计算机科学 2005Vol. 32№. 11

一种组播钥匙分配的分析和设计方法*)

胡 若[1,2] 钱省三[2]

(上海理工大学管理学院 上海 200093)[1] (宁夏大学数学计算机学院 银川 750021)[2]

摘 要 基于生成树组播钥匙分配方案的各种方法在提供有效的用户删除方法的同时努力将用户钥匙存储最小化。在这篇论文中,我们说明可以运用信息理论的基本概念来系统地研究生成树框架上的用户钥匙存储,同时可以将基于生成树组播钥匙分配问题作为最优化的问题提出,特别说明通过成员删除统计的平均信息量可以确定分配给一个成员钥匙平均数的最佳值。我们接着通过一个钥匙分配方案的实例说明了生成树上钥匙分配可以达到最优化但不能防止用户串通。

关键词 组控制器,最优化问题,密钥加密密钥,会话加密钥匙

A Method for Analysis and Design of Multicast Key Distribution

HU Ruo[1,2] QIAN Xing-San[2]

(College of Management, University of Shanghai for Science and Technology, Shanghai 200093)[1]

(College of Math and Computer, Ningxia University, Yinchuan 750021)[2]

Abstract All kinds of different methods based on rooted-tree multicast key distribution schemes make efforts to minimize the user key storage while providing efficient member deletion. In this paper, we show that the user key storage on rooted trees can be systematically studied using basic concepts from information theory, and the rooted-tree-based multicast key distribution problem can be posed as an optimization problem. In particular, we show that the entropy of member deletion statistics can quantifies the optimal value of the average number of keys to be assigned to a member. We then demonstrate the key distribution on rooted trees with an example of a key distribution scheme that attains optimality but fails to prevent user collusion.

Keywords Droup controller, Optimization problem, Key-encrypting key, Session-encrypting key

1 介绍

当一个相同的信息必须传递给多个接收者时,组播是一个较好的通信模型,组播通信减少了发送者和网络媒介的经常开支。密码学的应用对于作用在一个不可靠的网络媒介上的安全组播通信来说是一个实际可行的方法。当密码学用于安全通信时,一个会话加密钥匙(SEK)用于将数据加密。

由于数据分配给多个接受者,为了减少在发送者节点的加密数量同时最小化网络上数据包的数量,每个预期的接收者以及发送者应该共享一个相同的 SEK。为了确保只有组中合法的成员具有信息传递的权力,SEK 在下列情况时需要改变:a)当 SEK 的生命周期到期时,b)组的成员关系有一个改变,或者 c)一个或更多的成员被泄露时。

由于以下原因,SEK 需要更新:a)当一个新成员加入时,要确保新成员不能传递组过去的信息 b)当一个新成员离开或删除时,要确保离开或删除的成员不能执行组未来的通信。

由于组分布在不可靠的网络中,每当 SEK 失效时,需要有称为密钥加密密钥(KEK)——可以将更新的 SEK 加密后传递到合法的组成员。

我们说明可以将基于生成树的 KEK 分配问题作为最优化问题来研究,我们接着说明了最优化问题可以抽象地等同于信息理论中的最优名称长度选择问题。通过组控制器(GC)来更新最佳 KEKs,组控制器作用在成员删除情况下,最佳 KEKs 与成员删除统计平均数相关。基于生成树的 KEK 分配方案建立在钥匙分配的一个虚拟或逻辑的树层次上。

2 基于生成树的钥匙分配

这里提到针对安全组播通信的基于生成树的钥匙分配方法、一个二叉生成树及钥匙图的方法,这些方法都构造了一个基于组大小的逻辑树或钥匙图,而没有对这些钥匙之间的相互关系作任何假设。这两个方案的 GC 的钥匙存储需求以 O(n)的速度增长,同时钥匙更新的通信以及用户的存储需求以 O(log$n$)的速度增长。现在讨论基于生产树的分配方案。

2.1 二叉树上的钥匙分配

图 1 提供了一个针对 8 个成员分组的钥匙分配二叉树,这个逻辑树构筑的每个组成员都分配了树的唯一叶子节点。这里 K_{01},K_{02}、K_{03}、K_{04}、K_{05}、K_{06}、K_{07}、K_{08} 分别是成员 M_1、M_2、M_3、M_4、M_5、M_6、M_7、M_8 的叶子钥匙,K_{21}、K_{22}、K_{11}、K_{12}、K_{13}、K_{14} 分别是节点钥匙。这里给逻辑树的每个节点分配了一个 KEK,沿着一个叶子节点到根节点路径上所有节点的 KEKs 集合分配给和这个叶子节点相联系的成员。例如,在图 1 中,叶子节点为 K_{01},M_1 和 K_{01} 相对应,给 M_1 分配的密钥加密密钥(KEK)为{K_0,$K_{2,1}$ $K_{1,1}$,$K_{0,1}$}。

*)数字城市信息安全管理体系研究(上海市教委发展基金项目,编号:02GK09)。国家信息安全应用示范工程(S219)的子项目:信息安全集成系统研究(编号:2000-A32-08)。胡 若 博士生,主要研究方向:信息安全管理。钱省三 博导,教授。

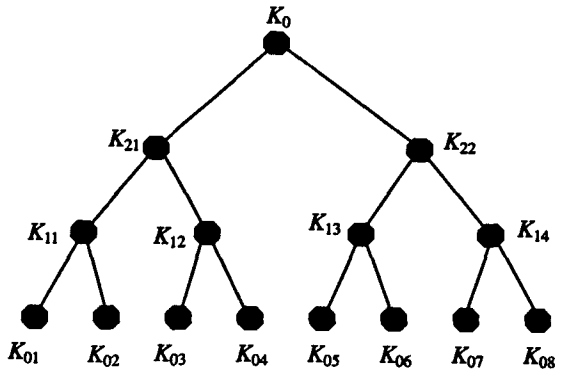

图 1　逻辑钥匙树

由于所有的成员共享根钥匙，如果组成员关系没有改变，K_0 可以用来更新所有成员的 SEK。一些基于树的钥匙分配方案使用根节点钥匙作为会话加密密钥和组 KEK，在安全领域，为了防止安全侵犯不允许使用相同的钥匙针对不同的功能。

基于树的结构也导致了成员间的一个自然的层次分组。通过把成员分配到合适的节点，GC 可以形成期望的成员层次群以及钥匙的选择性更新。例如在图 1 中，成员 M_5，M_6，M_7 和 M_8 唯一地共享钥匙 $K_{2,2}$，GC 可以使用钥匙 $K_{2,2}$ 来和这些成员进行有选择地通信，GC 可以决定基于特定应用需求树的成员群。实际上，在一个大小为 N 的组中 GC 可以传递到所有 2^N 个用户子集合。为了能够有选择地给组成员的一个子集合传播信息，GC 必须保证分配给一个子集合的公共钥匙没有分配给不属于这个子集合的任何成员。

GC 可以通过下面的传送有选择地发送一个消息 m 到成员 M_5，M_6，M_7 和 M_8：$GC \rightarrow M_5, M_6, M_7, M_8 : \{m\}_{K_{2,2}}$。

如果钥匙 $K_{2,2}$ 由于某种理由是无效的，GC 在为成员 M_5，M_6，M_7 和 M_8 使用一个公共钥匙之前必需更新无效的钥匙 $K_{2,2}$，这个树结构允许用两个更新消息来完成操作。GC 可以首先产生一个新版的 $K_{2,2}$ 表示为 $\hat{K}_{2,2}$，同时执行两个加密，一个用 $K_{1,3}$ 加密而另一个用 $K_{1,4}$ 加密。对于组的相关成员需要用下面的两个消息更新钥匙 $\hat{K}_{2,2}$：

$GC \rightarrow M_5, M_6 : \{\hat{K}_{2,2}\}_{K_{1,3}}$；　　$GC \rightarrow M_7, M_8 : \{\hat{K}_{2,2}\}_{K_{1,4}}$

2.2　生成树中的成员删除

由于 SEK 和作为根 KEK 的 K_0 对于组中所有的成员是共同的，每次当一个成员删除时，需要将它们作废，也由于其他有效的成员可能共享多个 KEK，需要将 KEK 更新。在大量成员删除的事件中，GC 必须执行下面的操作：a)识别所有的无效钥匙；b)发现最小的有效 KEK 数，这些钥匙可以用来传送更新的钥匙；c)用新钥匙更新有效的成员。

对于一个 d 叉树来说(在图 1 中 d=2)，在成员删除后，通过成员 M_1 在图 1 中的例子(在下面)讨论通用的规则。成员 M_1 通过四个钥匙的集合：$\{K_0, K_{2,1}, K_{1,1}, K_{0,1}\}$ 来标记，删除成员 M_1 导致这四个钥匙和 SEK 的无效、产生 $\log_d N$ 个新的密钥加密密钥，同时更新合适的有效成员，这些有效的成员与成员 M_1 共享这些无效的钥匙。当 M_1 删除时，下面的更新是必需的：a)所有的成员需要新的根钥匙 K_0 和新的会话加密钥匙 SEK；b)成员 $M_2 \rightarrow M_4$ 需要更新 $\{K_{2,1}\}$；c)成员 M_2 需要更新 $\{K_{1,1}\}$。对于生成树钥匙分配可以得出下面的结论：

• 由于每个成员要分配 $(2+\log_d N)=\log_d Nd^2$ 个钥匙，删除一单个成员要求 $(2+\log_d N)$ 个钥匙作废。

• 由于生成树的每个节点分配一个钥匙同时每个成员和多个成员共享 $\log_d N$ 个节点，在一个成员删除时需要更新的 KEKs 的数目是 $\log_d N$。

• 对于一个深度为 $h=\log_d N$ 的 d 叉树来说，GC 必需存储 $1+1+d+d^2+\cdots+d^h=\frac{d(N+1)-2}{(d-1)}$ 个钥匙。设 $d=2$ 时导致二叉树要求的存储量为：$\frac{2(N+1)-2}{2-1}=2N$。可以通过检查一个具有 N 个叶子 $2N-1$ 个节点的二叉树得到这个结果。因此，GC 必须保存 SEK 以及 $(2N-1)$ 个 KEKs，这导致了 $2N$ 个钥匙。

基于二叉生成树的钥匙分配要求 GC 保存 $2\log_2 N$ 个不同的钥匙，对于一个 d 叉树，要求 GC 保存 $d\log_d N$ 个钥匙，当一个成员删除时需要更新的钥匙数为 $\log_d N$，因此，当树是二叉树时，减少的存储数量为：$2(N-\log_2 N)$。对于大的 N，减少的存储为 $O(N)$，但是在多个成员删除时，它以增加安全问题为代价，这在后面的第 5 部分讨论。

3　初步结论

首先通过一个最坏情况的例子来说明需要优化生成树，考虑图 2 所示的二叉树，给每个成员分配一个唯一的叶子节点，这里假定组的大小为 N。在这个树中，当一个成员删除时平均更新的钥匙数计算为：

$$\frac{\sum_{i=1}^{N-1}(i+1)+N}{N}=\frac{N}{2}+1.5-\frac{1}{N} \qquad (1)$$

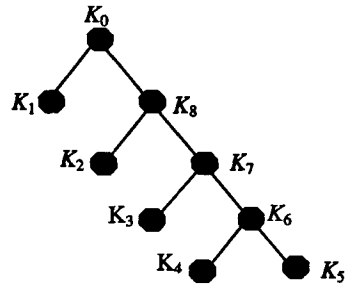

图 2　一个钥匙分配树

因此，对于这个模型来说要作废的钥匙平均数增长为 $O(N)$。一个虚拟树建立在大小为 N 的组之上，每个成员根据对 N 个成员的观察分配钥匙，对于一个 d 叉树，要更新的钥匙平均数增长为 $\log_d N$。

这里首先定义用公式表示优化问题的术语和过程。

3.1　一个自由覆盖的钥匙分配

在将钥匙分配给成员时，GC 也需要确信在多个成员非法串通时，要使这些成员不能覆盖分配给一个合法成员的所有钥匙。在生成树钥匙分配情况下，自由覆盖的特性要求不考虑有多少成员同时串通或被删除，每个合法的成员应该能够安全地和 GC 通信。这里正式定义集合间的安全覆盖特性如下：

定义：给定一个集合的收集：$\{S_1, \cdots, S_N\}$，一个非空集合 S_i 被 (k,a) 自由覆盖，如果满足

$$\frac{|S_i \setminus \bigcup_{j=1, j\neq i}^{k} S_j|}{|S_i|} \geqslant \alpha_i \qquad (2)$$

这里 $0 \leqslant \alpha_i \leqslant 1$，$1 \leqslant k \leqslant N$。当 $\alpha_i=0$ 时，对于钥匙集合 S_i 没有

自由覆盖的钥匙分配，对于自由覆盖条件，$\alpha>0$。$|S_i\setminus\bigcup_{j=1,j\neq i}^{k}S_j|$解释为：集合 S_i 中除去在$\{S_1,\cdots,S_j,\cdots,S_N\}$(其中 $j\neq i$)中出现的元素而得到剩下的元素个数。

3.2 用户标识和钥匙标识

设 $X_{n-1}X_{n-2}X_{n-3}\cdots X_0$ 表示二进制用户标识(UID)串，这里 X_i 取值"0"或"1"。将分配给成员的钥匙集合和形成一个钥匙标识(KID)联系在一起，每个成员应该有一个唯一的KID。

尽管KID和UID需要一一对应，一个KID需要满足附加的限制，而一个UID不需要满足。首先通过一个实例说明，考虑符号表{0,1}用作UID产生，而钥匙$\{K_1,K_2\}$用作KID产生。可以产生UIDs"01"和"10"并且唯一地分配给两个不同的成员。但是，KIDs钥匙标识 K_1K_2 和 K_2K_1 不能分配给两个不同的成员，分配给一个成员的钥匙可以完全地覆盖分配给另一个成员的钥匙。

尽管这是一个从钥匙改变得来的覆盖集合的特别类型，在定义KIDs时这是关键的。对于这个例子，我们注意到下面的KIDs特性：任何构成一个KID钥匙的假设将导致一个KID完全被初始的KID覆盖，使用这个特性来正式地定义KID：

KID定义：一个成员 M_i 的钥匙标识是一个串，分配给成员 M_i 的钥匙以任何连接顺序产生这个串，如果分配给成员 M_i 的钥匙数表示为 l_i，则通过这些 l_i 个钥匙可能产生 $l_i!$ 个不同的KID串。对于一个给定的KID，考虑自由覆盖特性，通过改变和结合它的钥匙而产生的所有的KIDs是相同的。如果产生 KID_1 的钥匙集合表示为 S_1，同时产生 KID_2 的钥匙集合表示为 S_2，我们得出：如果 $S_1=S_2$，$KID_1=KID_2$。

从图1中看到，给成员 M_1 分配4个成员$\{K_0,K_{2,1},K_{1,1},K_{0,1}$，由于 K_0 对于所有的成员是共同的，在定义KIDs时可以将它忽略，因此，M_1 的KID是 $K_{2,1}K_{1,1}K_{0,1}$。由于有六个不同的方法来连接这些钥匙，有五个额外的KIDs，这些KIDs是通过改变和连接钥匙而产生，这些钥匙构成 M_i 的KIDs。通过改变一个钥匙集合产生这些KIDs，这些KIDs之间的等价性是区分普遍的UIDs和KIDs的一个特征。

3.3 产生KIDs自由覆盖

如果一个KID分配是自由覆盖的，它必须是前束无约束的，我们说明前束无约束不能意指自由覆盖。构造一个前束无约束同时也是自由覆盖的生成树上的钥匙分配需要考虑给出自由覆盖条件的定义，在一个或多个成员删除时，由于每个合法成员需要能够安全地和GC通信，即使所有的$(N-1)$成员被删除，剩下的唯一的成员应该能够安全地通信。根据以前给定的参数(k,α)，设置 $k=(N-1)$ 导致对于一个合法成员 M_i 的自由覆盖条件所具有钥匙集合 S_i 满足：

$$\frac{|S_i\setminus\bigcup_{j=1,j\neq i}^{N-1}S_j|}{|S_i|}\geqslant\alpha_i \tag{3}$$

这里 $0<\alpha_i\leqslant 1$。由于 $\alpha_i>0$，α_i 的最低可能值为 $\frac{1}{|S_i|}$，这里实际的解释为集合 S_i 应该至少有一个钥匙，这个钥匙不同于所有其它钥匙集合的合并。

为了构造满足这个条件的基于树的钥匙分配方案，我们考虑将成员分配给一个逻辑树的方式，每个成员分配一个唯一的叶子节点。存在一个从叶子到根节点的唯一路径，每个成员和多个成员共享除叶子节点外的所有钥匙，因此，选择不同的叶子节点钥匙将确信成员 M_i 的钥匙集合 S_i 至少有一个元素不被其他钥匙集合的合并所覆盖。

因此，如果我们选择所有叶子节点钥匙都不同，基于生成树的无约束的KID分配对于下列情况将是充分必要的：a)防止用户串通以造成完全破坏安全通信；b)在任意个成员删除时，传递给一个有效的成员。由于有N个叶子节点，当在钥匙间没有额外的成员关系时GC保存的钥匙数以 $O(N)$ 增长。具有这些初步的结论，现在说明在成员删除时如何最小化GC重新产生的钥匙平均数。

4 成员删除的概率

用 l_i 表示成员 M_i 的KID长度，在基于生成树的钥匙分配模型中，成员 M_i 和两个或更多的其他成员共享 l_i 个KEKs，这个数 l_i 包含了根KEK而不包含成员 M_i 的叶子节点钥匙。在这个成员删除的事件中，更新的钥匙数也是 l_i，因此，如果可以最小化 l_i，就可以最小化用户的钥匙存储。

在基于生成树的钥匙分配中，每个成员和其他的成员共享 $\log_d N$ 个钥匙(不包括根钥匙和SEK)，当一个成员被删除时，将 $\log_d N$ 个钥匙更新并传送给其他成员。在成员删除时唯一不需要更新的钥匙是被删除成员的叶子节点钥匙，因此，在成员删除时，对于基于生成树的钥匙分配方案来说用户存储及更新的钥匙以 $O(\log_d N)$ 增长。根据成员删除过程的统计数字表明进一步减少用户钥匙存储，同时减少由此得出的钥匙更新需求。

4.1 成员删除的平均信息量

设 p_i 表示成员 M_i 删除的可能性，删除一个成员的可能性等同于删除它的节点钥匙以及它的KID的可能性。通过下面公式定义成员删除的平均信息量(也称为熵)：

定义 d 叉树成员删除的平均信息量 H_d 计算为[7,8]：

$$H_d=-\sum_{i=1}^{N}p_i\log_d p_i \tag{4}$$

4.2 给每个成员分配钥匙的最佳数

当一个成员 M_i 删除时具有一个可能性 p_i，组控制器必须产生和更新与其他成员共享的 l_i 个钥匙。因此，GC平均必须产生和更新

$$\sum_{i=1}^{N}p_il_i \tag{5}$$

个钥匙，这也是成员需要分配的平均钥匙数。GC必须找到一个最佳钥匙分配方案，这种分配方案将最小化更新钥匙的平均数。出现在生成树模型中的组播钥匙分配优化表示为最小化更新钥匙平均数：

$$\min_{l_i}\sum_{i=1}^{N}p_il_i \tag{6}$$

服从于限制 $\sum_{i=1}^{N}d^{-l_i}\leqslant 1$ (7)

这个问题可以表示为一个拉格朗日优化问题：

$$\min_{l_i}\{\sum_{i=1}^{N}p_il_i+\lambda(\sum_{i=1}^{N}d^{-l_i}-1)\} \tag{8}$$

这里 λ 是一个拉格朗日乘数。这个优化问题等同于信息理论中的最优名称长度选择问题[7]。

在基于生成树的钥匙分配环境中，更新钥匙的最佳数是成员删除过程的平均信息量，可以将这个结果总结为定理1(不再证明)：

定理1[9,10] 设 p_i 表示成员 M_i 的删除可能性，设组的大小为N，设钥匙分配生成树为d叉树，KEKs的最优平均数

表示为：$\sum_{i=1}^{N} p_i \log_d p_i$(分配给一个成员)，下面给出 d 叉树的成员删除事件平均信息量：

$$H_d = -\sum_{i=1}^{N} p_i \log_d p_i \tag{9}$$

包括根钥匙和 SEK，给定每个成员钥匙的最优平均数为：H_d+2。对于一个具有删除可能性 p_i 的成员 M_i，从式(8)中计算得到的分配钥匙最优数(不包括根钥匙和会话加密钥匙)为：

$$l_i^* = -\log_d p_i \tag{10}$$

具有删除可能性 p_i 的分配给成员 M_i 的钥匙数(包括 SEK 和根钥匙)计算为：

$$l_i^* + 2 = -\log_d p_i + 2 = \log_d \frac{d^2}{p_i} \tag{11}$$

引理 1 1)对比具有较低删除可能性的一个成员来说，应该给具有较高删除可能性的一个成员更少的钥匙。如果 $p_i > p_j$，则 $l_i(=-\log_d p_i) < l_j(=-\log_d p_j)$。

2)需至少有两个成员具有最长的 KID 串。

3)由于 GC 重新生成的钥匙数必须是一个整数同时 H_d 可能不是一个整数，每个成员真正的钥匙平均数与 H_d 最多在一个位上不同。(证明从略)

因此，KID 长度最多比成员删除平均数多一。由于 SEK 和根 KEK 对于所有成员来说是共同的，更新钥匙的平均数最多比成员删除事件的平均数多 3。

4.3 钥匙分配的最大平均信息量

现在解释基于生成树的钥匙分配结果，同时考虑到在前面的小节中得到的结果说明 GC 更新的钥匙平均数是：$H_d = -\sum_{i=1}^{N} p_i \log_d p_i$。现在发现产生的钥匙平均数的最大值为：

$$\max_{p_i} H_d = -\sum_{i=1}^{N} p_i \log_d p_i \tag{12}$$

服从于条件 $\sum_{i=1}^{N} p_i = 1$ (13)。

假定这个组的成员具有相同的删除可能性，对于这个成员删除可能性的值，给出最大的平均信息量值为：

$$H_d(\max) = -\sum_{i=1}^{N} N^{-1} \log_d N^{-1} = \log_d N$$

当成员 M_i 的删除可能性是 $p_i = N^{-1}$，分配给 M_i 的钥匙最优数是 $\log_d N$。

生成树上的每个成员分配 $\log_d N$ 个钥匙。由于说明平均信息量是分配给一个成员的钥匙平均数，同时当所有的成员具有相同的删除可能性时平均信息量得以最大限度地增加，钥匙分配对应于给每个成员分配最大钥匙平均数的策略。根据生成树方案的设计，每个成员分配 $\log_d N$ 个钥匙与假定最坏情况下钥匙分配的平均数相符合，下面的定理总结了这个问题以及解决方法。

定理 2 在一个基于 d 叉树的组播钥匙分配方案中，有 N 个成员同时成员 M_i 具有删除可能性 p_i，GC 重新产生的钥匙平均数上界为 $\log_d N$ 加上一个常数(会话加密钥匙)，当整个组具有相同的删除可能性时就达到钥匙平均数上界。

4.4 基于单向函数关系的钥匙选择

没有关于 KEKs 产生方式的详细说明，KEK 在生成树上使用伪随机单向函数进行构造。

一个混合方案是将一个组播分组分成大小为 M 的群，给每个群的成员分配一个使用伪随机函数产生的唯一钥匙，这个伪随机函数具有一个共同的种子，这样，就减少了这个群的 M 个成员的存储空间。由于假定每个群具有相同的大小 M，对于一个大小为 N 的组，有 $\left[\frac{N}{M}\right]$ 个群。由于给每个群分配一个 d 叉生成树的唯一叶子节点，这个树的高度或深度为：

$$h = \log_d \left[\frac{N}{M}\right] \tag{14}$$

对于大小为 $M = \log_d N$ 的群，GC 存储的钥匙数是：

$$1 + d + d^2 + \cdots + d^h = \frac{d^{h+1}-1}{d-1} = \frac{dN-M}{(d-1)M} \tag{15}$$

由于任何信息量不能大于成员删除事件所提供的平均信息量，基于平均信息量的公式将产生最低的平均费用，这个平均费用是当在钥匙间没有额外的成员关系时钥匙生成的平均费用。唯一能进一步减少在钥匙生成时的平均通信或存储空间的方法是在产生的钥匙间引进成员关系。在我们的公式中，由于将基于成员删除可能性的钥匙分配平均数最小化，当删除可能性 P_i 很小时，成员 M_i 的最优 KID 长度 $\log_d \frac{1}{p_i}$ 将大于 $\log_d N$，因此，在成员 M_i 删除时更新的钥匙数将会很大。

5 针对用户串通的钥匙分配

现在描述一个基于生成树的钥匙分配方案，这个方案满足分配给一个成员钥匙数的最大平均信息量边界 $\log_d N$，同时使 GC 的存储最小化，使用这个方案来说明当一个钥匙分配方案可能获得用户存储最优化时，它可能不是串通自由的。

设 $X_{n-1}X_{n-2}\cdots X_0$ 表示成员的一个二进制 UID，每个位 X_i 或者是一个 0 或者是一个 1，对于这种序列有 2^n 种不同的 UIDs。下面提出 KIDs 和 UIDs 间的直接映射，当 $N=8$ 时，$\log_2 8 = 3$ 位用来唯一地标识所有 8 个成员。由于 UID 的每个位 X_i 取两个值，这两个值可以映射到一对不同的钥匙，例如，当 X_i 是“0”时，它表示钥匙 K_{i0}，同时当 X_i 是“1”时，它表示钥匙 K_{i1}。表 1 说明了当 $N=8$ 时在标识(ID)位数字和钥匙之间的映射，在这里钥匙对 (K_{i0}, K_{i1}) 表示了成员标识位 X_i 两个可能的值。

表 1 N=8 时在标识(ID)位数字和钥匙之间的映射

标识(ID)位数字	钥匙	
X_0	K_{00}	K_{01}
X_1	K_{10}	K_{11}
X_2	K_{20}	K_{21}

钥匙分配对应的二进制树有特定的结构，在任何(从根开始的)给定的深度都使用两个新钥匙，在从根开始的深度 h，两个新钥匙 $K_{(\log_d N-h)0}$ 和 $K_{(\log_d N-h)1}$ 重复 h 次。例如在从根开始的深度 2，KEKs K_{10} 和 K_{11} 跨越树进行两次复制，在这个方案中 GC 存储的整个钥匙数是 $2\log_2 N$。对于一个 d 叉生成树，在这个方案中 GC 存储的整个钥匙数是 $d \log_d N$，每个成员只能保存 $\log_d N$ 个钥匙(不包括根钥匙和会话加密密钥)同时在成员删除时 GC 必需重新产生 $\log_d N$ 个钥匙。因此，考虑到单个成员删除时这个方案实际是一个优化解决方案，同时对比种子方案而言这个方案对于 GC 要求更小的存储空间。

尽管 GC 保存的钥匙总数为 $d \log_d N$，删除多个成员可能带来钥匙分配方案的终止。在图 3 的情况中，如果成员 M_0 和 M_7(或 M_1 和 M_6)需要删除时这种情况将发生。成员 M_0 的 KID 是 $K_{20}K_{10}K_{00}$ 同时成员 M_7 的 KID 是 $K_{21}K_{11}K_{01}$。形成这两个 KIDs 的钥匙合并包括了这个树上分配的所有钥

匙。因此，如果这两个成员需要同时删除，GC就没有钥匙可以安全地和其他的合法成员通信。

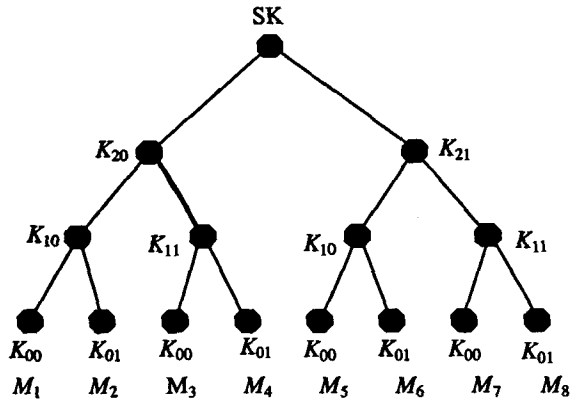

图3 钥匙分配[4,6]

离开成员删除，钥匙分配及变更也允许成员合作和协商，我们现在解释生成树上的用户串通。

5.1 对于最小钥匙需求数的说明

针对自由覆盖的充分条件要求所有 N 个叶子节点分配不同的钥匙。由于大小为 N 的组保存的钥匙总数是 $d\ \log_d N$，这个模型可以是自由覆盖的，如果：

$$d\ \log_d N \geqslant N \tag{16}$$

$$N \geqslant d^{\frac{N}{d}} \tag{17}$$

组的大小 N 应该是 d 的乘方，设置 $N=d^k$，这里 k 是不可决定的，导致了下面的等式：

$$kd=d^k \tag{18}$$

$$k=d^{k-1} \tag{19}$$

$k=1$ 满足这个等式，同时有 $N=d^k=d$，对于 $d\geqslant 2$ 表2总结了 k 的值针对不同的 d 满足 $k=d^{k-1}$。

表2 总结了对于 $d\geqslant 2$、k 的值针对不同的 d 满足 $k=d^{k-1}$

d=2	k=1 or k=2	N=2 or N=4
d>2	k=1	N=d

因此，已经说明如果在 $k=k_0$ 时，$K_0<d^{k_0-1}$，则对于 $\forall k \geqslant k_0$ 时，$k<d^{k-1}$。反过来暗示对于 $N=d^k\geqslant d^{\frac{N}{d}}$ 没有整数 $k>k_0$。由于这个整数满足部分次序，同时 $d\geqslant 2$ 是一个整数，如果对于 $d=d_0$ 同时 $k=k_0$ 时，$k_0<d_0^{k_0-1}$，则 $\forall d>d_0$ 时，

$$k_0<d_0^{k_0-1}<d^{k_0-1} \tag{20}$$

对于 $d=2$，$k=1,2$ 满足 $k=d^{k-1}$。当 $d=2$，如果 $k=3$，$k<d^{k-1}$，因此，大小为 N 的组不满足不等式(21)(下面的定理3)。

定理3 对于一个基于生成树的组播钥匙分配方案[9,10]，有一个大小为 N 的组同时成员删除可能性为 $\{p_i\}_{i=1}^{N}$，如果GC产生位的速度为 B，则单位时间可更新的钥匙长度 L：

$$L\leqslant \min\left\{\frac{B}{1+H_d}, \frac{B}{1-\log_d p_{\min}}\right\} \tag{21}$$

从不等式(17)中可以得出结论：对于一个二叉树来说，如果组的大小大于4，在这个模型下不可能有一个自由串通的钥匙分配。

如果设置 $d=3$ 同时 $k=2$，就有：$k=2<3^{2-1}=d^{k-1}=3$。

因此，对于 $k>1$，$d=3$，没有整数 $N>3$ 满足不等式(20)。根据事实如果 $k<3^{k-1}$ 则 $k<3^{k-1}<4^{k-1}<5^{k-1}\cdots$，得出结论对于 $d>2$，$N\geqslant d^{\frac{N}{d}}$ 中 N 唯一拥有的值是 $N=d$。

因此，我们已经说明如果组的大小 $N=d$ 时，钥匙分配将是自由串通的。

5.2 串通问题的另一种解释

串通问题的第二种解释是根据集合的概念，组的每个成员都有一个唯一的识别钥匙，这个唯一的钥匙分配给组中每个成员而不包括这个钥匙识别的成员，当一个成员删除时，就将成员的删除标识广播。在下一个会话中，所有合法的成员针对成员的删除设置钥匙作为新的会话加密密钥。在这个模型下，对于一个具有 N 个成员的集合，所有的成员将有($N-1$)个钥匙对应于其他的成员而没有成员拥有对应于自己的钥匙。如果考虑任何两个成员，它们所保存的钥匙的结合将覆盖整个组保存的钥匙，因此，这个钥匙分配必须考虑成员的删除。

如果使用符号 $(k_j,\bar{k}_j)$ 来表示唯一的钥匙对，它代表UID位 X_j 持有的两个可能的二进制值，分别持有钥匙 k_j 和 $\bar{k}_j$ 的两个成员的串通或协商将损害钥匙对 $(k_j,\bar{k}_j)$ 的完整性。在一个基于d叉生成树的钥匙分配中，每个位取得可能的值在(1,d)之间，这些值的总和给定为 $\frac{d(d-1)}{2}$。设一个成员 M_i 的第 b 个位地址的值表示为 b_i，如果：

$$b_1+b_2+\cdots+b_k\equiv 0 \bmod \frac{d(d-1)}{2} \tag{22}$$

则 k 个成员可以串通和协商对应于这个位地址 b 的所有钥匙。

结论和探讨 这篇文章说明了基于生成树的安全组播钥匙分配方案可以使用基本的信息理论概念进行系统的研究。通过运用成员删除事件作为公式的基础，说明给一个成员分配的最佳钥匙数和成员删除统计的平均信息量相关。得到钥匙分配自由串通的充分必要条件以及“自由覆盖”的充分必要条件，特别说明当钥匙间没有额外的成员关系时生成树上的自由覆盖条件要求叶子节点都是不同的。在这个条件下，组控制器的存储要求针对组的大小 N 是线性的。接着证明了基于生成树的策略，因此，这个钥匙分配方案对应于最大-最小钥匙分配策略。同时也得到一个平均钥匙长度、成员删除事件可能性和硬件位产生速度之间的关系。

参考文献

1 Maurer U M. Secret key agreement by public discussion from common information. Trans. Inform. Theory, May, 1993, 39 (1462):729～752

2 Ballardie A. Scalable multicast key distribution. report, RFC. 1949, May 1996

3 Balenson D, McGrew D A, Sherman A. Key establishment in large dynamic groups: One-way function trees and amortized initialization. IETF Draft, draft-balenson-groupkeyment-oft-00. txt, Feb. 1999

4 Caronni G, Waldvogel M, Sun D, Plattner B. Efficient security for large and dynamic groups. In: Proc. 7th Workshop Enabling Technologies. Cupertrino, CA: IEEE Comp. Soc. Press, 1998

5 Brumester M, Desmedt Y. A secure and efficient conference key distribution system. In: Advance in Cryptology-Eurocrypt ' 94 (Lecture Notes in Computer Science). Berlin, Germany: Springer-Verlag, 1994, 50: 269～284

6 Chang I, Engel R, Kandlur D, PendarakisD, Saha D. Key management for secure internet multicast using Boolean function minimization techniques. In: Proc. IEEE INFOCOM '99. 687～699

7 Cover T, Thomas J. Elements of Information Theory. New York: Wiley, 1991

8 Massey J L. An information-theoretic approach to algorithms. In: Impact of Processing Techniques in Communications, ser. NATO Advanced Study Institutes Ser. E91, 1985. 2～26

一种基于身份的群签名方案

董 亮 肖国镇

(西安电子科技大学 ISN 综合业务网国家重点实验室 西安 710071)

摘 要 基于身份的密码系统(Identity-based Cryptosystem)是为了简化基于证书的密码系统繁琐的密钥管理过程而提出的。群签名能对签名者提供很好的匿名性,它在电子商务、匿名电子选举等方面有重要应用。本文利用双线性对的性质构造了一种新的基于身份的群签名方案。对该方案的安全性及其效率的分析表明,方案是安全有效的。

关键词 基于身份,双线性映射,群签名

An ID-based Group Signature Scheme

DONG Liang XIAO Guo-Zhen

(Information Security and Privacy Institute in ISN, Xidian University, Xi'an 710071)

Abstract Identity-based (ID-based) cryptosystem can simplify key management procedures of certificate-based cryptosystem. Group signature is very useful to provide the signer's anonymity, so it is plays an important role in building e-commerce, anonymous electronic voting etc. In this paper, we propose a new ID-based group signature scheme based on the bilinear pairings. The analysis shows that our scheme has higher safety and better efficiency.

Keywords ID-based, Bilinear map, Group signature

1 引言

1984 年 Shamir[1] 为了简化电子邮件系统中的证书管理问题,提出了基于身份的密码系统(IBC)。系统中每个用户都有一个身份,其中身份可以是姓名、住址、电子邮件地址等,用户的公钥可以由任何人根据其身份计算出来,而私钥则由可信第三方生成,这里,可信第三方称为私钥生成器 PKG (Private Key Generator)。IBC 系统不需要保存每个用户的公钥证书,避免了使用证书带来的存储和管理开销的问题,简化了基于证书的密码系统繁琐的密钥管理过程。最近,利用椭圆曲线上的 Weil 对的双线性性质,一些基于身份的加密[2]、签名方案[3~7]相继被构造出来。

群签名能对签名者提供很好的匿名性,它可以由可信中心协助执行,中心掌握各签名人与所签名之间的相关信息,并为签名人匿名签字保密;在有争执时,可以由可信中心打开签名而识别出签名人,所以群签名在电子商务、匿名电子选举及网上投标等场合有重要应用。本文是在 SOK-IBS (Sakai-Ogishi-Kasahara Identity Based Signature)[6] 的基础上,利用椭圆曲线上的 Weil 对的双线性性质,构造了一种新的基于身份的群签名方案。

2 双线性映射和 GDH 群

在文[2~7]的加密和签名系统中使用了双线性映射和 GDH 群,下面简要介绍它们的定义及性质。

2.1 双线性映射

设 G_1 是一个由 P 产生的循环加法群,其阶为 q,G_2 是一个阶为 q 的循环乘法群,a,b 是 Z_q^* 的元素,我们假定 DLP 问题在 G_1 和 G_2 上是困难的。e 是由椭圆曲线上的 Weil 对产生的,双线性映射 $e:G_1\times G_2\rightarrow G_2$ 具有以下性质:

(1)双线性:对任意的 $P,Q,R\in G_1$,$e(P,Q+R)=e(P,Q)e(P,R)$,$e(P+Q,R)=e(P,R)e(Q,R)$。对于任意的 $a\in Z_q^*$,用 aP 表示 P 自加 a 次,则对任意的 $a,b\in Z_p^*$,有 $e(aP,bQ)=e(P,Q)^{ab}$。

(2)非退化性:存在 $P,Q\in G_1$,使得 $e(P,Q)$ 不等于 1。

(3)可计算性:存在一个高效的算法计算 $e(P,Q)$,其中 $P,Q\in G_1$。

2.2 GDH 群

设 G 是一个由 P 生成的阶为素数 q 的加法循环群,假定在 G 上乘法和逆在单位时间内可以计算出来,$a,b,c\in Z_q^*$。则:

(1)CDHP(计算上的 Diffie-Hellman 问题):已知(P,aP,bP),计算 abP。

(2)DDHP(决定性的 Diffie-Hellman 问题):已知(P,aP,bP,cP),判断 $c=ab \bmod q$ 是否成立。

(3)CDH 假设:不存在有效的多项式时间算法可以解决 CDHP。

(4)GDH 群:在素数阶循环群 G 上,DDHP 在多项式时间内能被解决,但没有算法可以解决 CDHP,即 CDH 假设成

9 Wallner D M, Harder E J, Agee R C. Key management for multicast: Issues and architectures. Internet draft, Sept. 1998
10 Wong C K, Gouda M, Lam S S. Secure group communications using key graphs. IEEE/ACM Trans. Networking, Feb. 2000, 8: 15~36. Also In: Proc. ACKM SIGCOMM '98, Vancouver, BC, Canada, Sept. 1998
11 Gallager R. Information Theory and Reliable Communication. New York: Wiley, 1968
12 Harney H, Muckenhirn C. GKMP architecture. Request for Comments(RFC), 1997, 2093
13 戴宗坤,罗万伯. 密钥管理. 信息系统安全. 电子工业出版社, 2002
14 冯登国. 计算机通信网络安全. 清华大学出版社, 2001
15 Stallings W 著. 杨明,等译. 密码编码学与网络安全. 电子工业出版社, 2001