域 Z_p 上的运算。

(3) 系统在公告牌上公开 α 及有序数组 $(y_1,y_2,\cdots,y_n)$。

2.2 秘密恢复

假设任意 k 个子密钥持有者 (不妨设为 $P_1,P_2,\cdots,P_k$) 欲恢复系统密钥。那么只须每个成员 P_i 在公告牌上查到 α 和 y_i 后，$i=1,2,\cdots,k$，利用其秘密子密钥 s_i 计算：$x_i=\alpha^{s_i}$，并提交 x_i(x_i 称为 P_i 的屏蔽子密钥，相应地 s_i 称为 P_i 的秘密子密钥)。在汇总所有的 (x_i,y_i) 之后，利用 Lagrange 内插公式：

$$h(x)=\sum_{i=1}^{k}y_i\prod_{1\le j\le k,j\ne i}\frac{x-x_j}{x_i-x_j}$$

确定出 $h(x)$，进而可恢复出系统密钥 $K=h(0)$。

3 性能分析

(1) 可行性：由于 $x_i=\alpha^{s_i}$ 及 $y_i=h(\alpha^{s_i})$ 可知 (x_i,y_i) 是 $h(x)$ 上的一个点。再注意到 α 是本原元及 $s_i\ne s_j(i\ne j)$，故有 $x_i\ne x_j$。于是在恢复密钥时，k 个成员 $P_1,P_2,\cdots,P_k$ 恰好可以给出多项式 $h(x)$ 上的 k 个不同的点 $(x_i,y_i)(i=1,\cdots,k)$，而 $h(x)$ 是 $k-1$ 次多项式，故利用 Lagrange 内插公式可由这 k 个点将 $h(x)$ 确定出来，因此上述方案可行。

(2) 安全性：本方案的安全性基于 Shamir 方案的安全性与离散对数问题的难解性。第一，由本方案的初始化过程可知，少于 k 个成员时，汇聚他们的子密钥至多可得到 $k-1$ 次多项式 $h(x)$ 上的 $k-1$ 个点，再由 Shamir (k,n)-门限方案的安全性可知，这对恢复系统密钥没有任何帮助；第二，由于在恢复密钥时，每个成员 P_i 提交的是其屏蔽子密钥 $x_i=\alpha^{s_i}$，根据离散对数问题的难解性，其他成员无法通过 α 及 x_i 求出 P_i 的秘密子密钥 s_i。即每个成员的秘密子密钥并没有因为系统密钥的恢复而被公开，从而可继续使用；第三，同样是根据离散对数问题的难解性，任何成员无法通过系统公开的信息 α 及有序数组 $(y_1,y_2,\cdots,y_n)$ 来获取其他成员的秘密子密钥及其屏蔽子密钥。

(3) 系统更新：分下述两种情形考虑

(a) 原系统密钥 K 尚未被恢复，只是出于某种原因而需要更换系统密钥：

此时系统只须重新选择一个 $k-1$ 次多项式 $h'(x)$，满足 $h'(0)=K'$ 为新的系统密钥。然后利用新的 $k-1$ 次多项式 $h'(x)$ 更新公告牌上的有序数组 $(y_1,y_2,\cdots,y_n)$ 即可。

(b) 原系统密钥 K 已被恢复，现在要保存新的系统密钥：

此时系统选择一个新的本原元 $\alpha'(\alpha'\neq\alpha)$ 及 $k-1$ 次多项式 $h'(x)$，满足 $h'(0)=K'$ 为新的系统密钥。然后利用新的 α' 及 $h'(x)$ 更新公告牌上的 α 及有序数组 $(y_1,y_2,\cdots,y_n)$ 即可。

由于 α 是 Z_p 的任意本原元，故每个成员的子密钥可以无限制地被多次使用。

(4) 确认欺骗者：该方案可以很容易修改为一个可确认欺骗者的动态 (k,n)-门限方案。此时系统只须为每个成员 P_i 公开一个检测信息：$v_i=\alpha^{x_i}$，其中 $x_i=\alpha^{s_i}$。在恢复密钥时，其他成员可以通过检验等式 $v_i=\alpha^{x_i}$(此处 x_i 是 P_i 的屏蔽子密钥) 是否成立来确认成员 P_i 是否为欺骗者。注意根据离散对数问题的难解性，任何成员无法通过系统公开的信息 α,v_i 及 y_i 来获取成员 P_i 的秘密子密钥 s_i 和屏蔽子密钥 x_i。

(5) 增删成员

(a) 当有新成员 P_{n+1} 加入时，系统只须为新成员随机地生成一个 s_{n+1} 作为其秘密子密钥，并在公告牌上的有序数组中增加一个元素 y_{n+1}，其中 $y_i=h(\alpha^{s_{n+1}})$ 即可，而无须更改其他成员的秘密子密钥；

(b) 当要删除某个成员 P_{i_0} 时，系统只须重新选择一个 $k-1$ 次多项式 $h'(x)$，满足 $h'(0)=K$ 为系统密钥，然后利用新的 $k-1$ 次多项式 $h'(x)$ 更新公告牌上的有序数组 $(y_1,y_2,\cdots,y_n)$。此时无须计算 y_{i_0}(可令 y_{i_0} 仍为原值或置 y_{i_0} 项为空)，那么 P_{i_0} 原有的子密钥 s_{i_0} 即可无效。在这一过程中其他成员的秘密子密钥没有被更改。

(6) 更新个别成员的秘密子密钥：当某个成员 P_{i_0} 的秘密子密钥泄密时，系统只须为该成员重新分配子密钥 s'_{i_0}，之后重新选择一个 $k-1$ 次多项式 $h'(x)$，使 $h'(0)=K$ 为系统密钥，并利用新的 $s_{i_0}=s'_{i_0}$ 和 $h'(x)$ 更新公告牌上的有序数组 $(y_1,y_2,\cdots,y_n)$，而不必更改其它成员的子密钥。

参 考 文 献

[1] A. Shamir, How to share a secret, Commun. ACM, 1979, 22(11), 612-613.

[2] G. R. Blackley, Safeguarding cryptographic keys, Proc. Nat. Computer Conf. AFIPS Conf. Proc., USA, 1979, 313-317.

[3] E. D. Karnin, J. W. Green, M. E. Hellman, On secret sharing systems, IEEE Trans. on IT, 1983, 24(1), 231-241.

[4] He. J., E. Dawson, Multistage secret sharing based on one-way function, Electron. Lett., 1994, 30(19), 1591-1592.

[5] L. Harn, Comment: Multistage secret sharing based on one-way function, Electron. Lett., 1995, 31, (4), 262-263.

[6] 刘焕平，杨义先，杨放春，基于单向函数的多级密钥共享方案，电子科学学刊，1999, 21(4), 561-564.

[7] R. G. E. Pinch, Online multiple secret sharing, Electron. Lett., 1996, 32(12), 1087-1088.

[8] 谭凯军，诸鸿文，基于单向函数的动态秘密分享机制，通信学报，1999, 20(7), 81-84.

A DYNAMIC (k, n)-THRESHOLD SECRET SHARING SCHEME BASED ON DISCRETE LOGARITHM

Liu Huanping* **　Ji Zhenzhou*　Hu Mingzeng*　Fang Binxing*　Yang Yixian***

*(*Harbin Institute of Technology, Harbin* 150001, China)

**(*Harbin Normal University, Harbin* 150080, China)

***(*Beijing University of Posts and Telecommunications, Beijing* 100876, China)

Abstract　A dynamic (k,n)-threshold secret sharing scheme based on discrete logarithm is proposed in this paper. It can reconstruct the different system secrets for many times without any restriction. Any cheater can be checked out. When some participants' secret sharing values are revealed, they can be renewed without any effect on the others. It is convenient to add or to delete one or more participants. The system secret can be recovered with a parallel process.

Key words　Data safety, Cryptography, Secret sharing, Discrete logarithm

刘焕平：　男，1965 年生，博士、副教授，信息安全、密码学.
季振洲：　男，1965 年生，博士、教授，计算机体系结构.
胡铭曾：　男，1935 年生，教授、博士生导师，计算机体系结构.
方滨兴：　男，1960 年生，教授、博士生导师，计算机体系结构、信息安全.
杨义先：　男，1961 年生，教授、博士生导师，信息安全、信号理论、编码理论.