鲁棒且容错的大型动态组播系统的密钥管理*)

刘 璟 周明天

(电子科技大学计算机科学与工程学院 成都610054)

Robust And Fault-Tolerant Key Management in Large Dynamic Multicast Group

LIU Jing ZHOU Ming-Tian

(College of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054)

Abstract Current research activities on the problem of key management in group communication system include tow areas: one to solve the problem of key updating and distribution in large dynamic multicast groups; another to solve the problem of key agreement in the DPGs(dynamic peer groups). The two research areas are independent with each other. In this paper, we show a new research direction, under which we can effectively combine the corresponding techniques in these two fields to solve new problems, and also give a good demonstration that through combining the GDH techniques of key agreement in DPGs and our techniques of key management in large dynamic multicast groups, we present a robust and fault-tolerant key management protocol suite in large dynamic multicast group. The protocol suite solve the problem of SPOF(single point of failure)which universally exists in center-oriented key management protocols and is robust and fault-tolerant.

Keywords Group communication, Large dynamic multicast groups, Peer dynamic groups, Key management, Robust, Fault-tolerance

1 引言

随着 Internet 的迅猛发展,越来越多的分布式应用开始依赖可靠的组通信模式。例如:视频会议、视频点播、分布交互模拟、网络游戏、以及协同工作等等。同时组通信的安全问题也日益成为人们关注的焦点。其中组通信的密钥管理问题是其它组通信安全问题的基础,因而得到学术界的广泛关注,涌现了大量的研究结果。目前,针对组通信系统的密钥管理研究分为两个领域:一个解决大型动态组播系统的密钥更新和发布[1~6];一个解决所谓的动态对等群组(Dynamic Peer Groups,DPGs)的密钥协商[7,8]。两个领域的研究是相对独立的。

两个研究领域试图解决的问题实际上是不同应用环境下的同一个问题:为了防止组通信被非授权用户访问,所有的组内成员必须共享一个组通信密钥,所有的组通信都通过这个组通信密钥加密。每当有用户离开或加入群组时,这个组通信密钥必须更新。以使离开后的成员无法访问目前的通信(称为后向安全性),而一个新加入的成员无法访问以前的通信(前向安全性)。但是,由于应用环境的不同,这同一个问题又具有不同的内涵。

1.1 大型动态组播系统的密钥更新和发布

大型动态组播系统的用户数量一般可达十万甚至上百万。一般为一点到多点的通信模式。典型的应用是视频点播、实时信息服务(股票行情等)、网络游戏和软件在线升级等。由于用户数量巨大,底层的组通信机制通常使用可靠的组播通信。为了保证前面所述的组通信的前向和后向安全性,每当有成员变动(加入、退出、网络出错、以及强制逐出等)时都要更新组通信密钥,由于用户数量大以及成员变动频繁,密钥更新所付出的系统开销相当巨大。这就是目前大型组播系统密钥管理问题面临的挑战。目前的研究结果多为分级的集中式管理方案[1~6]。由一个组管理器或 TTP 产生组通信密钥并通过分级结构将密钥分发到各端用户。这类管理方案存在的问题包括:SPOF(single point of failure);组管理器或 TTP 往往很容易成为系统攻击的突破点;只适用于一点到多点的通信模式;系统的鲁棒性和容错性差,一旦组管理器或 TTP 崩溃,整个系统就将彻底瘫痪。

1.2 动态对等群组(DPGs)的密钥协商

和大型动态组播群组相比,DPGs 的用户数量一般不大,最多达到几百。一般为多点到多点的通信模式。典型的应用为镜像服务器(例如数据库、Web 服务器、时间服务器镜像等)、视频会议、支持协同工作的应用等。底层的组通信机制可以使用组播通信也可以使用传统的点到点的 Unicast 通信。为了保证前面所述的组通信的前向和后向安全性,DPGs 同样存在组通信密钥的更新问题。在 DPGs 中,目前的研究结果均是将两方的 Diffie-Hellman 密钥交换协议向动态对等群组扩展。我们统称为 GDH(Group Diffie-Hellman)协议。在 GDH 协议中,DPGs 中的每一个成员都对组通信密钥的生成作出同等的贡献(共享密钥的计算结果中含有每个成员自己的个人秘密值),各成员通过协商产生一个共享的组密钥。GDH 中也存在一个组管理器,可由组成员担任。这个组管理器不是固定的而是动态浮动的(可以由每次新加入的成员担任)。如果组管理器崩溃,系统只需付出很小的代价就可以动态生成一个新的组管理器并快速恢复。因此不存在 SPOF 问题。系统的鲁棒性和容错性好。应该说这些优点都是对等协商所带来的。这类协议的缺点是不适合用户数目巨大的动态群组。当用户数目超过20的时候系统性能就有明显下降[9]。

2 Diffie-Hellman 密钥交换协议在 DPGs 中的扩展

本节我们简介文[8]中提出的 CLIQUES 协议簇。它成功

*)本文得到国家863信息安全技术应急计划项目(公共安全接口技术研究863-301-7-9)基金支持。刘 璟 博士生,主要研究领域为计算机网络安全、分布对象技术;周明天 教授,博士生导师,主要研究领域为计算机网络、分布对象技术、并行分布处理和网络与信息系统安全。

地将 Diffie-Hellman 密钥交换协议扩展到 DPGs 上。表1给出了协议中要用到的一些记号。CLIQUES 协议簇包括两类协议，一类称为 IKA(Initial Key Agreement)即初始密钥协商协议。顾名思义，即在组刚刚成立后的起始阶段，用 IKA 协议产生共享的组密钥。另一类称为 AKA(Auxiliary Key Agreement)即辅助密钥协商协议。这类协议包括除 IKA 产生共享的组密钥外的所有密钥协商操作。例如在有成员变动(加入，离开等)情况下在 IKA 的基础上用于更新组密钥的协商操作。IKA 协议如图1所示，协议共执行 n 轮。实际上序号最大的成员 M_n 充当了组管理器的作用。在第一阶段($n-1$轮)执行后 M_n 经过计算 $K_n=(g^{N_1\cdots N_{n-1}})^{N_n}=g^{N_1\cdots N_n}$ 得到了共享组密钥，第二阶段 M_n 向各成员 M_i 发送中间值，各 M_i 通过计算 $K_n=(g^{\frac{N_1\cdots N_N}{N_i}})^{N_i}=g^{N_1\cdots N_n}$ 得到共享组密钥。

表1 CLIQUES 协议中用到的记号

n:协议参与者(组成员)的数目
i,j,k:区别组成员的下标
M_i:第 i 个组成员;$i\in[1,n]$
q:代数群的阶
g:指数基;代数群中阶为 q 的生成元
N_i:M_i 产生的随机(秘密)指数
S,T:$\{N_1,\cdots,N_n\}$的子集
$\prod(S)$:集合 S 中所有元素的乘积
K_n:n 个成员共享的组密钥

$$M_i \xrightarrow{\{g^{\frac{N_1\cdots N_i}{N_k}}\mid k\in[1,i]\},\,g^{N_1\cdots N_i}} M_{i+1}$$

Stage1(Upflow):round i;$i\in[1,n-1]$

$$M_i \xrightarrow{\{g^{\frac{N_1\cdots N_i}{N_k}}\mid i[1,n]\}} M_n$$

Stage 2(Broadcast):round n

图1 初始密钥协商协议 IKA

AKA 协议包括四个协议，篇幅所限我们只简介其中的两个协议即：有成员加入时的 AKA1协议和有成员离开时的 AKA2协议。分别如图2和图3所示。

在 AKA1协议中 M_{n+1} 为新加入的成员，组管理器职能从原来的 M_n 迁移给新加入的 M_{n+1}(考虑到新加入的成员可能不可信任，组管理器可以固定，参见文[8])，其中 $\hat{N}_n$ 为 M_n 新生成的个人秘密值。协议第一轮 M_n 将其重新生成的一组中间值传给 M_{n+1}。M_{n+1} 通过计算 $K_{n+1}=(g^{N_1\cdots\hat{N}_n})^{N_{n+1}}=g^{N_1\cdots\hat{N}_nN_{n+1}}$ 得到新的组密钥。协议第二轮 M_{n+1} 再将它计算后得到的一组新的中间值发送给各组成员。各组成员根据相应的中间值计算得到新的组密钥 K_{n+1}。

在 AKA2协议中，M_p 为离开或被逐出的成员。M_n 在生成新的个人秘密值 $\hat{N}_n$ 后，将其新生成的一组中间值传给各组成员。由于中间值中没有 $g^{N_1\cdots N_{p-1}\cdots N_{p+1}\cdots N_{n-1}\hat{N}_n}$，$M_p$ 无法根据中间值生成新的组密钥 $K_{new}=g^{N_1\cdots N_{p-1}\cdots N_{n-1}\hat{N}_n}$。如果离开或被逐出的成员为 M_n 时，情况有些不同(参见文[8])。

$$M_n \xrightarrow{\{g^{\frac{N_1\cdots\hat{N}_n}{N_k}}\mid k\in[1,n]\},\,g^{N_1\cdots\hat{N}_n}} M_{n+1}$$

Upflow:round 1

$$M_i \xrightarrow{\{g^{\frac{N_1\cdots\hat{N}_nN_{n+1}}{N_i}}\mid i\in[1,n]\}} M_{n+1}$$

Broadcast:round 2

图2 有成员加入时的 AKA1协议

$$M_n \xrightarrow{\{g^{\frac{N_1\cdots\hat{N}_n}{N_i}}\mid i\in[1,n-1]\wedge i\neq p\}} M_i$$

Broadcast:round 1

图3 有成员离开时的 AKA2协议

综上所述，在 IKA 协议执行并生成共享组密钥后(n轮)，当有成员变动时，AKA 更新组密钥所付出的计算和通信代价相对较小(AKA1协议仅执行两轮，AKA2协议仅执行一轮)。在下一节中，我们将利用这三个协议来构造新的大型动态组播组的密钥管理协议。

3 一个鲁棒且容错的组通信密钥管理协议簇

如图4所示，整个系统为一个三级结构。第三级为各分组内部的子组，如图4中分组1中的1号子组subgroup11，各子组

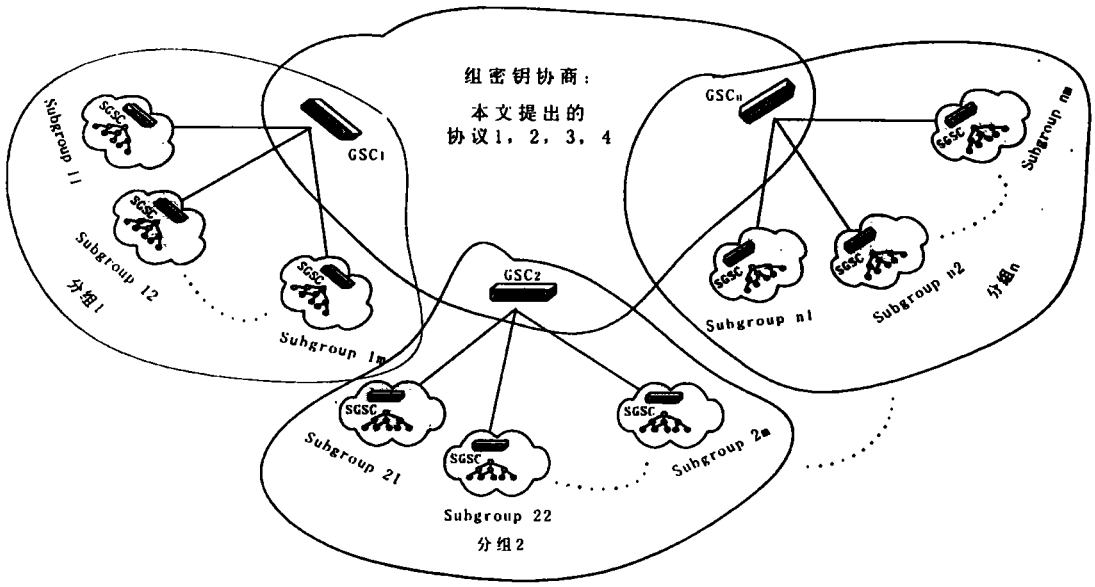

图4 系统框架图

由最终用户组成并由各分组内部的子组安全控制器 SGSC 管理(细节及协议参见文[1])。第二级为分组,各分组由组安全控制器 GSC 管理(细节及协议参见文[1])。第一级为组密钥协商级,组密钥的管理工作(产生及更新)通过3.1节提出的协议1,2,3,4完成。GSC 在系统的第二级上完成自己的职责,即:生成分组密钥,通过 GSC 密钥树在各子组安全控制 SGSC 间安全更新组密钥;SGSC 在系统的第三级上完成自己的职责,即:生成子组密钥,通过 SGSC 密钥树在各端用户间安全更新组密钥(参见文[1])。而组密钥的生成和更新则是在系统的第一级即组密钥协商级上进行的。

3.1 组密钥的生成及更新

组密钥的生成 各分组 GSC_i 作为组密钥协商级的成员参与组密钥的协商和更新。协商后生成的组密钥,通过分组密钥加密,由 GSC_i 通过 GSC_i 密钥树安全发送给各分组中的 $SGSC_{ik}$;各 $SGSC_{ik}$ 再通过 SGSC 密钥树将组密钥安全发送给各最终用户。具体协议(协议1)如下:

1. GSC_i ($i=1,\cdots,n$)作为 IKA 协议的各成员,执行 IKA 协议(参见图1),协商得到组密钥 K_n。

2. $GSC_i \rightarrow SGSC_{ik}$: $\{K_n\}_{K_{GSC_i}}$ 以组播方式发送。其中 K_{GSC_i} 表示各分组的分组密钥,$\{K_n\}_{K_{GSC_i}}$ 表示 Kn 被 K_{GSC_i} 加密。

3. $SGSC_{ik} \rightarrow m_{ij}$: $\{K_n\}_{K_{GSC_i}}$ 以组播方式发送。其中 m_{ij} 为第 i 个分组的第 j 个最终用户。

组密钥更新分为以下三种情况:

A)有分组加入(协议2)

1. 执行 AKA1协议(新加入分组的 GSC_{n+1} 相当于 AKA1 协议中的 M_{n+1}),协商得到新的组密钥 K_{n+1}。

2. $GSC_i \rightarrow SGSC_{ik}$: $\{K_{n+1}\}_{K_{GSC_i}}$ 以组播方式发送。

3. $SGSC_{ik} \rightarrow m_{ij}$: $\{K_{n+1}\}_{K_{GSC_i}}$ 以组播方式发送。

B)有分组离开(协议3)

1. 执行 AKA2协议(离开分组的 GSC_r 相当于 AKA2协议中的 M_r),协商得到新的组密钥 K_{new}。

2. $GSC_i \rightarrow SGSC_{ik}$: $\{K_{new}\}_{K_{GSC_i}}$ 以组播方式发送。

3. $SGSC_{ik} \rightarrow m_{ij}$: $\{K_{new}\}_{K_{GSC_i}}$ 以组播方式发送。

C)分组内部成员变动(端用户的加入、离开等情况)(协议4)

$$M_n \xrightarrow{\{g^{\frac{N_1\cdots N_n}{N_i}} \mid i\in[1,n-1]\}} M_i$$

Broadcast: round 1

图5 AKA3协议

1. 据分组内部成员变动情况(端用户加入或离开等)分别执行文[1]中的协议3,协议4或协议5更新分组密钥。

2. 执行 AKA3协议(由于在第一级组密钥协商级上没有成员的加入和离开,只需要更新组密钥即可。为此我们根据AKA2协议的思想,提出了图5所示的 AKA3协议。),得到新的组密钥 K_{new}。

3. $GSC_i \rightarrow SGSC_{ik}$: $\{K_{new}\}_{K'_{GSC_i}}$ 以组播方式发送,其中 K'_{GSC_i} 为第1步更新后的分组密钥。

4. $SGSC_{ik} \rightarrow m_{ij}$: $\{K_{new}\}K'_{GSC_i}$ 以组播方式发送。

由于篇幅所限,在组密钥协商级上,还有另外几种情形我们没有讨论:聚集加入(多个分组同时加入);组融合(例如群组1和群组2融合);子群组离开或者说组分裂(例如由分组1,2,3组成的一个子群组离开原来的组)。这些情况下的具体协议参见文[8]。

3.2 系统的鲁棒性和容错性分析

如前所述,集中式密钥管理方案[1~3,5,6]存在 SPOF(single point of failure)问题;为此我们借鉴 GDH 协议的思想,提出了上述的一组鲁棒及容错的大型动态组播系统的密钥管理协议。

文[1]中所述的方案一旦在组安全控制器 GSC 崩溃或变得不可信时,将造成全局性的影响(文[2,3,5,6]也存在同样的问题),为了确保系统安全运行,各 SGSC 只能等待 GSC 恢复并更新组密钥后才能继续工作。而恢复组系统必须重建组安全控制器 GSC 和 GSC 密钥树。由于系统恢复代价大,对端用户来说将有一个很大的延迟。如果某个 SGSC 崩溃或变得不可信(受到攻击等情况下),需要运行文[1]中的协议4,以在系统恢复期间临时性地更新组密钥。而恢复崩溃的 SGSC 所在的子组,需要重建 SGSC 及 SGSC 密钥树,并运行文[1]中的协议3。

而在本方案下,如图4所示,如果某个 GSC_i 崩溃或变得不可信,这时仅需执行协议3即可在系统恢复期间临时性地更新组密钥。而恢复子组 i 仅需要将 GSC_i 下属的各 $SGSC_{ij}$ 视为密钥协商级的成员,运行文[8]中聚集加入情况下的 AKA 协议即可更新组密钥。若某个 $SGSC_{ij}$ 崩溃或变得不可信,那么执行协议3临时性地更新组密钥以用于加密系统恢复期间继续的所有组通信。而恢复崩溃的 SGSC 所在的子组,情况要复杂一些,这时需要重建 SGSC 及 SGSC 密钥树,并运行本文中的 AKA2。由于 AKA 协议相对简单,执行效率高,所以系统恢复快,端用户感觉不到恢复延迟。

综上所述,引入了 IKA 和 AKA 协议后的大型动态组播系统的密钥管理将具有很高的鲁棒性和容错性。

结论 从上述的讨论中,可以看出大型动态组播系统的密钥管理和动态对等群组(DPGs)的密钥协商两个研究领域在多方面可以互补。遗憾的是目前学术界仍将其作为不同的研究领域予以区别对待。本文首次提出了可将两个领域的相关技术进行有效结合的研究方向,并作出了成功尝试:提出了一组鲁棒及容错的大型动态组播系统的密钥管理协议。很好地解决了集中式密钥管理协议存在的 SPOF 问题,使得协议具有良好的鲁棒性和容错性。

参考文献

1 刘璟,周明天. 大型动态多播群组的密钥管理和访问控制. 软件学报,2002,13(2):291~297

2 刘璟,周明天. 大型动态群组的多播安全机制. 计算机科学,2001,28(1):84~88

3 Wong C K, Gouda M, Lam S S. Secure Group Communication using Key Graphs. University of Texas at Austin: [Computer Science Technical report TR 97~23]

4 Mittra S. IOLUS: A Framework for Scaleable Secure Multicast. ACM Computer Communication, 1997, 27(3): 277~288

5 Harkins D, Doraswamy N. A Secure, Scalable Multicast Key Management Protocol (MKMP). IETF Internet draft (work in progress), Mar. 1998. http: // www. ipmulticast. com/tech-cent. htm

6 Balenson, McGrew D, Sherman A. Key Management for Large Dynamic Groups: One-Way Function Trees andAmortized Initialization. Internet Draft. http: // www. ipmulticast. com/tech-cent. htm

7 Steiner M. Tsudik G, Waidner M. Diffie-Hellman key distribution extended to groups. In ACM Symposium on Computer and Communication Security, March 1996

8 Steiner M, Tsudik G, Waidner M. Cliques: A new approach to group key agreement. In: IEEE Conf. on Distributed Computing Systems, May 1998

9 Agarwal D A, Chevassut O, Tompson M R, Tsudik G. An Integrated Solution for Secure Group Communication in Wide-Area Networks. In IEEE Symposium on Computers and Communications, 2001