基于CBT组播路由协议的组密钥管理协议分析与改进

Analyses and Improvement on Group Key Management Protocol Based on CBT Multicast Routing Protocol

张 斌 邬江兴

(国家数字交换系统工程技术研究中心 郑州450002)

Abstract In RFC1949, A. Ballardie proposed a method of scalable multicast key distribution based on the Core Based Tree multicast protocol fitting for wide-area IP multicasting. This paper analyses the method deeply, points out that the model can't resist the key compromise of a single member effectively, and proposes two kinds of improvement. One is the distributed generating and distributing group data encryption key, the other one is partitioning the group into subgroups. These improvements can increase the security and lifetime of the group key, and decrease the loss of compromise of a single group key encryption key.

Keywords Group key management protocol(GKMP), Core based tree(CBT), Multicast key distribution, Multicast security

1 引言

随着Internet网的迅猛发展,组播应用越来越广泛,对其安全性要求也日益提高。为了进行安全组播,一个重要的问题就是组播密钥的安全高效分配。采用传统的密钥分配中心KDC执行密钥分配功能只适合于小型、局域网组播环境,对于广域网组播效率很低,不具有可放缩性。组密钥管理协议GKMP提出了进行组播密钥分配的一种方法,此方法并不依赖于一个集中式的KDC,而是把密钥管理的任务交给一个组成员,不同的组可以有不同的组成员充当组控制器,用以提高密钥分配的可放缩性,但由于每个组密钥分配的负担只交给一个组成员,仍然有可放缩性问题[1,2]。为了进一步提高密钥分配的可放缩性,Ballardie在RFC1949中提出了基于CBT组播路由协议[3]的可放缩组播密钥分配方法,提供了一种简单、低代价可放缩方案[4]。CBT是一种适合于本地和广域网IP组播的新型路由协议,它为每个组提供一个共享传递树,在CBT中构成传递树的所有路由器知道其树上的邻居路由器,允许数据报文的逐跳认证[3]。CBT树中核心的存在提供了一种自然的授权中心,使用CBT组播路由协议进行密钥分配并不排除使用其它的组播路由协议进行实际的组播通信。本文对这种组播密钥分配方法进行了分析,指出了由于其强可放缩性而带来的不足之处并提出了一些改进意见。

2 CBT组播密钥分配模型

以CBT组播路由协议为基础结构进行组播的密钥分配并不依赖于一个集中式KDC的存在,每个组都有它自己的组密钥分配中心GKDC,而且这种密钥分配功能可以传递给已加入CBT树的其它节点[4],因此这种方法具有分布式密钥分配的能力,具有可放缩性。RFC1949建议初始化时,由CBT主核心承担GKDC功能,由组启动者负责创建组播组访问控制表ACL,为了对这张表进行认证需对它进行数字签名,之后组启动者向主核心单播ACL。如果ACL中组成员信息为保密信息,ACL在被发送前将用主核心公钥进行加密;另外,ACL中还可包括可信路由器信息,用以决定是否这些路由器将成为CBT树中的节点,具备密钥分配能力,然后,主核心创造一个用以接收和认证含有访问控制表消息的组密钥grpkey和一个组密钥加密密钥KEK,KEK用于周期性地更新老的grpkey和老的KEK,任何拥有ACL、grpkey和KEK的树节点都有能力认证一个要求加入的成员,因此具有安全地分布式加入树的能力和组播密钥分配能力。

2.1 有关的术语定义

PK-A:实体A公钥;

SK-A:实体A私钥;

张 斌 博士生,主要研究方向为计算机网络安全。邬江兴 教授,博士生导师,主要研究方向为交换理论、通信网、计算机网络安全和高速路由器等。

token：令牌，强认证交换的组成部分，由时标 t 和随机数 r 组成；

A→B：表示从 A 向 B 发送消息；

{A}^B：表示用 B 对 A 进行加密；

[A]^B：表示用 B 对 A 进行数字签名；

SAparams：安全集合参数，包括认证算法、算法模式、密钥生存周期和敏感级别等，由主核心或主核心与组启动者相结合建立组通信时使用的安全参数；

grpAP：从已证实树节点向要求加入节点发送的组访问报文，其组成部分为：

[token-sender，[ACL]^SK-core，{[grpkey，KEK，SAparams]^SK-core}^PK-originhost，{[grpkey，KEK，SAparams]^SK-core}^PK-next-hop]^SK-sender。

2.2 加入认证和组播密钥分配模型

为了便于说明问题，假定存在一个全局的公钥管理体制 RSA，ACL 已被安全地传递到主核心，主核心已经过安全集合建立协议获得了 grpkey、KEK 和安全参数索引 SPI。如图1所示，主机 h 要求加入组播组 G，本地组播路由器 A 还没有加入组 G 的 CBT 树，其加入认证和密钥分配过程如下：

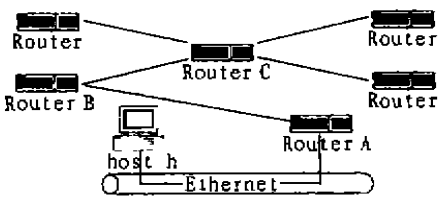

图1 基于 CBT 的组播密钥分配网络示意图

(1)要加入组 G 的主机 h 发送一个 IGMP 组成员报告[5]：

h→DR(A)：[[token-h]^SK-h，IGMP 组成员报告]，其中 DR 为指定路由器。

(2)本地指定路由器 A 接收 IGMP 报告，认证 h 的封装令牌 token，如果认证成功，路由器 A 形成一个 CBT 加入请求，目标为主核心路由器 C，这个请求消息包括它自己的令牌以及主机 h 签名的令牌，A 向到达 C 的路径上的最佳下一跳路由器 B 单播发送这个消息：

A→B：[[token-A]，[token-h]^SK-h，join-request]^SK-A

(3)B 认证 A 的加入请求，如果成功 B 将重复(2)，直至到达最接近主核心的路由器。对于图1的情况，将把从 A 接收到的消息中的 token-A 变为 token-B，token-h 不变，用 SK-B 进行数字签名。

B→C：[[token-B]，[token-h]^SK-h，join-request]^SK-B

(4)C 认证 B 的加入请求，又因为 C 是主授权点和 GKDC，C 也认证主机 h。C 首先检查 h 是否在组 G 的 ACL 中，如果不在不需要认证 h，从 C 向 B 发送加入失败 join-nack 消息，这个消息最终将到达本地指定路由器 A，如果 C 成功地认证了 B 和 h，C 形成组访问报文 grpAP，封装在 join-ack 消息中，同时对这个消息用 SK-C 数字签名。Join-ack 中也应包括 h 的令牌，用以让 h 所在子网的路由器识别新加入树中的成员。

C→B：[[token-h]^SK-h，grpAP，join-ack]^SK-C

(5)B 认证来自 C 的 join-ack，从 grpAP 中用其自己的私钥 SK-B 解密抽取(grpkey，KEK)密钥对，认证主核心，使用本地钥对解密后的密钥对重新加密，然后证实 ACL 的数字签名，把 ACL 存储在本地适当的表中。B 对接收到的 grpAP 进行如下变化，即把原来用 PK-B 加密的(grpkey，KEK)密钥对改用 join-ack 要经过的下一跳路由器的公钥，对图1就是用 PK-A 加密，把 token-C 换为 token-B，对组访问报文改用发送者的私钥即 SK-B 签名。

B→A：[[token-h]^SK-h，grpAP，join-ack]^SK-B

(6)A 认证来自 B 的 join-ack 消息，操作过程和(5)类似，唯一不同的是只需从接收的 grpAP 中抽取加密的源主机密钥对直接转发给 h，不再生成新的 grpAP，因为下一跳是主机 h 而不是路由器。

A→h：[[token-h]^SK-h，{grpkey，KEK，SAparams}^PK-h]

主机 h 接收到 A 送来的消息后，解密密钥对，在其 SA 表中储存 SA 参数。

由此过程可知，只要是 CBT 密钥分配树上的路由器，都可以充当 GKDC，例如，假定 A 已加入 CBT 树，h 向 A 发出 IGMP 成员报告后，A 将直接向 h 发送(6)中的消息，由此获得(grpkey，KEK，SAparams)。

3 CBT 组播密钥分配模型安全性分析

为了支持密钥管理的可放缩性，CBT 组播密钥分配模型使用单个组控制器即主核心负责产生组密钥对：组密钥 grpkey 和组密钥加密密钥 KEK，同时为了达到可放缩性，CBT 无条件地信任全部组成员，允许任何已加入组的有效成员接受新成员并分配密钥对，每个组成员包括组控制器拥有相同的单个组密钥和单个组密钥加密密钥，这种方法极大地减轻了主核心路由器作为 GKDC 分配组密钥对的负担，同时提供了分布式分配密钥的能力，为想加入组播组的用户提供了一种方便高效的手段，然而一旦一个组成员发生泄露就将泄露这两个密钥，从而危及未来的整个组通信。显然，需要对 CBT 传递树上的全部路由器都信任持怀疑

态度，有可能一个或几个路由器遭到攻击导致泄露组密钥和组密钥加密密钥，单独泄露组密钥可以让主核心产生新的组密钥对，然后用老的KEK加密组播传送到整个组，以此抵抗这种攻击，而KEK或整个密钥对的泄露则只能由主核心产生新的密钥对重新执行加入认证和密钥分配过程，导致组通信延迟增加和效率降低。

为了提高安全性，RFC1949建议所有的加入请求必须被一个核心路由器认证，即到达CBT树上路由器的加入请求将向核心路由器转发，只有核心路由器才具有密钥分配的能力，这就降低了这种模型的分布式分配密钥的能力，同时也降低了效率，然而这种改进也没有解决KEK泄露的问题。

4 对CBT组播密钥分配模型的改进

为了提高组密钥的安全性，增加其生存周期，一旦一个或几个成员泄露了组密钥和KEK能妥善处理泄露问题，这可以从以下几个方面加以考虑。

4.1 利用共享组密钥分布式产生组数据加密密钥

为了在组成员间进行保密通信，可以利用每个组成员都拥有的已建立的组密钥grpkey加密传送，然而为了提高grpkey的安全性，延长其生存周期同时也延长KEK生存周期，要尽量减少用其加密的数据量，限制密码分析者攻击时所能得到的同一密钥下加密的密文量。当然，每个成员发送者要求发送数据时可以向核心路由器申请新的组数据加密密钥S，由核心路由器产生S，用组密钥加密组播传送至每个组成员，这种方法虽可达到目的，但无疑加重了组核心路由器的负担，同时申请和传送密钥S也占用了网络带宽。以下给出一种分布式产生S的方法，具有一次一密、防止重放攻击的优点，即使S被泄露，由于使用S加密的数据量有限因而影响不大，同时不增加核心路由器负担，不需要传送密钥S，S只在需要时通过协议建立，降低了密钥S的分配存储量。

4.1.1 方法描述

(1)任何想发送消息的组成员产生一个随机数n；

(2)产生组数据加密密钥S；

S=SHA(control-information,grpkey,n)

其中control-information为与组相关的非保密信息，可以包括发送成员的用户标识符、密钥S的使用信息等一些控制信息，SHA为抗冲突的强单向Hash函数。

(3)用S加密要发送的消息，可使用任何对称密钥加密算法，例如DES、IDEA等；

(4)如果需要认证发送者，使用以下公式产生数字签名；

SIG=[SHA(control-information,n)]⁻ SK-sender

其中SK-sender表示发送者私钥，对于每个发送者n不能重复。

(5)同时向每一个组成员组播发送用S加密的消息、控制信息、随机数n以及如果需要的话发送数字签名SIG；

(6)接收者收到控制信息、随机数n之后，由于已知grpkey，可根据(2)中求S公式得出S，利用S解密经过组播送来的加密消息。

4.1.2 安全性分析

分布式产生组数据加密密钥方法的安全性取决于组密钥的安全性和抗冲突的强单向Hash函数SHA或MD5的安全性。组密钥在CBT组播密钥分配模型中是由主核心负责产生并由公钥密码体制传递，具有很高的安全性。例如若采用RSA体制，公钥大小为1024位，则若想获得这个密钥须至少执行2^{80}次操作。SHA是一种标准Hash安全算法，其输入消息长度小于2^{64}位，输出压缩值为160位，伪造一个消息其Hash值与给定的Hash值相同在计算上是不可行的，找出两个不同消息具有同一Hash值在计算上也是不可行的，能对抗穷举攻击，因此也具有很高的安全性。

4.2 划分子组减少KEK泄露带来的损失

前面已提到KEK一旦泄露可通过重新执行加入认证和密钥分配得到新的KEK，然而这种方法用的是公钥密码体制，例如RSA体制，运算效率很低，因此可通过将组成员根据一定的规则分组产生子组密钥，在组成员加入组时进行分配，这样一旦一个或几个成员泄露了KEK，可首先找到其所在的分组，对这些分组还要执行初始分配过程，而对其它分组则可在核心路由器产生新的KEK后用子组密钥加密分别组播传送至各个不同的分组，极大地节约了时间和空间开销，对密钥分配过程的修改如下：

(1)ACL中的主机要根据一定的规则进行分组，例如可根据IP地址，具有相同子网地址的主机分为一组，核心路由器第一次接到来自成员h的加入请求信息后，除了产生grpkey、KEK还要根据其分组信息产生子组密钥subgrpkey，以后再收到成员加入信息时，首先判断是否已产生了其所在分组的subgrpkey，如果没有则产生之，并建立一张子组用户和subgrpkey的对应表，

(2)组访问报文grpAP变为：

[token-sender,[ACL]⁻ SK-core{[grpkey,KEK,SAparams,subgrpkey]⁻ SK-core}⁻ PK-originhost,{[grpkey,KEK,SAparams,subgrpkey]⁻ SK-core}⁻ PK-next-hop]⁻ SK-sender

(下转第32页)

假)组织相应的回复邮件并发回给会议组织者。如果接收到的是重发的会议通知邮件,Agent 将提示被通知者已经接收到了来自会议组织者的催促,要求其作出回复。

下面是会议通知系统运作过程的框图,从中可以很清楚地了解到各种邮件格式和其关联处理 Agent 是如何交织来完成整个工作的。

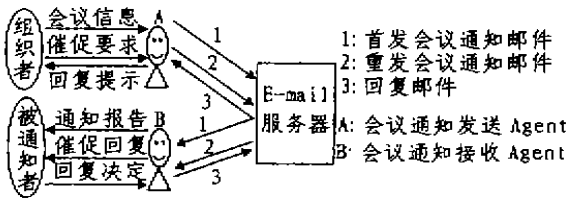

图3 会议通知系统工作框图

3.4 管理和智能提醒功能

新的应用模式改进了用 E-mail 来发送会议通知的方式,但是繁多的会议通知信息和回复信息仍需要有效的辅助。因此,针对会议通知这个领域的应用还应该具备对繁多复杂的会议通知进行有效的管理和智能化的提醒的功能。

管理的功能主要体现在对所有已收或已发会议通知的系统记录,有序列举和各种信息的随时待查,这些信息可以包括已收会议通知的相关信息,已做的决定,已发会议通知的相关信息和各位被通知者的回复情况等等。

当收到的会议通知或是组织的会议比较多的时候,就可能会出现遗忘的现象。智能提醒功能针对这一问题,对每个会议都设定了若干个提醒点,一旦到达某个提醒点就会主动提醒用户此会议即将召开,同时展示与本会议相关的各种信息。智能提醒一方面可以提醒组织者和决定参加者准时到会,另一方面又可以督促组织者查询回复情况以及时作出催促的决定。

这样看来,这不仅仅提供了一种更有效、更自然的通过 E-mail 发送会议通知的手段,从某种意义上来说它更像是一个助手,能辅助你完成诸多与会议通知有关的事项,并且做得更好。

结束语 本文针对目前 E-mail 的应用现状及 E-mail 本身的应用潜力提出了一种新的 E-mail 应用模式的设想。实现了一个中间件平台 EAMS,并设计了一个邮件格式化语言 EDL。EAMS 是一个为了能够更有效地实现基于 E-mail 处理的应用系统而设计的中间件平台系统,我们在其上以会议通知领域为典型应用说明了新的应用模式。

利用新的应用模式下会议通知领域的功能以 E-mail 为通信途径发送会议通知,不仅克服了现有使用 E-mail 过程中的种种弊端,提供了一种更有效、自然的通知途径,还具有系统化的管理会议信息和智能化的提醒功能。

会议通知领域只是各种各样领域中的一个,文中举例的目的是为了抛砖引玉,我们在其它很多的领域中诸如通讯评审、数据库离线查询、高性能计算资源共享等等都可以构建各自的领域应用,从而以新的应用模式来更好地使用 E-mail 这种通讯手段。

参 考 文 献

1 Standard for Format of ARPA Internet Text Messages. Ceocker,RFC822,1982
2 Postel J B. Simple Mail Transfer Protocol,RFC821,1982
3 Rose M Post Office Protocol-Version 3,RFC1460,1993
4 Genesereth M R,Ketchpel S P. Software Agents. Comm. ACM,1994,37(7):48～53
5 Maes P. Agents that Reduce Work and Information Overload. Comm. ACM,1994,37(7):30～40
6 Lemay L. Web Publishing with HTML. Sams. net Publishing,1996
7 柳杨. 基于 Internet 的 E-mail Agent 中间件系统.[南京大学硕士论文]. 1999

(上接第35页)

结论 基于 CBT 组播路由协议的组播密钥分配方法是一种适合于广域网组播的可选组播密钥分配方法,具有很强的可放缩性,每个经过认证的 CBT 传递树上的路由器都可成为组密钥分配中心 GKDC。本文对其组播密钥分配模型进行了详细分析,指出了其优缺点并从提高这种方法安全性和可用性的角度提出了两种改进措施,克服了原方法存在的一些缺点,具有一定的实用价值和推广意义。

参 考 文 献

1 Harney H,Muckenhirn C. Group Key Management Protocol(GKMP) Specification. RFC2093,1997
2 Harney H,Muckenhirn C. Group Key Management Protocol(GKMP)Architecture RFC2094,1997
3 Ballardie A,Cain B,Zhang Z. Core Based Tree(CBT Version3) Multicast Routing Protocol Specification. Internet draft,1998
4 Ballardie A. Scalable Multicast Key Distribution. RFC 1949,1996
5 Fenner W. Internet Group Management Protocol,Version 2 RFC2236,1997