一种基于身份标识的 MANET 组密钥协商协议

宋 震[1], 周贤伟[2], 窦文华[1]
(1. 国防科技大学计算机学院, 湖南长沙 410073; 2. 北京科技大学信息工程学院, 北京 100083)

摘 要: 组密钥是安全组通信中实现信息机密性和完整性的关键. 适应于 MANET 有限的计算、通信资源, MANET 组密钥管理协议应具有较少的计算量, 较低的运算强度. 分析了 MANET 组密钥管理方案所应具备的性质; 结合固定网络环境下具有最小通信量的组密钥协商协议 STR 协议及基于身份标识的公钥密码技术, 提出了一个基于身份标识的贡献式 MANET 组密钥协商管理协议 CEAGKP, 具有较小的通信量、较强的安全性与可扩展性, 能够很好地适应 MANET 环境的要求. 仿真结果证明了 CEAGKP 具有较好的伸缩性.

关键词: MANET; 身份标识; 组密钥; 协商
中图分类号: TP393.08 文献标识码: A 文章编号: 0372-2112 (2008) 10-1862-07

A New ID-Based MANET Group Key Agreement Protocol

SONG Zhen[1], ZHOU Xian-wei[2], DOU Wen-hua[1]
(1. *Computer School, National University of Defense Technology, Changsha 410073, China;*
2. *School of Information Engineering, University Science and Technology of Beijing, Beijing 100083, China*)

Abstract: Confidentiality and Integrity are realized by the group key in secure group communications. Adapt to the limited computation and communication resources, the group key management protocol in MANET environment should have less computation and less complexity. The necessary property that the group key management protocol should have is analyzed. A new ID-based group key agreement protocol CEAGKP is put forward by introducing the key tree of STR protocol that has minimum traffic in fixed network environment combined with the ID-based public key cryptography. The protocol has strong security and scalability at less traffic and well suits the requirement of MANET. The simulation proves that CEAGKP scales well in MANET.

Key words: MANET; identity; group key; agreement

1 引言

组密钥是所有参与组播的成员之间共享的秘密密钥, 用以对组播数据进行加/解密、认证等操作, 以满足数据机密性、源认证及完整性等安全需求[1].

组密钥管理是指为组成员生成、分发和更新组密钥, 其基本任务是: 为合法的组成员分配和维护密钥, 实现组播通信时秘密信息在合法组成员或某个成员子集之间共享, 而非组成员因不能解密而无法知道该秘密信息. 同时, 利用组密钥对组播内容进行加密, 确保得到通信内容的实体是群组成员, 从而在确保数据机密性的同时, 达到一定程度上的数据源认证(保证了组播源为组成员, 称之为群组认证).

1.1 组密钥的安全性

与单播密钥管理机制相比, 组密钥安全具有一些自身的要求. 总结起来, 主要包括:

(1) 密钥的保密性(Key Secrecy): 保证攻击者获得一个组密钥在计算上是不可能的.

(2) 前向保密性(Forward Secrecy): 主动退出或被强制退出的组播结点(比如恶意结点)不能继续参与组播, 即无法利用它们知道的密钥解密后续组播数据或生成有效的加密数据. 这种要求称为前向保密性或前向机密性. 通常为了达到前向保密性, 当有组成员退出后, 通信组更新组密钥以保证前向保密性. 由于密钥更新消息同样可以被以前的组成员获得, 必须防止以前的组成员从密钥更新消息中得到新密钥.

(3) 后向保密性(Backward Secrecy): 新成员加入群组后, 新加入的组成员无法破解其加入前的组播数据. 这种要求称为后向保密性或后向机密性. 当新成员加入时, 对组密钥进行更新可以实现后向保密性.

(4) 密钥独立性(Key Independence): 密钥独立性保证了当某敌手获得了一些组密钥时, 它不能发现其它未知的组密钥.

(5) 防止同谋破解: 组播密钥管理不仅要防止某个

收稿日期: 2007-12-10; 修回日期: 2007-12-25
基金项目: 国家自然科学基金(No. 60573050, 60603061)

结点破解系统, 还要防止某几个结点联合起来破解. 如果几个恶意结点联合起来, 掌握了足够多的密钥信息, 使得无论系统如何更新, 都可以获得更新后的组密钥, 从而导致组密钥前向保密性和后向保密性失败, 或者使得恶意结点可以冒充其他结点进行欺骗(从而攻破系统的认证性), 这称为同谋破解. 组密钥管理要杜绝同谋破解的发生或降低发生的概率.

1.2 MANET 组密钥管理方案

评价组密钥管理方案的标准通常包括:

(1) 所能够提供的安全性: 前向保密性、后向保密性、防止同谋破解等. 在不同的应用环境中, 对组密钥管理方案的安全性要求可能不同.

(2) 可扩展性: 是否能够适应组播环境下群组大小的变化. 由于通信组的成员数目通常是可变化的, 因此, 组密钥管理协议能否适应这种成员数目的变化成为了重要评价标准之一.

(3) 协议开销: 通常包括三类:

(ⅰ)计算开销: 即协议的计算复杂性, 各种成员事件下更新组密钥的计算量. 有时还需要计算顺序运算次数, 即完成单一协议运行所需要计算量;

(ⅱ) 通信开销: 主要包括(组密钥生成及组成员加入或离开时) 协议的圈数、消息数(分为广播消息数与单播消息数)、消息大小等;

(ⅲ)存储开销: 组控制器和组成员拥有的密钥数是衡量存储资源需求的重要指标, 特别是在资源有限的系统中, 其关系到协议的效率与可扩展性.

虽然组密钥管理问题一直是组播技术研究的一个热点, 但是, 按照上述组密钥管理方案的评价标准, 结合 MANET 自身拓扑结构动态变化、成员退出与加入操作可能非常频繁、资源有限等特点, 可以发现: 这些方案并不能完全适应于 MANET 应用, 理想的 MANET 组密钥管理方案还应具有以下性质:

(1) 分布式: MANET 的分布式特性决定了不存在永久固定的群组服务器, 因此, 理想的 MAENT 组密钥管理方案必须采用分布式处理方式.

(2) 贡献式(Contributory): 通常认为 MANET 中通信各方的身份完全平等, 且由于组成员关系可能经常发生改变, 其组密钥管理方案必须是贡献式的密钥协商机制.

(3) 低通信量: 适应于较低的网络带宽, 多跳、局部广播的通信方式, 以及动态变化的动态拓扑结构, 协议应具有较少的通信量(通信圈数少, 消息尺寸小), 以保证具有较高的成功率, 特别是在成员关系发生变动时的通信量, 要尽可能地小.

(4) 低计算量: 适应于有限的计算资源, 协议应具有较少的计算量, 这里较少的计算量是指协议每一个个体应具有较少的计算量, 另外, 考虑到公平性因素, 执行协议的个体之间计算量差异不能太大(在异构的网络中, 计算量可以有一些差异).

另外, 在安全性要求较高的战术环境等条件下, 密钥的认证性也是一个必须条件.

MANET 环境中的组密钥管理问题是 MANET 安全技术研究的一个热点. 理想的 MANET 组密钥管理协议应具有分布式、低通信量、低计算量以及认证性等特点. 但是, 通常情况下, 低通信量与低计算量的要求是无法同时满足的, 哪种因素更为重要与 MANET 应用环境及自身特点密切相关. 有研究表明[20,21], 典型嵌入式平台上, 传输信息比执行计算更消耗电能, 每发送一个比特信息所消耗的能量足以执行数千条计算指令(具体的试验数据与测试平台相关). 另外, 依赖于制造工艺等技术的不断进步, 处理器计算能力与功耗的比值呈不断上升趋势, 提高较为迅速.

为此, 为提高网络的生存能力, 针对 MANET 自身特点, 本文结合固定网络环境下具有最小通信量的组密钥协商协议 STR 协议[2] 及基于身份标识的公钥密码技术[3], 提出了一个基于身份标识的贡献式 MANET 组密钥协商管理协议 CEAGKP, 具有较小的通信量, 能够很好地适应 MANET 环境的要求.

2 网络模型假设

研究基于如下的网络模型假设展开:

(1) 系统存在一个离线的可信任机构 PKG: 网络在部署阶段存在一个可信任机构 PKG, 负责产生系统的密码参数及公/私钥对, 并为系统初始用户分发秘密密钥. 完成所有工作后, PKG 可以离线.

(2) 所有组成员遵循同一安全框架(具有相同的安全参数). 每一个组成员处均维护着一个组成员资格列表, 表示着当前具有合法资格的组成员. 这与其它文献中的组成员资格证书具有相同的作用.

(3) 每个结点都有唯一的可相互区别的身份标识: 例如在文献[4]中, MANET 结点使用了所谓的统计唯一密码可验证(Statistically Unique and Cryptographically Verifiable, SUCV)地址, 结点地址与其公开密钥绑定在一起, 具有不可复制性; 另外, 在结点加入网络时, 结点可以产生/分配不可预计的随机标识; 最后, 在具有明确安全要求的应用环境下, MANET 结点具有唯一的可辨识身份标识既是可行的, 也是必须的.

(4) 每个结点都具有某种监视机制, 可以对网络结点(尤其是其一跳邻居结点的异常) 做出判断: 这一假设对于 MANET 网络是可能的, 由于无线网络的广播特性, 可以利用对邻居结点通信的监听来判断邻居结点的异常. 在 MANET 入侵检测技术、公平路由及合作机制等研

究中普遍性地使用了这一假设.

(5) 基本的组通信机制能够抵抗 fail-stop 故障: 可靠组通信机制符合 VS 模型[6], 可以向所有的群组成员提供一致的组成员关系(所有的组成员将"看到"相同的组成员事件序列), 并且通信满足可靠性和因果顺序.

(6) 组成员之间相互信任. 由于每个结点都具有监视机制, 且密钥能够被妥善地保存而不被窃取, 因此, 在某组成员被攻陷后, 该组成员能够迅速地被排除出通信组.

3 基于身份标识的 MANET 组密钥管理方案

3.1 密钥树

CEAGKP 协议将所有组成员组织成一棵(虚拟的)逻辑二叉树, 树中所有的叶子结点均代表着组成员, 所有内部结点为虚拟结点, 其子结点中最多只有一个内部结点. 以四个组成员的组播组为例, 形成如图 1 所示的密钥树(下文中 CEAGKP 协议的描述均以此组播组密钥协商的过程为例).

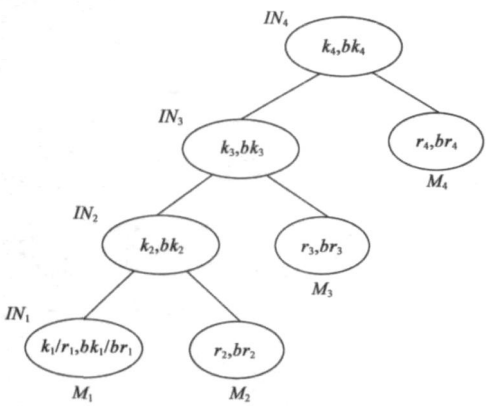

图 1 CEAGKP 协议密钥树示意图

CEAGKP 协议中, 每次成员事件发生之后都会形成新的逻辑密钥树.

3.2 符号描述

假设组播组中有 N 个成员, 其身份标识为 $M_1, \cdots, M_N$. 方案描述中所用到的符号定义如下:

(1) M_i: 组成员结点身份标识, $i = 1, 2, \cdots, N$;

(2) IN_i: 第 i 层的内部结点. 方便描述起见, 假设最底层的左叶子结点为内部结点 IN_1;

(3) LN_i: 成员 M_i 所对应的叶子结点;

(4) k_i: 第 i 层内部结点所持有的临时密钥, k_i 通过两个子结点的密钥计算得到;

(5) r_i: 在进行组密钥协商时, 组成员结点 M_i 选择会话随机数(Session Random) r_i.

3.3 系统建立与组密钥的生成

假设 PKG 为密钥协商双方生成共享的系统安全参数 $parmas$:

$$parmas = \langle G_1, G_2, \hat{e}, p, P, P_{pub}, H, H_1, H_2 \rangle$$

其中: G_1, G_2 为大素数阶 p 的椭圆曲线群; 双线性映射 $\hat{e}: G_1 \times H_2 \rightarrow H_2$; $P, P_{pub} \in G_1$, $P_{pub} = sP$; Hash 函数 H: $\{0,1\}^* \rightarrow G_1$(表述方便起见, 记 $Q_{ID} = H(ID)$), H_1: $G_2 \rightarrow Z_p^*$. Hash 函数, H_2: $\{0,1\}^* \rightarrow Z_p^*$ 用于消息"签名"的计算.

PKG 生成系统的主密钥为 $s \in Z_p$, 并为用户 M_i 计算秘密密钥 $S_i = sQ_i$(公开密钥为 $Q_i = H(M_i)$). 完成所有用户的密钥生成工作后, PKG 可以离线退出. 主密钥必须保持严格的保密, 或者被直接销毁.

假设群组初始有 n 个组成员 $M_1, \cdots, M_n$ ($0 \leqslant n \leqslant N$). 当发起组密钥协商时, 每一个组成员 M_i 随机选择一个会话随机数 $r_i \in Z_p^*$, 计算 $\langle U_i, V_i \rangle = \langle r_i P, H_2(U_i) S_i + r_i P_{pub} \rangle$ 并广播($U_i = r_i P$ 也称为盲会话随机数(Blinded Session Random), 在图 1 中记为 br_i).

每一个组成员 M_i 计算:

$$\begin{aligned} \hat{e}(V_i, P) &= \hat{e}(H_2(U_i) Q_i + r_i P, P_{pub} \\ &= \hat{e}(H_2(U_i) Q_i + U_i, P_{pub} \end{aligned} \tag{1}$$

对参与协议运行的所有组成员进行认证.

成员 M_1 计算:

$$\begin{aligned} &k_2 = \hat{e}(U_2, P_{pub})^{r_1} = \hat{e}(P, P)^{sr_1 r_2}, \\ &\langle IU_2, IV_2 \rangle = \langle H_1(k_2) P, H_2(IU_2) S_1 + H_1(k_2) P_{pub} \rangle, \\ &k_3 = \hat{e}(U_3, P_{pub})^{H_1(k_2) r_3} = \hat{e}(P, P)^{sH_1(k_2) r_3}, \\ &\langle IU_3, IV_3 \rangle = \langle H_1(k_3) P, H_2(IU_3) S_1 + H_1(k_3) P_{pub} \rangle, \\ &k_4 = \hat{e}(U_4, P_{pub})^{H_1(k_3) r_4} = \hat{e}(P, P)^{sH_1(k_3) r_4}, \\ &\cdots \quad \cdots \end{aligned} \tag{2}$$

并广播 $\langle IU_2, IV_2 \rangle, \langle IU_3, IV_3 \rangle, \cdots, \langle IU_{n-1}, IV_{n-1} \rangle$.

$\langle IU_i, IV_i \rangle$ 是内部结点临时密钥的签名, 在此用 M_1 成员的私钥进行了"承诺", 这是因为假设合法组成员之间是相互信任的($IU_i = H_1(k_i) P$ 也称为盲密钥(Blinded Key), 在图 1 中记为 bk_i).

每一个组成员 M_i 计算:

$$\begin{aligned} \hat{e}(IV_i, P) &= \hat{e}(H_2(IU_i) Q_1 + H_1(r_i) P_{pub}, P) \\ &= \hat{e}(H_2(IU_i) Q_1 + IU_i, P_{pub} \end{aligned} \tag{3}$$

通过 M_1 的"签名", 验证内部结点 IN_i 盲密钥的认证性.

每一个组成员 M_i 可以相应性地计算:

$$k_i = \hat{e}(IU_{i-1}, P_{pub})^{r_i} = \hat{e}(P, P)^{sH_1(k_{i-1}) r_i} \tag{4}$$

$$k_j = \hat{e}(U_j, P_{pub})^{H_1(k_{j-1})} = \hat{e}(P, P)^{sH_1(k_{j-1})}, \ i+1 \leqslant j \leqslant n \tag{5}$$

通过上述公式递归计算, 可得到密钥树中根结点的密钥, 即为组播组的组密钥.

CEAGKP 协议中引入了一个辅助的管理结点, 称为 Sponsor 结点 M_s, 该结点根据树的形状动态选取, 用于辅

助进行组密钥的更新等操作. 根据组成员事件的不同, CEAGKP 协议相应地改变密钥树的形状, 并在辅助管理结点的协助下, 进行组密钥的更新. 由于子组合并和分割协议以加入和离开算法为基础, 同时密钥更新算法又是离开算法的特例(无成员离开), 所以本文只讨论加入和离开算法.

3.4 成员加入

假设组播组中有 n 个成员$\{M_1, \cdots, M_n\}$. 希望加入群组的主机 M_{n+1}首先向群组成员发送加入请求, 加入请求中包括主机的认证信息, 组成员可以根据组成员资格列表(或群组安全策略)来接受或拒绝.

此时, Sponsor 结点 M_s 为原根结点的右子结点, 即 M_n.

成员加入操作算法执行过程如下:

(1) 新成员 M_{n+1}广播一个加入请求消息, 其中包含着$\langle U_{n+1}, V_{n+1}\rangle$;

(2) 更新密钥树, 创建一个新的根结点, 原来的根结点成为新结点的左子结点, 新成员对应新结点的右子结点;

(3) Sponsor 生成新的会话随机数, 计算并广播$\langle U_n, V_n\rangle$及$\langle IU_n, IV_n\rangle$;

(4) 所有结点计算新的组密钥, 分为三种情况:

(ⅰ) M_{n+1}对$\langle IU_n, IV_n\rangle$进行验证并计算:

$$k_{n+1} = \hat{e}(IU_n, P_{pub})^{r_{n+1}} = \hat{e}(P, P)^{sr_{n+1}H_1(k_n)} \quad (6)$$

(ⅱ) M_n 对$\langle IU_{n+1}, IV_{n+1}\rangle$对进行验证并计算:

$$\hat{e}(U_{n+1}, P_{pub})^{H_1(k_n)} = \hat{e}(P, P)^{sr_{n+1}H_1(k_n)} = k_{n+1} \quad (7)$$

(ⅲ) 其余结点对$\langle U_{n+1}, V_{n+1}\rangle$, $\langle U_n, V_n\rangle$进行验证并计算:

$$\hat{e}(U_n, P_{pub})^{H_1(k_{n-1})} = \hat{e}(P, P)^{sr_nH_1(k_{n-1})} = k_n \quad (8)$$

$$\hat{e}(U_{n+1}, P_{pub})^{H_1(k_n)} = \hat{e}(P, P)^{sr_{n+1}H_1(k_n)} = k_{n+1} \quad (9)$$

其伪代码如算法 1 所示.

算法 1　*CEAGKP* 组密钥更新算法: 成员加入

```
1:  M_{n+1} broadcast JoinRequest(M_{n+1}, ⟨U_{n+1}, V_{n+1}⟩) to Group \ 圈 1
2:  if M_{n+1} is not in the MemberList then
3:  deny M_{n+1} JoinRequest \ M_{n+1}不是合法的组成员
4:  return
5:  else
6:  The Sponsor M_s(s = n) updates its session random r_s;
7:  M_s computes ⟨U_s, V_s⟩ and new k_s and⟨IU_s, IV_s⟩, then broadcasts
    ⟨U_s, V_s⟩ and ⟨IU_s, IV_s⟩;
8:  for all M_i(i ∈ 1, 2, ···, n+1) do\ 圈 2
9:  updates the tree by inserting M_{n+1}
10: end for
11: if i ∈ {i, ···, n−1} then
12:  M_i verifies ⟨U_{n+1}, V_{n+1}⟩ and⟨U_s, V_s⟩. if succeed, computes k_n and
     k_{n+1} by (8) and (9);
13: else if i = n+1 then \ 新加入成员结点
14:  M_{n+1} verifies⟨IU_s, IV_s⟩. if succeed, computes k_{n+1} by (6);
15: else if i = s then \ ponsor 结点
16: M_s computes k_{n+1} by (7);
17: end if
18: n ← n+1;
19: end if
```

具体以图 1 所示密钥树为例, 当有新成员加入时, 密钥树的变化情况如图 2 所示.

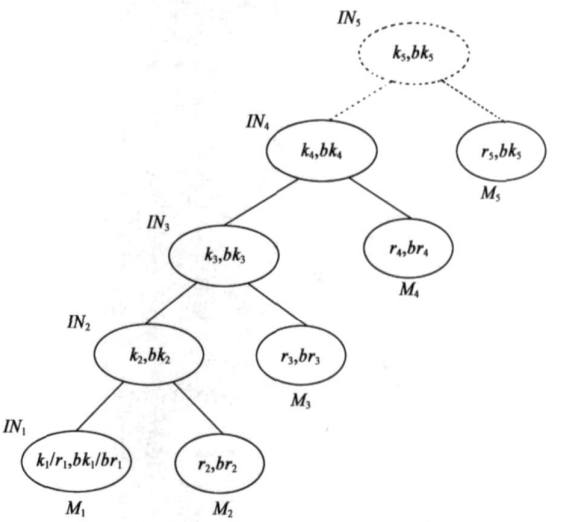

图 2　CEAGKP:成员加入操作

3.5 成员离开

假设群组中当前有 n 个成员, 成员 $M_d(d \leqslant n)$要离开群组. 如果 $d > 1$, 则 Sponsor 结点 M_s 是要离开成员下面的叶结点, 即 M_{d-1}, 否则 Sponsor 结点将是 M_2.

当成员 M_1 离开组播组时, 其执行过程与组密钥初始生成的协议类似, M_2 将作为最左叶结点进行组密钥的生成.

在其它情况下, 当接收到成员离开事件通知后, 每个剩余成员执行的算法过程如下:

(1) 更新密钥树, 删除 M_d 对应的结点及其父结点, 用 M_d 原来的兄弟结点代替 M_d 的父结点;

(2) Sponsor 结点 M_{d-1}生成新的会话随机数 r_{d-1}, 计算所有高层内部结点的临时密钥 $k_{d-1}, k_{d+1}\cdots$, 计算盲密钥(及其"签名")$\langle IU_j, IV_j\rangle (j = d-1, d+1, \cdots, n)$与自己的盲会话随机数(及其"签名")$\langle U_{d-1}, V_{d-1}\rangle$并广播给其余组成员;

(3) 组成员 M_i 计算组密钥:

(ⅰ) $i = d+1$: M_{d+1}验证 Sponsor 结点 M_{d-1}发送来

的所有$\langle IU_j, IV_j\rangle$及$\langle U_{d-1}, V_{d-1}\rangle$, 并计算:

$$k_{d+1}=\hat{e}(IU_{d-1}, P_{pub})^{r_{d+1}} \quad (10)$$

$$k_j=\hat{e}(U_j, P_{pub})^{H_1(r_{j-1})}, j=d+2, \cdots, n \quad (11)$$

(ii) $i>d+1$: M_i 验证 Sponsor 结点 M_{d-1}发送来的所有$\langle IU_j, IV_j\rangle$及$\langle U_{d-1}, V_{d-1}\rangle$, 并计算:

$$k_i=\hat{e}(IU_{i-1}, P_{pub})^{r_i} \quad (12)$$

$$k_j=\hat{e}(U_j, P_{pub})^{H_1(r_{j-1})}, j=i+1, \cdots, n \quad (13)$$

(iii) $i<d-1$: M_i 验证 Sponsor 结点发送来的所有$\langle IU_j, IV_j\rangle$及$\langle U_{d-1}, V_{d-1}\rangle$, 并计算:

$$k_{d-1}=\hat{e}(U_{d-1}, P_{pub})^{H_1(k_{d-2})} \quad (14)$$

$$k_{d+1}=\hat{e}(U_{d+1}, P_{pub})^{H_1(k_{d-1})} \quad (15)$$

$$k_j=\hat{e}(IU_j, P_{pub})^{H_1(k_{j-1})}, j=d+2, \cdots, n \quad (16)$$

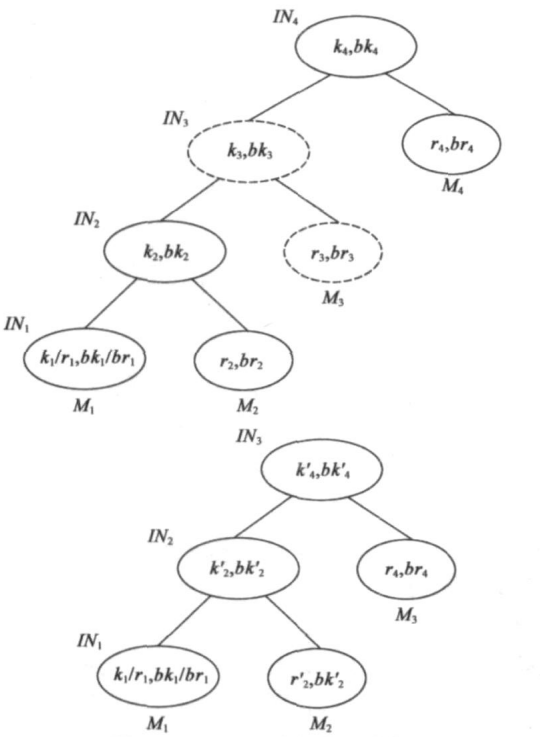

图 3 CEAGKP: 成员离开操作

其伪代码如算法 2 所示.

算法 **2** CEAGKP 组密钥更新算法: 成员离开

```
for all M_i (i ∈ 1, 2, ..., n) \ d do
updates the tree by removing M_d
end for
Sponsor M_{d-1} updates its session random r_{d-1} and computes ⟨U_{d-1}, V_{d-1}⟩;
Sponsor M_{d-1} computes ⟨IU_i, IV_i⟩, ∀ i ∈ [d-1, n-1] \ {d};
Sponsor M_{d-1} broadcast ⟨U_{d-1}, V_{d-1}⟩ and ⟨IU_i, IV_i⟩, ∀ i ∈ [d-1, n-1] \ {d};
for all M_i (i ∈ {1, 2, ..., n} \ d) do
if i = d+1 then
  M_{d+1} computes k_n by (10) and (11);
else if i > d+1 then
  M_i computes k_n by (12) and (13);
else if i > d-1 then
  M_i computes k_n by (14), (15) and (16).
end if
end for
n ← n-1
```

具体以图 1 所示密钥树为例, 当成员 M_3 离开时, 密钥树的变化情况如图 3 所示.

4 比较与分析

主要从协议开销、安全性以及协议仿真与性能分析等三个方面对 CEAGKP 进行分析.

4.1 协议开销分析

表 1 所示为 CEAGKP 协议的开销情况(其中, 由于对运算是 CEAGKP 协议的主要运算环节, 计算量开销以对运算的次数衡量).

表 1 显示: CEAGKP 协议在组密钥建立、成员加入与退出操作时, 协议执行圈数为常数, 其中, 成员加入操作需要 1 轮广播, 协议运行 2 轮, 只需要 3 条消息; 成员退出操作只需要 1 轮广播, 协议只需要运行 1 轮, 需要 $2n-4$(退出成员为 M_1)或 $2n-2d$(退出成员 M_d($d\neq 1$))条消息. 计算开销方面, 成员加入操作时, 每个组成

表 **1** CEAGKP 协议开销评价指标

协议操作 \ 评价指标		计算量		存储量	通信量		
		运算次数	顺序运算次数		圈数	广播数	消息大小
组密钥建立		n^2+n-2	$2n-2$	$n-1$	2	$n+1$	$4n-4$
成员加入		$4n$	4	$n-1$	$2v$	1	3
成员退出	$d=1,2$	$\frac{n^2+n-6}{2}$	$3n-8$	$n-1$	1	1	$2n-4$
	$d\geqslant 3$	$\frac{n^2-d^2+n+3d-6}{2}$	$3n-8$	$n-1$	1	1	$2n-2d$

员需要进行 4 次 Pairing 运算(其中 2 次是进行身份验证);成员退出时每个组成员所执行的运算开销均为 $O(n)$*.

从上述分析可以看出, CEAGKP 协议执行圈数为常数, 协议通信量较小, 且协议的可扩展性较强.

4.2 安全性分析

4.2.1 CEAGKP 协议认证性分析

CEAGKP 协议中, 用每个用户 M_i 广播的消息 U_i 作为组密钥协商的组件之一, 而 $U_i = r_iP$ 的认证性是通过公开承诺 $V_i = H_2(U_i)S_i + r_iP_{pub}$ 保证的, 其验证方程是:

$$\hat{e}(V_i, P) = \hat{e}(H_2(U_i)Q_i + U_i, P_{pub})$$

在 ROM 模型中考虑其认证安全性, 即把所有的 Hash 函数视为随机预言, 然后, 基于文献[25]给出的 Forking Lemma 提供的 Oracle 重放攻击技术来证明上述承诺是不可伪造的. 简要证明过程如下:

假设存在一个概率多项式时间的图灵机 E(敌手), 它以$\langle U_i, V_i\rangle$作为输入, 并且能够以不可忽略的概率输出一个伪造的"消息承诺"对$\langle U_i^*, V_i^*\rangle$, 则可以证明:存在一个多项式时间算法 E' 解决弱 Diffie-Hellman 问题**, 如下:

根据文献[6]给出的 Forking Lemma, E 可以获得同一消息的两个伪造承诺: V_i 与 V_i^u:

$$V_i = H_2(U_i)S_i + r_iP_{pub},\ V_i^u = H_2'(U_i)S_i + r_iP_{pub}$$

因此, $V_i - V_i^u = (H_2(U_i) - H_2'(U_j))S_i$

然后, $\hat{e}(V_i - V_i^u, P) = \hat{e}(Q_i, P_{pub}^{H_2(U_i) - H_2(U_i)})$

可以看出, E' 可求 Pairing 的逆. 因此, 存在一个多项式时间算法 $f: G_2 \rightarrow G_1$. 令 g_2 是一个 G_2 的生成元, 则 $g_1 = \hat{e}(f(g_2), f(g_2))$ 也是 G_1 的生成元. $\hat{e}(f((g_2)^\lambda), f((g_2)^\lambda)) = (g_1)^{\lambda\mu}$. 也就是说, 给定$(g_2)^\lambda$与$(g_2)^\mu$, 计算 $(g_1)^{\lambda\mu}$, 这就得到了一个 G_2 中的弱 Diffie-Hellman 问题求解, 而一般情况下均假设是难解的, 无法在多项式时间内以不可忽略的概率求解.

因此, 敌手无法以不可忽略的概率伪造"消息承诺", CEAGKP 协议的认证性得证.

4.2.2 CEAGKP 协议组密钥保密性分析

不失一般性, 考虑 2 个用户 M_1 与 M_2 的情形, 该协议可以简化为

$$M_1 \xrightarrow{U_1 = r_1P} M_2$$

$$M_1 \xrightarrow{U_2 = r_2P} M_2$$

则 M_1 计算 $k_1 = \hat{e}(U_2, P_{pub})^{r_1}$, M_2 计算 $k_2 = \hat{e}(U_1, P_{pub})^{r_2}$.

可以看出

$$k_1 = \hat{e}(P, P_{pub})^{r_1r_2} = \hat{e}(P, P)^{sr_1r_2} = k_2$$

如果攻击者能够在仅知道 $P_{pub} = sP$, r_1P, r_2P 的情况下, 计算得到会话密钥 $k = \hat{e}(P, P)^{sr_1r_2}$, 那么它必须能够以不可忽略的概率解决 *BDH* 问题. 而一般认为 BDH 是难解的, 无法在多项式时间内以不可忽略的概率求解.

因此, 两方用户进行密钥协商的情况下, 组密钥的保密性得证. 而在 CEAGKP 中, 在多个组用户进行组密钥协商时, 其组密钥的保密性能够递归推导证明. 据此, CEAGKP 组密钥的保密性得到证明.

另外, 当成员加入群或离开群组时, 新的组密钥中都包含了随机生成的新信息, 这就保证了新密钥和其它密钥之间的相互独立, 即使成员的长期密钥丢失了, 也不会引起旧组密钥的泄漏, 即 CEAGKP 协议提供了密钥独立性、前向保密性和后向保密性.

4.3 试验仿真及性能分析

试验仿真运用 Linux 环境下的网络仿真软件 ns-2[7], 对 CEAGKP 协议进行仿真, 仿真参数如下: MAC 层协议采用 IEEE 802.11, 协议采用 AODV 协议, 数据传输带宽为 2Mbps, 结点的传输距离为 200m, 场景范围为 500 × 500.

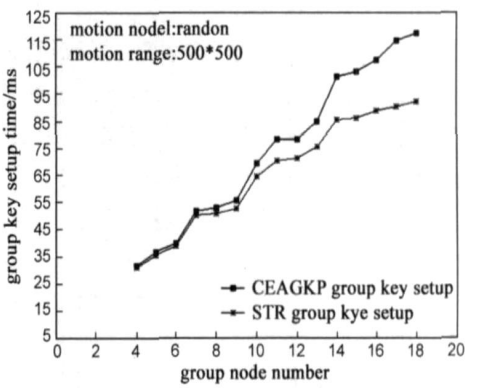

图 4　不同结点数的组密钥建立时间

图 4 所示为不同结点数的组密钥建立时间. 可以看出, CEAGKP 组密钥建立协议的时间效率劣于 STR 协议, 但 CEAGKP 组密钥建立时间大概为为组规模的一次函数, 在实际应用中可以接受, 另外相比 STR 协议, CEAGKP 提供了密钥前向保密性以及认证性, 结点越多, 认证的次数越多, 时间开销也越大, 因此, 组规模越大, CEAGKP 协议与 *STR* 协议的时间差也越大.

图 5 所示为不同运动场景下的相同组规模的组密钥建立时间. 对于相同的组规模, 组密钥建立时间随着结点的运动速度大小的变化很小, 组中的结点可以根

---

* 表中"运算次数"表示了协议需要完成的典型计算的次数, 它表示了协议的复杂度, 反映了协议总的运算开销; 表中"顺序运算次数"表示了协议必须串行完成的典型运算的次数, 它表示了每一个实体所执行协议的最大复杂度, 主要反映了协议的最大执行时间.

** 给定$(P, Q, aP), a \in Z_q^*$, 计算 aQ.

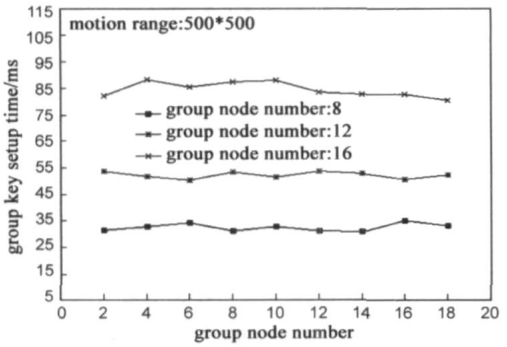

图 5 不同运动场景下的组密钥建立时间

据需要选择不同的运动速度, 协议的灵活性、可适应性很强; 组密钥的建立时间主要由组的规模决定, 规模越大, 组密钥建立时间越长. 组规模的选择应该根据实际情况而定, 否则, 太小可能导致结点频繁的加入或离开该组, 从而花费更多的开销; 太大又可能带来很大的管理开销, 同时丧失了灵活性.

5 结束语

组密钥管理是组播安全技术研究中的重要课题之一, 也是目前的研究热门. 针对固定环境下的网络, 研究人员提出了大量的组密钥管理方案, 但对于具有动态变化的拓扑结构、有限的网络资源、组成员加入/退出操作频繁等特点的 MANET 网络, 传统的组密钥管理方案往往会遇到不相适应的问题. 理想的 MANET 组密钥管理方案应具备分布式处理、贡献式密钥生成、低通信量、低计算量等性质. 这种性质之间往往存在冲突, 因此, 不存在能够满足一切要求的 MANET 组密钥管理方案, 必须在具体的应用环境中选择具有特定优势的 MANET 组密钥管理方案.

本文结合固定网络环境下具有最小通信量的组密钥协商协议 STR 协议及基于身份标识的公钥密码技术, 提出了一个认证的基于身份标识的贡献式 MANET 组密钥协商管理协议, 很好地适应了 MANET 自身的特点及无线通信环境的要求. 当群组成员增加时, 新成员的秘密份额可以作为计算群组密钥的一个参数, 而且原有成员的秘密份额不需改变; 当群组成员减少时, 离开的群组成员的秘密份额不会参与新群组密钥的计算, 协议具有较小的通信量; 同时, 认证的密钥协商过程, 确保了协议能够适应高安全性要求的环境.

参考文献:

[1] Debby M Wallner, Eric J Harder, Ryan C Agee. Key Management for Multicast: Issues and Architectures[S]. RFC2627, 1999.

[2] Yongdae Kim, Adrian Perrig, Gene Tsudik. Communication-efficient group key agreement[A]. Proceedings of the 17th International Information Security Conference IFIP SEC. 01[C], 2001. 229-244.

[3] Adi Shamir. Identity-based cryptosystems and signature schemes [A]. Advances in cryptology-Proceedings of CRYPTO'84 [C]. vol. 196 of Lecture Notes in Computer Science, Springer-Verlag, 1984. 47-53.

[4] Rekesh Babu Bobba, Laurent Eschenauer, Virgil Gligor, William Arbaugh. Bootstrapping Security Associations for Routing in Mobile Ad-Hoc Networks[R]. Technical Report 2002-44, University of Maryland, 2002.

[5] Yair Amir, Giuseppe Ateniese, Damian Hasse, Yongdae Kim, Cristina Nita-Rotaru, Theo Schlossnagle, John Schultz, Jonathan Stanton, Gene Tsudik. Secure Group Communication in Asynchronous Networks with Failures: Integration and Experiments [A]. Proceedings of the 20th IEEE International Conference on Distributed Computing Systems (ICDCS 2000)[C]. 2000. 330-343.

[6] David Pointcheval, Jacques Stern. Security arguments for digital signatures and blind signatures[J]. Journal of Cryptology: the journal of the International Association for Cryptologic Research, 2000, 13(3): 361-396.

[7] Network simulator ns-2[OL]. http://www.isi.edu/nsnam/ns/.s

作者简介:

宋 震 男, 1976 年 10 月生于陕西三原, 国防科技大学计算机学院博士研究生, 研究方向为网络与信息安全, 路由协议.

E-mail: songzhen@nudt.edu.cn

周贤伟 男, 1963 年 5 月生于四川成都, 北京科技大学信息工程学院教授, 博士生导师, 主要研究领域为网络安全、移动通信、无线传感器网络.

E-mail: xwzhouli@sina.com

窦文华 男, 国防科技大学计算机学院教授, 博士生导师, 主要研究领域为网络, 信息安全等.