等级系统中的访问控制方案研究

王丽娜[1],费如纯[2],董晓梅[2]
(1. 武汉大学软件工程国家重点实验室,湖北武汉 430071;2. 东北大学信息科学与工程学院,辽宁沈阳 110004)

摘 要: 本文基于 Lagrange 插值多项式,提出了等级系统中的一个访问控制方案,并从空间复杂度和时间复杂度角度分析了其性能. 该方案具有很强的安全性,并且允许所有用户自主选择秘密密钥. 提出了基于门限秘密共享体制的一般性访问控制方案,阐述了一般性的访问控制方案的基本思想、方案的构造算法及安全性.

关键词: 等级系统; 访问控制; 安全类; 秘密密钥; 安全性
中图分类号: TP309.7 **文献标识码**: A **文章编号**: 0372-2112 (2003) 02-0290-04

Research on Access Control Scheme in a Hierarchy System

WANG Li-na[1], FEI Ru-chun[2], DONG Xiao-mei[2]
(1. *State Key Laboratory of Software Engineering, Wuhan University, Wuhan, Hubei* 430072, *China*;
2. *School of Information Science and Engineering, Northeastern University, Shenyang, Liaoning* 110004, *China*)

Abstract: Access control is the key problem in a hierarchy system. An access control scheme based on Lagrange's interpolation polynomial is proposed, and the property is analyzed from the view of time and space complexity. The scheme is of very strong security, and it allows all users to select their own secret keys. A general access control scheme in a hierarchy is proposed based on threshold secret sharing scheme. Its basic idea, constructing algorithm and security are discussed.

Key words: hierarchy system; access control; security class; secret key; security

1 引言

1983 年,Akl 和 Taylor[1]首次将密钥和等级系统的访问控制问题相结合,他们的基本思想是:前趋安全类的用户可以获得所有后继安全类用户的秘密密钥,可以访问后继安全类用户的信息,反之则不行. Akl-Taylor 方案较简单,但实现起来却比较困难. 1985 年 MacKinnon[2]等人对 Akl-Taylor 方案进行了改进,但需要占用大量存储. 目前,等级系统中比较有代表性的访问控制方案是 Chang[3]方案,该方案对存储量的要求很低,且简单而高效. 但是,Chang 方案的普适性稍差,而且该方案是由权力中心为等级系统中的所有安全类用户强制性分配秘密密钥而不允许用户自主选择秘密密钥.

2 等级系统访问控制的基本概念

假设 $C=\{C_1, C_2, \cdots, C_n\}$ 是等级系统中全部 n 个安全类构成的集合,"$\geq$"是 C 上的一个偏序关系,$C_i \geq C_j$ 表示 C_i 的安全级别不低于 C_j,$C_i > C_j$ 表示 $C_i \geq C_j$ 且 $C_i \neq C_j$. 当 $C_i > C_j$ 且不存在 C_k 使得 $C_i > C_k > C_j$ 时,则称 C_i 是 C_j 的直接前趋,C_j 是 C_i 的直接后继,否则称 C_i 是 C_j 的间接前趋,C_j 是 C_i 的间接后继. 直接前趋和间接前趋合称前趋,直接后继和间接后继合称后继. 如果 C_i 和 C_j 具有相同的直接前趋,则称 C_i 和 C_j 互为兄弟. 综上所述,一个等级系统可以用一个偏序集 $(C, \geq)$ 来表示. 实际上,用偏序集的哈斯图来表示一个等级系统更加简单直观. 哈斯图由一些结点和一些弧构成,结点表示等级系统中的安全类,弧表示安全类之间的关系,弧尾安全类是相应弧头安全类的直接前趋,弧头安全类是相应弧尾安全类的直接后继.

一个等级系统要求满足:当且仅当 $C_i \geq C_j$ 时 C_i 可以访问 C_j 的信息. 要满足上述要求,可以采用 Akl 和 Taylor 所提出的思想,它要求:当且仅当 $C_i \geq C_j$ 时 C_i 可以得到 C_j 的密钥.

3 访问控制方案及性能分析

本文所提出的等级系统中的访问控制方案,是基于有限域上的 Lagrange 插值多项式,可以普遍适用于各种等级系统,具有很强的安全性,并且允许用户自主选择自己的秘密密钥.

假设等级系统的 n 个安全类依次为 $C_1, C_2, \cdots, C_n$,CA 是一个权力机构,它根据所有用户选择的秘密密钥为每一个安全类用户生成公共参数,一个安全类用户可以使用这些公共参数计算得到其后继安全类用户的秘密密钥,进而可以访问后继安全类用户的信息.

3.1 初始化算法

权力机构 CA 根据所有安全类用户的秘密密钥,通过初始化算法生成所有安全类用户的公共参数. 假设 p 是一个充分大的素数,$H: GF(p) \rightarrow GF(p)$ 是一个单向 Hash 函数,则等级系统中的访问控制方案的初始化算法步骤如下:

(1) 用一个哈斯图表示等级系统;

收稿日期:2002-04-19;修回日期:2002-10-26
基金项目:国家自然科学基金项目(No. 90104005, No. 66973034, No. 60173051)

(2)所有安全类分别自主选择自己的秘密密钥 K_i ($i = 1, 2, \cdots, n$);

(3)对哈斯图按层次自顶向下进行遍历,在遍历到的每一个非叶节点 C_i (假设其秘密密钥为 K_i)进行如下处理:

(a)找出 C_i 的所有 m 个直接后继节点 $C_{i1}, C_{i2}, \cdots, C_{im}$,假设他们的秘密密钥依次为 $K_{i1}, K_{i2}, \cdots, K_{im}$;

(b)CA 用 $(0, H(K_i) + tag_i), (1, H^2(K_i)), \cdots, (m-1, H^m(K_i)), (m, K_{i1}), \cdots, (2m-1, K_{im})$ 构造一个唯一的 $2m-1$ 次多项式 $L(x)$;

(c)CA 计算 $y_{ij} = L(2m-1+j)$,将 y_{ij} 作为 C_{ij} 的公共参数 $(j = 1, 2, \cdots, m)$,将 tag_i 登记在 C_i 节点.

上述初始化算法中 H^t 表示 Hash 函数的复合,例如 $H^3(K) = H(H(H(K)))$. 第(3)步的(b)中 tag_i 按照如下方式取值:如果

$$\sum_{t=0}^{m-1}(H^{t+1}(K_i)/\prod_{\substack{j=0\\ j\neq t}}^{2m-1}(t-j)) + \sum_{t=1}^{m}(K_{it}/\prod_{j=0}^{2m-1}(m+t-1-j)) = 0$$

成立,则 $tag_i = 1$,否则 $tag_i = 0$,这样就可以保证所构造的多项式 $L(x)$ 必定是 $2m-1$ 次的.

CA 执行完初始化过程以后,公开所有的公共参数,而各个安全类用户秘密保存其秘密密钥. 注意:如果一个安全类具有多个直接前趋,则该安全类的公共参数也有多个.

3.2　密钥恢复算法

等级系统中的任意一个安全类用户可以使用密钥恢复算法计算其后继安全类用户的秘密密钥. 安全类用户 C_i 获得其直接后继 C_{ij} 的秘密密钥的过程如下:

设安全类用户 C_i 的秘密密钥为 K_i,其直接后继依次为 $C_{i1}, C_{i2}, \cdots, C_{im}$,相应的公共参数依次为 $y_{i1}, y_{i2}, \cdots, y_{im}$. C_i 用 $(0, H(K_i) + tag_i), (1, H^2(K_i)), \cdots, (m-1, H^m(K_i)), (2m, y_{i1}), \cdots, (3m-1, y_{im})$ 构造一个唯一的 $2m-1$ 次多项式 $L(x)$,则 $C_{ij}(1 \leq j \leq m)$ 的秘密密钥 $K_{ij} = L(m-1+j)$. 依理类推,C_i 可计算出其所有的后继(包括间接后继)的秘密密钥.

3.3　有关计算方法

由初始化算法的(b)、(c)两步,根据 Lagrange 插值[4]公式,可知

$$\begin{aligned} y_{ij} &= L(2m-1+j) \\ &= (H(K_i) + tag_i)\, T_m(0, j) \\ &\quad \cdot \sum_{s=1}^{m-1} H^{s+1}(K_i)\, T_m(s, j) + \sum_{s=1}^{m} K_{is} T_m(m-1+s, j) \end{aligned}$$

其中

$$T_m(u, j) = \prod_{\substack{v=0\\ v\neq u}}^{2m-1} \frac{2m-1+j-v}{u-v}$$

初始化算法计算 $T_m(u, j)$ 的计算量为 $2m-1$ 次求逆和 $4m-3$ 次乘法,则计算 y_{ij} 的计算量为 $O(m^2)$,其中大部分时间用于计算针对于不同 u 的 $T_m(u, j)$. 为了提高计算速度,CA 可以在原来的初始化算法之前加上预处理算法,用于计算所有的 $T_m(u, j)$,这样,在计算 y_{ij} 时就只需要 $2m$ 次乘法和 $2m-1$ 次加法. 设哈斯图节点的最大直接后继数为 MAX,最小直接后继数为 MIN,则预处理算法可构造 MAX - MIN + 1 个表 $T_{MIN}, T_{MIN}+1, \cdots, T_{MAX}$,其中表 T_m 是一个 $2m \times m$ 的二维数组.

由密钥恢复算法可知,C_{ij} 的秘密密钥 K_{ij} 为

$$\begin{aligned} K_{ij} &= L(m-1+j) \\ &= (H(K_i) + tag_i)\, W_m(0, j) \cdot \sum_{s=1}^{m-1} H^{s+1}(K_i)\, W_m(s, j) \\ &\quad + \sum_{s=1}^{m} y_{is} W_m(2m-1+s, j) \end{aligned}$$

其中

$$W_m(u, j) = \prod_{\substack{0\leq v\leq m-1 \text{或} \\ 2m\leq v\leq 3m-1 \\ v\neq u}} \frac{m-1+j-v}{u-v}$$

与初始化算法同样道理,密钥恢复算法计算 K_{ij} 的时间复杂度为 $O(m^2)$,其中大部分时间用于计算针对于不同 u 的 $W_m(u, j)$. 为了提高计算速度,CA 可以在初始化过程结束后加上密钥恢复算法的预处理过程,用于计算所有的 $W_m(u, j)$. 密钥恢复算法的预处理过程只进行一次,但预处理的结果却能为很多次的密钥恢复过程服务,这样,在计算 Kij 时就只需要 $2m$ 次乘法和 $2m-1$ 次加法. 设哈斯图节点的最大直接后继数为 MAX,最小直接后继数为 MIN,则密钥恢复算法的预处理过程可构造 MAX - MIN + 1 个表 $W_{MIN}, W_{MIN}+1, \cdots, W_{MAX}$,其中表 W_m 是一个 $2m \times m$ 的二维数组.

3.4　性能分析

3.4.1　空间复杂度和时间复杂度　设等级系统中共有 n 个安全类,相应哈斯图的边数为 e,节点的最大直接后继数为 d. Chang 方案的空间复杂度为 $O(n)$,初始化算法的时间复杂度为 $O(nd^2)$,一个安全类用户计算一个直接后继的秘密密钥的密钥恢复算法的时间复杂度为 $O(d^2)$. 本文所提出的访问控制方案存储 T 表和 W 表的空间复杂度为 $O(d^3)$,存储 n 个安全类的公共参数的空间复杂度为 $O(e)$,总的空间复杂度为 $O(e+d^3)$. 初始化算法的预处理过程和密钥恢复算法的预处理过程的时间复杂度均为 $O(d^4)$. 初始化算法的时间复杂度为 $O(nd)$,初始化过程总的时间复杂度为 $O(nd+d^4)$. 一个安全类用户计算一个直接后继的秘密密钥的密钥恢复算法的时间复杂度为 $O(d)$. 综上所述,本文的访问控制方案具有很高的时间和空间效率.

3.4.2　安全性讨论　假设安全类 C_i (设其秘密密钥为 Ki)的所有 m 个直接后继 $C_{i1}, C_{i2}, \cdots, C_{im}$ (设他们的秘密密钥依次为 $K_{i1}, K_{i2}, \cdots, K_{im}$)合谋猜测 C_i 的秘密密钥,他们已知的信息为 $(m, K_{i1}), (m+1, K_{i2}), \cdots, (2m-1, K_{im}), (2m, y_{i1}), (2m+1, y_{i2}), \cdots, (3m-1, y_{im})$,可以重新构造一个唯一的 $2m-1$ 次多项式 $L(x)$,进而可以计算出 $H(K_i) + tag_i = L(0)$, $H^2(K_i) = L(1)$, $\cdots$, $H^m(K_i) = L(m-1)$,但根据单向 Hash 函数的性质,他们得不到 C_i 的秘密密钥 K_i. 因此,本文的访问控制方案的安全性依赖于单向 Hash 函数的安全性,如果单向 Hash 函数是无条件安全的,则本文的访问控制方案在抵抗后继安全类用户合谋猜测他们的前趋安全类用户的秘密密钥方面也是无条件安全的.

假设一个安全类的直接后继中部分安全类用户合谋猜测同级的其他兄弟安全类用户的秘密密钥,不妨设 C_i 的 m 个

直接后继中 C_{i2}, C_{i3}, …, C_{it} $(t < m)$ 合谋猜测 C_{i1} 的秘密密钥, 他们已知的信息为 $(m+1, K_{i2})$, $(m+2, K_{i3})$, …, $(m+t-1, K_{it})$, $(2m, y_{i1})$, $(2m+1, y_{i2})$, …, $(3m-1, y_{im})$, 不能重构唯一的一个 $2m-1$ 次多项式, 由秘密共享的 Shamir[5] 方案可知 C_{i1} 的秘密密钥可取 $GF(p)$ 上的任意值, 猜测 K_{i1} 失败.

针对等级系统中访问控制方案的各种攻击均可归结为如上两种类型. 综上所述, 本文的访问控制方案可以抵抗各种攻击, 具有很强的安全性.

本文所讨论的等级系统中的访问控制方案中使用了单向 Hash 函数[6], 单向 Hash 函数的安全性对访问控制方案的安全性至关重要. 目前已经存在很多安全性很好的单向 Hash 函数, Rivest[7] 设计了一个称为 MD4 的 Hash 函数, 之后又设计了 MD5[8]. 另外, 美国国家标准与技术研究所 (NIST) 和美国国家安全局 (NSA) 也设计了安全 Hash 标准[9]. 这些 Hash 函数均可以用于等级系统访问控制方案, 除此以外, 也可以采用指数函数或安全的幂函数做 Hash 函数.

4 基于门限秘密共享体制的一般性访问控制方案

本节基于秘密共享体制[10]构造了等级系统中的一般性的访问控制方案, 在方案中没有使用具体的秘密共享体制. 使用者在具体实现等级系统中的访问控制方案时只要方案中的有关部分替换为具体的秘密共享体制即可.

4.1 一般性的访问控制方案的基本思想

假设安全类 C_i 的所有直接后继安全类依次为 C_{i1}, C_{i2}, …, C_{it}, 他们所选择的秘密密钥依次为 SK_{i1}, SK_{i2}, …, SK_{it}, 权利机构 CA 通过计算分配给他们的公共参数为 PD_{i1}, PD_{i2}, …, PD_{it}.

SK_{i1}, SK_{i2}, …, SK_{it}, PD_{i1}, PD_{i2}, …, PD_{it} 均被看作一个门限秘密共享体制中的片段, 它们在门限秘密共享中的地位是平等的. 因此, 已知足够数量的片段可以利用门限秘密共享体制的恢复算法计算出一个合成的秘密信息, 再利用门限秘密共享体制的片段分配算法就可以计算出其他的一些片段.

在进行公共参数的初始化计算时, 权利机构 CA 使用 $(2t+1, 3t+1)$ 门限秘密共享体制的恢复算法, 用 SK_{i1}, SK_{i2}, …, SK_{it} 以及 C_i 所掌握的 $t+1$ 个片段合成一组中间数据, 再使用片段的分配算法计算出 PD_{i1}, PD_{i2}, …, PD_{it}. 当 C_i 需要计算得到他的 t 个直接后继的秘密密钥时, C_i 使用与初始化过程相类似的计算步骤, 用他所掌握的 $t+1$ 个片段以及作为公开参数的 t 个片段 PD_{i1}, PD_{i2}, …, PD_{it} 计算得到 SK_{i1}, SK_{i2}, …, SK_{it}.

4.2 一般性的访问控制方案的构造

假设等级系统的 n 个安全类依次为 C_1, C_2, …, C_n, CA 是一个权力机构, 它根据所有用户选择的秘密密钥为每一个安全类用户生成公共参数.

4.2.1 初始化算法 权利机构 CA 根据所有安全类用户的秘密密钥, 通过初始化算法生成所有安全类用户的公共参数. 假设 p 是一个充分大的素数, $H: GF(p) \rightarrow GF(p)$ 是一个单向 Hash 函数, 则等级系统中的一般性的访问控制方案的初始化算法步骤如下:

(1) 用一个哈斯图表示等级系统;

(2) 所有安全类分别自主选择自己的秘密密钥 SK_i $(i = 1, 2, \cdots, n)$;

(3) 对哈斯图按层次自顶向下进行遍历, 在遍历到的每一个非叶节点 C_i (假设其秘密密钥为 SK_i) 进行如下处理:

(a) 找出 C_i 的所有 t 个直接后继节点 C_{i1}, C_{i2}, …, C_{it}, 假设他们的秘密密钥依次为 SK_{i1}, SK_{i2}, …, SK_{it};

(b) CA 使用 $(2t+1, 3t+1)$ 门限秘密共享体制, 用 SK_i, $H(SK_i)$, …, $H^t(SK_i)$, SK_{i1}, …, SK_{it} 作为 $2t+1$ 个片段 (设这些片段在门限秘密共享体制中的公开信息依次在 $x_0, x_1, \cdots, x_t, x_{t+1}, \cdots, x_{2t}$), 计算相对于公开信息 x_{2t+j} 的片段 PD_{ij} 并将 PD_{ij} 作为 C_{ij} 的公共参数 $(j = 1, 2, \cdots, t)$, 这里 x_{2t+j} $(j = 1, 2, \cdots, t)$ 是公开的.

CA 执行完初始化过程以后, 公开所有的公共参数, 而各个安全类用户秘密保存其秘密密钥. 如果一个安全类具有多个直接前趋, 则该安全类的公共参数也有多个.

4.2.2 密钥恢复算法 等级系统中的任意一个安全类用户可以使用密钥恢复算法计算其后继安全类用户的秘密密钥. 安全类用户 C_i 获得其直接后继 C_{ij} 的秘密密钥的过程如下:

设安全类用户 C_i 的秘密密钥为 SK_i, 其直接后继依次为 C_{i1}, C_{i2}, …, C_{it}, 相应的公共参数依次为 PD_{i1}, PD_{i2}, …, PD_{it}. C_i 使用 $(2t+1, 3t+1)$ 门限秘密共享体制, 用 SK_i, $H(SK_i)$, …, $H^t(SK_i))$, PD_{i1}, PD_{i2}, …, PD_{it} 作为 $2t+1$ 个片段 (这些片段在门限秘密共享体制中的公开信息依次在 $(x_0, x_1, \cdots, x_t, x_{2t+1}, \cdots, x_{3t})$, 计算相对于公开信息 x_{t+j} 的片段 SK_{ij} 并将 SK_{ij} 作为 C_{ij} 的秘密密钥 $(j = 1, 2, \cdots, t)$, 这里 x_{t+j} $(j = 1, 2, \cdots, t)$ 是公开的. 依理类推, C_i 可计算出其所有的后继 (包括间接后继) 的秘密密钥.

4.3 安全性分析

上述访问控制方案是安全的. 假设安全类 C_i 的所有直接后继依次为 C_{i1}, C_{i2}, …, C_{it}, 因为 C_i 的任意 v $(1 \leq v \leq t)$ 个直接后继安全类均不拥有 $(2t+1, 3t+1)$ 门限秘密共享体制恢复算法所要求的 $2t+1$ 个片段, 他们最多拥有 $2t$ 个片段. 如果访问控制方案中所使用的门限秘密共享体制是完善的, C_i 的任意多个直接后记安全类合作均得不到有关 C_i 以及他们的兄弟安全类的秘密密钥的任何部分信息. 上述一般性的访问控制方案满足等级系统访问控制的要求, 即: 任何安全类能够并且仅能够获得他的后继安全类的秘密密钥.

5 结论

本文对等级系统中的访问控制问题进行了讨论, 提出了基于 Lagrange 插值的访问控制方案, 它具有如下特点: 算法简洁, 易于实现; 普遍适用于各种偏序集所表示的等级系统; 能抵抗后继安全类用户合谋猜测前趋安全类用户的秘密密钥; 能抵抗部分安全类用户合谋猜测同级的其他兄弟安全类用户的秘密密钥; 允许所有安全类用户自主地选择一般性的访问控制方案, 在该方案中使用任何一个具体的门限秘密共享体

制均可.

参考文献:

[1] S G Akl, P D Taylor. Cryptographic solution to a problem of access in a hierarchy[J]. ACM trans. Compute. Syst. 1983, 1(3): 239 - 248.

[2] S J Mackinnon, P D Taylor. An optimal algorithm for assigning cryptographic keys to access control in a hierarchy[J]. IEEE Trans. Compute. 1985, 34(9): 797 - 802.

[3] C C Chang, R J Hwang. Cryptographic key assignment scheme for access control in a hierarchy[J]. Information Systems, 1992, 17(3): 243 - 247.

[4] Richard L Burden, J Douglas Faires. Numerical Analysis (Seventh Edition), 数值分析(第七版影印版)[M]. 北京:高等教育出版社, 2001.

[5] A Shamir. How to share a secret[J]. Communications of the ACM, 1979, 22(1): 612 - 613.

[6] 李克洪, 王大玲, 董晓梅. 实用密码学与计算机数据安全[M]. 沈阳:东北大学出版社, 1997.

[7] R L Rivest. The MD4 message digest algorithm[A]. Advances in Cryptology—CRYPTO '90 Proceedings[C]. Springer-Verlag, 1991. 303 - 311.

[8] R L Rivest. The MD5 message digest algorithm[A]. RFC 1321[C]. 1992.

[9] B Schneier. 应用密码学——协议、算法与 C 源程序[M]. 北京: 机械工业出版社, 2000.

[10] 许春香, 陈恺, 肖国镇. 安全的矢量空间秘密共享方案[J]. 电子学报, 2002, 30(5): 715 - 718.

作者简介:

王丽娜　女, 1964 年 10 月出生于辽宁省营口市, 博士, 副教授, 主要研究领域为计算机网络安全, 发表论文 20 余篇, 编著 2 部, 获辽宁省政府科技进步一等奖.

费如纯　男, 1969 年 5 月出生于河北省昌黎县, 硕士, 讲师, 主要研究领域为计算机安全及密码学, 发表论文 7 篇.