文章编号:1001 - 9081(2007)03 - 0611 - 03

基于簇结构的 Ad Hoc 网络安全密钥管理方案

叶永飞,余梅生
(燕山大学 信息科学与工程学院,河北 秦皇岛 066004)
(yeyongfei005@126.com)

摘 要:针对移动自组网络,提出了一种基于簇结构的分布式安全密钥管理方案,将系统私钥与簇私钥结合起来,增强了网络的认证、机密性、可用性及鲁棒性等多方面的安全性。

关键词:移动 Ad Hoc 网;簇结构;(r, n)门限方案;密钥更新
中图分类号: TP393.08 **文献标识码**:A

Secure key management scheme based on cluster architecture in mobile Ad Hoc networks

YE Yong-fei, YU Mei-sheng
(*College of Information Science and Engineering, Yanshan University, Qinhuangdao Hebei* 066004, *China*)

Abstract: Mobile Ad Hoc Network (MANET) is a new wireless network in certain domain. With its unique characteristics, people pay more attention to its security coupling with its adaptable application scope. Combining the system private key and the cluster private key, a completely distributed secure key management scheme based on cluster architecture was proposed. It enhanced the authenticity, confidentiality, availability and robustness.

Key words: mobile Ad Hoc network; cluster architecture; (r, n) threshold scheme; key update

0 引言

移动自组网络(Mobile Ad Hoc Network,MANET)是由一组带有无线收发装置的移动终端组成的一个多跳的临时自治性无线局域网系统。它具备两个主要优点:1)不需基础设施(如基站等)的支持就能快速自动组网;2)通过中间节点的多跳转发保证了网络的覆盖范围,同时也节约了每个终端的能源。网络中移动终端同时具有主机和路由器的功能,这种网络可以以末端子网的形式接入现有网络。但由于它又具有拓扑结构不稳定、终端能源有限、有限带宽及自组织等不同于有线网络的特点,使 Ad Hoc 网络的安全问题成为了一个极具挑战性的研究课题。完善的密钥管理机制是安全寻由及机密数据可靠传输的基础,因而成为安全机制中最重要和最复杂的问题。

集中式的密钥管理(如 PKI 管理)在依靠无线信道进行信息传输的 Ad Hoc 网络中易引发单点失败。而分布式密钥管理是由多个节点管理系统私钥共同承担 CA(Certificate Authority)的认证功能,能减小私钥泄露的风险和实现信任分散,很适合 Ad Hoc 网络。Shamir 1979 年提出了秘密共享的概念[1],1999 年 Hass 和 Zhou 首次将秘密共享的思想应用到分布式 CA 中。

Ad Hoc 网络有平面和分级两种结构。平面结构又称为对等结构,假设网络中的各个节点能力、地位都平等,只适用于小规模的网络,应用范围狭窄。分级结构中,网络被划分为簇,每个簇由一个簇头和多个簇成员及多个网关节点组成。簇头和网关节点所负责任大些。分级结构体现了一种层次,真实反映了现实中的情况,比如在战场上,在一个抢险救灾过程中,实体之间都存在着一定的身份及责任的差别。簇结构的使用使节点的初步认证限定在本地范围内,所以有较高的可行性。在 Ad Hoc 网络中进行分布式密钥管理,前人已经做了很多研究工作[2,3],但这些研究都是基于平面结构提出的密钥管理方案,不适于在分级结构网络中扩展。Bechler 等人于 2004 年首次提出基于簇结构的分布式密钥管理模型,解决了认证和访问控制问题,但对一个新节点赋予的权限以新节点得到证人数量的多少为依据,没有体现公平机制。本文对其改进,提出一种基于簇结构的完全分布式安全密钥管理方案。

1 基于簇结构的安全密钥管理方案

1.1 Shamir 的 (t,n) 门限机制[1]

设组中有 n 个节点,t 为门限,$SK(SK \geqslant 0)$ 为私钥,把 SK 秘密分割成 n 份,任意等于或大于 t 个密钥份额能恢复出私钥 SK。

SK 的秘密分割过程:选择一个素数 $p' > \max(SK, n)$,随机选取秘密素数 a_j,且 $0 \leqslant a_j \leqslant p' - 1$, $0 \leqslant j \leqslant t - 1$, 令 $a_0 = SK$。构造一个以 a_j 为系数的 $t - 1$ 次多项式 $g(x)$。计算多项式 $y_i \equiv g(x_i) \bmod p'$, $i = 1, 2, \cdots, n$,可得组中每个节点的私钥子份额 y_i。然后将 a_j 销毁,将 p' 公布,将 (x_i, y_i) 秘密发送给节点 P_i。

私钥 SK 恢复过程:根据拉格朗日内插值定理,任 t 个不同的点 (x_i, y_i) 可以计算出多项式系数 a_j,重构多项式:

$$g(x) = \sum_{i=1}^{t} \left(y_i \prod_{j=1, j\neq i}^{t} \frac{x - x_j}{x_i - x_j} \right) (\bmod p') \tag{1}$$

$$SK = g(0) = \sum_{i=1}^{t} \left(y_i \prod_{j=1, j\neq i}^{t} \frac{-x_i}{x_i - x_j} \right) (\bmod p') \tag{2}$$

这样只有至少 t 个份额持有者共同合作,才能恢复私钥 SK。经私钥 SK 签名后,用与之对应的公钥 PK 可验证秘密分割的正确性。门限机制体现了一种单点不可信、点的集合是可信的思想。

收稿日期:2006 - 09 - 18;**修订日期**:2006 - 12 - 03

作者简介:叶永飞(1978 -),女,河北张家口人,硕士研究生,主要研究方向:信息安全; 余梅生(1943 -),男,江西人,教授,主要研究方向:信息安全、模式识别.

1.2 基于簇结构的安全密钥管理方案

本方案的改进思想就是将系统的私钥以秘密共享的方式分发到网络中的每个节点上,将系统私钥与簇私钥相结合,并设置了一个安全阈值 β。假设每个节点在网络内有唯一的非零身份标识 ID_i,并拥有一对公/私密钥对(Npk_i/Nsk_i),$Npk_i = H(ID_i)$,H 是一个单向函数,节点的公钥 Npk_i 向网内公开,私钥 Nsk_i 用于签名。网络拓扑结构形成后先由各簇头以一种完全自组的形式合作形成系统的公私密钥对(PK/SK)及网络私钥 SK 的共享份额 $S_i(i = 1,2,\cdots,n$,n 为网络中簇的数目),由网络中至少 k 个不同的簇可以重构网络的私钥,而少于 k 个簇则不能重构私钥。然后由一可信任者 Dealer 将网络的私钥份额 S_i 作为簇 CLS_i 的簇私钥,并形成与簇私钥 S_i 相对应的簇公钥 P_i,同时将 S_i 再以一种秘密共享的方式分发到簇内不同的 t 个小集合中。每个簇中的门限值 t 是根据自己簇内所拥有的节点数自行决定的。请求加入簇的新节点的身份认证只需要欲加入簇中 t 个不同的簇私钥份额联合签名即可。方案中使用阈值 β 对漫游节点的身份认证做了限制。经过认证后的节点与它的身份相符的访问权限,这与传统的只将系统私钥份额分发到每个簇的簇头上是不同的,本方案具有更高的安全性。

2 方案实现

本方案的具体实现要求先形成系统公/私密钥对,然后产生簇的公/私密钥对。私钥份额产生以后,节点的身份才能得到认证。对付“移动攻击”采用簇私钥份额的配置适应性的方法。

2.1 网络初始化

图 1 所示的拓扑结构是我们方案适用环境的示例。网络形成时按节点所处的地理位置划分成不同的簇,各簇在网络中有唯一的标识 $CLS_i(i = 1,2,\cdots,n)$。本文采用文献[4]的安全方案对簇进行安全管理。簇头 $CLSH_i$ 在不需第三方参与的情况下自组产生网络的公/私密钥对(PK/SK)。公钥 PK 在全网络系统内公开,私钥 SK 以(k,n) 门限方式存在各个簇中。

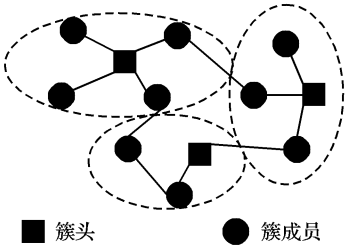

图 1 基于簇结构的网络拓扑结构

2.2 网络系统公/私密钥对的形成

网络系统私钥 SK 产生步骤:

1) 每个簇头 $CLSH_i$ 随机选取一个秘密 X_i 和一个 $k-1$ 阶多项式:$f_i(z) = x_i + a_{i,1}z + a_{i,2}z^2 + \cdots + a_{i,k-1}z^{k-1} \bmod q$,$i = 1,2\cdots,n$,$f_i(0) = x_i$,其中 q 是一个大于网络私钥和最大门限值的大素数,g 是有限域 Z_q 的生成元。$SK = \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} f_i(0)$ 是簇头合作产生的系统私钥。

2) 簇头 $CLSH_i$ 计算分发给簇 CLS_j 的网络系统子份额:$SS_{ij} = f_i(j)$,$j = 1,2,\cdots,n$,并将 SS_{ij} 秘密发送给一个 Dealer(Dealer 收到足够多的 SS_{ij} 后会计算出簇 CLS_j 的簇私钥 S_j),同时计算 $C_{ij} = g^{a_{i,j}} \bmod q(i = 1,2,\cdots,n, j = 0,1,\cdots, k-1$,其中 $a_{i,0} = X_i$) 并将 C_{ij} 公布。

3) Dealer 计算簇 CLS_j 的关于网络私钥 SK 的份额 S_j(亦为簇 CLS_j 的簇私钥):$S_j = \sum_{i=1}^{n} SS_{ij} = \sum_{i=1}^{n} f_i(j)$,$j = 1,2,\cdots, n$;$C_j = \prod_{i=1}^{n} C_{ij} = g^{\sum_{i=1}^{n} a_{i,j}}$,$C(x) = C_0 C_1^x C_2^{x^2} \cdots C_k^{x^k}$,验证:$g^{S_j}$ 与 $C(j)$ 是否模 q 同余,如同余则接受 S_j 为簇 CLS_j 所拥有系统私钥份额(亦为本簇私钥);否则需重新申请,执行 1) ~ 3)。所有份额计算完成后销毁 $a_{i,k-1}$,$i = 1,2,\cdots, n$。

4) 网络私钥被秘密分发之后,任意 k 个拥有私钥份额的簇组合可以恢复网络私钥 SK:$SK = \sum_{i=1}^{k} S_i L_i(0) \pmod q$,其中,$L_i(0) = \prod_{j=1, j\neq i}^{k} \frac{-z_j}{z_i - z_j}$ 是拉格朗日系数。

系统公钥 PK 的产生:

份额产生完毕,Dealer 将 S_iP 公布,P 是一个基于身份的公共参数[5]。

则网络系统的公钥为:$PK = \sum_{i=1}^{n} S_i P$

将系统公钥 PK 在网内公布,用于验证系统签名的正确性。

2.3 簇 CLS_i 私钥的管理

本方案借鉴文献[6]中对私钥份额分发及更新的算法实现对簇私钥的管理。

2.3.1 簇 CLS_i 私钥 S_i 的秘密分发

系统私钥份额 S_i 成为簇 CLS_i 的私钥被按照秘密共享的思想分发到簇中的 t 个集合中,每个集合中的成员都拥有一个相同的簇私钥份额。集合的分类是按照将节点的公钥值 Npk_i 模 t 后落入的范围确定的。

每个簇的簇头 $CLSH_i$ 首先将自己簇内拥有的成员数 M_i 安全传输给可信任者 Dealer,然后由 Dealer 完成每个簇私钥 S_i 的秘密分发,过程如下:

1) 产生 RSA 密钥公/私密钥对 $< pk_i = (e, N)$,$sk_i = (d, N) >$,其中 $d = s_i$;

2) 选择最初的门限值 t(t 是个正奇数);

3) 产生 t 个随机数 $r_i \in [-(N-1)/2, (N-1)/2]$。$\sum_{i=0}^{t-1} r_i \equiv 0 \bmod \Phi(N)$,其中 $\Phi(N)$ 是欧拉函数;

4) 计算私钥份额 $SS_i \equiv (-1)^i \cdot d + r_i \bmod \Phi(N)$;

5) 计算 $w_i \equiv (-1)^i + e \cdot r_i \bmod \Phi(N)$,$i = 0,1,2,\cdots, t-1$,$w_i$ 是一个与私钥份额 SS_i 绑定的证人,用于接收者验证收到的份额的真伪;

6) 通过安全信道将份额 SS_i 分发到至少 t 个成员,节点 ID_i 通过下式进行验证:

如果 $(a^{SS_i})^e \equiv a^{w_i} \bmod \Phi(N)$(对任意非零整数 a),则可证明收到的份额为真;否则为假,拒绝接受 SS_i。

7) 公开$\langle PK_i, t, w_0, \cdots, w_{t-1}\rangle$;

8) 销毁其他计算值。Dealer 也不再需要,从网络中撤出。

2.3.2 节点请求加入簇

当有节点 IDx 请求加入簇 CLS_i 时,簇头 $CLSH_i$ 通过 NIODS 算法[4] 查找 IDx 的信息,如果 IDx 是一个新节点则执行以下过程:

通过簇内每个集合(共 t 个) 中任一名成员 ID_{ji},$i = 1,2,\cdots, t-1$ 用自己拥有的私钥份额 SS_i 对 IDx 提供的认证请求消息 m 进行签名。收到 t 个签名后,IDx 重构签名 $sig(m) \equiv \sum_{i=0}^{t-1} (m)^{SS_i} \bmod \Phi(N)$。

如果 $sig(m)^e \equiv m \bmod \Phi(N)$，则收到的簇签名是正确的，节点 IDx 的身份得到认证，被赋予与节点 IDx 的 ID 值模 t 后的值相同的集合的簇私钥份额，同时给予与身份相符的访问权限。如验证不成功，则签名失败。IDx 通过检查式 $sig_{ji}(m)^e \equiv m^{w_i} \bmod \Phi(N)$ 是否成立可验知是哪部分签名无效，然后向签名失败的集合重新申请部分签名，减少了计算量。

如果请求加入节点为一个诚实的漫游节点，则需检查漫游次数 r 是否超过 β，若没有超过，则在本簇中按一个新加入节点来对待对其进行重新认证，并分配关于本簇的簇私钥份额。若超过 β，IDx 需经过网络系统私钥 SK 的签名后方可得到认证。若请求节点为恶意节点，则簇拒绝它的加入请求。β 值越小网络越安全，但同时也增大了计算量。

经过认证后的节点会得到由分布式的 CA 签名的证书，证书将节点的身份和公钥绑定。经过相互认证的节点可安全通信。

2.3.3　簇私钥份额的配置适应性

分布式的 Ad Hoc 网络密钥管理易遭受“移动攻击”。对付“移动攻击”最有效的办法就是周期性或实时地更新私钥份额。当我们利用监测系统检测出离开某个簇的节点累计数目接近于簇私钥份额的门限值或者簇中新加入成员已达到一定数目时，对本簇的簇私钥份额门限进行 $t \to t+k$ 的更新，按如下步骤完成份额的更新：

1）每次更新时由簇 CLS_i 中可信度最高的节点对其所拥有的份额 SS_i 进行劈分。

2）选择一个正奇数 k；

3）随机选择一个份额 $SS_j(j \neq i)$，然后产生 $k+1$ 个新份额：$SS'_i = SS_i + u_0$，$SS_{t+g-1} = (-1)^g SS_i + u_g, g = 1, \cdots, k$。证人：$w_{t+g-1} = (-1)^g w_i + e \cdot u_g, w'_i = w_i + e \cdot u_0$ 和一个更新因子 $SS''_i = -SS_i + u_{k+1}$，且 $\sum_{j=0}^{k+1} u_j = 0$。$SS'_j = SS_j + SS''_i$，$w'_j = w_j - w_i + e \cdot u_{k+1}$。

4）这样簇 CLS_i 中又产生了 k 个新簇私钥份额 $SS_{t+g-1}(g = 1, \cdots, k)$，同时 SS_i 和 SS_j 关于簇私钥份额也被更新为 SS'_i 和 SS'_j。按前文所述的方法进行验证可知新生成的密钥份额是否正确。

通过扩展可以更新簇 CLS_i 内所有簇私钥份额，这需要产生一个更新因子 u_i 且满足 $\sum_{j=0}^{n} u_j = 0$，t 是当前簇中所拥有的簇私钥份额数。

3　结语

本文提出的完全分布式的安全密钥管理方案是基于簇结构的。经过分析，本方案具备如下安全性：

1）可用性。网络能在被俘获节点数低于门限值的情况下安全运行，而且如果网络中大部分节点由于灾难性故障不能恢复系统私钥时，所有簇头联合可生成系统私钥。

2）抗单点失败性。秘密共享思想的运用消除了集中管理方式中 CA 和簇头被俘导致系统私钥泄露的隐患。

3）信任分散。本方案将系统私钥份额分发到了每个节点上。重构系统私钥时，需要至少 k 个簇中分别派出本簇内门限个以上的节点共同参与才能恢复系统私钥，入侵者要想获得系统私钥是非常困难的。

4）防密谋攻击机制。为防止离开节点联手密谋攻击网络私钥或恶意节点通过漫游进行移动攻击，我们采用设立阈值 β 和重新配置门限值进行密钥份额更新两种安全机制。

如何在安全性与计算量之间寻求一个平衡点是我们亟待解决的问题，这也是我们以后研究的重点。

参考文献：

[1] SHAMIR A. How to Share a Secret [J]. Communication of the ACM, 1979, 22(11): 612 - 613.

[2] 许峰，谢冬莉，黄皓，等. 一个基于权限的移动自组网门限信任模型[J]. 计算机应用，2006，26(3)：574 - 576.

[3] 刘志远，毛胜利. 一个新的 Ad Hoc 安全组密钥管理方案[J]. 微计算机信息，2006，22(4)：3，4，25.

[4] 张晓宁，冯登国. 无线自组网中基于簇结构的安全方案[J]. 计算机研究与发展，2006，43(2)：238 - 243.

[5] BONH D, FRANKLIN M. Identity-Based Encryption from Weil Pairing[A]. Advances in Cryptology, CRYPTO2001, LNCS2139[C]. Springer Verlag, 2001. 213 - 229.

[6] DI PIETRO R. Efficient and Adaptive Threshold Signatures for Ad hoc networks[M]. Elsevier, 2006.

（上接第 596 页）

通常证书的接收方可以借助验证“签发用户证书的认证中心的证书”的正确性来验证该认证中心。如果该认证中心又是由其他认证中心所签发的，则重复验证证书的正确，不断地重复此验证的步骤，一直验证到接收方所信任的认证中心为止，这就是 X.509 中定义的证书路径（又称证书链），信任传递如图 1 所示。因此，本方案中跨域认证能否通过，主要在于票据签发者的证书是否在一个被该应用服务器信任的证书路径上。

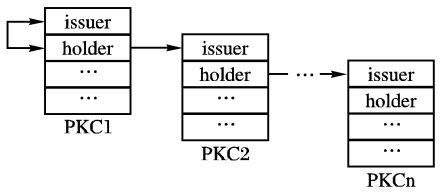

图 1　证书路径

跨域认证步骤如下：1）验证票据是否过期，地址是否正确；2）验证该认证服务器证书的有效性；3）验证票据签名的正确性；4）执行域内票据验证协议。

3　结语

本文提出的单点登录方案，能够满足证书和口令认证并存的安全需求，同时由于基于公钥设计了票据协议，从而消除了对称密钥在实现域内认证和跨域认证时复杂的密钥管理，为单点登录系统的设计提出了一种新思路。

本文提出的单点登录方案，已在公安部移动数据接入系统和我校信息门户系统中获得了成功应用。

参考文献：

[1] Microsoft. . net passport review guide[EB/OL]. http://www.Microsoft.com/net/services/passport/reviewguide.asp, 2003 - 03 - 13.

[2] Libery Alliance Project. Liberty architecture overview[EB/OL]. http://www.projectliberty.org/specs/liberty-architecture-overview-v1.1.pdf, 2003 - 01 - 15.

[3] TUNG B. Public Key Cryptography for Initial Authentication in Kerberos[EB/OL]. draft-ietf-cat-kerberos-pk-init-15.txt, 2006.

[4] SIRBU M, CHUANG J. Distributed Authentication in Kerberos Using Public Key Cryptography[A]. Symposium on Network and Distributed System Security[C]. 1997.

[5] 文铁华，谷士文. 增强 Kerberos 协议安全性的改进方案[J]. 通信学报，2004，25(6)：76 - 79.