自认证公钥体制Ad hoc网络密钥管理方案

胡荣磊[1, 2]，刘建伟[1]，张其善[1]

(1. 北京航空航天大学电子信息工程学院 北京 海淀区 100191; 2. 北京电子科技学院通信工程系 北京 丰台区 100070)

【摘要】基于自认证公钥体制和门限密码机制，为Ad hoc网络提出了一种新的分布式密钥管理方案。方案中节点公钥具有自认证功能，不需要证书管理，降低了网络节点的存储和通信需求；解决了基于身份公钥体制方案中的密钥托管问题，提高了系统安全性；将组合公钥的思想引入到门限密钥分发的随机数选择，简化了传统ElGamal型门限签名方案在签名前协商随机数的过程，大大降低了网络节点的通信量和计算量。分析表明，同以往提出的基于公钥密码体制的密钥管理方案相比，该方案有更高的效率和安全性。

关 键 词 Ad hoc网络; 密码学; 数据保密; 密钥管理; 自认证公钥

中图分类号 TN915.08; TN918 **文献标识码** A doi:10.3969/j.issn.1001-0548.2009.06.010

Key Management Scheme for Ad hoc Networks Using Self-Certified Public Key System

HU Rong-lei[1,2], LIU Jian-wei[1], and ZHANG Qi-shan[1]

(1. School of Electronics and Information Engineering, Beijing University of Aeronautics and Astronautics Haidian Beijing 100191;

2. Department of Communication Engineering, Beijing Electronic Science and Technology Institute Fengtai Beijing 100070)

Abstract A new distributed key management scheme based on self-certified public key system and threshold cryptography is proposed for Ad hoc network. The storage space and the communication overheads can be reduced because the public key is self certified and the certificate is unnecessary. There is no key escrow problem since the key distribution center (KDC) does not know the users' private keys. The idea of composite public key (CPK) is introduced for selecting random number for threshold key distribution. It reduces the process of generating a random number before threshold signature is issued in traditional ElGamal type threshold signature and so it reduces the communication and computation overheads of network nodes. The analysis shows that the scheme is more secure and efficient than previous works implemented with public key systems.

Key words Ad hoc networks; cryptograph; data privacy; key management; self-certified key

Ad hoc网络的特点使其安全问题比传统无线网络更难解决，节点密钥管理作为安全问题的基础，逐渐成为研究热点。目前为Ad hoc网络提出的非对称密码体制的密钥管理方案有：基于证书公钥密码体制(certificate based public key system，CBS)方案[1-2]、基于身份的公钥密码体制(identity based public key system，IDBS)[3]方案和基于自认证公钥体制(self-certified public key based system，SCKBS)的方案[4-6]。CBS方案需要一个部分分布式[1]或完全分布式[2]的认证中心(certificate authority，CA)。节点证书的获得需要多个CA节点共同协作完成，每个CA节点对证书进行部分签名。CA节点需要保存所有节点的证书，当节点证书更新或撤销时，每个节点的证书目录都需要同步刷新。这种证书管理的方法对网络节点的计算、存储和通信要求都比较高。IDBS方案不需要证书管理，每个节点身份信息(如Email、IP、MAC地址等)皆可作为节点公钥，并且不需要有一个公共文件存储公钥信息。但是IDBS方案由分布式密钥分发中心(key distribution center，KDC)统一产生用户私钥，KDC知道所有用户的私钥，使得IDBS方案初始就具有密钥托管(key escrow)的功能，安全性大大降低。文献[7]提出自认证公钥的概念。SCKBS方案与IDBS方案类似，不需要证书保证公钥的可靠性，公钥自身具有认证功能。它也要依赖于KDC，但是KDC不直接生成用户私钥，只产生与用户身份对应的部分私钥，用户自己把部分私钥和一些秘密信息结合，得到实际私钥，有效地解决了密钥托管问题。SCKBS方案简化了CBS方案中的证书管理，

收稿日期：2008 – 06 – 12；修回日期：2009 – 01 – 13
基金项目：国家自然科学基金(60672102)；北京电子科技学院重点实验室基金(YZDJ0710，YZDJ0805)
作者简介：胡荣磊(1977 –)，男，在职博士生，工程师，主要从事无线网络安全方面的研究.

减少了计算量和通信量，提高了执行效率，而且可以避免IDBS方案的密钥托管问题，实现公钥和签名的验证过程在逻辑单步内同时完成，因而SCKBS方案在Ad hoc网络中具有良好的应用前景。

目前为Ad hoc网络提出的SCKBS密钥管理方案非常少，并且这些方案的计算量和通信量都比较大，因为在这些方案中，有一个重要的问题没有解决。SCKBS方案基于ElGamal型门限签名(包括ECC ElGamal型签名)，在签名前需要签名各方协商一个随机数，该过程需要多个节点相互配合，多次通信，运算量也比较大，大大降低了密钥分发的效率和成功率。本文提出一种新的SCKBS方案，引入了组合公钥(composite public key，CPK)[8]思想，很好地解决了这个问题，使节点运算量和通信量大大降低。

1 自认证公钥体制

文献[7]提出自认证公钥概念后，一些关于SCKBS的方案被陆续提出[9-11]。文献[9]提出的方案比较简单且具有代表性。该方案采用单KDC结构，系统参数 p 和 q 是两个大素数，且 $q \mid p-1$，Z_q^* 是 Z_p^* 的 q 阶子群，g 是群 Z_p^* 中阶为 q 的生成元，p、q 和 g 是公开的。KDC拥有公/私钥对 (k_s, k_p)，其中 $k_p = g^{k_s} \bmod p$。

KDC向用户 A 分发密钥的步骤为：

(1) KDC选取随机数 $\ell \in Z_q^*$，计算 $\omega' = g^{\ell} \bmod p$，传送 ω' 给用户 A。

(2) 用户 A 选取随机数 $\alpha \in Z_q^*$，计算 $\omega = \omega' g^{\alpha} \bmod p$，传送 (I_A, ω) 给KDC，I_A 为用户 A 的身份标识。

(3) KDC用自己的私钥 k_s 和Schnorr签名方案计算参数 $x'_A = k_s h(I_A, \omega) + \ell$，传送 x'_A 给 A，其中 $h(\cdot)$ 为单向hash函数。

(4) A 计算 $x_A = (x'_A + \alpha) \bmod q$，通过等式 $g^{x_A} = k_p{}^{h(I_A,\omega)} \omega \bmod p$ 校验其有效性。如果校验通过，则 A 接受 $(x_A, y = g^{x_A} \bmod p)$ 为其公/私钥对。

2 提出的方案

2.1 系统模型

网络在系统初始化阶段需要一个可信第三方(trusted third party，TTP)完成网络组建和系统参数设置，初始化完成后，TTP退出网络。网络在运行阶段，需要一个分布式KDC，由网络中 n 个节点共同承担，系统私钥采用 (t,n) 门限管理方式，即由 n 个网络节点共享，每个节点持有部分私钥，任意 $t(t \leqslant n/2)$ 个节点联合可以恢复系统私钥，但任何少于 t 个节点联合都不能恢复私钥。

对网络节点属性需要做一些假设：(1) 节点在网络期间有唯一的ID号，唯一是指不同节点的ID号不能冲突，并且在网络期间保持不变，节点不能冒充使用别的节点的ID号，比如IP地址。(2) 节点具有发现邻居节点，并能获得邻居节点身份信息的功能。(3) 节点具有监控邻居节点的功能，可以发现邻居节点是否为恶意节点或可疑节点。

2.2 网络初始化

网络初始化阶段如图1所示。

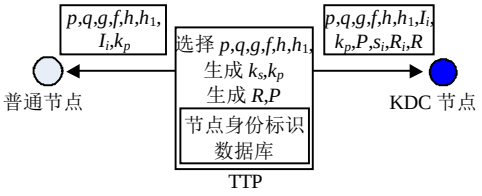

图1 网络初始化阶段

初始化过程如下：

(1) TTP选择如上节所述的系统参数 p、q、g。TTP选择安全的hash函数 h，h_1 和安全单向函数 f。公布参数 (p,q,g,h,h_1,f)。

(2) TTP选择网络中 n 个节点作为分布式KDC节点，记为 $\{u_1, u_2, \cdots, u_n\}$。

(3) TTP产生系统公/私钥对 $(k_s, k_p = g^{k_s} \bmod p)$，公布 k_p，利用文献[12-13]可验证秘密共享方案，将私钥 k_s 分割成 n 份，秘密分发给 n 个KDC节点，每个节点 u_i 仅持有部分私钥 s_i。具体步骤如下。

步骤 1 TTP选择一个 $t-1$ 次多项式 $f_T(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1}$，满足 $f_T(0) = a_0 = k_s$，其中 $a_i \in Z_q^*$ $(i = 0,1,\cdots,t-1)$；

步骤 2 TTP计算节点 u_i 的部分私钥 $s_i = f_T(I_i)$ $(i = 1,2,\cdots,n)$，其中 I_i 为节点 u_i 的身份标识。将 s_i 秘密发送到 u_i，并广播 $g^{a_i} \bmod p$ $(i = 0,1,\cdots,t-1)$，其中 $g^{a_0} \bmod p = k_p$；

步骤 3 u_i 收到 s_i 后，验证等式：

$$g^{s_i} = \prod_{j=0}^{t-1} (g^{a_j})^{(I_i)^j} \bmod p \tag{1}$$

如果验证失败，则广播 s_i，并对TTP提出投诉；

步骤 4 利用式(1)，其他的KDC节点验证提出投诉的 s_i，如果验证通过，拒绝提出投诉的KDC节点；如果验证失败，则要求TTP重新计算并传送 s_i 给节点 u_i。

(4) TTP产生一个随机数矩阵 $\boldsymbol{R}$ 和相应的随机数公钥矩阵 $\boldsymbol{P}$：

$$\boldsymbol{R}=\begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1v} \\ r_{21} & r_{22} & \cdots & r_{2v} \\ \vdots & \vdots & \cdots & \vdots \\ r_{u1} & r_{u2} & \cdots & r_{uv} \end{pmatrix}$$

$$\boldsymbol{P}=\begin{pmatrix} g^{r_{11}} \bmod p & g^{r_{12}} \bmod p & \cdots & g^{r_{1v}} \bmod p \\ g^{r_{21}} \bmod p & g^{r_{22}} \bmod p & \cdots & g^{r_{2v}} \bmod p \\ \vdots & \vdots & \cdots & \vdots \\ g^{r_{u1}} \bmod p & g^{r_{u2}} \bmod p & \cdots & g^{r_{uv}} \bmod p \end{pmatrix}$$

其中 $r_{ij} \in Z_q^* (i=1,2,\cdots,u, j=1,2,\cdots,v)$。TTP公布$\boldsymbol{P}$，使用与过程(3)相同的方案将$\boldsymbol{R}$中的每个元素 r_{ij} 分割成 n 份，秘密分发给 n 个KDC节点。其中 $f_T(x)$ 可以采用过程(3)中使用的多项式，也可另选。节点 u_i 仅持有部分随机数矩阵$\boldsymbol{R_i}$:

$$\boldsymbol{R_i}=\begin{pmatrix} (r_{11})_i & (r_{12})_i & \cdots & (r_{1v})_i \\ (r_{21})_i & (r_{22})_i & \cdots & (r_{2v})_i \\ \vdots & \vdots & \cdots & \vdots \\ (r_{u1})_i & (r_{u2})_i & \cdots & (r_{uv})_i \end{pmatrix}$$

密钥初始化过程可以不需要TTP，由网络节点使用多方可验证秘密共享方案[12-13]产生系统公/私钥对和随机数矩阵，但这样计算量和通信量比较大。因为TTP在初始化完成后退出网络，所以不影响网络的安全性。

2.3　分布式密钥服务

分布式密钥服务是指网络节点向分布式KDC申请获取或更新公/私钥对，如图2所示。图中●为KDC节点，○为普通节点。

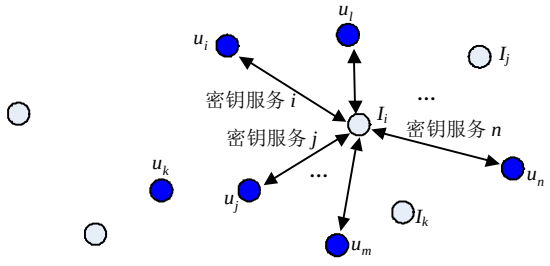

图2　网络运行阶段的密钥服务

假设用户 A 向KDC申请公/私钥对 $(x_A, y_A = g^{x_A} \bmod p)$，该过程描述如下：

(1) A 选择随机数 $\alpha \in Z_q^*$，广播 $g^{\alpha} \bmod p$、I_A 和 $Q_A = h_1(I_A \| \text{expire\_time})$ 提出密钥申请。I_A 为用户 A 身份标识。expire_time 为密钥失效时间，是由系统确定的，在同一时间段内对所有的节点都相同。

(2) 收到密钥申请的KDC节点 u_i 计算 $l_z = f^{(z)}(Q_A \| g^{\alpha}) \bmod u \ (z=1,2,\cdots,v)$，$f$ 是安全单向函数，可以采用hash函数，$f^{(z)}$ 表示函数 f 计算 z 次。u_i 在自己持有的$\boldsymbol{R_i}$的第 $z(z=1,2,\cdots,v)$ 列取出第 l_z 个随机数，然后求和，则 u_i 为部分签名选取的随机数为 $\ell_i = \sum_{z=1}^{v} (r_{lz})_i \bmod q$，$u_i$ 同时将随机数公钥矩阵$\boldsymbol{P}$中 v 个相同位置的数值选出，做模乘运算：$\prod_{z=1}^{v} g^{r_{lz}} \bmod p = g^{\sum_{z=1}^{v} r_{lz}} \bmod p = g^{\ell}$，$\ell = \sum_{z=1}^{v} r_{lz}$ 即是KDC为密钥分发签名在$\boldsymbol{R}$中选定的随机数。

(3) u_i 用持有的部分私钥 s_i 计算签名，仍采用Schnorr签名方案 $(x'_A)_i = s_i h(Q_A, g^{\ell+\alpha}) + \ell_i$，$u_i$ 将 $(x'_A)_i$、$g^{\ell+\alpha} \bmod p$ 传送给 A。

(4) A 收到 t 个以上的KDC节点的签名信息后，计算自己的私钥 $x_A = \sum_{i=1}^{t} \lambda_i (x'_A)_i + \alpha = k_s \cdot h(Q_A, g^{\ell+\alpha}) + \ell + \alpha$，其中 λ_i 为Lagrange系数。设 $\omega = g^{\ell+\alpha}$，A 校验等式 $g^{x_A} = k_p^{h(Q_A,\omega)} \omega \bmod p$，如果校验通过，则接收 $(x_A, y_A = g^{x_A} \bmod p)$ 为其公/私钥对。因为有：

$$x_A = \sum_{i=1}^{t} \lambda_i (x'_A)_i + \alpha =$$

$$\sum_{i=1}^{t} \lambda_i [s_i \cdot h(Q_A, g^{\ell+\alpha}) + \ell_i] + \alpha =$$

$$\sum_{i=1}^{t} \lambda_i s_i \cdot h(Q_A, g^{\ell+\alpha}) + \sum_{i=1}^{t} \sum_{z=1}^{v} \lambda_i (r_{lz})_i + \alpha =$$

$$k_s \cdot h(Q_A, g^{\ell+\alpha}) + \sum_{z=1}^{v} r_{lz} + \alpha =$$

$$k_s \cdot h(Q_A, g^{\ell+\alpha}) + \ell + \alpha =$$

$$k_s \cdot h(Q_A, \omega) + \ell + \alpha \tag{2}$$

$$g^{x_A} = g^{k_s \cdot h(Q_A,\omega)} \cdot g^{\ell+\alpha} =$$

$$k_p^{h(Q_A,\omega)} \cdot \omega \bmod p \tag{3}$$

所以可知上述方案的正确性。

2.4　系统私钥与随机数矩阵更新

如果网络存在的时间足够长，攻击者可能会捕获 t 个以上KDC节点，从而计算出系统私钥 k_s 和随机数矩阵$\boldsymbol{R}$。为了对抗这种攻击，可以在不改变 k_s 和$\boldsymbol{R}$的前提下由所有的KDC共同周期性地、自主地更新 s_i 和$\boldsymbol{R_i}$。更新基于以下原理[14]：如果 $(s_1^1, s_2^1, \cdots, s_n^1)$ 是秘密 k_1 用 (t,n) 门限机制分割后得到的子秘密，而 $(s_1^2, s_2^2, \cdots, s_n^2)$ 是秘密 k_2 的 (t,n) 子秘密，那么 $(s_1^1 + s_1^2, s_2^1 + s_2^2, \cdots, s_n^1 + s_n^2)$ 是 $k_1 + k_2$ 的 (t,n) 子秘密，令 k_2 为0，就得到 k_1 的一个新的 (t,n) 共享分割。攻击者不能联合旧的子秘密与新的子秘密求出秘密 k_1。$\boldsymbol{R}$包含 uv 个元素，每次更新矩阵所有的元素计算量和通信量太大，根据CPK的特性[8]，每次只更新$\boldsymbol{R}$的一列，可以达到相同的效果。更新过程需要所

有KDC节点联合执行。假设更新矩阵$\boldsymbol{R}$的第β $(1\leqslant\beta\leqslant v)$列：

(1) 每个节点u_i $(i=1,2,\cdots,n)$选取随机多项式$f_i(x)=a_{i,1}x+\cdots+a_{i,t-1}x^{t-1}$，$a_{i,k}\in Z_q^*(k=0,1,\cdots,t-1)$，计算$f_i(I_j)\bmod q$ $(j=1,2,\cdots,n)$，将其秘密发送到u_j。

(2) 每个u_i $(i=1,2,\cdots,n)$收到其他的$n-1$个节点传送的数据，计算$s_i'=\left[s_i+\sum_{z=1}^{n}f_z(I_i)\right]\bmod q$，生成新的部分私钥$s_i'$。

(3) 每个u_i $(i=1,2,\cdots,n)$对矩阵$\boldsymbol{R_i}$中第β列的每个元素都加上$\sum_{z=1}^{n}f_z(I_i)\bmod q$，生成新的部分随机数矩阵$\boldsymbol{R_i}'$。

为了增强方案的安全性，在更新部分私钥s_i和部分随机数矩阵$\boldsymbol{R_i}$时，可以选用不同的随机多项式。

2.5 密钥撤销

为了增强网络的效率和安全性，规定每个网络节点的公/私钥对在同一时间段内，即同一个expire_time只允许更新一次。KDC节点保存它已经分发过密钥的节点的Q_A值，对于已有的Q_A值不再分发相同时间段内的密钥，Q_A在expire_time到期时清除。

网络节点在其密钥失效前，要向KDC节点申请更新自己的公/私钥对，如果节点退出网络，则不再申请密钥。如果某个节点被监测为恶意节点，该节点的信息在全网公布，其他的节点不再与其通信，KDC也不再为此节点分发密钥，其现有密钥到期后自动退出网络。

3 方案分析

3.1 安全性

本文提出的密钥分发方案包含Pedersen可验证秘密共享方案、门限Schnorr签名和CPK技术3部分内容。文献[12]和文献[8]分别对Pedersen可验证秘密共享方案和CPK的安全性给出了证明。CPK的核心优势是由少量的种子密钥可以组合成海量的密钥空间(当u=32，v=32时，矩阵$\boldsymbol{R}$拥有1024个元素，随机数容量可达到$(2^5)^{32}=2^{160}\approx10^{48}$)[8]，从攻击者的角度讲，签名者从$\boldsymbol{R}$中选择随机数，与从$Z_q^*$中选择随机数是不可区分的。文献[4]给出了一种门限Schnorr签名方案在Random Oracle模型下的安全性证明，采用同样的方法可以证明本文提出的门限签名方案也是安全的。基于以上基础方案的安全性：

(1) 在网络初始化时，每个KDC节点在可验证秘密共享方案中不能获得其它KDC节点的s_i和$\boldsymbol{R_i}$，也不能通过门限Schnorr签名信息获得系统私钥k_s的任何信息。同样，其他的网络节点也不能获得系统私钥的任何信息。

(2) 在IDBS门限密钥分发方案中，t个以上KDC节点共谋可以获得所有用户的私钥，这是IBE方案的一个弱点。而本文方案中，即使t个以上KDC共谋只能获得用户私钥的一部分$\sum_{i=1}^{t}\lambda_i(x_A')_i$，而不能获得用户私钥$x_A$，KDC获得私钥的难度相当于求解离散对数难题。因此本文方案有更高的安全性。

(3) 节点可以对其私钥的正确性进行检验，如果KDC或攻击者篡改参数g^{α}、$(x_A')_i$或者ω，可以被及时检测出来，因此节点私钥不能被伪造。

(4) 节点不能私自更换为其分发的公/私钥对，它的难度相当于在不知道私钥的情况下模仿Schnorr签名。

虽然在初始化阶段有TTP的存在，但TTP并不知道网络节点的公/私钥对，并且在初始化后立刻退出网络，并不影响系统的安全性。方案中每个KDC节点掌握的s_i和$\boldsymbol{R_i}$都是独立的，单个节点失效或密钥泄漏对其它节点间的认证没有影响。根据网络的规模设定门限值t，只要密钥泄漏的KDC节点数小于t个，系统就是安全的。并且KDC节点可以根据不同的安全等级，设定系统私钥与随机数矩阵更新的时间间隔T，攻击者需要在一个更新周期T内破解大于等于t的节点，由此系统有很强的抗毁性。

3.2 效率

文献[4-6]都是为Ad hoc网络提出的SCKBS密钥分发方案。本文方案在KDC节点收到密钥申请后，只需要自己计算，不需要确定由哪些节点产生部分密钥，节点间也不需要交互通信，任何收到密钥申请的KDC节点都可以产生部分密钥，可以避免随机数协商过程中出现的错误，也能快速有效地解决产生部分密钥过程中出现的错误，因此有更高的效率和成功率。在大规模的Ad hoc$(t>v)$网络中，该方案更能体现出其优越性，如表1所示。

表1 门限密钥分发效率比较

方案	模幂运算(次/节点)	模乘/除运算(次/节点)	加/减运算(次/节点)	通信次数(收+发/节点)
文献[4]方案	$8t^2-7t+2$	$6t^2-2t+2$	$2t^2+2t-1$	$5t-1$
文献[5]方案	$t+2$	$2t+4$	t	t
文献[6]方案	$t+2$	$2t+2$	t	t
本文方案	1	$v+t$	$v+t-1$	2

表中，t为签名节点个数；v为随机数矩阵列数。表中给出了几个方案在一次门限密钥分发过程中，每个KDC节点的计算量和通信次数的比较，其中忽略了hash函数和因产生错误而附加的计算量。可以看出本文方案计算量小，通信次数少。这是因为以前提出的方案在门限密钥分发之前，都要进行随机数协商，需要事先确定由哪些KDC节点协作完成，协商过程中节点不能随意更换，如果因某种原因更换节点，则整个协商过程要重新开始，效率比较低。

4 结 束 语

本文基于自认证公钥机制和门限密码机制，为Ad hoc网络提出了一种安全性和效率都比较高的密钥管理方案。该方案不需要证书管理，解决了CBS方案中网络节点存储证书的负担，同时避免了因传递证书而造成的通信开销；方案解决了IDBS方案中的密钥托管问题，提高了系统安全性；方案简化了传统ElGamal型门限签名方案协商随机数的过程，同以往提出的基于自认证公钥体制的密钥管理方案相比，减轻了节点计算量和通信量。本文提出的方案也可以使用椭圆曲线密码ECC机制实现，使节点计算量和通信量更小。

参 考 文 献

[1] ZHOU L, HAAS Z J. Securing Ad hoc networks[J]. IEEE Network Journal, 1999, 13(6): 24-30.

[2] LUO H, LU S. Ubiquitous and robust authentication service for Ad hoc wireless network[R]. [S.l.]: UCLA Computer Science Technical Report 200030, 2000.

[3] KHALILI A, KATZ J, ARBAUGH W. Toward secure key distribution in truly Ad hoc networks[C]//2003 Symposium on Applications and the Internet Workshops (SAINT 2003). Orlando, Florida: IEEE Computer Society, 2003: 342-346.

[4] STINSON D R, STROBL R. Provably secure distributed Schnorr signatures and *a* (*t*,*n*) threshold scheme for implicit certificates[C]//Information Security and Privacy ACISP'01, Lecture Notes in Computer Science, LNCS2119. London: Springer-Verlag, 2001: 417-434.

[5] LI F, XIN X, HU Y. Key management in Ad hoc networks using self-certified public key system[J]. International Journal of Mobile Communications, 2007, 5(1): 94-106.

[6] 李海峰, 刘云芳. 移动Ad hoc网络中应用自认证的(*t*,*n*)门限群签名方案[J]. 北京联合大学学报, 2006, 20(3): 19-22.
LI Hai-feng, LIU Yun-fang. The application of self- certified (*t*,*n*) threshold group signature scheme in mobile Ad hoc networks[J]. Journal of Beijing Union University (Natural Sciences), 2006, 20(3): 19-22.

[7] GIRAULT M. Self-certified public keys[C]//Advances in Cryptology-Eurocrypt'91, Lecture Notes in Computer Science, LNCS547. New York: Springer-Verlag, 1991: 490-497.

[8] 唐　文, 南湘浩, 陈　钟. 基于椭圆曲线密码系统的组合公钥技术[J]. 计算机工程与应用, 2003, 21: 1-3.
TANG Wen, NAN Xiang-hao, CHEN Zhong. Elliptic curve cryptography-based combined public key technique[J]. Computer Engineering and Applications, 2003, 21:1-3.

[9] PETERSEN H, HORSTER P. Self-certified keys-concepts and application[C]//Third Conference on Communication and Multimedia Security. Athens: Chapman & Hall, 1997: 102-116.

[10] CHANG Y, WU T C, HUANG S C. ElGamal-like digital signature and multi-signature schemes using self-certified public keys[J]. Journal of Systems and Software, 2000, 50(2): 99-105.

[11] HSU C L, WU T S. Self-certified threshold proxy signature schemes with message recovery, non-repudiation, and traceability[J]. Applied Mathematics and Computation, 2005, 164(1): 201-225.

[12] PEDERSEN T P. A threshold cryptosystem without a trusted party[C]//Advances in Cryptology-Eurocrypt'91, Lecture Notes in Computer Science, LNCS547. New York: Springer-Verlag, 1991: 522-526.

[13] PEDERSEN T P. Distributed provers with applications to undeniable signatures[C]//Advances in Cryptology-Eurocrypt'91, Lecture Notes in Computer Science, LNCS547. New York: Springer-Verlag, 1991: 221-242.

[14] HERZBERG A, JARECKI S, KRAWCZYK H, et al. Proactive secret sharing or: how to cope with perpetual leakage[C]//Advances in Cryptology-Crypto'95. Berlin: Springer-Verlag, 1995: 339-352.

编　辑　蒋　晓