移动自组网络分布式组密钥更新算法*

况晓辉[1,2+], 朱培栋[1], 卢锡城[1]

[1](国防科学技术大学 计算机学院,湖南 长沙 410007)

[2](北京系统工程研究所,北京 100101)

Distributed Group Rekeying Algorithms for Mobile Ad-Hoc Networks

KUANG Xiao-Hui[1,2+], ZHU Pei-Dong[1], LU Xi-Cheng[1]

[1](School of Computer, National University of Defense Technology, Changsha 410073, China)

[2](Beijing Institute of System Engineering, Beijing 100101, China)

+ Corresponding author: Phn: +86-10-66356599, E-mail: xiaohui_kuang@hotmail.com

Received 2003-04-10; Accepted 2003-06-04

Kuang XH, Zhu PD, Lu XC. Distributed group rekeying algorithms for mobile ad-hoc networks. *Journal of Software*, 2004,15(5):757~766.

http://www.jos.org.cn/1000-9825/15/757.htm

Abstract: Many emerging mobile wireless applications depend upon secure group communication, in which a secure and efficient group rekeying algorithm is very important. In this paper, a rekeying algorithm named DGR (distributed group rekeying algorithm) is proposed, which is based on DGKMF (distributed group key management framework). DGR algorithm generates a group key with local secrete information, and is suitable for mobile ad hoc networks. In order to further reduce the communication complexity, the DGR algorithm is improved on by generating a cluster dynamically in the rekeying process, and the CDGR (cluster distributed group rekeying algorithm) is proposed. The security, correctness, and completeness of the two algorithms are discussed in this paper, and their message complexity costs are evaluated. Simulation results demonstrate that the two algorithms are better than other algorithms and protocols such as CKD, GDH v.2 and BD in the group rekeying success ratio and delay, and the CDGR is better than GDR in the group rekeying delay because it uses the cluster in the rekeying process.

Key words: Mobile Ad-hoc Networks; secure group communication; DGKMF (distributed group key management framework); group rekeying; clustering

摘 要: 安全性是移动自组网络组通信的基本需求,安全、高效的组密钥更新算法是保证组通信安全的关键.在移动自组网络分布式组密钥管理框架(distributed group key management framework,简称 DGKMF)的基础上,提出了一种组密钥更新算法——DGR(distributed group rekeying)算法.该算法能够利用局部密钥信息更新组密钥,适合拓扑结构变化频繁、连接短暂、带宽有限的移动自组网络.为了进一步降低算法的通信代价,通过在组

* Supported by the National Natural Science Foundation of China under Grant Nos.90104001, 90204005 (国家自然科学基金)

作者简介: **况晓辉**(1975－),男,湖南新化人,博士生,主要研究领域为计算机网络,信息安全;**朱培栋**(1971－),男,博士,副教授,主要研究领域为网络路由,组播技术,高性能路由器;**卢锡城**(1946－),男,教授,博士生导师,中国工程院院士,主要研究领域为先进网络技术,高性能计算,并行与分布处理.

密钥更新时动态生成组密钥更新簇,对 DGR 算法进行了改进,提出了 CDGR(cluster distributed group rekeying)算法,并讨论了上述算法的安全性、正确性和完备性,分析了算法的通信代价.最后,利用 ns2 模拟器对算法的性能进行了分析.模拟结果显示,DGR 和 CDGR 算法在组密钥更新成功率和延迟等方面均优于其他算法,并且由于采用簇结构,CDGR 算法的更新延迟低于 DGR 算法.

关键词: 移动自组网络;安全组通信;分布式组密钥管理框架;组密钥更新;簇结构

中图法分类号: TP309 **文献标识码**: A

移动自组网络(mobile ad hoc network,简称 MANET)[1]是一种不依赖于任何固定基础设施的新型无线网络.与需要中心控制的设备,如基站或访问服务点的蜂窝移动通信网络和无线局域网相比,移动自组网络具有自组性、多跳性、无基础设施要求、易铺设等特点,可被广泛用于军事战场信息系统建设、民用紧急救助、执法等场合[2~5].

安全性是移动自组网络组通信的基本要求.在组通信中,信息机密性和完整性通过组密钥加密实现,而组成员的变化都需要更新组密钥[6,7],因此,如何安全、高效地更新组密钥是安全组通信的关键问题.目前有线网络的安全组通信研究取得了许多进展[6~8],然而由于移动自组网络拓扑结构的频繁变化、连接的短暂性以及带宽有限等特点,有线网络的组密钥管理协议和算法在移动自组网络中难以获得较好的性能[9,10],移动自组网络的安全组通信研究正日益受到人们的重视.针对大规模移动自组网络的安全组通信问题,S. Griffin 和 B. DeCleene 基于层次密钥管理框架[4],提出了 DR,SR,IR 以及 FEDRP[11]等域间组密钥更新算法,但这些算法依赖于固定的密钥管理节点生成和分发组密钥;Carman[12]通过模拟的方法比较了组密钥管理协议在 DSP 网络能源消耗的情况,并提出了低功耗的密钥分发算法,但是该算法未考虑网络节点移动的情况.Stefano Basagni[5]针对节点运算能力较弱的移动自组网络提出了基于对称加密体制的组密钥更新算法,但是该算法不考虑退出组成员对组通信安全的威胁,并且不提供点到点通信的鉴别机制.

DGKMF(distributed group key management framework)[13]是一种以组密钥生成中心为核心的分布式组密钥生成框架.本文在 DGKMF 的基础上,针对所有节点均可移动且属于同一个通信组的自组网络环境,提出了两种组密钥更新算法:DGR(distributed group rekeying)算法和 CDGR(cluster distributed group rekeying)算法.DGR 算法仅利用局部密钥信息更新组密钥,避免了移动自组网络拓扑结构变化频繁、连接短暂等对组密钥更新的影响.在 DGR 算法的基础上,CDGR 算法在组密钥更新时动态生成组密钥更新簇,从而进一步降低了组密钥更新的通信代价.最后,通过模拟验证了算法的有效性和正确性.

1 分布式组密钥管理框架

分布式组密钥管理框架以 RSA 非对称密码体制和基于拉各朗日插值的秘密共享方案为基础,为带宽有限、通信信道可靠性不高、节点自由移动的自组网络提供组密钥管理支持.在部署移动自组网络之前,由离线的控制节点根据网络的应用需求对通信组进行划分,并初始化节点的密钥信息.在网络部署以后,由门限个节点组成组密钥生成中心,对组成员的资格进行管理.离线的组控制节点也可为新的组成员颁发组成员资格证书[13].为简化算法的描述,假设网络中的所有节点属于同一个通信组.

假设移动自组网络由 N 个节点组成,离线的组控制节点生成组的 RSA 密钥对为 $\{SK,PK\}$,其中 SK 为私有密钥,PK 为公开密钥,并选择组通信密钥种子生成函数 $g(x)$以及初始值.在网络部署之前,组控制节点预先为每个组成员生成并分发一定的密钥信息.

假设节点的全局惟一标识为 $v_i, i \in 1,2,...,N$, 拥有的密钥信息包括:

- 节点的共享密钥 GCK_i;
- 节点的私钥和公钥对:$\{sk_i,pk_i\}$;

- 节点的组成员资格证书$(cert_j)_{SK}$*;
- 组通信密钥种子生成函数 $g(x)$;
- HASH 函数;
- 初始组密钥:$HASH((g(m))_{SK})$,m 表示组密钥种子初始序号.

节点的共享密钥由组控制节点根据拉各朗日插值秘密共享方案生成,生成过程如下:令 $SK=(d,n)$,门限为 k,且$1<k<\frac{N}{2}$.组控制节点随机选择 k–1 阶多项式:

$$f(x)=d+\sum_{i=1}^{k-1}f_i x^i,$$

因此有 $f(0)=d$.节点 v_i 的 $GCK_i=f(v_i) \bmod n$,其中 $i=1,2,...,N$.

节点的公钥、私钥对用于组密钥生成与分发时的安全通信.节点的组成员资格证书由离线控制节点利用组私钥签发,用于验证节点的公钥以及组成员资格.组成员的删除通过证书废除列表实现.

在框架中,组通信密钥 TEK(traffic encrypt key)由组通信密钥种子经过组私有密钥加密后的密文,再经过散列函数(用 HASH 表示)变换后获得.根据秘密共享方案,任何 k 个以上的成员都可以在不暴露组私有密钥的情况下生成 TEK.因此,在该框架中,任意 k 个组成员都能够组成 TEK 生成中心,使 TEK 的生成与分发不受网络拓扑变化的影响,避免了多跳连接不可靠以及单点失效等问题.此外,组成员节点可以利用组密钥生成种子序列号的单调性来维护 TEK 的一致性.

2 TEK 更新算法

在组通信过程中,数据的私密性通过 TEK 加密实现.当节点加入或退出时,组成员节点都需要更新 TEK,以保证通信的后向私密性和前向私密性[6,7].

在 DGKMF 中,请求加入的组成员需要由离线组控制节点或门限个组成员节点为其颁发组成员资格证书[13].利用 TEK 更新算法,新加入的组成员与其邻居节点生成新的组密钥以后,用已有的组密钥加密新的组密钥,向组中所有节点广播,以更新组密钥.

组成员退出分为主动退出和强制退出两种情况.当组成员主动退出时,它向全组广播退出请求.强制退出请求由发现异常的节点广播.接到退出请求的组成员利用 TEK 更新算法更新组通信密钥,同时将退出组成员的资格证书加入到证书废除列表中,以防止已退出的组成员参与密钥更新过程.

高效、安全的组密钥更新算法对于安全组通信至关重要.在 DGKMF 的基础上,本文提出了两种组密钥更新算法:DGR 算法和 CDGR 算法.这两种算法均只需利用局部密钥信息更新组密钥,避免了移动自组网络拓扑结构变化频繁、连接短暂等特点对组密钥更新的影响;其主要区别在于密钥更新的方式不同.在 DGR 算法中,每个节点均需要与门限个以上的邻居节点通信,以更新组密钥,局部通信代价较大.而 CDGR 算法在组密钥更新时动态生成组密钥更新簇,通过建立层次结构降低了组密钥更新的通信代价.

假设节点在密钥更新过程中时钟同步,节点在更新过程中安全可靠,且节点通信具有松散的同步机制支持.

2.1 DGR算法

在 DGR 算法中,节点在接收到密钥更新广播以后,利用邻居节点的共享密钥信息更新组密钥.所有节点采用相同的更新过程:每个组成员节点通过邻居发现过程获得 k–1 个邻居节点标识,然后向邻居节点发送部分密钥请求(PARTKEY_REQ),在接收到 k–1 个部分密钥(PART_KEY)响应以后,利用组密钥生成算法合成 TEK.在 DGR 算法中,节点状态和行为的有限状态机如图 1 所示.

初始状态(init state).节点在新的密钥更新开始前的状态称为初始状态.此时,所有节点的 TEK 均为 $HASH((g(i))_{SK})$,其中 $g(i)$为 TEK 对应的组密钥种子,其序号为 i,节点的状态标识 *Rekey* 置为 0,表示节点处于初始

* $(m)_{SK}$ 表示 m 由私钥 SK 加密.

状态,*Rekey_Round* 置为 1,用于控制通信的范围.

当节点接收到组密钥更新请求(*REKEY*)时,进入等待更新状态.

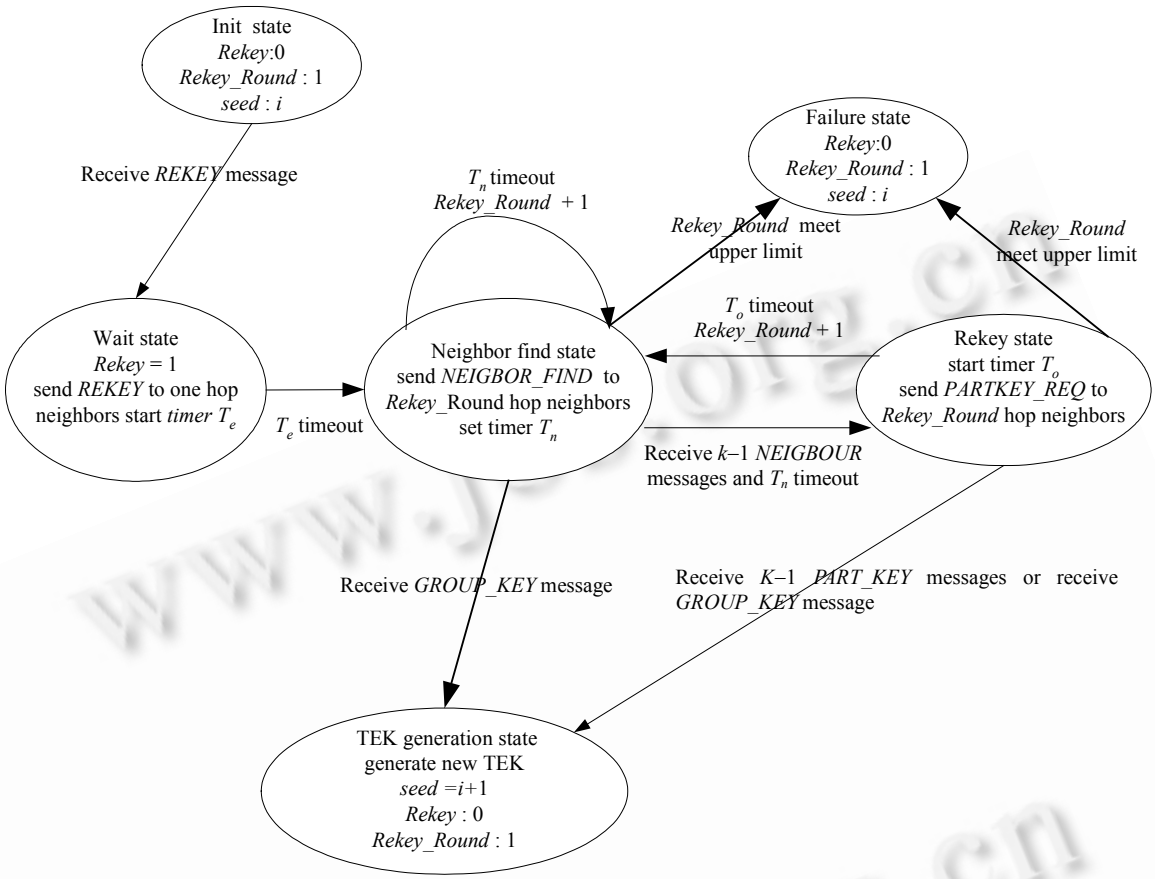

Fig.1 Finite state machine diagram of DGR algorithm

图 1 DGR 算法的有限状态机

等待更新状态(wait state).节点进入等待更新状态以后,首先将 *Rekey* 置为 1,并向邻居节点广播 *REKEY*.同时启动计时器 T_e,该计时器用于避免节点同时进入邻居发现状态,而导致无线信道过度拥塞.当 T_e 超时后,节点进入邻居发现状态.如果 *Rekey* 为 1,节点不再响应 *REKEY* 消息.

如果节点接收到 *NEIGBOR_FIND* 消息,首先将请求节点标识加入响应节点集合中,以避免在同一次密钥更新过程中多次响应同一个节点请求,然后向请求节点返回邻居响应消息(*NEIGBOR*).

邻居发现状态(neighbor find state).为了正确更新组密钥,节点需要与门限个邻居组成员合作,节点在获取其门限个邻居节点标识的过程中所处的状态称为邻居发现状态.节点进入邻居发现状态以后,向其 *Rekey_Round* 跳范围内的邻居节点广播 *NEIGBOR_FIND*,其中包含组密钥种子序号 *i*+1,然后等待 *NEIGBOR* 消息.同时启动计时器 T_n,该计时器用于控制节点等待响应消息的时间.当 T_n 超时以后,节点仍未获得 *k*–1 个邻居节点标识,则将 *Rekey_Round* 加 1,重复上述过程.当 *Rekey_Round* 超过其上限,邻居发现过程失败,进入失败状态.

如果节点接收到 *NEIGBOR* 消息,则将其中的节点标识加入到邻居节点列表中,当邻居节点的数量达到 *k*–1 时,在 T_n 超时后节点进入更新状态.

如果节点接收到更新组密钥消息(*GROUP_KEY*),则进入组密钥生成状态.

更新状态(rekey state).节点进入更新状态以后,向其 *Rekey_Round* 跳范围内的邻居节点广播 *PARTKEY_REQ*,其中包含节点的组成员资格证书、邻居节点标识以及组密钥种子序号 *i*+1,并用节点的私钥签

名保证消息的完整性,然后等待接收 *PART_KEY* 消息.

启动组密钥更新计时器 T_o,该计时器用于控制节点等待响应消息的时间.当 T_o 超时以后,节点仍未获得 k–1 个部分组密钥,则返回到邻居发现状态,并将 *Rekey_Round* 加 1.当 *Rekey_Round* 超过其上限时,进入失败状态.反之,当节点获得 k–1 个部分组密钥时,则进入组密钥生成状态.

如果节点接收到 *GROUP_KEY* 消息,则直接进入组密钥生成状态.

组密钥生成状态(TEK generation state).如果节点接收到 *GROUP_KEY* 消息,则利用私钥解密后直接获得组密钥,否则,节点利用获得的 k–1 个部分组密钥,加上自己生成的部分组密钥,根据组密钥生成算法[13]生成新的组通信密钥 $HASH((g(i+1))_{SK})$,而后将 *Rekey_Round* 置为 1,*Rekey* 置为 0,恢复到初始状态,完成此次组密钥更新.

失败状态(failure state).将 *Rekey_Round* 置为 1,*Rekey* 置为 0,TEK 对应的组密钥种子序号仍为 i.此时,节点可能因为移动而变为孤立节点,从而导致组密钥不一致.当节点接收到组通信信息或发送组通信信息时,可利用 DGKMF 的基于序列组密钥一致性机制获得组密钥.

处于邻居发现状态或更新状态的节点在接收到 *PARTKEY_REQ* 消息以后,首先验证消息的完整性,然后利用组成员资格证书检测请求发送节点是否为合法的组成员,如果节点标识在 *PARTKEY_REQ* 消息的邻居列表中,则根据其中的组密钥种子序号生成部分组密钥,利用请求节点的公钥加密,向请求节点返回 *PART_KEY* 消息.

当节点处于组密钥生成状态时,在接收到 *PARTKEY_REQ* 或 *NEIGBOR_FIND* 消息以后,如果其组密钥生成种子的序号等于或高于请求中的序号,则直接向请求节点返回 *GROUP_KEY* 消息.

通过上述描述可以看出,DGR 算法中每个节点的密钥更新过程完全相同,都需要首先获取 k 个以上的邻居节点标识,而后通过发送部分密钥更新请求,并通过合成大于 k 个部分组密钥完成组密钥更新,因此算法的通信代价较高.

2.2 CDGR算法

针对 DGR 算法局部通信代价较高的问题,CDGR 算法在密钥更新过程中动态生成多个组密钥更新簇.每个组密钥更新簇包含一个簇首和多个簇成员节点,其中簇首节点为发起组密钥更新簇生成过程的节点.在密钥更新过程中,簇首节点通过向邻居节点广播 *NEIGBOR_FIND* 消息生成组密钥更新簇,然后向邻居节点发送 *PARTKEY_REQ* 请求.当簇首节点接收到 K–1 个 *PART_KEY* 响应以后,利用组密钥生成算法合成 TEK,然后通过安全信道点播发送给簇成员节点.除非簇首节点点播新的组密钥超时,否则簇成员节点不需要获取邻居节点标识及向其邻居节点请求部分组密钥.因此,CDGR 算法有效地降低了组密钥更新的通信代价.CDGR 算法的节点状态和行为的有限状态机如图 2 所示.

初始状态(init state).同 DGR 算法,簇首标识 *clusterhead* 为空.

等待更新状态(wait state).同 DGR 算法.其中当 T_e 超时以后,节点进入更新簇生成状态而非邻居发现状态.如果节点接收到 *NEIGBOR_FIND* 消息,则进入簇成员状态.

更新簇生成状态(clustering state).节点进入更新簇生成状态以后,首先将节点标识赋给 *clusterhead*,而后向其 *Rekey_Round* 跳范围内的邻居节点广播 *NEIGBOR_FIND* 请求,其中包含节点的组成员资格证书以及组密钥种子序号 i+1,然后等待邻居节点响应 *NEIGBOR* 消息.

启动计时器 T_n,该计时器用于控制节点等待响应消息的时间.当 T_n 超时以后,节点仍未获得 k–1 个邻居节点标识,则将 *Rekey_Round* 加 1,重复上述过程.当 *Rekey_Round* 超过其上限时,邻居发现过程失败,进入失败状态.

如果节点接收到 *NEIGBOR* 消息,则将其中的标识加入到邻居节点列表和簇成员节点列表中;如果接收到 *NEIGBOR_UNJOIN* 消息,则仅将其中的节点标识加入到邻居节点列表中.如果邻居节点的数量达到 k–1,则 T_n 超时后,节点进入簇首状态.

若节点接收到 *GROUP_KEY* 消息,则当 T_n 超时后直接进入组密钥生成状态.

簇成员状态(cluster member state).处于等待更新状态的节点在接收到 *NEIGBOR_FIND* 消息后,进入簇成员状态.节点首先启动 T_c 计时器,该计数器用于避免节点始终处于簇成员节点.再将 *NEIGBOR_FIND* 消息中的节点标识赋 *clusterhead*,并将该标识加入响应节点集合中,以避免在同一次密钥更新过程中多次响应同一个节

点请求,然后向邻居节点返回 *NEIGBOR* 消息.当 T_c 超时后,将其 *Rekey_Round* 置为上限,进入更新簇生成状态.

若节点接收到 *GROUP_KEY* 消息,则直接进入组密钥生成状态.

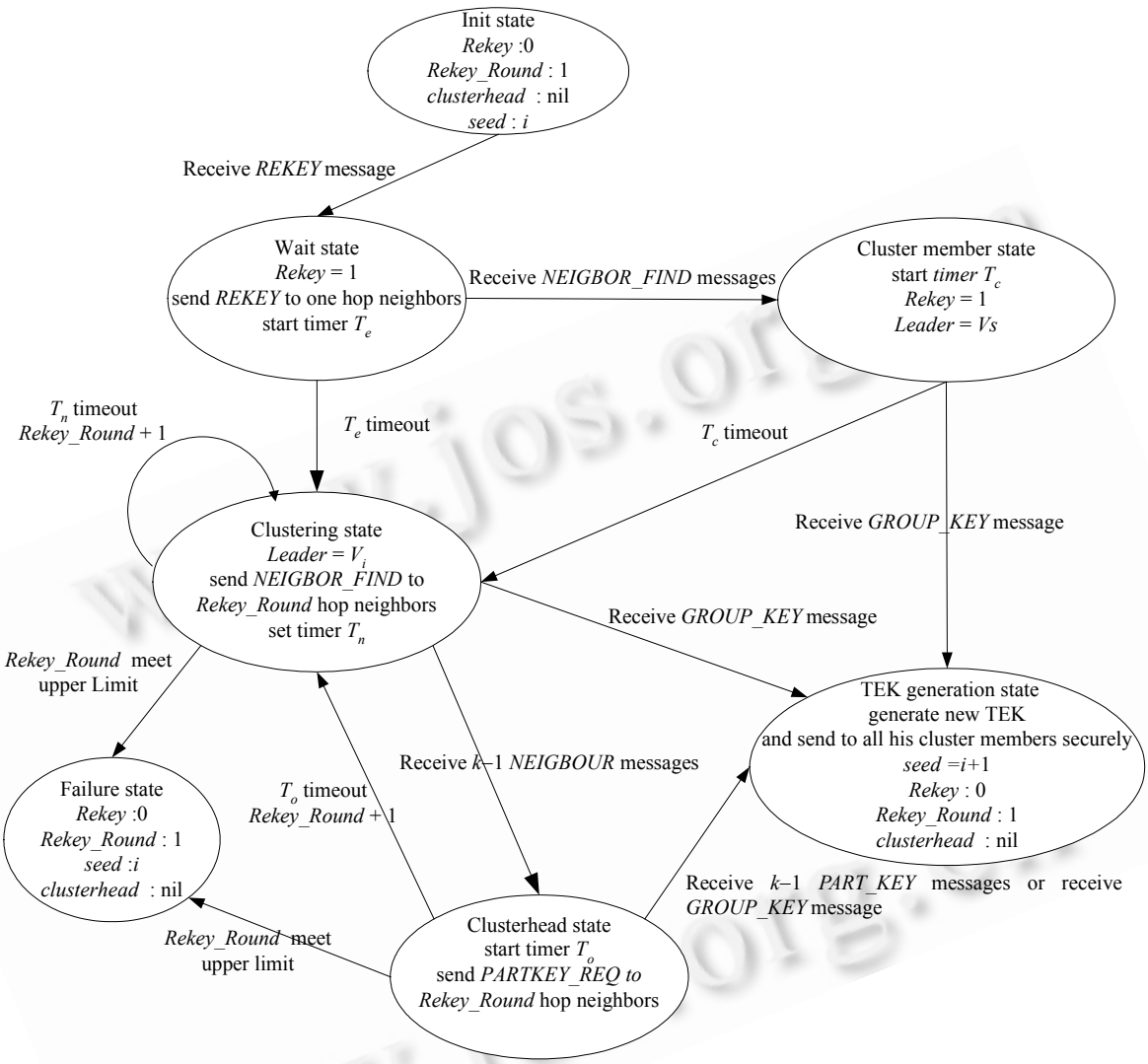

Fig.2 Finite state machine diagram of CDGR algorithm

图 2 CDGR 算法的有限状态机

簇首状态(clusterhead state).节点进入簇首状态,向其 *Rekey_Round* 跳范围内的邻居节点广播 *PARTKEY_REQ* 请求,其中包含节点的组成员资格证书、邻居节点标识以及组密钥种子 i+1,并用节点的私钥签名保证消息的完整性.同时,启动组密钥更新计时器 T_o,控制节点等待响应的时间.

如果接收到 *PART_KEY* 消息,则将接收到的部分密钥数量加 1.当 T_o 超时后,节点仍未获得 k–1 个部分组密钥,则返回到更新簇生成状态,并将 *Rekey_Round* 加 1.当 *Rekey_Round* 超过失败次数的上限后,该节点的密钥更新失败.当节点获得 k–1 个部分组密钥,则进入组密钥生成状态.若节点接收到 *GROUP_KEY* 消息,也直接进入组密钥生成状态.

组密钥生成状态(TEK generation state).如果节点接收到 *GROUP_KEY* 消息,则直接获得组密钥,否则节点利用获得的 k–1 个部分组密钥,加上自己生成的部分组密钥,根据组密钥生成算法生成新的组通信密钥 $HASH((g(i+1))_{SK})$,而后向其所有簇成员节点发送 *GROUP_KEY* 消息,其中包含更新后的组密钥,再将

Rekey_Round 置为 1,*Rekey* 置为 0,*clusterhead* 置为空,恢复到初始状态,完成此次组密钥更新.

失败状态(failure state).同 DGR 算法,并将 *clusterhead* 置为空.

处于簇成员状态、更新簇生成状态或簇首状态的节点,在接收到 *NEIGBOR_FIND* 消息后,首先验证消息的完整性,如果节点为合法的组成员,并且节点不在响应节点集合中,则向发送节点返回 *NEIGBOR_UNJOIN* 消息.

处于簇成员状态、更新簇生成状态或簇首状态的节点,在接收到 *PARTKEY_REQ* 消息后,首先验证消息的完整性,然后检测请求发送节点是否为合法的组成员,如果节点在消息的邻居节点标识列表中,则向该节点返回 *PART_KEY* 消息.

当节点处于组密钥生成状态时,在接收到 *PARTKEY_REQ* 或 *NEIGBOR_FIND* 消息后,如果其组密钥生成种子的序号等于或高于请求中的序号,则直接向请求节点返回 *GROUP_KEY* 消息.

算法中采用了多个计时器,其中计时器 T_e 在组密钥更新时随机选取,用于避免无线信道的拥塞,其最大值与信道的容量、带宽有关.计时器 T_n 和 T_o 的取值与无线信道的带宽和节点的响应速度有关,并随着 *Rekey_Round* 的增加而增大.*Rekey_Round* 的上限为 *k*,保证在网络连通的情况下,*Rekey_Round* 跳范围内的节点数量不小于门限 *k*.而 CDGR 算法中的计时器 T_c 应大于簇首节点密钥更新的最大时间.

2.3 算法的分析与讨论

(1) 算法的复杂性

由于在密钥更新过程中,通信具有局部性,因此通信复杂性主要考虑消息的复杂性.在最优的情况下,每个节点的邻居节点均大于或等于 *k*,在此仅考虑等于 *k* 的情况.在 DGR 算法中,节点仅需要与单跳节点通信即可完成组密钥更新过程.CDGR 算法还需要组密钥更新簇均匀分布才可获得最优通信性能.在最坏的情况下,网络拓扑为链状结构.节点邻居发现过程平均需要$\lceil k/2 \rceil$次广播,且广播的跳数不断增大,才可获得 *k*–1 个组成员节点标识,对于 CDGR 算法,每个节点都是簇首节点.需要强调的是,此种网络拓扑出现的概率很低,并且 CDGR 算法中节点随机选择处于等待更新的时间,每个节点都是簇首节点的概率几乎为 0.假设网络的平均邻居节点的数量为 $r(r<k)$,CDGR 算法中簇成员节点的平均数量为 $m(m<k)$,算法的通信复杂性见表 1.

Table 1 Message complexity analysis of the two algorithms

表 1 两算法的通信复杂性分析

	Message complexity			
	DGR		CDGR	
	Broadcast	Unicast	Broadcast	Unicast
Best case	$O(N)$	$O(N\cdot k)$	$O(N/k)$	$O(N)$
Worst case	$O(N\cdot k)$	$O(N^2)$	$O(N\cdot k)$	$O(N^2)$
Common case	$O(N\cdot\log,k)$	$O(N\cdot k)$	$O(N\cdot\log,k/m)$	$O(N\cdot k/m)$

通过分析可以看出,CDGR 算法的通信复杂性在平均情况下均优于 DGR 算法,其优化的程度与簇成员节点的数量有密切的关系.在最坏情况下,CDGR 算法的通信复杂性与 DGR 算法相当.

算法的运算复杂性需要考虑加/解密、签名和验证的代价.在算法中,签名和验证的代价与节点接收和发送的消息数量有关,加/解密操作用于保证部分组密钥的私密性.因此,签名和验证的代价可参见通信复杂性.对于 DGR 算法,加/解密的代价为 $O(N\cdot k)$次,而在 CDGR 算法中,簇首节点的加/解密代价为 $O(k)$,簇成员节点的代价为 $O(1)$.因此,CDGR 算法的运算复杂性也优于 DGR 算法.

(2) 算法的安全性

在算法执行过程中,由于通信量较少,可利用非对称加密机制保证通信的私密性和完整性.根据假设,组中所有节点的时钟同步,可防止重放攻击.

在组密钥的合成过程中,部分组密钥的正确性可以通过可验证秘密共享机制验证[14].当 *k*–1 个节点退出时,需要更新成员的共享密钥以防止退出组成员的合谋.共享密钥更新机制[15]能够更新所有节点的共享密钥,而不改变组的私钥.当共享密钥更新以后,即使退出组成员节点超过门限,也不能合成组的私钥,从而避免了门限个退出成员节点的合谋问题.

(3) 算法的正确性和完备性

通过算法的状态转换图可以看出,在网络密度较小、邻居节点数较少的情况下,邻居发现过程通过不断增加邻居发现广播的 TTL 值扩大邻居发现的范围,直到发现门限个邻居节点为止.在邻居发现过程中,节点可利用与 DSR 协议类似的机制建立其与多跳邻居节点的路由,也可以利用已有的路由协议实现节点间的多跳通信.在 CDGR 算法中,由于节点的移动性,处于簇首状态的节点在向其成员节点发送更新组密钥时,需要路由协议支持.

通过一阶时序逻辑可以证明在网络未分割并且连接可靠的情况下,上述算法的正确性和完备性.考虑到移动自组网络的连接可靠性和节点移动等因素,通过模拟可以看出,算法能够最大限度地保证每个组成员节点 TEK 的一致性.当 TEK 不一致时,DGKMF 的基于序列的组密钥生成机制能够迅速恢复组密钥的一致性.

当多个节点加入或退出时,将同时触发多个组密钥更新过程.由于 TEK 对应的组密钥更新种子具有单调性,当多个更新过程并发时,组成员选择最新的组密钥更新种子更新组密钥.在算法执行过程中,处于中间状态的节点在接收到新的组密钥更新消息后,恢复算法初始状态重新执行.

3 模拟实验

在实际环境下,移动自组网络的连通性以及链路的可靠性难以保证.因此,为了验证算法的有效性,本文采用 ns2 网络模拟器比较了 DGR 算法和 CDGR 算法在节点加入、退出时 TEK 更新的成功率和更新延迟,并比较了它与组密钥管理协议 CKD,GDH v2 以及 BD 协议的性能.

模拟环境的链路可靠性为 90%,节点的平均速度为 10m/s,节点停等时间为 5s.网络规模以节点数量表征,节要点数量是 30~100,幅度 10 均匀变化.模拟空间随网络节点数量变化,以保证网络的连通性,协议模拟时间为 1 500s.

由于组密钥更新算法的性能与网络中节点的密度有密切的关系,我们将网络中组成员的平均邻居数量称为网络的连通强度,用 D 标识.当模拟时,D=5,即平均每个节点的邻居数量为 5.

3.1 模拟结果

根据上一节所述,网络连通强度为 5,网络中节点的数量分别为 30,40,…,100,链路可靠性为 90%,节点的最大移动速度为 5m/s,节点停等时间为 5s.如图 3~图 6 所示分别为门限 k=5 时节点退出、节点加入时的模拟结果.

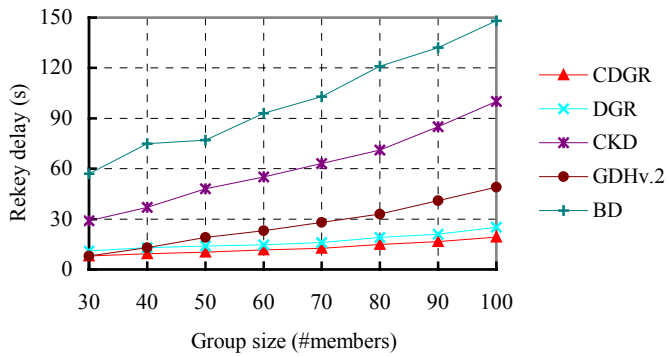

Fig.3 Node leave—TEK rekey delay

图 3 节点退出时 TEK 更新延迟

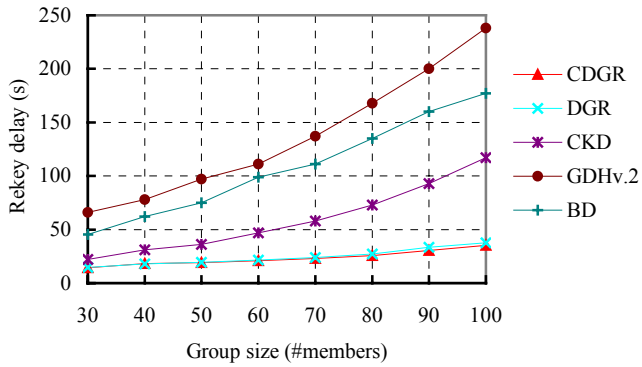

Fig.4 Node join—TEK rekey delay

图 4 节点加入时 TEK 更新延迟

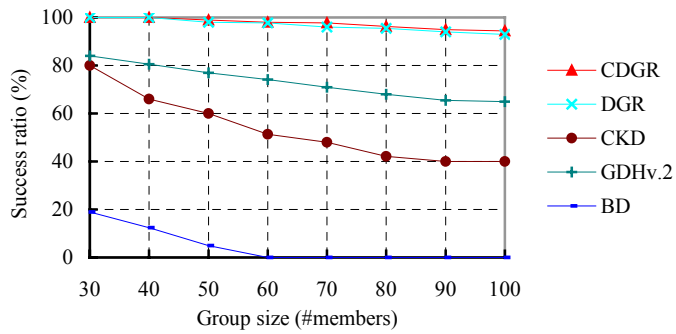

Fig.5 Node leave—TEK rekey success ratio

图 5 节点退出时 TEK 更新成功率

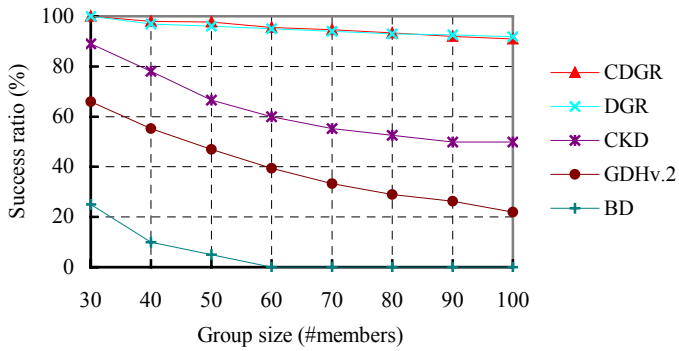

Fig.6 Node join—TEK rekey success ratio

图 6 节点加入时 TEK 更新成功率

从上述模拟实验结果可以看出,DGR 算法和 CDGR 算法依赖于局部通信更新组密钥,延迟在 30s 左右,随着网络规模的扩大略有提高.由于组成员加入时,DGR,CDGR 算法的密钥更新过程类似,均需要为新加入的节点分发共享密钥,因此算法的性能相当.在节点退出时,CDGR 算法由于在组密钥更新时动态生成更新簇,利用簇首节点局部生成并分发组密钥,减少了本地通信代价,因此 CDGR 算法的延迟低于 DGR 算法.在节点加入、退出时,DGR 算法和 CDGR 算法的 TEK 更新成功率均接近于 100%.

在相同的模拟条件下,CKD,GDH v2 以及 BD 等[4,13]组密钥管理协议和算法的更新延迟平均在 80s 左右,组密钥更新的成功率均低于 90%,并且随着组的规模增加性能急剧下降.其密钥更新成功率和延迟均比 DGR 算法和 CDGR 算法要差.

4 结束语

基于分布式组密钥管理框架,本文提出了两种移动自组网络组密钥更新算法:DGR 算法和 CDGR 算法.这两种算法都能够根据局部密钥信息更新组密钥,非常适合拓扑结构变化频繁、多跳连接不可靠且带宽有限的移动自组网络.CDGR 算法在组密钥更新过程中动态生成组密钥更新簇,进一步减少了节点退出时组密钥更新的通信代价,因而组密钥更新的延迟更低.模拟结果验证了上述算法在移动自组网络中的有效性.

当节点加入或退出过于频繁时,将导致组密钥过度更新,从而造成网络拥塞.定时更新是防止组密钥更新过于频繁的有效方法.在今后的工作中,我们将就如何利用定时更新机制解决移动自组网络中组密钥过度更新的问题做进一步的研究.

References:

[1] Perkins CE. Ad Hoc Networking. London: Addison-Wesley, 2001.

[2] Zhou LD, Hass ZJ. Securing in ad-hoc networks. IEEE Networks, 1999,13(6):24~30.

[3] Stajano F, Anderson R. The resurrecting duckling: Security issues for ad-hoc wireless networks. In: Proc. of the 7th Int'l Workshop on Security Protocols. Berlin: Springler-Verlag, 1999. http://www.cl.ac.uk/~fms27/duckling/duckling.htm

[4] Griffin S, DeCleene B, Dondeti L, Flynn R, Kiwior D, Olbert A. Hierarchical key management for mobile multicast members. Technical Report, Northrop Grumman Information Technology, 2002.

[5] Basagni S, Herrin K, Bruschi D, Rosti E. Secure pebblenets. In: Proc. of the 2001 ACM Int'l Symp. on Mobile Ad Hoc Networking and Computing. New York: ACM Press, 2001. 156~163.

[6] Hardjono T, Tsudik G. IP Multicast security: Issues and directions. Julho-Agosto: Annales de Telecom, 2000. 324~334. http://www.nr.no/~abie/Multicasting.htm

[7] Moyer MJ, Rao JR, Rohatgi P. A survey of security issues in multicast communications. IEEE Network Magazine, 1999,13(1): 12~23.

[8] Wallner D, Harder E, Agee R. Key Management for Multicast: Issues and Architectures. Request for Comments (Informational) 2627, Internet Engineering Task Force, 1999. http://www.faqs.org/rfcs/rfc2627.html

[9] Kuang X, Hu H. Performance analysis of group key management protocol in mobile ad hoc network. Computer Engineering and Science, 2004,26(3):4~7 (in Chinese with English abstract).

[10] Hietalahti M. Efficient key agreement for ad-hoc networks [MS. Thesis]. Espoo: Department of Computer Science and Engineering, Helsinki University of Technology, 2001.

[11] Zhang C, DeCleene B, Kurose J, Towsley D. Comparison of inter-area rekeying algorithms for secure wireless group communications. Performance Evaluation, 2002,49(1-4):1~20.

[12] Carman DW. Constraints and approaches for distributed sensor network security. Technical Report, #00-010, NAI Labs, 2000.

[13] Kuang X, Lu X. Secure group communications for mobile ad-hoc networks. Journal of Computer Research and Development, 2004,41(4):704~710 (in Chinese with English abstract).

[14] Schoenmakers B. A simple publicly verifiable secret sharing scheme and its application to electronic voting. In: Advances in Cryptology—CRYPTO'99. Vol. 1666 of Lecture Notes in Computer Science, Berlin: Springer-Verlag, 1999. 148~164.

[15] Herzberg A, Jarecki S, Krawczyk H, Yung M. Proactive secret sharing or: How to cope with perpetual leakage. In: Advances in Cryptology—CRYPTO'95. Vol. 963 of Lecture Notes in Computer Science, London: Springer-Verlag, 1995. 339~352.

附中文参考文献:

[9] 况晓辉,张念,胡华平.移动自组网络环境下组密钥管理协议性能分析.计算机工程与科学,2004,26(3):4~7.

[13] 况晓辉,胡华平,卢锡城.移动自组网络的组密钥管理框架.计算机研究与发展,2004,41(4):704~710.