安全组播中密钥分配问题的研究*

朱文涛+, 熊继平, 李津生, 洪佩琳

(中国科学技术大学 电子工程与信息科学系,安徽 合肥 230027)

A Study of the Key Distribution in Secure Multicast

ZHU Wen-Tao+, XIONG Ji-Ping, LI Jin-Sheng, HONG Pei-Lin

(Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China)

+ Corresponding author: Phn: 86-551-3603634, Fax: 86-551-3601334, E-mail: wtzhu@263.net

http://eeis.ustc.edu.cn/

Received 2003-02-27; Accepted 2003-09-15

Zhu WT, Xiong JP, Li JS, Hong PL. A study of the key distribution in secure multicast. *Journal of Software*, 2003,14(12):2052~2059.

http://www.jos.org.cn/1000-9825/14/2052.htm

Abstract: Multicast is a preferred network communication technique in the case of multiple recipients, whose importance has been increasingly highlighted with the development of the Internet. IGMP, the multicast management protocol, does not provide access control of the users. In order to protect communication confidentiality, traffic in secure multicast is encrypted with a Session Encryption Key (SEK) which is known only to the certificated group members. Whenever there is a change in the group membership, the SEK has to be dynamically updated, thus the key distribution becomes a key problem in the research of secure multicast. In designing key distribution algorithms, communication costs, key storage, protection against attacks and computation complexity are considered as the four important factors. A group key distribution scheme utilizing a polynomial expansion is proposed, which features in no traditional encryption and decryption. Analyses show that it performs well in small scale multicast. This polynomial expansion based algorithm is then integrated with the Logical Key Hierarchy; while preserving the logarithmic communication cost with the group size, the presented PE-LKH scheme lowers the computation complexity observably, thus is scalable to large dynamic groups.

Key words: secure multicast; communication confidentiality; key distribution; polynomial expansion; logical key hierarchy

摘 要: 组播是面向组接收者的首选网络通信技术,其重要性随着 Internet 的发展日益突出.组管理协议 IGMP 不提供成员接入控制.为了保护通信机密性,安全组播使用仅为认证组成员所知的会话加密密钥(SEK)来加密业务数据.每当组成员关系发生变化时,都应动态更新 SEK,密钥分配也就成为安全组播研究的关键问题.在设计密

* Supported by the National Natural Science Foundation of China under Grant No.60272043 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2002AA121067 (国家高技术研究发展计划(863))

第一作者简介: 朱文涛(1979－),男,江西临川人,博士生,主要研究领域为通信协议,网络安全.

钥分配算法时,通信开销、存储开销、抗冲击性和计算开销被认为是 4 个重要因素.提出了一种利用多项式展开的组密钥分配方案,其特点是不使用传统加密和解密.分析表明,其在小型组播中可获得较好的性能.将基于多项式展开的该算法与逻辑密钥层次结合,又提出了一种 PE-LKH 方案,在保留通信开销随组规模呈对数增长的同时,其计算复杂度有效降低,可适用于大规模动态群组.

关键词: 安全组播;通信机密性;密钥分配;多项式展开;逻辑密钥层次

中图法分类号: TP309 **文献标识码**: A

1 组播及组的管理

组播是基于 UDP/IP 协议、面向多接收者的数据分发方式[1,2].图 1 是组播源 S 向成员 M_1~M_5 组播数据的示意图,S 传至路由器 R_1 的每一份 UDP 报文都被复制为 3 份,一份给 M_1,一份送至 M_2 和 M_3 所在的网段(广播链路),一份给路由器 R_2,R_2 再将报文生成两份拷贝分发给 M_4 和 M_5.组播数据仅在执行组播路由协议的路由器处进行最小次数拷贝(组播树的"分叉"),与单播相比,能有效地节省服务器资源和网络带宽.随着 Internet 的宽带化、多媒体化和商业化,组播应用也越来越广泛,如远程会议、联网游戏、电视直播以及一些军用场合等.

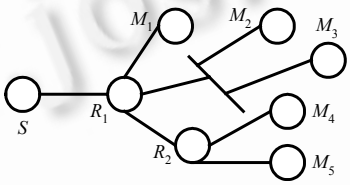

Fig.1 A source multicasting data to a group with five members
图 1 组播源向 5 个成员组成的群组发送数据

因特网组管理协议 IGMP 用于管理组播.当前的 IGMPv2[3]规定了主机向路由器注册组播组的操作规程,例如图 1 中 M_1~M_3 向组播路由器 R_1 注册,M_4 和 M_5 向 R_2 注册.IGMPv2 报文由 TTL 值为 1 的 IP 包封装,并像 ICMP 一样被认为是 IP 协议的一部分.下面给出 IGMPv2 报文的格式,长度为 8 字节.

Type	Max Resp Time	Checksum
Group Address		

其各域的含义说明如下:

• "类型"表示 IGMP 报文种类,有成员报告、退出通告和成员查询 3 类,其中成员报告和退出通告由主机发送,成员查询由路由器发送;成员查询又可细分为普通查询(路由器仅欲查询子网内哪个群组还存在成员)和特定查询(路由器查询某一特定群组);

• "最大响应时间"只用于查询报文,是成员应答查询的最大允许延时,单位为 0.1s;

• "校验和"对作为 IP 协议净荷的整个 IGMP 报文提供保护;

• "组播地址"表明路由器特定查询及主机加入或退出的是哪个组,普通查询时该值置 0.

IGMP 第 3 版[4]的报文格式要稍微复杂一些.它提供对 IGMPv2 的兼容.其主要改进是增加了源地址绑定,除了允许主机指定接收来自特定地址的组播数据以外,还允许其拒收来自某地址的数据.IGMPv3 才标准化不久,要得到设备生产商的支持还需一段时日.

2 安全组播和密钥分配问题

如前所述,组管理协议并不提供成员接入控制,用户只要获知特定业务使用的组播地址就可以向路由器发送 IGMP 成员报告,不经审核地加入群组并获得 UDP 数据的拷贝.保护组播数据机密、建立安全通信系统是安全组播研究的主要目标.与端到端的单播情形相比,组播通信的安全问题更为复杂.将现有单播安全技术直接移植到组播应用上往往不可行或是低效的.安全组播在协议设计、策略控制及算法应用等各个方面都存在大量需要研究的问题[5,6].

支持安全组播的基本方法是所有成员共享一个不为未被授权用户所知的业务加密密钥 SEK(session encryption key)或称为 TEK(traffic encryption key).SEK 是对称密钥,组内所有通信都使用该密钥进行加密和解密.每当有用户加入或离开群组时,必须更新 SEK,以使新加入成员无法访问过去的历史数据(后向安全性*),且离开的成员无法解读当前及将来的通信(前向安全性).这一过程称为 rekey,其目的是为了在成员关系变动、密钥过期或被泄露时维持组通信的机密性.SEK 的分发和更新是安全组播研究的核心问题,必须采用合理的密钥分配算法,以减少系统付出的开销.

密钥分配的研究通常基于集中式管理和分布式管理两类基本模型.后者的特点是,所有群组成员通过密钥协商(key agreement)来共同建立 SEK,其缺点是计算复杂导致延时很大,故难以适用于成员关系频繁变化的大规模动态群组,因而不在本文的讨论范围之内.

集中式密钥管理模型的特点是存在一个专门的组控制者 GC(group controller),负责生成通信密钥 SEK,并通过特定算法分发给规模为 N 的群组中的每个成员,是一种被广为采用的方法.这样,组播模型就可以用图 2 来表示,其中左图是组播基本模型的示意,右图是引入数据加密之后的安全组播的示意.应该指出的是,从网络实际构成来看,路由器位于组播源和组成员之间,负责拷贝和转发 UDP 数据(如图 1 所示),但从功能上可以认为组播源是直接将数据(明文或用 SEK 加密后的密文)分发给组成员.在电视直播等场合,组播源只有一个且自身不必作为成员加入组播,而在另外一些场合,如在联网游戏中,所有用户在功能上都既是组播源又是数据接收者.在如图 2 所示的安全组播功能模型中,GC 同时向组播源和组成员播送 SEK.

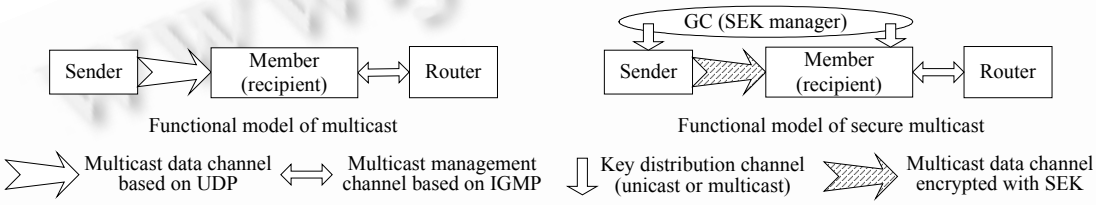

Fig.2 Basic functional model of multicast and secure multicast
图 2 组播与安全组播的基本功能模型

一般而言,由新成员的加入所引起的 rekey 操作相对较容易,难点是成员删除时触发的密钥更新**.例如,当新成员加入时,GC 用自己和该成员共享的密钥加密 SEK 后单播给它,并用原有成员皆知的旧 SEK 加密新 SEK 后向其余成员组播***.

我们认为,应该从通信开销、存储开销、抗冲击性和计算开销这 4 点来对密钥分配进行综合评价.第 3 节将结合几种具体的密钥分配算法来对以上观点加以论述.第 4 节将简要介绍具有树型结构的集中式密钥管理.第 5 节引入一种基于多项式展开(PE)的密钥管理方案,其特点是密钥分配算法中不使用通常的加解密(如 DES).第 6 节将 PE 算法与树型密钥管理相结合,提出适用于大规模动态群组的 PE-LKH 方案.第 7 节对全文进行总结.

3 密钥分配的 4 个因素

简单密钥分发中心 SKDC(simple key distribution center)是最为直观的密钥分配算法.在 SKDC 中,GC 使用与第 i 个成员 M_i 共享的密钥加密密钥 KEK_i(key encryption key)来逐个加密 SEK,并单播给每个成员****,记为

* 在一些组播应用中可以不考虑后向安全性,而只需当成员退出时才更新 SEK,见文献[7]中的有关论述.

** 因此,下文中如未作特殊说明,rekey 均指成员退出事件所触发的密钥更新.

*** 这涉及到组播方式传递密钥的可靠性问题,为简明起见,安全组播(secure multicast)的研究中通常不考虑可靠组播,但也有个别文献,如文献[8],将安全组播和可靠组播一并考虑.IETF 为研究可靠组播而另外成立的 Reliable Multicast Transport 工作组见 http://www.ietf.org/html.charters/rmt-charter.html.

**** KEK 本身则经过专门的身份认证后登记,例如离线注册,或者通过公钥体制在线传送.

$KEK_i(SEK)$*.GC 存储所有 N 个 KEK 并负责产生 SEK,M_i 则存储仅为自己与 GC 所知的 KEK_i,并用其解密出 SEK.这样,添加一个成员需要 GC 执行 N+1 次加密传送,排除一个成员需要执行 N–1 次加密传送.很容易对 SKDC 作算法渐进分析,GC 的存储复杂度和通信复杂度都是 $O(N)$,故只适合小规模的组通信.可以看出,rekey 通信复杂度和密钥存储复杂度是密钥管理的两个重要方面[9],其中以降低 rekey 通信次数尤为重要,一方面是为了节省网络带宽,另一方面是为了减少密钥更新时的延时.通信复杂度是当前组播应用的最大瓶颈[7],降低通信次数是当前密钥分配研究所致力的目标.

已有的研究指出,密钥分配往往是在通信和存储这对开销中取得一个折衷[7,10].考虑一种极端的密钥分配算法,设 M_0~M_{N-1} 为所有可能的 N 个潜在用户,GC 生成 2^N 个密钥,编号为 $SEK_i, i\in[0,2^N-1]$,依次对应于 2^N 种可能的成员组合 G_i.新成员 $M_j(j\in[0,N-1])$加入组播时从 GC 处获得并存储 $SEK_i, i\in\{x|x\equiv y(\mathrm{mod}\ 2^{j+1}), x\in[0,2^N-1], y\in[2^j,2^{j+1}-1]\}$,共 2^{N-1} 个,依次对应于其余 N–1 个潜在用户加入或不加入某一特定群组的 2^{N-1} 种情况;群组启用对应于当前成员组合 G_g 的 SEK_g.当需要从 G_g 排除用户 M_k 时,GC 明文组播指令"删除 M_k",其余用户就自动地从自己的密钥集中选取 SEK_{g-K} 作为新的 SEK,$K=2^k$.算法的存储复杂度是 $O(2^N)$,通信复杂度是 $O(1)$.它以庞大的指数级存储开销达到了理论上最少的通信次数.

本节讨论的第 3 种方案是补足变量法(complementary variable)[11],GC 除了初始化 SEK 以外,还负责生成对应于各成员 M_i 的补足变量 V_i(共 N 个),并将 V_i 分发给除了 M_i 以外的其他成员,从而使得 M_j 拥有的补足变量集合是$\{V_i|i\in[1,N], i\neq j\}$.我们以删除成员 M_{20} 为例来说明 GC 和各组成员所进行的操作,GC 组播一则消息"删除 M_{20}",新的 SEK 由除了 M_{20} 以外的所有成员利用旧 SEK 和 V_{20} 各自计算出 $SEK^{new}=f(SEK,V_{20})$.由于 V_{20} 被 M_{20} 以外的所有成员持有,所以 M_{20} 自身不能得出 SEK^{new},从而被排除出组.此方案的优点是 GC 不使用额外的密钥即 KEK,GC 也无须存储补足变量 V_i,并且 rekey 通信复杂度降低为 $O(1)$:GC 只需组播一条简单的控制指令("删除 M_{20}")而且不需要加密.然而,该方案有两个主要的缺点:① 所有成员都必须存储 N–1 个额外的 V_i;② 当删除多个成员时,例如 M_1 和 M_2,它们可以相互勾结(collusion),各自向对方提供对方不知道的补足变量(M_1 向 M_2 提供 V_2,M_2 向 M_1 提供 V_1),从而都能计算出新的 SEK,也就摆脱了 GC 的控制.由此可见,密钥分配算法还必须考虑其抗冲击性,即新 SEK 不能由非法成员或其联盟所持的历史信息中推算出.

最后,文献[12]给出了一种基于中国剩余定理的安全锁(secure lock)方案,可以将 SEK 安全广播到每个成员.该算法的 rekey 报文个数不随组规模 N 增长,但安全锁的生成需要巨大的计算开销,且安全锁的自身长度及计算量与 N 成比例,故该方案只适用于小的群组.这也说明,计算开销是设计组播密钥分配算法中不可忽视的第 4 个因素,尤其是在群组成员计算能力有限的场合.在后续章节中,我们将兼顾以上 4 个因素对密钥分配作综合论述.

目前已经形成协议标准的密钥管理包括 SMKD[13]和 GKMP[14].以上密钥分配算法的共同特点是只适合于小型组播,随着组规模 N 的增大,算法性能都显著下降.例如,SKDC 中通信次数随着 N 呈线性增长,是一个明显的瓶颈.为了支持大规模安全组播,密钥管理方案应具有可扩展性,需要引入树型密钥管理来适应组成员关系的高度变化并处理大型群组的频繁密钥更新[11].

4 树型密钥管理方案

文献[11,15]各自提出了倒置树(rooted-tree)式的逻辑密钥层次 LKH(logical key hierarchy).它们以及由此派生出来的一系列方案统称为树型(tree-based)密钥管理[10].

如图 3 所示为一棵拥有 8 个叶节点的二叉密钥树,对应于一个成员个数为 8 的群组:树根代表 GC,它所拥有的 K_0 即为业务加密密钥 SEK**;叶节点代表组成员,它们分别拥有的 $K_{3.1}$~$K_{3.8}$ 为 M_i 与 GC 共享的 $KEK_i, i\in[1,8]$;虚拟的内部节点对应于中介 KEK,用于 SEK 更新时向组内特定范围的组成员传递新密钥(新密钥既包括 SEK,也包括那些需要更换的中介 KEK).每个成员拥有从代表它的叶节点到根节点这条路径上的所有密钥,例如成

* $K(X)$表示用 K 来加密 X,例如 KEK_i(SEK)表示用第 i 个成员的 KEK_i 加密 SEK.X 可以是多则待加密的消息.

** K_0 为所有成员共享,故可以取 K_0 作为 SEK.有些文献,如文献[9],认为 SEK 应再额外设置,并用 K_0 加密 SEK.

员 M_1 拥有的密钥是$\{K_{3.1},K_{2.1},K_{1.1},K_0\}$,其中 K_0 是为所有成员共有的 SEK,$K_{3.1}$ 是仅为 GC 和 M_1 所知的 KEK_1,而 $K_{1.1}$ 和 $K_{2.1}$ 则从 GC 处获得,并用来限制密文消息只组播到包含 M_1 在内的有限集合(例如,M_5~M_8 不能解读用 $K_{1.1}$ 加密的消息;M_3 和 M_4 能解读用 $K_{1.1}$ 加密的消息,但不能解读用 $K_{2.1}$ 加密的消息).

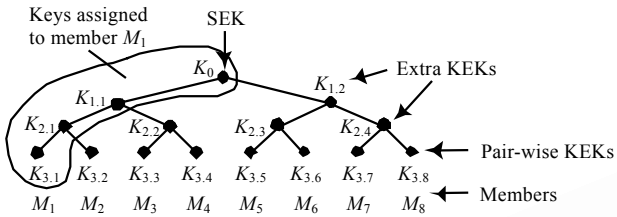

Fig.3 A binary logical key tree with 8 leaf nodes

图 3 一棵度为 2、叶节点个数为 8 的逻辑密钥树

使用 LKH 能有效地降低组成员关系变动时的 rekey 通信次数.例如,为了删除成员 M_7,GC 需向 M_1~M_6 及 M_8 发送 SEK^{new},并更换 M_7 所拥有的中介 KEK($K_{2.4}$ 和 $K_{1.2}$,它们被认为是已泄密的),为此需要作 3 次 rekey 通信:① 向 M_8 单播 $K_{3.8}(K_{2.4}^{new},K_{1.2}^{new},K_0^{new})$;② 向 M_5 和 M_6 组播 $K_{2.3}(K_{1.2}^{new},K_0^{new})$;③ 向 M_1~M_4 组播 $K_{1.1}(K_0^{new})$.不难分析,对一棵度为 d、深度为 $\log_d N$ 的密钥树*,更新 SEK(即 K_0)所需的通信次数是

$$C_{LKH}=(d-1)\log_d N. \tag{1}$$

第 3 节已指出,降低 C 是密钥分配研究的主要目标.对于给定的 N,显然,当 d=2 时,式(1)有最小值 $\log_2 N$,故通常取二叉树 LKH 的情况进行研究.图 3 中 d=2,N=8,故 C=(2−1)×3=3.

下面讨论存储开销.GC 需要存储所有的 KEK(包括 K_0、各中介 KEK 以及各 KEK_i),总量是

$$S_{LKH}=\sum_{i=0}^{\log_d N} d^i=\frac{dN-1}{d-1}>N. \tag{2}$$

可以认为 LKH 是对 SKDC 的层次化扩展,因为当 d=N 时,式(1)变为 C=N−1,式(2)变为 S=N+1,此时,LKH 就退化为 SKDC 方案.与 SKDC 相比,LKH 通过牺牲密钥存储开销(GC 和各成员都必须存储额外的中介 KEK),换来了 $O(\log N)$的通信开销,且该算法具有抗冲击性和可以接受的计算开销.

5 基于多项式展开的 PE 算法

本文提出一种基于多项式展开(polynomial expansion,简称 PE)的密钥分配算法,其特点是不使用通常的加解密算法(如 DES 等),该思想部分地来自文献[16].文献[16]的密钥管理可简述如下:SEK 和所有 KEK_i 均为 B 比特长,不失一般性,考虑删除 M_N 的情况,为此,GC 向剩余成员 M_1~M_{N-1} 组播的 rekey 报文包含随机种子μ和大数值α,新 SEK 包含在α中:

$$\alpha=SEK+\prod_{i=1}^{N-1}\left[2^B+h(KEK_i,\mu)\right], \tag{3}$$

其中,h 是以μ为参量的单向函数,μ和 h 运算结果亦要求为 B 比特长.$SEK<2^{B+1}$,故对 M_i 有

$$\alpha=SEK\left(\mathrm{mod}\left[2^B+h(KEK_i,\mu)\right]\right). \tag{4}$$

本文取文献作者名缩写,称该方案为 TSPL 算法.其缺点在于,α是 N−1 个数求积后与 SEK 的和,其值会非常大,以至于无论 GC 按式(3)计算α,还是各成员按式(4)计算 SEK,其开销都是非常可观的.此外,α值过大还可能导致 rekey 报文(包含μ和α)过长,从而还必须拆分成多个协议数据单元进行网络传送.

本文采用类似的思想,令 GC 为所有 N−1 个剩余成员 M_1~M_{N-1} 构造多项式:

$$P(x)=SEK+\prod_{i=1}^{N-1}\left[x-h(KEK_i,\mu)\right]. \tag{5}$$

将多项式展开为式(6),并显然有式(7)成立:

* 研究中总是假定树为全满即 N 为 d 的整数次幂;若不为全满,为了提高效率应为平衡树[8],且树深为$\lceil\log_d N\rceil$.

$$P(x)=x^{N-1}+a_{N-2}x^{N-2}+a_{N-3}x^{N-3}+...+a_1x+a_0, \tag{6}$$

$$P\left(h\left(KEK_i,\mu\right)\right)=SEK,\ i\in[1,N-1]. \tag{7}$$

GC 只要按式(6)组播 rekey 报文$\{\mu,a_{N-2},a_{N-3},...,a_1,a_0\}$,剩余成员 M_1~M_{N-1} 就可以各自按式(7)求出新 SEK.显然,PE 算法不涉及通常密钥分配算法中使用的加密和解密过程.

密钥分配算法的抗冲击性要求能抵抗外部和内部两方面的攻击,必须考虑为来自内部的攻击提供可靠的安全屏障[17].由式(5)和式(7)有

$$P(x)-P\left(h\left(KEK_i,\mu\right)\right)=\prod_{i=1}^{N-1}\left[x-h\left(KEK_i,\mu\right)\right],\ i\in[1,N-1]. \tag{8}$$

令式(8)等于 0,这样,M_1~M_{N-1} 之一就存在利用$\{\mu,a_{N-2},a_{N-3},...,a_1,a_0\}$推算出其余 $h(KEK_i,\mu)$的可能.此即引入单向函数 h 的原因,以便保证无法进而推算出其余成员的 KEK_i.进一步地,为了防止被删除成员利用事先获得的他人的 $h(KEK_i,\mu)$按式(7)计算后续的 SEK,GC 每次发布 rekey 报文$\{\mu,a_{N-2},a_{N-3},...,a_1,a_0\}$时必须更换随机种子$\mu$,从而确保算法的抗冲击性.

容易看出,多项式系数$\{a_{N-2},a_{N-3},...,a_1,a_0\}$的值都很大,例如

$$a_{N-2}=-\sum_{i=1}^{N-1}h\left(KEK_i,\mu\right),\ a_0=SEK+\prod_{i=1}^{N-1}h\left(KEK_i,\mu\right). \tag{9}$$

为此,令 Pr 为小于 2^B 的最大质数,式(5)~式(9)中的所有运算都调整为结果需对 Pr 取模*.例如在式(9)中,根据$(x+y)(\mathrm{mod}\ Pr)\equiv((x\ \mathrm{mod}\ Pr)+(y\ \mathrm{mod}\ Pr))(\mathrm{mod}\ Pr)$,$a_{N-2}$ 的求和过程中最大中间和至多是 $2(Pr-1)$,而根据$(xy)(\mathrm{mod}\ Pr)\equiv((x\ \mathrm{mod}\ Pr)(y\ \mathrm{mod}\ Pr))(\mathrm{mod}\ Pr)$,$a_0$ 的求积过程中最大中间积至多是$(Pr-1)^2$.相比之下,式(3)中虽然也不出现加密和解密操作,但求积的结果不小于 $2^{(B+1)(N-1)}$,显然有$\alpha>>(Pr-1)^2$.

这样,系数$\{a_{N-2},a_{N-3},...,a_1,a_0\}$的计算以及式(7)的求值就大为简化,rekey 报文的长度也调整为 $B\times N$ 比特(与式(3)基本持平).通常取 B=128 比特,故 rekey 报文长 16N 字节.假定网络由以太网互联而成(如宽带城域网),最大 IP 包长为 1 500 字节,扣除 20 字节 IP 头标和 8 字节 UDP 头标,最大净荷长度是 1 472 字节,大于 16×90 字节,故 rekey 通信次数可取为

$$C_{\mathrm{PE}}=\left\lceil\frac{16N}{16\times90}\right\rceil=\left\lceil\frac{N}{90}\right\rceil, \tag{10}$$

简记为 $C=N/90$.将 TSPL 与 PE 作对比,结果见表 1.两者的抗冲击性均由单向函数的不可逆来保证.

Table 1 Comparison between two flat schemes TSPL and PE

表 1 两种平坦型方案 TSPL 与 PE 的对比

	Communication cost	Storage cost	Computation cost
TSPL	Around BN bits	GC: $O(N)$, member: $O(1)$	max *medi-sum*>$2^{B(N-1)}$, max *medi-product*>$2^{B(N-1)}$
PE	Around BN bits	GC: $O(N)$, member: $O(1)$	max *medi-sum*<2^{B+1}, max *medi-product*<2^{2B}

分别计算 SKDC,LKH(d=2)和 PE 这 3 种密钥分配算法的 rekey 通信量,结果见表 2.

Table 2 Comparison of SKDC, LKH and PE on rekey communication cost

表 2 SKDC,LKH 与 PE 在更新密钥所需通信开销上的比较

N	100	200	300	400	500	600	700	800	900	1 000
SKDC: $C=N-1$	99	199	299	399	499	599	699	799	899	999
LKH: $C=\log_2N$	7	8	9	9	9	10	10	10	10	10
PE: $C=N/90$	2	3	4	5	6	7	8	9	10	12

由表 2 可见,PE 算法与 SKDC 相比有显著的改善,当 $N\le900$ 时,PE 的通信次数亦不多于二叉树 LKH.PE 算法不涉及加密和解密操作,故不存在中介 KEK,因而存储开销小于 LKH.它与 SKDC 一样达到了最低存储量.3 种方案都具有抗冲击性.最后,从计算方式来看,SKDC 使用 KEK 加密 SEK,LKH 使用 KEK 加密 SEK 和 KEK,PE 算法则以模 Pr 的多项式运算及单向函数计算代替了加解密过程.由此可见,在小规模组播中,PE 算法在各方面都获得了较为理想的性能.

* 这样,SEK 也需要限制在 0~Pr–1 之间.在后面的 PE-LKH 算法中则无此限制.

6 面向大型群组的 PE-LKH 算法

由式(10)可知,PE 算法的通信复杂度仍为 $O(N)$,故与所有非树型(non-tree-based)方案一样,不适合于大型群组.这一点也已经反映在表 2 中.解决此问题的办法是将 PE 算法与 LKH 结合,本文称为 PE-LKH 算法.仍以图 3 为例,GC 为删除成员 M_7 所构造的多项式依次如下,其含义则注明在公式的左侧:

M_8 获得 $K_{2.4}^{\text{new}}$ $$P_{2.4}(x)=K_{2.4}^{\text{new}}+\left[x-h\left(KEK_{3.8},\mu\right)\right],\tag{11}$$

M_8 获得 $K_{1.2}^{\text{new}}$ $$P_{1.2-2.4}(x)=K_{1.2}^{\text{new}}+\left[x-h\left(KEK_{2.4}^{\text{new}},\mu\right)\right],\tag{12}$$

M_5 和 M_6 获得 $K_{1.2}^{\text{new}}$ $$P_{1.2-2.3}(x)=K_{1.2}^{\text{new}}+\left[x-h\left(KEK_{2.3},\mu\right)\right],\tag{13}$$

M_5,M_6 和 M_8 获得 K_0^{new} $$P_{0-1.2}(x)=K_0^{\text{new}}+\left[x-h\left(KEK_{1.2}^{\text{new}},\mu\right)\right],\tag{14}$$

M_1~M_4 获得 K_0^{new} $$P_{0-1.1}(x)=K_0^{\text{new}}+\left[x-h\left(KEK_{1.1}^{\text{new}},\mu\right)\right].\tag{15}$$

可见,PE 算法与 LKH 结合以后,多项式全部降级为一次式.进一步地,模 Pr 的加减法也可以全部使用异或代替,从而式(11)~式(15)简化为

M_8 获得 $K_{2.4}^{\text{new}}$ $$P_{2.4}(x)=K_{2.4}^{\text{new}}\oplus\left[x\oplus h\left(KEK_{3.8},\mu\right)\right],\tag{11′}$$

M_8 获得 $K_{1.2}^{\text{new}}$ $$P_{1.2-2.4}(x)=K_{1.2}^{\text{new}}\oplus\left[x\oplus h\left(KEK_{2.4}^{\text{new}},\mu\right)\right],\tag{12′}$$

M_5 和 M_6 获得 $K_{1.2}^{\text{new}}$ $$P_{1.2-2.3}(x)=K_{1.2}^{\text{new}}\oplus\left[x\oplus h\left(KEK_{2.3},\mu\right)\right],\tag{13′}$$

M_5,M_6 和 M_8 获得 K_0^{new} $$P_{0-1.2}(x)=K_0^{\text{new}}\oplus\left[x\oplus h\left(KEK_{1.2}^{\text{new}},\mu\right)\right],\tag{14′}$$

M_1~M_4 获得 K_0^{new} $$P_{0-1.1}(x)=K_0^{\text{new}}\oplus\left[x\oplus h\left(KEK_{1.1}^{\text{new}},\mu\right)\right],\tag{15′}$$

这样,在 PE-LKH 算法中,GC 只构造一次式,各成员只需存储 $\log_d N$ 个 KEK,并相应地计算 $h(KEK,\mu)$,这就极大地降低了计算复杂度.不难分析,rekey 通信复杂度像所有树型密钥分配算法一样,为 $O(\log N)$,适合于大型群组:

$$C_{\text{PE-LKH}}=d\log_d N-1.\tag{16}$$

在图 3 中,d=2,N=8,故 C=2×3−1=5,对应于式(11)~式(15).式(16)与式(1)相比,$C_{\text{PE-LKH}}$ 略大于 C_{LKH}.LKH 的中介 KEK 在 PE-LKH 中变成了 GC 和各成员进行多项式展开所使用的中间变量,其中 GC 存储量与式(2)结果相等,为

$$S_{\text{PE-LKH}}=\frac{dN-1}{d-1}.\tag{17}$$

PE-LKH 算法的抗冲击性由 PE 和 LKH 各自具有的抗冲击性来保证.最后,正如式(11′)~式(15′)所示,PE-LKH 算法不使用加密和解密,只涉及单向函数及异或运算,故计算开销显著下降.

7 结 语

安全组播正在逐渐成为一个活跃的研究领域.针对组播密钥管理这一核心问题,国际学术界提出了各种解决方案.本文较为系统地概述了已有的研究成果,指出密钥分配应综合考虑通信开销、存储开销、抗冲击性和计算开销这 4 个方面.在考查各种具体算法时,我们也始终以这 4 个方面为基本出发点.

本文提出了一种基于多项式展开、不使用加密和解密的密钥分配算法 PE,并论证得出它是一种面向小规模组播(N≤900)的高性能密钥管理方案.将该算法与树型密钥管理结合,我们又给出了一种适合于大规模动态群组的 PE-LKH 方案,其特点是通信复杂度为 $O(\log N)$且计算开销很小.随着 Internet 的发展,组播将具有广泛的应用前景,本文所做的工作具有实用意义,并对安全组播的发展具有研究价值.

References:

[1] Deering S. Host extensions for IP multicasting. IETF RFC1112, 1989.

[2] Quinn B, Almeroth K. IP multicast applications: Challenges and solutions. IETF RFC 3170, 2001.

[3] Fenner W. Internet group management protocol, version 2. IETF RFC2236, 1997.

[4] Cain B, Deering S, Kouvelas I, Fenner B, Thyagarajan A. Internet group management protocol, version 3. IETF RFC3376, 2002.

[5] Krusus PS, Macker JP. Techniques and issues in multicast security. In: Proceedings of the Military Communications Conference. Boston, 1998. 1028~1032.
[6] Canetti R, Pinkas B. A taxonomy of multicast security issues. Internet Draft, 2000.
[7] Snoeyink J, Suri S, Varghese G. A lower bound for multicast key distribution. In: Proceedings of the IEEE INFOCOM 2001. Anchorage, 2001. 422~431.
[8] Tanaka S, Sato F. A key distribution and rekeying framework with totally ordered multicast protocols. In: Proceedings of the 15th International Conference on Information Networking. Beppu City, 2001. 831~838.
[9] Li M, Poovendran R, Berenstein C. Design of secure multicast key management schemes with communication budget constraint. IEEE Communications Letters, 2000,6(3):108~110.
[10] Poovendran R, Baras JS. An information-theoretic approach for design and analysis of rooted-tree-based multicast key management schemes. IEEE Transactions on Information Theory, 2001,47(7):2824~2834.
[11] Wallner D, Harder E, Agee R. Key management for multicast: Issues and architectures. IETF RFC2627, 1999.
[12] Chiou G, Chen W. Secure broadcasting using the secure lock. IEEE Transactions on Software Engineering, 1989,15(8):929~934.
[13] Ballardie A. Scalable multicast key distribution. IETF RFC1949, 1996.
[14] Harney H, Muckenhirn, C. Group key management protocol (GKMP) architecture. IETF RFC2094, 1997.
[15] Wong CK, Gouda M, Lam SS. Secure group communications using key graphs. IEEE/ACM Transactions on Networking, 2000,8(1): 16~30.
[16] Trappe W, Song J, Poovendran R, Liu KJR. Key distribution for secure multimedia multicasts via data embedding. In: Proceedings of the Acoustics, Speech, and Signal Processing. Salt Lake City, 2001. 1449~1452.
[17] Ghanem SM, Abdel-Wahab H. A simple XOR-based technique for distributing group key in secure multicasting. In: 5th IEEE Symposium on Computers and Communications. Antibes-Juan les Pins: IEEE Computer Society, 2000. 166~171.