适用于大规模群组 Ad hoc 网络的组密钥管理方案*

曹　帅，张串绒，宋程远

（空军工程大学 电讯工程学院，西安 710077）

摘　要：提出一种适于大规模群组 Ad hoc 网络的组密钥管理方案。该方案基于地理位置建立了分层分组的网络模型，将大部分消息的传输距离限制在一个小组范围内，降低了通信开销，结合密钥分发和密钥协商的各自优点来生成、分发和更新组密钥，在避免单点失效的同时提高了效率；同时，椭圆曲线上的离散对数难题保证了方案的安全性。分析表明方案具有较强的可用性。

关键词：Ad hoc 网络；大规模群组；组密钥管理；密钥协商；密钥分发

中图分类号：TP393　　**文献标志码**：A　　**文章编号**：1001-3695(2012)04-1420-04

doi:10.3969/j.issn.1001-3695.2012.04.061

Group key management scheme for large mobile Ad hoc network

CAO Shuai , ZHANG Chuan-rong, SONG Cheng-yuan

(*Telecommunication Engineering Institute*, *Air Force Engineering University*, *Xi'an* 710077, *China*)

Abstract: This paper proposed a new hierarchical group key management scheme for large Ad hoc network. In the scheme, divided group based on geographical location into several sub-groups, that reduced communication overhead for limiting message transmission distance within a sub-group. It adopted the combination of key agreement and distribution to generate, distribute and update group key, that made it avoid single point of failure while improving efficiency. Based security of the scheme on solving elliptic curve discrete logarithm problem. Through analysis, the scheme provides higher availability.

Key words: Ad hoc network; large group; group key management; key agreement; key distribution

Ad hoc 网络是一种多跳无线移动网络，它不依赖固定的基础设施，具有自组织和分布式等特点，在军事和民用领域有广泛的应用。然而其动态变化的网络拓扑、受限的无线传输带宽和移动终端的局限性等固有属性，决定它比传统网络更容易受到各种安全威胁。由于 Ad hoc 网络中的很多网络功能，包括路由、邻居发现、密钥分发和拓扑控制都是面向群组的，因此为安全组通信服务的组密钥管理成为 Ad hoc 网络急需解决的安全问题之一。现有成熟的传统网络组密钥管理方案主要可以分为集中式的组密钥管理[1,2]、分布式的组密钥管理[3~5]和分散式的组密钥管理[6~8]三类。然而依据 Ad hoc 网络的受限性和开放性，对于大规模群组的组密钥管理，如果采用完全集中式组密钥分发方案，存在单点失效的问题；如果采用完全分布式协商组密钥的方式，虽然具有较高的安全性，但计算量和通信量较大，而且对于整个群组来说，消息的传输范围比较大，需要 Ad hoc 网络节点的多次多跳转发，容易造成全网通信量增大导致的碰撞、拥塞和延迟，严重影响组密钥的分发和更新效率。同时，Ad hoc 网络中的节点是动态移动的，若一个节点加入或者退出都需要更新组密钥，就会产生“1 影响 n”的不利因素，导致计算和通信开销的增加。针对以上问题，本文提出了一种适于大规模群组的组密钥管理方案。

1　理论基础

椭圆曲线上的点乘运算通常是指，对某一点 P 计算其整数倍，比如 k 为整数。计算 kP，本章将详细介绍文献[9]提出的已知椭圆曲线上某点 P 的阶为素数 q 时，根据正整数 k 及其与 P 的点乘结果 Q，如何求解点 P 这种计算方法，其计算方法将被应用到本文的密钥协商中。

定理 1[10]　若群中元素的 P 阶是 q，则 $|P^k| = r/\gcd(k,q)$。

由于 $|P| = q$，则根据定理 1 $|Q| = |kP| = q/\gcd(k,q)$，其中 q 是素数，此时存在两种情况：a）当 $k = nq$ 时，$|Q| = 1$，n 是任意正整数，此时 Q 是无穷远点 O，P 可以是群内的任何一点；b）当 $k \neq nq$ 时，$|Q| = q$，此时 P 是群内确定的点。下面讨论这种情况下 P 的求解方法。

定理 2[11]　如果 k 和 q 互素，则存在唯一的整数 $x < q$ 满足 $kx \equiv 1 \pmod q$。

由于 k 和 q 互素，即 $\gcd(k,q) = 1$，则由定理 2 可知，必存在唯一的 $x < r$ 使 $kx \equiv 1 \pmod r$ 成立，也就是说 $kx - 1$ 是 q 的倍数，假设 $kx - 1 = qy$，其中 y 是正整数。$kx - 1 = qy$ 是一次不定方程，该方程有解的充要条件是 $\gcd(k,q) = 1$，而由已知条件 k 和 q 互素，所以该方程必有解。运用 Euclid 算法确定 x 的值。最后对等式 $Q = kP$ 两边同时作倍标量 $|G|$ 乘运算：

$$xQ = (kx)P = (qy + 1)P = qyP + P = P, \text{即 } P = xQ$$

2　网络模型

将网络中的节点基于地理位置信息分为多个子组，各个子组的通信范围为一跳或者两跳，每个子组中存在一个组长节

收稿日期：2011-08-19；**修回日期**：2011-10-25　**基金项目**：国家自然科学基金资助项目(60873233)

作者简介：曹帅(1987-)，男，甘肃平凉人，硕士研究生，主要研究方向为信息安全、移动自组网络安全(shuaics187@163.com)；张串绒(1965-)，女，陕西眉县人，教授，博士(后)，主要研究方向为密码学、信息安全；宋程远(1987-)，女，山东东营人，硕士研究生，主要研究方向为信息安全、传感器网络安全.

点,子组内存在参与子组密钥协商的节点和直接由组长节点分发子组密钥的节点,所有组长节点构成一个高一级的群组参与协商组密钥,如图 1 所示。

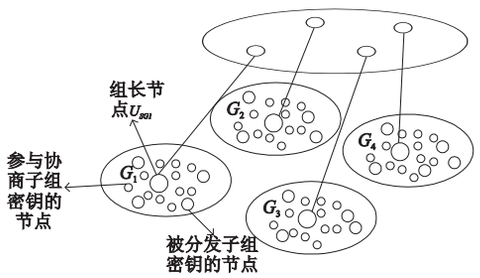

图1　大规模群组Ad hoc网络模型

3　方案描述

本方案由初始化阶段、组密钥生成和分发协议、组密钥更新协议三部分组成。

3.1　初始化阶段

3.1.1　子组的生成

假设网络中每个节点有唯一的身份标志符 ID_i,而且每个节点持有 GPS 设备,可以随时获取自身当前位置信息 l_i。首先,每个节点广播自己的身份标志符 ID 和其位置信息 l,收到其他节点广播信息后,计算 $|l_i - l_j|$,把 $|l_i - l_j| \leqslant R$ 不等式成立的节点记为其邻居节点,其中 R 表示节点的两跳通信距离;其次,所有节点交流其两跳范围内的邻居节点数量,选择邻居数量最大的节点为组长节点;最后,组长节点广播消息“I am a leader”,其他节点加入该子组则广播申请消息“I am a member”,组长节点收到子组其他成员的加入请求后,构造子组成员列表 List$M$,并广播给子组其他成员。子组形成后,所有的组长节点互相交流,得到整个网络中组长节点列表 List L。

3.1.2　系统参数

假设群组中的节点数为 n,被分成 k 个子组,每个子组内的节点数为 m。第 i 个子组内参与协商子组密钥的节点数为 t,节点集合为 $B = \{U_{i1}, U_{i2}, \cdots, U_{it}\}$,参与子组密钥分发的节点数为 $m - t$,节点集合为 $A = \{U_{i,t+1}, U_{i,t+2}, \cdots, U_{i,m}\}$,将第 i 个子组的组长节点记做 U_{sc_i},组长节点的集合为 $C = \{U_{sc_1}, U_{sc_2}, \cdots, U_{sc_k}\}$。设 E 为定义在有限域 Z_q^* 上的椭圆曲线,G_1 是 E 上阶为 q 的循环加法群,P 为曲线 E 上的基点,安全的 hash 函数 $H: \{0,1\}^* \rightarrow Z_P^*$,$H_1: G_1 \rightarrow \{0,1\}^*$。$(pk_{i,j}, sk_{i,j})$ 为网络中节点 U_{ij} 的公私钥对,$k_{i,j}(t+1 \leqslant j \leqslant m)$ 为第 i 子组内参与子组密钥分发的节点和组长节点的共享密钥。

3.2　组密钥生成和分发协议

第一轮:集合 $B = \{U_{i,1}, U_{i,2}, \cdots, U_{i,t}\}$ 中的每个节点选择随机数 $r_{ij} \in (1,q)(1 \leqslant j \leqslant t)$,计算 $z_{ij} = r_{ij}P$,在子组内广播消息 $m_{ij} = ID_{ij} \| z_{ij}$。

第二轮:组长节点 U_{sc_i} 收到广播消息 m_{ij} 后,首先选择随机数 $r_{sc_i} \in (1,q)$,计算 $T_{i,j} = r_{sc_i} z_{ij}$、子组密钥 $k_{sc_i} = r_{sc_i}P + \sum_{j=1}^{t} r_{ij}P$ 和过滤多项式 $f(x) = A(x)k_{sc_i}$,其中 $A(x) = (\prod_{c=t+1}^{m}(x - H(k_{i,c})) + 1)$ 为混淆多项式,并整理成标准形式 $f(x) = \sum_{i=0}^{n-t+1} a_i x^i$;然后在其子组范围内广播消息 $\{T_{i,1} \| T_{i,2} \| \cdots T_{i,t} \| ID_{sc_i} \| f(x)\}$;最后选择随机数 $m_i \in (1,q)$,计算 $R_i = H_1(k_{sc_i})$ 和 $A_i = R_i P$,在整个群组范围内广播消息 $m_{sc_i} = ID_{sc_i} \| A_i$。

第三轮:a)集合 B 中的每个节点 $U_{ij}\{1 \leqslant j \leqslant t\}$ 收到组长广播的消息 $\{T_{i,1} \| T_{i,2} \| \cdots T_{i,t} \| ID_{sc_i} \| f(x)\}$ 和其他节点广播的消息 z_{ij} 之后,根据定理 2 从 $T_{i,j}$ 中计算出 $r_{sc_i}P$,然后计算子组密钥 $k_{sc_i} = r_{sc_i}P + \sum_{j=1}^{t} z_{ij}$;b)集合 A 中的节点 $U_{ic}\{t+1 \leqslant c \leqslant m\}$ 得到过滤多项式 $f(x)$ 后,将它与组长节点的共享密钥的散列值 $H(k_{i,c})$ 代入多项式 $f(x) = A(x)k_{sc_i}$,计算得到子组密钥 $k_{sc_i} = f(H(k_{i,c}))$。c)其他组长节点 U_{sc_j} 收到消息 $m_{sc_i} = ID_{sc_i} \| A_i (1 \leqslant i \leqslant k, i \neq j)$ 后,组播消息 $m'_{sc_j} = ID_{sc_j} \| W_{j,1} \| W_{j,2} \cdots \| W_{j,j-1} \| W_{j,j+1} \cdots \| W_{j,k} (1 \leqslant j \leqslant k, j \neq i)$ 给其他组长节点,其中 $W_{j,i} = m_j A_i \{1 \leqslant i \leqslant k, i \neq j\}$。

第四轮:组长节点 U_{sc_i} 收到广播消息 $m'_{sc_j}(1 \leqslant j \leqslant k, j \neq i)$ 后,根据定理 2 计算出其他组长节点的 $m_j P(1 \leqslant j \leqslant k, j \neq i)$,就可以计算出组密钥 $K = \sum_{i=1}^{m} m_i P$。用它们的子组密钥 k_{sc_i} 加密组密钥 K 得到 $E_{k_{sc_i}}(K)$,并组播给子组内其他节点。

组密钥的计算:子组内的所有节点收到组播消息 $E_{k_{sc_i}}(K)$ 后,可以用它们拥有的子组密钥 k_{sc_i} 解密 $E_{k_{sc_i}}(K)$,最终得到组密钥 K。

3.3　组密钥更新协议

3.3.1　节点加入密钥更新协议

1)单个普通节点加入　新加入节点广播加入请求消息包括其身份信息 ID_{m+1} 和它的位置信息 l_{m+1}。邻居节点收到它广播的加入请求后传递给它所属的组长节点,该组长节点发送响应请求及其位置信息 l_j 给新节点。新节点计算 $|l_{m+1} - l_j|$,选择离自己比较近的组长节点加入并发送响应消息,组长节点收到响应消息后与其他组长节点交流数据,判断该节点是否也为其他组长节点发送了响应消息。若多于两个的组长节点收到其响应消息,则拒绝其加入,并列入不诚实节点列表;若只有一个组长节点收到响应消息,则组长节点更新子组成员列表 List M,并将其列入参与分发子组密钥的集合 A 中,进行该子组密钥更新操作。在此假设新节点为 $U_{i,m+1}$,加入子组 SC_i。

子组 SC_i 的组长节点 U_{sc_i} 重新选择新的随机数 $r'_{sc_i}, m'_i \in (1,q)$,计算新的子组密钥 $k'_{sc_i} = r'_{sc_i}P + \sum_{j=1}^{t} r_{ij}P$ 和新的组密钥 $K' = \sum_{j=1}^{k} m_j P + m'_i P$,用节点 $U_{i,m+1}$ 的公钥加密新的子组密钥 k'_{sc_i} 单播消息 $E_{pk_{i,m+1}}(k'_{sc_i})$ 给新节点 $U_{i,m+1}$,在子组范围内组播消息 $E_{k_{sc_i}}(k'_{sc_i}) \| E_{k'_{sc_i}}(K')$,组播消息 $\{ID_{sc_i} \| W'_{i,1} \| W'_{i,2} \cdots \| W'_{i,i-1} \| W'_{i,i+1} \cdots \| W'_{i,k}\}$,其中 $W'_{i,j} = m'_i A_j = m'_i a_j P(1 \leqslant j \leqslant t, j \neq i)$ 给其他组长节点。其他组长节点 U_{sc_j} 收到广播消息 $W'_{i,j}$ 后,根据定理 2 计算出新的 $m'_i P$,进而计算出新的组密钥 $K' = \sum_{j=1, j \neq i}^{k} m_j P + m'_i P$;然后组播 $E_{k_{sc_j}}(K')$ 给它们的子组内的节点。

组密钥计算:新加入的节点 $U_{i,m+1}$ 收到单播消息 $E_{s_{i,m+1}}(k'_{sc_i})$ 后,利用自己的私钥解密 $E_{pk_{i,m+1}}(k'_{sc_i})$,得到新的子组密钥 k'_{sc_i};子组 SC_i 内其他节点收到组播消息 $E_{k_{sc_i}}(k'_{sc_i})$,可以用旧的子组密钥 k_{sc_i} 解密得到新的子组密钥 k'_{sc_i},然后解密 $E_{k'_{sc_i}}(K')$ 得到新组密钥 K';同理其他子组的节点同样可以得到新组密钥。

2)整个子组加入　假设新加入的子组为 SC_{k+1},其组长节点为 $U_{sc_{k+1}}$,该子组的子组密钥为 $k_{sc_{k+1}}$。网络中原有组长节点随机选择一个组长节点作为密钥更新发起者,假设为 U_{sc_r}。$U_{sc_{k+1}}$ 随机选择 $m_{k+1} \in (1,q)$,计算 $R_{k+1} = H_1(k_{sc_{k+1}})$,$A_{k+1} = R_{k+1}P$,单播消息 $A_{k+1} \| ID_{k+1}$ 给组密钥更新发起者 U_{sc_r}。U_{sc_r} 计算 $H_1(K) = m'_r$,$W_{r,k+1} = m'_r R_{k+1} P$,发送单播消息 $\{A_r \| ID_r \| W_{r,k+1}\}$ 给 $U_{sc_{k+1}}$。组长节点 $U_{sc_{k+1}}$ 收到消息 $\{A_r \| ID_r \| W_{r,k+1}\}$

后,根据定理 2 从 $W_{r,k+1}$ 中计算出 m'_rP,之后计算新组密钥 $K'=m_{k+1}P+H_1(K)$。计算 $W_{k+1,r}=m_{k+1}R_rP$,发送单播消息 $\{\mathrm{ID}_{k+1}\parallel W_r\}$ 给发起者 U_{sc_r}。发起者 U_{sc_r} 收到消息 $\{\mathrm{ID}_{k+1}\parallel W_r\}$ 后,根据定理 2 计算出 $m_{k+1}P$,然后计算新组密钥 $K'=m_{k+1}P+H_1(K)$,并在全局内广播消息 $E_K(K')$。

3.3.2 节点离开密钥更新协议

1)子组普通节点离开 假设子组 SC_i 内的普通节点 $U_{i,d}$ 离开,其中 $U_{i,d}$ 为参与子组密钥协商的节点。组长节点 U_{sc_i} 重新选择随机数 r'_{sc_i},计算 $T'_{ij}=r'_{sc_i}z_{ij}$,$j\neq d$ 和过滤多项式 $f'(x)=A(x)k'_{sc_i}$,整理为标准式,在子组内广播消息 $\{T'_{i,1}\parallel T'_{i,2}\parallel\cdots T'_{i,d-1}\parallel T'_{i,d+1}\cdots T'_{i,t}\parallel \mathrm{ID}_{sc_i}\parallel f'(x)\}$。子组中参与子组密钥协商的节点 $U_{i,j}$ 收到消息 $T_{i,j}$ 之后,根据定理 2 可以计算出新的子组密钥 $k'_{sc_i}=\sum_{j=1,j\neq d}^{t}r_{ij}P+r'_{sc_i}P$,子组内参与子组密钥分发的节点也可以计算出新的子组密钥 $k'_{sc_i}=f'(H(k_{i,x}))$。组密钥的更新过程与节点的加入组密钥更新过程一样。

2)组长节点离开 如果一个组长节点主动离开组播组,那么向其子组成员以及网络中其他的组长节点发送一个离开通知消息,并指定子组内某一个成员作为新的组长节点。由于新的组长节点之前已经收到过其他普通节点发送的密钥的一部分,因此子组密钥更新过程与子组普通节点离开事件子组密钥更新过程一样,组密钥更新过程与节点加入组密钥更新过程一样。另外,当组长节点准备离开时,若发现子组内成员数目较少时,可以向邻居组长节点发出请求,将两个子组合并。

3.3.3 子组间跨区漫游

成员可能离开目前所属的子组但并不离开组播组,只是漫游进入组播组内其他子组,这种情况称为子组间的漫游。虽然该节点仍然是组播组的成员,但是为了防止在后续的密钥更新中该节点得到其不该得到的组密钥,需要更新其离开子组的子组密钥。例如在 t 时刻节点离开子组 SC_1,加入子组 SC_2。若不更新子组 SC_1 的子组密钥,在 $t+3$ 时刻,若节点离开组播组,而子组 SC_1 并没有节点离开或加入,子组密钥还是该节点离开时的子组密钥。显然,节点可以利用所知道的子组 SC_1 的子组密钥,得到它离开组播组 $t+3$ 时刻之后的组密钥。因此需要更新子组 SC_1 的子组密钥,更新过程与节点离开时子组密钥更新过程一样。

3.3.4 子组分裂

当子组的成员过多时,为了降低延迟,需要将子组分裂成两个子组,由原子组的组长节点选择一个新的节点为另一个子组的组长节点,由这两个组长节点分别发起子组密钥的更新,其更新过程与节点离开时的密钥更新过程类似。

3.3.5 子组合并

当两个子组的成员较少时,需要合并这两个子组。密钥更新过程如下:选择其中一个子组的组长节点为合并组的组长节点,首先这两个组长节点交流各自的成员列表;然后用对方的公私钥加密各自的子组密钥发送给对方,这两个组长节点收到该消息后解密并计算合并组的子组密钥为这两个子组密钥的 hash 值;最后用原有的子组密钥加密新子组密钥分发给组中节点。例如:假设子组 SC_i 和 SC_j 两个子组合并,选择子组 SC_j 的组长节点 U_{sc_j} 为新子组的组长节点。两个子组的组长节点 U_{sc_i} 和 U_{sc_j} 互相发送 $E_{sk_{sc_i}}(k_{sc_i})$ 和 $E_{sk_{sc_j}}(k_{sc_j})$ 给对方,U_{sc_i} 和 U_{sc_j} 收到并解密,计算合并后子组的子组密钥 $k=H(k_{sc_i}/\!/k_{sc_j})$。最后 U_{sc_i} 用原有密钥 k_{sc_i} 加密新子组密钥组播消息 $E_{k_{sc_i}}(k)$ 给子组 SC_i 中的节点;同理节点 U_{sc_j} 也可以组播消息 $E_{k_{sc_j}}(k)$ 给 SC_j 中的节点,这两个子组内的节点解密消息后可以得到新的子组密钥 k。

4 方案分析

4.1 正确性分析

定理 3 在整个组密钥生成和分发过程中,只要节点计算正确,都可以确定节点计算出来的组密钥是正确的。

证明 依据定理 2,参与协商子组密钥的节点根据广播消息 $T_{i,j}=r_{sc_i}z_{ij}=r_{sc_i}r_{ij}P=r_{ij}r_{sc_i}P$ 和自己选择的随机数 $r_{ij}\in(1,q)$,可以计算得到 $r_{sc_i}P$ 和子组密钥 $k_{sc_i}=r_{sc_i}P+\sum_{j=1}^{t}r_{ij}P$。

未参与密钥协商的子组节点依据自己和组管理者的共享密钥 $k_{i,j}(t+1<j<m)$ 可以从广播多项式 $f(x)=A(x)k_{sc_i}$ 中计算出正确的子组密钥 k_{sc_i},因为此时 $A(x)=1$。

同上的分析,在组密钥协商过程中,各个组长节点依据自己已知的 R_i 和广播信息 $W_i=m_jA_i=m_jR_iP$ 可以计算得到其他组长节点的 m_jP,进而计算出正确的组密钥 $K=\sum_{i=1}^{m}m_jP$。

各个子组的普通节点利用其所拥有的子组密钥 k_{sc_i} 解密 $E_{k_{sc_i}}(K)$ 就可以得到正确的组密钥 K。

密钥更新阶段的正确性与组密钥生成分发阶段的正确性分析一样,在此不再详述。

4.2 安全性分析

4.2.1 机密性

1)子组密钥协商和分发是安全的 子组密钥协商的安全性基于椭圆曲线上的离散对数困难问题。攻击者虽可以搭线窃听到公共信道传输的消息 $z_{ij}=r_{ij}P$ 和 $T_{i,j}=r_{sc_i}z_{ij}$,但它首先不能从广播的消息中计算出 r_{sc_i};其次作为非子组成员不拥有合法的随机数 r_{ij},也就不能根据定理 2 计算出 $r_{sc_i}P$,得不到 r_{sc_i} 或者 $r_{sc_i}P$,也就得不到子组密钥 $k_{sc_i}=r_{sc_i}P+\sum_{j=1}^{t}z_{ij}$。从过滤多项式 $f(x)=A(x)k_{sc_i}$,其中 $A(x)=(\prod_{c=t+1}^{m}(x-k_{i,c})+1)$ 的构造可以看出,只有合法的子组节点利用自己和组长节点的共享密钥才可以计算出正确的子组密钥,而对于非子组节点,由于不具有合法的共享密钥,计算出的 $A(x)\neq 1$ 是随机数,进而不能计算出正确的组密钥。

2)组密钥协商和分发是安全的 组密钥协商的安全性也是建立在椭圆曲线上的离散对数困难问题。首先攻击者从已知的消息 $W_{sc_i}=m_jA_i$ 和 $A_i=R_iP$ 中是不能计算出各个组长节点自己选择的秘密随机数 m_j;其次在不拥有子组密钥 k_{sc_i} 的情况下也就不能根据定理 2 计算出节点的 m_jP,组密钥分发都是用各个子组的子组密钥加密进行传输的,非组内节点没有合法子组密钥也就不能得到组密钥。

4.2.2 后向安全性

新加入节点想得到加入之前的组密钥,就需要得到旧子组密钥和组密钥中包含的旧随机数 r_{sc_i}、m_i 或者 $r_{sc_i}P$、m_iP。显然,节点加入组后虽能知道新的 $r'_{sc_i}P$、m'_iP,但节点拥有关于旧子组密钥以及旧组密钥的信息与加入之前拥有的信息是一样的,从上面对子组密钥协商和组密钥协商的安全性分析可知,新加入节点不能得到旧子组密钥,也就不能得到旧组密钥。因此,方案满足后向安全性。

4.2.3 前向安全性

无论是子组普通节点的离开还是子组组长节点的离开,想

得到离开之后的组密钥,首先需要知道新的子组密钥 k'_{sc_i},要知道 k'_{sc_i}就必须知道新子组密钥中含有的随机数 r'_{sc_i}或 $r'_{sc_i}P$。显然根据椭圆曲线上的离散对数难题,从广播的消息 T'_{ij}中不能计算出随机数 r'_{sc_i}或 $r'_{sc_i}P$,进而也就不能得到退出之后的子组密钥和组密钥。因此,方案满足前向安全性。

4.3　性能分析

下面从通信消息数、计算开销和存储开销三个方面将本方案与两种比较典型的集中式[1]和分布式组播密钥管理方案[5]进行比较分析,如表 1 ~3 所示。

表 1　组密钥生成和分发开销比较

方案	指标		
	通信消息数	计算量	存储量
LKH[1]	$n(U)$	…	组控制者:$2n-1$ 普通成员:$\log_2(n)+1$
TGDH[16]	$2n(B)$	$3h-3$(EXP)	$h+1$
本方案	$t(SB)$ $2k+1(B)$	组长节点:$(t+2)k$(MUL) 普通节点:1(MUL)	普通成员:2 组长成员:3

表 2　节点加入密钥更新开销比较

方案		指标	
		通信消息量	计算量
LKH[1]		$2\log_2 n(M),1(U)$	…
TGDH[5]		$3(B)$	3h − 3(EXP)
本方案	整个子组加入	$2(U),1(B)$	新加入组的组长节点:2(MUL) 组密钥更新发起组长节点:1(MUL)
	普通节点加入	$1(SB),1(U),2(B)$	加入节点所属的组长节点:k(MUL)

表 3　节点离开密钥更新开销比较

方案		指标	
		通信消息量	计算量
LKH[1]		$2\log_2 n(M)$	…
TGDH[5]		$1(B)$	$3h-3$(EXP)
本方案	组长节点离开	$1(SB),2(B)$	新的组长节点:$t+k-1$(MUL)
	普通节点离开	$1(SB),2(B)$	离开节点所属的组长节点:$t+k-1$(MUL)

表中 n 为整个群组的节点数;k 为网络中子组的个数;m 为子组中节点的个数;t 为子组中参与子组密钥协商的节点数数;B 为全局广播;SB 为子组内广播;U 为单播;MUL 为椭圆曲线上的点乘运算;EXP 为模指数运算;h 为逻辑密钥树的高度。其中在计算量的统计中,本文忽略了花费资源较少的多项式运算、hash 运算以及对称加/解密运算,因此表中 LKH 的计算量为 0。

从表 1 可看出,本方案在组密钥生成和分发阶段通信与计算开销较集中式方案 LKH 有所增加,但是方案中普通节点只需 1 次椭圆曲线上的点乘运算,明显优于 TGDH 协议的有限域上的 $3h-3$ 次指数运算。在存储量方面,该方案不像 LKH 和 TGDH 方案随着节点数目的增多存储量也随之增加。本方案中的通信由组长间的组播和子组内广播组成。由于子组的构造特点,保证子组广播是在一跳或者两跳范围内就可以到达,无须节点的多跳转发,因此比 TGDH 方案的通信量有所降低。

由表 2 和 3 可知,在节点加入和离开导致密钥更新所产生的开销中,本方案的通信开销优于 LHK 和 TGDH 方案:在集中式 LKH 方案中,随着网络中节点数目的增多通信开销也随之增加;本方案和 TGDH 方案通信次数虽都是常数,但是 TGDH 的全局内广播消息需要节点的多跳转发。在计算开销方面,本方案的大多数计算负担都在处理能力较强的组长节点上,降低了普通节点的计算开销,较分布式的 TGDH 有明显的优势。

4.4　可用性分析

在本方案中,由于子组内的所有节点都是在一跳或两跳通信范围内,子组内的广播消息无须转发直接到达,避免了在全网中进行过多的消息传递,降低了通信开销;各个子组的子组密钥协商和分发都是并行的,降低了网络延迟;在子组中采用密钥协商与密钥分发相结合的方式生成和分发子组密钥,避免了单个节点发生故障而导致整个协议不可用,提高了子组密钥分发和更新的效率;在各个分组中建立独立的子组密钥,使得一个子组内节点动态加入和退出群组不会影响到其他子组密钥,避免了“1 影响 N”的不利因素。因此,方案具有较强的可用性。

5　结束语

针对 Ad hoc 网络中大规模群组安全组通信面临的组密钥管理问题,本文提出一种基于地理位置的分层分组式组密钥管理方案。该方案基于节点的地理位置信息,将一跳或者两跳范围内的节点组成一个子组,采用分布式密钥协商与集中式密钥分发相结合的方式生成、分发和更新子组密钥,由各个子组中组长节点组成更高一级的群组参与协商组密钥,在避免单点失效的同时,提高了密钥生成和分发的效率,而且当节点动态加入或离开时,无须其他子组节点更新其子组密钥,避免了“1 影响 N”的不利因素,具有较强的可用性。同时,该方案基于椭圆曲线上的离散对数难题,用较短的密钥满足了 Ad hoc 网络组密钥管理的安全需求。

参考文献:

[1] WALLNER D, HARDER E, AGEE R. IETF RFC 2627, Key management for multicast: issues and architectures[S]. 1999.

[2] SHERMAN A T, McGREW D A. Key establishment in large dynamic groups using one-way function trees[J]. IEEE Trans on Software Engineering, 2003, 39(5): 444-458.

[3] STEINER M, TSUDIK G, WAIDNER M. Key agreement in dynamic peer groups[J]. IEEE Trans on Parallel and Distributed Systems, 2000, 11(8): 769-780.

[4] KIM Y, PERRIG A, TSUDIK G. Simple and fault-tolerant key agreement for dynamic collaborative groups[C]//Proc of the 7th ACM Conference on Computer and Communication Security. New York: ACM Press, 2003: 235-244.

[5] KIM Y, PERRIG A, TSUDIK G. Group key agreement efficient in communication[J]. IEEE Trans on Computers, 2004, 53(7): 905-921.

[6] RENUKA A, SHET K C. Hierarchical approach for key management in mobile Ad hoc networks[J]. Computer Science and Information Security, 2009, 5(1): 87-95.

[7] MITTRA S. Iolus: a framework for scalable secure multicasting [C]//Proc of ACM SZGCOMM'97. New York: ACM Press, 1997: 277-288.

[8] HERNANDEZ-SERRANO J, PEGUEROLES J, SORIANO M. GKM over large MANET[C]//Proc of SNPD/SAWN'05. Washington DC: IEEE Computer Society, 2005: 484-490.

[9] 冯涛, 王毅琳, 马建峰. 一种新的基于椭圆曲线密码体制的 Ad hoc 组密钥管理方案[J]. 电子学报, 2009, 37(5): 918-924.

[10] 扬子胥. 近世代数[M]. 北京:高等教育出版社, 2004:41.

[11] 卢开澄. 计算机密码学[M]. 北京:清华大学出版社, 2003.