

## 基于分簇的 Ad Hoc 网络组密钥建立方案

张小彬, 韩继红, 王亚弟, 刘 敏

(信息工程大学 电子技术学院, 郑州 450004)

(zhxbnhjd@sina.com)

**摘 要:**针对移动 Ad Hoc 网络的特点,提出了一个基于分簇的移动 Ad Hoc 网络组密钥建立方案。该方案首先采用部分协商和部分分发相结合的方式建立簇密钥,然后采用完全协商的方式来建立组密钥。协商过程采用椭圆曲线密码体制和双线性对来实施,能够对不诚实节点进行检测和鉴别。簇密钥分发则采用成员过滤技术来实施。另外,还提出了一个签名方案来提供认证性。提出的组密钥建立方案有效地降低了组密钥建立过程的计算和通信开销,并具有较高的安全性和可用性。

**关键词:**Ad Hoc 网络;分簇;混合式;组密钥分发;组密钥协商

**中图分类号:**TP393.08 **文献标志码:**A

## Cluster-based group key establishment scheme in Ad Hoc network

ZHANG Xiao-bin, HAN Ji-hong, WANG Ya-di, LIU Min

(Institute of Electronic Technology, Information Engineering University, Zhengzhou Henan 450004, China)

**Abstract:** Based on the characteristics of Ad Hoc networks, this paper proposed a cluster-based group key establishment scheme in Ad Hoc networks. Firstly, a combination of part negotiation and part distribution was adopted to establish the cluster key, then, a full negotiation method was used to establish the group key. Elliptic curve cryptography and bilinear pairing were employed in the negotiation process; therefore, the dishonest node could be detected and identified. Member filter technique was employed in the cluster key distribution. Moreover, in order to authenticate the messages, a signature scheme was proposed. The proposed group key establishment scheme reduces both the computational overhead and communication costs, and has high security and usability.

**Key words:** Ad Hoc network; clustering; composite; group key distribution; group key agreement

### 0 引言

移动 Ad Hoc 网络是一种新型的移动多跳无线网络。与传统网络不同的是,它不依赖固定的基础设施,通过移动节点间的相互协作进行组网,具有动态变化的网络拓扑、受限的无线传输带宽、移动终端的局限性、分布式控制等特征。显然传统的组密钥建立方案不能直接应用于移动 Ad Hoc 网络。理想的组密钥管理方案应该具有分布式、低通信量、低计算量、高可扩展性、高可用性等性质。

文献[1]提出了一个固定轮数的组密钥协商协议 BD 协议,但此协议中假定节点间的距离都是一跳。直接应用于全网组密钥的协商时,由于节点间距离可能较远,全网的广播会造成较大的通信量,且采用指数运算造成节点的计算量相对较大,另外缺乏对协商过程中不诚实节点的检测和鉴别。文献[2]提出了非平衡二叉树结构的 STR 协议,此协议具有较小的通信量,但是节点的计算量相对较大,当节点数较多时,处于树的最底层的节点的退出将会造成很大的计算量。文献[3]给出了一个采用成员过滤技术的组密钥分发方案,由簇首生成组密钥,使用节点私钥的哈希值来构造成员过滤函数并分发,成员节点利用此过滤函数计算出组密钥。但此方案中由簇首单个节点生成组密钥并进行分发,存在集中式的信

任和服务的可用性问题,且此方案中退出成员结合退出前后的成员过滤函数仍然可以计算出更新后的组密钥。另外,这三个方案直接应用于平面结构的移动自组网组密钥建立时都存在可扩展性和可用性的问题。

### 1 预备知识

#### 1.1 双线性对

$G_1$  和  $G_2$  是两个阶为  $q$  的群,  $G_1$  是一个加法群,而  $G_2$  是一个乘法群。

**定义** 双线性对。是这样映射  $e: G_1 \times G_1 \rightarrow G_2$ , 该映射具有如下的属性。

1) 双线性性。对于所有的  $P, Q \in G_1, a, b \in Z_q^*, e(aP, bQ) = e(P, Q)^{ab}$ 。

2) 非退化性。如果  $P$  是  $G_1$  的产生元,则  $e(P, P)$  是  $G_2$  的产生元。

3) 可计算性。存在一个有效的算法来计算  $e(P, Q)$ 。

双线性 Diffie-Hellman 问题假定 (BDHP 假定) 给定  $P, aP, bP, cP \in G_1$ , 而  $a, b, c \in Z_q^*$ 。计算  $e(P, P)^{abc}$  是困难的,也就是说不存在概率多项式时间算法来解决此问题。

#### 1.2 Joux 的三方协商协议

Joux 给出的  $A, B, C$  三方密钥协商协议<sup>[4]</sup>如下。

收稿日期:2009-02-07;修回日期:2009-03-25。

**作者简介:**张小彬(1982-),男,云南曲靖人,硕士研究生,主要研究方向:计算机网络安全;韩继红(1966-),女,山西定襄人,教授,主要研究方向:计算机网络安全、信息系统安全;王亚弟(1953-),男,甘肃兰州人,教授,博士生导师,主要研究方向:计算机网络安全、信息系统安全;刘敏(1986-),女,山东潍坊人,硕士研究生,主要研究方向:计算机网络安全。

1) 交换信息。  $A \rightarrow B, C; aP; B \rightarrow A, C; bP; C \rightarrow A, B; cP$ 。  
 $a, b, c$  是  $A, B, C$  选择的随机数且  $a, b, c \in Z_p^*$ 。

2) 计算组密钥。  $A$  计算  $e(bP, cP)^a$ ,  $B$  计算  $e(aP, cP)^b$ ,  $C$  计算  $e(aP, bP)^c$ , 最终三方得到相同的组密钥  $e(P, P)^{abc}$ 。

### 1.3 BD 协议<sup>[1]</sup>

$G$  是一个阶为  $p$  有限循环群,  $g$  是  $G$  的一个生成元。协议过程如下。

1) 每个用户选择一个随机数  $r_i, 1 \leq r_i \leq p-2$ , 计算并广播  $z_i = g^{r_i} \pmod{p}$ 。

2) 每个用户收到  $z_{i+1}$  和  $z_{i-1}$  后, 计算并广播  $X_i = (z_{i+1}/z_{i-1})^{r_i} \pmod{p}$ 。

3) 每个用户计算会话密钥:

$$K_i = (z_{i-1})^{m_i} \cdot X_i^{n-1} \cdot X_{i+1}^{n-2} \cdot \dots \cdot X_{i-2} \pmod{p}。$$

### 1.4 安全过滤器

安全过滤器<sup>[5]</sup>是 Galois 域  $GF(p)$  上的一个关于随机变量  $x$  的多项式,  $p$  是一个公开的大素数, 集合  $C = \{k_0, k_1, \dots, k_{i-1}\}$  中元素和数  $gk$  都属于  $Z_p^*$ , 可以构建一个多项式  $sf$ ,  $sf(x) = (x - h(k_0))(x - h(k_1)) \dots (x - h(k_{i-1})) + gk$ 。显然, 如果  $k_j \in C, sf(h(k_j)) \equiv gk \pmod{p}$ , 因此称  $sf$  是在参数集  $C$  上  $gk$  的一个过滤器, 其中  $h$  是一个单向哈希函数。为了安全, 将  $sf$  整理为  $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$  标准形式, 如无参数集中的参数, 则无法直接从  $sf$  推导出  $gk$ 。

## 2 基于分簇的混合式的组密钥建立框架

直接在全网中部署一个集中式分发或分布式协商的组密钥管理方案存在扩展性和可用性的问题, 而分层分簇的网络结构中一个簇的成员关系的变化不会影响到其他簇的簇密钥, 这样可以带来良好的扩展性, 而且进行簇的划分后, 在簇内实现密钥协商或分发相对来说容易, 这样可以增强方案可用性。基于此, 本文提出了一个基于分簇的移动自组网混合式组密钥建立框架, 如图 1 所示。在网络中, 首先对网络分簇, 各簇簇首形成高一级的网络。在每个低级簇内建立一个簇密钥, 各簇的簇密钥之间相对独立。由簇首组成的高级簇中也建立一个簇密钥, 高级簇的簇密钥作为全网组密钥向下分发, 最终所有节点得到一个共享的组密钥。

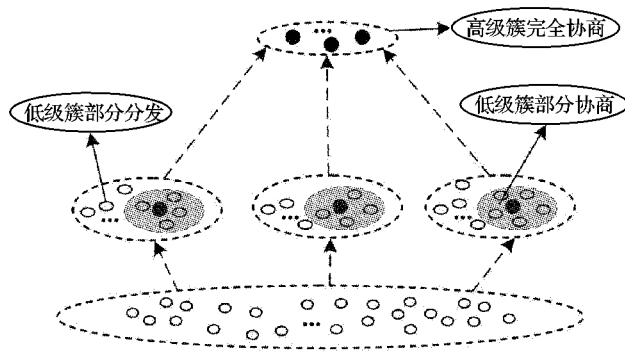


图1 基于分簇的混合式组密钥建立框架

根据移动自组网的特点, 在低级簇中, 如果完全采用集中式的密钥分发存在集中式信任和服务的可用性问题。如果完全采用分布式协商的方式, 虽然具有较高的安全性, 但由于低级簇内节点数目较多, 协商时不仅要求所有节点全部在线, 且会造成较大的计算量和通信量。为了增强方案的可用性并兼顾效率, 在低级簇内采用密钥协商和密钥分发相结合的方式

来生成组密钥。首先由簇首根据簇内节点的通信状况及可信度选择一些相对集中的节点连同自身一起组成低级簇协商组, 协商组生成低级簇簇密钥后, 再将簇密钥分发给未参与协商的节点。在高级簇中, 由于簇由低级簇的簇首组成, 节点间相对独立, 距离相对较远, 可能要经过多跳通信才能到达, 遭受攻击的可能性也比较大, 而簇首节点数量相对较少, 地位相对来说比较重要, 因此为了增强安全性并体现组密钥的共同参与性, 在高级簇采用完全分布式协商的方式来生成组密钥。

## 3 基于分簇的组密钥建立方案

组密钥的建立包含三个阶段: 第一个阶段是低级簇簇密钥的建立; 第二个阶段是高级簇簇密钥的建立, 即组密钥的生成; 第三个阶段是将生成的组密钥在低级簇中进行分发。

### 3.1 相关假设及符号定义

#### 3.1.1 相关假设

1) 假设已有一个基于节点可信度及能量的分簇算法来对网络分簇, 低级簇的簇首形成高一级的簇。

2) 假设簇首在本簇内周期性地广播簇内成员名单。

3) 假定簇内节点间的最大距离为两跳。

4) 假设节点向簇首注册成为簇成员时, 利用 DH 协议和簇首已建立了两方会话密钥。

#### 3.1.2 基本符号定义

$ID_i$ : 节点  $i$  的身份标志;

$n$ : 组中节点数;

$k$ : 低级簇的节点数;

$t$ : 低级簇中参与协商的节点数;

$u$ : 低级簇中未参与协商的节点数,  $u = k - t$ ;

$m$ : 高级簇中的节点数;

$e$ : 对映射  $G_1 \times G_1 \rightarrow G_2$ , 其中  $G_1$  和  $G_2$  是阶为大素数  $p$  的椭圆曲线群;

$P$ : 群  $G_1$  的基点;

$r_i$ : 节点  $i$  选取的随机数作为临时私钥,  $r_i \in [2, p-2]$ ;

$Q_i, Z_i$ : 节点  $i$  的临时公钥;

$x_i$ : 节点  $i$  的私钥,  $x_i \in [2, p-2]$ ;

$Y_i$ : 节点  $i$  的公钥,  $Y_i = x_i P$ ;

$CH_i$ : 第  $i$  个簇的簇首标志;

$K_g$ : 组密钥;

$C_s$ : 第  $s$  个簇;

$K_{C_s}$ : 第  $s$  个簇的簇密钥;

$K_{s,j}$ : 第  $s$  个簇的簇首和该簇内节点  $j$  共享的会话密钥;

$f(x)$ : 成员过滤函数;

$h$ : 三叉树的高度, 初始建立时  $h = \lceil \frac{m+1}{2} \rceil$ ;

$(l, v)$ : 三叉树中第  $l$  层的第  $v$  个节点,  $(l, 1)$  为该层的内部节点;

$K_{(l,v)}$ : 叶子节点  $(l, v)$  所在簇的簇私钥;

$Q_{(l,v)}$ : 叶子节点  $(l, v)$  所在簇的簇公钥,  $Q_{(l,v)} = K_{(l,v)} P$ ;

$k_l, bk_l$ : 分别为第  $l$  层内部节点的密钥和盲密钥, 其中  $bk_l = H(k_l) P$ ;

$g_l$ : 第  $l$  层的节点数;

$H$ : 强单向哈希函数,  $H: \{0, 1\}^* \rightarrow Z_p^*$ ;

$X_{co}(w)$ : 点  $w$  的横坐标值。

### 3.2 低级簇簇密钥的建立

在低级簇中,根据提出的组密钥建立框架,采用部分协商和部分分发相结合的方式建立簇密钥。首先由协商组成员利用扩展的认证的 BD 协议生成簇密钥,然后使用改进的成员过滤分发方法将簇密钥分发到未参与协商的节点。簇内成员身份标志分别为  $\{ID_1, ID_2, \dots, ID_k\}$ , 以簇  $C_1$  为例,簇密钥的建立过程如下。

1) 簇首根选择一些相对集中的节点连同自身一起组成低级簇协商组  $G = \{ID_1, ID_2, \dots, ID_t\}$ , 协商组形成一个环,然后簇首将协商组成员名单在簇内进行广播。

2) 协商组中每个节点  $ID_i$  选择随机数  $r_i$ ,  $1 \leq r_i \leq p-2$ , 计算并广播  $Q_i = r_i P$ 。

3) 协商组中每个节点  $ID_i$  收到  $Q_{i+1}$  和  $Q_{i-1}$  后,计算并广播  $Z_i = r_i(Q_{i+1} - Q_{i-1})$ 。

4) 协商组中每个节点  $ID_i$  计算会话密钥:

$$K_i = tr_i Q_{i-1} + (t-1)Z_i + (t-2)Z_{i+1} + \dots + Z_{i-2} = (r_1 r_2 + r_2 r_3 + \dots + r_{i-1} r_i + r_i r_1) P$$

5) 密钥确认。

① 协商组中每个节点  $ID_i$  计算出会话密钥后,计算并广播  $\sigma_i = H(ID_i, K_i, G)$ 。

② 协商组中每个节点  $ID_i$  计算  $H(ID_j, K_i, G)$  并与  $\sigma_j$  比较,如相等则所有协商组成员间建立起相同的簇密钥  $K_{C_1} = X_{\omega}(K_i)$ , 执行 7); 如不相等则簇密钥建立失败,表明有不诚实的参与者存在,执行 6)。

6) 不诚实参与者的鉴别。

① 协商组中每个节点  $ID_i$  广播本次协商中选择的  $r_i$  到组中其他节点。

② 协商组中每个节点  $ID_i$  收集到所有的  $r_j$ , 计算  $Q'_j = r_j P$  和  $Z'_j = r_j(Q_{j+1} - Q_{j-1})$ , 然后和以前收集到的节点  $ID_j$  广播的  $Q_j$  和  $Z_j$  进行比较,如相等,则不进行操作,否则,认为节点  $ID_j$  为一个不诚实节点,从协商组  $G$  中去除节点  $j$ 。最终协商组中所有诚实成员得到了相同的新协商组  $G'$ , 重新进行协商。

7) 簇首选择一个随机数  $r_c$ , 利用和未参与协商节点共享的会话密钥构造部分过滤函数如下:

$$f'(x) = (x - r_c)(x - H(K_{1,t+1}))(x - H(K_{1,t+2})) \dots (x - H(K_{1,k})) \pmod{p}$$

然后将其整理成标准形式的  $f'(x) = \prod_{i=0}^{k-t+1} a'_i x^i \pmod{p}$ , 使用簇密钥  $K_{C_1}$  加密  $f'(x)$  发送到协商组中的成员。

8) 协商组中的成员解密可得到  $f'(x)$ , 然后构造完整过滤函数  $f(x) = f'(x) + K_{C_1} \pmod{p}$ , 将其整理成标准形式

$$f(x) = \prod_{i=0}^{k-t+1} a_i x^i \pmod{p}, \text{ 由簇首指定其中一个节点在簇内广播 } f(x)。$$

9) 簇内未参与协商的节点  $ID_i (t+1 \leq i \leq k)$  收到此过滤函数  $f(x)$  后,计算其和簇首共享会话密钥的哈希值  $H(K_{1,i})$ , 并将其作为变量值代入到  $f(x) = \prod_{i=0}^{k-t+1} a_i x^i \pmod{p}$  中得到的结果就是簇密钥  $K_{C_1}$ 。

10) 未参与协商的节点  $ID_i (t+1 \leq i \leq k)$  计算  $H(ID_j, K_{C_1}, G)$ , 其中  $ID_j \in G$ , 并与收集到的协商组  $G$  中节点广播的  $\sigma_j$  进行比较: 如正确则获得了正确的簇密钥; 如不正确, 向  $G$

中其他成员申请分发簇密钥直至获得正确的簇密钥  $K_{C_1}$ 。

### 3.3 高级簇簇密钥的建立

高级簇中,采用完全协商的方式来建立簇密钥,由于节点间距离相对较远,通信开销较大,因此应重点考虑减少通信量。本文采用二叉树来代替 STR 协议<sup>[2]</sup>中的二叉树,使用 Joux 的三方密钥协商和两方椭圆曲线 DH 协商来代替 STR 中的两方 DH 协商。图 2 为六个成员的不平衡二叉树,最下层为第一层,根节点为第  $h$  层,叶子节点为成员节点。

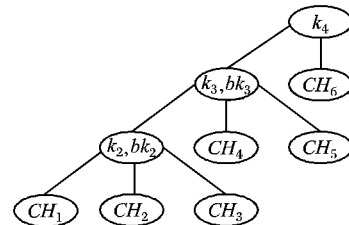


图2 六个成员的二叉不平衡密钥树

协商过程中各叶子节点并不贡献新的随机数,而是用其对应的低级簇簇密钥模  $p$  后的值代替,即  $K_{(l,v)} = K_{C_s} \pmod{p}$ 。初始时,根据节点数的不同,第  $h-1$  层的节点数可能为 2 或 3,而其以下层的节点数则都为 3(随节点的加入离开,以后除根外各层的节点数可能为 2 或 3)。高级簇初始簇密钥(即组密钥)建立过程如下。

1) 每个成员节点  $(l, v)$  计算并广播  $Q_{(l,v)} = K_{(l,v)} P$ 。

2) 成员节点  $(1, 1)$  计算所有内部节点的密钥值及盲密钥,其中  $bk_1 = Q_{(1,1)}$ , 计算:

$$k_2 = e(Q_{(1,2)}, Q_{(1,3)})^{K_{(1,1)}}, bk_2 = H(k_2) P,$$

$$k_3 = e(Q_{(2,2)}, Q_{(2,3)})^{H(k_2)}, bk_3 = H(k_3) P,$$

$\vdots$

$$k_{h-1} = e(Q_{(h-2,2)}, Q_{(h-2,3)})^{H(k_{h-2})}, bk_{h-1} = H(k_{h-1}) P,$$

计算  $k_h$ ,

$$\text{如果 } g_{h-1} = 2, \text{ 则 } k_h = H(k_{h-1}) Q_{(h-1,2)},$$

$$\text{如果 } g_{h-1} = 3, \text{ 则 } k_h = e(Q_{(h-1,2)}, Q_{(h-1,3)})^{H(k_{h-1})},$$

$$\text{然后广播 } BK = (bk_2, bk_3, bk_4, \dots, bk_{h-1})。$$

3) 高级簇中每个组成员  $(l, v)$ , 逐层向上计算:  $k_{l+1} = e(bk_l, Q_{(l,v)})^{K_{(l,v)}}, Q_{(l,v)}$  表示第  $l$  层的另一个叶子节点  $(l, v')$  (在第一层时不为  $(1, 1)$ ) 所在簇的簇公钥。

$$k_{l+2} = e(Q_{(l+1,2)}, Q_{(l+1,3)})^{H(k_{l+1})}$$

$\vdots$

$$k_{h-1} = e(Q_{(h-2,2)}, Q_{(h-2,3)})^{H(k_{h-2})}$$

$$\text{如果 } n_{h-1} = 2, \text{ 则 } k_h = H(k_{h-1}) Q_{(h-1,2)};$$

$$\text{如果 } n_{h-1} = 3, \text{ 则 } k_h = e(Q_{(h-1,2)}, Q_{(h-1,3)})^{H(k_{h-1})}。$$

4) 高级簇中采用和低级簇中相似的密钥确认方法进行密钥确认,以检测成员节点  $(1, 1)$  的诚实性。确认通过,高级簇中的节点都可以计算出组密钥  $K_g = X_{CO}(k_h)$ 。

### 3.4 组密钥的分发

当组密钥生成后,高级簇中节点将组密钥向下分发,分发过程如下:

1) 高级簇中的节点  $CH_i$  利用其所在的低级簇簇密钥  $K_{C_i}$  加密组密钥  $K_g$ , 然后组播发送到低级簇中的节点;

2) 低级簇中的节点使用其簇密钥解密得到组密钥;

至此,网络中的每个节点都拥有相同的组密钥。

### 3.5 认证的组密钥建立方案

在前面方案中,节点间协商过程中交换的信息缺乏认证,易遭受假冒、伪造、中间人攻击等,而且在密钥协商失败时,无法确定是由于信息遭篡改引起还是由于节点不诚实引起,进而导致不诚实节点的鉴别失败。因此,基于节点的公钥证书,本文提出了一个签名方案来提供认证性。

假设每个节点  $i$  都知道离线权威机构的公钥且拥有该机构颁发的数字证书  $Cert_i = \{ID_i, T_i, Y_i, e_i, (D_i, E_i)\}$ , 其中  $T_i$  为证书的签发时间和有效期,  $e_i$  为消息  $ID_i \parallel T_i \parallel Y_i$  经哈希运算后的摘要,  $(D_i, E_i)$  为  $e_i$  利用椭圆曲线数字签名生成的整数对。 $TS_i$  为时戳,  $S_i$  为签名项,其他符号见 3.1.2 节。

#### 3.5.1 签名过程

假定节点  $i$  要对信息  $M \in \{0,1\}^*$  进行签名,则:

- 1) 节点  $i$  选择一个随机数  $a_i, 2 \leq a_i \leq p-2$ ;
- 2) 计算  $R_i = a_i P, S_i = a_i + x_i H(R_i, M, TS_i) \bmod p$ ;
- 3) 节点  $i$  发送签名消息  $\{R_i, M, Cert_i, S_i, TS_i\}$ 。

#### 3.5.2 验证过程

节点  $j$  收到签名消息后,进行如下操作:

- 1) 检查时间戳  $TS_i$  的正确性;
- 2) 验证  $Cert_i$  的正确性;
- 3) 从  $Cert_i$  中取出节点  $i$  的公钥  $Y_i$  验证  $R_i = S_i P - Y_i H(R_i, M, TS_i)$  是否成立,如成立,接受签名,否则拒绝。

#### 3.5.3 签名方案正确性证明

$$\begin{aligned} S_i P - Y_i H(R_i, M, TS_i) &= \\ [a_i + x_i H(R_i, M, TS_i) \bmod p] P - Y_i H(R_i, M, TS_i) &= \\ a_i P + x_i H(R_i, M, TS_i) P - Y_i H(R_i, M, TS_i) &= \\ a_i P = R_i \end{aligned}$$

对于协商过程中发送的消息,使用提出的签名方案对其签名后再发送,可以确保消息的认证性,从而将提出的方案转化为一个认证的组密钥建立方案。

## 4 方案分析

### 4.1 安全性分析

#### 4.1.1 签名方案的安全性分析

签名方案中,假设攻击者想要找到一个  $S_i$  来使随后的消息验证通过,则攻击者先要选择一个随机数  $a_i$ , 计算  $R_i = a_i P$ , 然后,必须找到  $d$  使其满足验证等式  $dP = R_i + Y_i H(R_i, M, TS_i)$ , 但是找到  $d$  等价于解决椭圆曲线离散对数难题,从而攻击者很难找到这样一个  $d$ , 即无法伪造一个有效的签名消息来欺骗消息的接收者。

另外,攻击者获得某次签名  $S_i$ , 其想从签名等式  $S_i = r_i + x_i H(R_i, M, TS_i) \bmod p$  中得到用户的私钥  $x_i$  进而去签名消息。从签名等式可以看出,攻击者要想得到用户的私钥,必须得到  $r_i$ , 而  $r_i$  并没有直接在网络中传输,攻击者想要从  $R_i$  中计算出  $r_i$  等价于解决椭圆曲线离散对数难题,因此攻击者无法从签名过程中得到用户的私钥。

#### 4.1.2 组密钥建立方案的安全性分析

组密钥建立过程的安全性分析主要包括三部分。

##### 1) 低级簇簇密钥建立的安全性。

低级簇内协商组密钥协商采用的是扩展的椭圆曲线密钥体制的 BD 协议,协议的安全性基于椭圆曲线离散对数问题

(ECDLP) 和哈希数的单向性。基于 ECDLP 的难解性,攻击者无法从节点广播的  $Q_i$  和  $Z_i$  中计算出节点选择的随机数  $r_i$ , 自然也无法计算出协商组协商的簇密钥。基于哈希函数的单向性,攻击者无法从节点广播的密钥确认消息  $H(ID_i, K_i, G)$  中得出  $K_i$ , 也就得不到簇密钥。低级簇中簇密钥的分发是基于成员过滤函数的安全性,成员过滤函数实际上属于有限域上的模运算问题,根据文献[5]可以得出分发过程也是安全的。

##### 2) 高级簇簇密钥建立的安全性。

在高级簇中,各簇首使用低级簇的簇密钥模  $p$  后的值替代选取的随机数来进行协商。协商基于 Joux 的三方密钥协商协议和两方的椭圆曲线密钥交换协议进行,而这两者的安全性分别建立在 BDHP 和 ECDLP 的难解性之上。基于低级簇簇密钥协商的安全性,攻击者无法得到低级簇的簇密钥,也就无法计算出全网的组密钥,因此高级簇中的簇密钥也是安全的。

##### 3) 全网组密钥分发的安全性。

基于低级簇簇密钥的安全性,使用对称加密来分发高级簇协商的组密钥。除非攻击者能破解对称加密模式,否则组密钥的分发过程是安全的。

### 4.2 执行效率分析

下面从计算量和通信量两个方面来对所提方案的执行效率进行分析。计算量的大小通过方案中主要运算的总的执行时间来表示,而通信量的大小则通过方案中通信轮数、广播次数和发送的消息数来表示。由于 BD 协议和 STR 协议都未提供认证性且没有进行密钥确认,因此,对本文方案的开销进行统计时不计入签名和密钥确认所带来的开销。为了比较方便,假定低级簇中每个簇的节点数都为  $k$ , 则低级簇簇的个数为  $m = \lceil n/k \rceil$ , 高级簇中密钥树的高度为  $h = \lceil \frac{n+k}{2k} \rceil$ 。下面对所提方案的计算量和通信量进行统计,各种运算的符号定义如表 1 所示。

表 1 符号定义

符号	定义
$T_{\text{MUL}}$	执行一次模乘运算的时间
$T_{\text{EXP}}$	执行一次模指数运算的时间
$T_{\text{ADD}}$	执行一次模加运算的时间
$T_{\text{EC-MUL}}$	执行一次椭圆曲线上乘法运算的时间
$T_{\text{pair}}$	执行一次双线性对运算的时间

根据文献[5],低级簇簇密钥分发过程的计算量为  $O(u^2)T_{\text{MUL}} + O(u^2)T_{\text{ADD}}$ 。方案中协商过程的主要运算环节是模指数运算、椭圆曲线上的点乘运算、对运算,根据文献[6-7],  $T_{\text{EXP}} \approx 240T_{\text{MUL}}$ ,  $T_{\text{EC-MUL}} \approx 29T_{\text{MUL}}$ ,  $T_{\text{pair}} = 1.5 \sim 3T_{\text{EXP}}$ , 相对以上三种运算,忽略协商过程中其他运算产生的计算量。该方案的计算量和通信量与 BD 协议和 STR 协议的比较如表 2 所示,其中  $h = \lceil \frac{n+k}{2k} \rceil$ 。从表 2 中可以看出,所提方案与

BD 和 STR 协议相比具有较高的计算效率。通信量方面,本方案的通信轮数虽然较 BD 和 STR 协议有所增加,但方案中节点只在簇内进行广播,并不像 BD 和 STR 协议那样需要在全网进行广播,从而大大地减少了需要转发的消息,有效地减少了通信开销。另外,当节点的加入与离开引起组密钥更新时,本方案中只需对受影响簇的簇密钥进行更新,而不需要对

其他簇的簇密钥进行更新,从而能够大大减少组密钥更新时的计算和通信开销。限于篇幅,本文对组密钥更新不作详细讨论。

表 2 方案的执行效率比较 ( $O(k)$  表示计算复杂度为  $k$ )

方案	通信轮数	交换消息数	总的计算量
BD 协议	2	$2n$	$O(n^2)T_{\text{EXP}}$
STR 协议	2	$n+1$	$O(n^2)T_{\text{EXP}}$
本文方案	7	$2n - \frac{2m \cdot n - 4n - k}{k}$	$O(nk)T_{\text{EC-MUL}} + O(k^2)T_{\text{pair}}$

#### 4.3 可用性分析

本文方案中,采用分簇结构,在各簇中建立独立的簇密钥,使得一个簇的成员关系变化不会影响到其他簇的簇密钥。在低级簇簇密钥建立时,采用部分协商和部分分发相结合,不要求协商时簇内所有节点同时在线,同时避免了由单个节点生成并分发组密钥所带来的集中式信任和服务可用性问题。对不诚实节点进行的检测和鉴别可以防止内部不诚实节点所造成的组密钥建立失败,以上这些都大大增强了方案的可用性。

## 5 结语

随着移动 Ad Hoc 网络的应用和发展,其组密钥管理问题日益成为研究的热点。本文提出了一个基于分簇的认证的安全高效的组密钥建立方案,在低级簇簇密钥的建立过程中,采用部分协商和部分分发相结合的方式建立簇密钥,利用椭圆曲线密码体制对 BD 协议进行改进和扩展,给出了一个能够检测并鉴别不诚实节点的协商协议,并采用改进的成员过滤方法由簇内参与协商成员向未参与协商成员分发簇密

钥。在组密钥的协商过程中,采用二叉树结构进行协商,协商过程中利用两方 ECDH 和 Joux 的三方协商来代替 STR 中的两方 DH。另外,本方中还提出了一个签名方案用于在密钥协商过程中提供认证。与 BD 和 STR 方案相比,本方案有效地降低了组密钥建立过程中的计算开销和通信开销,并增强了方案的安全性、可扩展性和可用性。

#### 参考文献:

- [1] BURMESTER M, DESMETS Y. A secure and efficient conference key distribution system[C]// Proceedings of Eurocrypt'94. Berlin: Springer-Verlag, 1994: 275-286.
- [2] KIM Y, PERRIG A, TSUDIK G. Group key agreement efficient in communication[J]. IEEE Transactions on Computers, 2004, 53(7): 905-921.
- [3] TSAUR W-J. Dynamic key management scheme for secure group communication based on hierarchical clustering in mobile Ad Hoc networks[C]// ISPA 2007, LNCS 4743. Berlin: Springer, 2007: 475-484.
- [4] JOUX A. A one round protocol for tripartite Diffie-Hellman[J]. Journal of Cryptology, 2004, 17(4): 263-276.
- [5] WU KUEN-PIN, RUAN SHANG-JANG, LAI FEIPEI. On key distribution in secure multicasting[C]// 25th Annual IEEE Conference on Local Computer Networks. New York: IEEE, 2000: 208-212.
- [6] BARRETO P S L M, KIM H Y, LYNN B, et al. Algorithms for pairing-based cryptosystems[C]// Advances in Cryptology: Crypto 2002, LNCS 2442. Berlin: Springer-Verlag, 2002: 354-368.
- [7] KOBLITZ N, MENEZES A J, VANSTONE S A. The state of elliptic curve cryptography[J]. Designs, Codes and Cryptography, 2000, 19(2): 173-193.

(上接第 2212 页)

#### 4.2.3 加密速度

由表 4 可看出本文提出的加密算法具有较高的运算效率,主要有以下两个原因。

1) 本算法加密只包含异或运算、加法、正弦值(可通过查表实现)等简单的算术运算以及少数次的 DES 加密处理。对于每个像素来说,在一轮加密中只有一次异或、一次简单的波加密操作。然而对于文献[3]的算法,在 1 轮加密中需要 3 次标准映射、1 次混沌迭代;文献[4]的算法需要 4 次标准映射(同时进行了值替换)和 1 次混沌迭代。

2) 本算只要两轮加密就足以得到很好的加密性能和安全性。而文献[3]的算法需要 6 轮,文献[4]的算法虽然只需要 1 轮,但其 1 轮的操作比本文提出的算法的计算量大很多。

因此,从加密数据量和加密速度来看,本算法具有较高的性能。

## 5 结语

本文提出了一种新的基于波传播的图像加密算法。该算法易于实现、计算简单,并可同时实现高速、高安全性、高敏感性。波传播加密,顾名思义,是一种通过模拟波传播来改变像素灰度值的方法。仿真实验证明了本算法的高性能(尤其在敏感性和速度上)和安全性。可以很明显地看出,NPCR 和密钥敏感性在第一轮加密后就已非常高了,只需 2 轮加密就能完全满足加密的性能和安全性要求。

#### 参考文献:

- [1] MANICCAM S S, BOURBADIS N G. Lossless image compression and encryption using SCAN[J]. Pattern Recognition, 2001, 34(6): 1229-1245.
- [2] CHENG H, LI XIAOBO. Partial encryption of compressed images and videos[J]. IEEE Transactions on Signal Processing, 2000, 48(8): 2439-2451.
- [3] LIAN SHIGUO, SUN JINSHENG, WANG ZHIQUEN. A block cipher based on a suitable use of the chaotic standard map[J]. Chaos, Solitons & Fractals, 2005, 26(1): 117-129.
- [4] WONG K-W, KWOK B S-H, LAW W-S. A fast image encryption scheme based on chaotic standard map[J]. Physics Letters A, 2008, 372(15): 2645-2652.
- [5] WONG K-W, KWOK B S-H, YUEN C-H. An efficient diffusion approach for chaos-based image encryption[EB/OL]. [2009-02-01]. <http://lib.physcon.ru/download/p1264.pdf>
- [6] CHEN R-J, CHEN Y-H, CHEN C-S, et al. Image encryption/decryption system using 2-D cellular automata[C]// IEEE Tenth International Symposium. [S.l.]: IEEE, 2006: 1-6.
- [7] 张晓岩, 王超, 孙志人, 等. 基于二维 CA 和 CWQ 方法的图像加密方案[J]. 南京师大学报: 自然科学版, 2008, 31(1): 1-7.
- [8] CHEN R-J, LAI J-L. Image security system using recursive cellular automata substitution[J]. Pattern Recognition, 2007, 40(5): 1621-1631.
- [9] CHEN LINFEI, ZHAO DAOMU. Image encryption with fractional wavelet packet method[J]. Optik - International Journal for Light and Electron Optics, 2008, 119(6): 286-291.