第 42 卷 第 6A 期
2015 年 6 月

计 算 机 科 学
Computer Science

Vol. 42 No. 6A
June 2015

基于节点间信任评估算法的无线传感器网络密钥管理方案

陈 昊[1,2] 黄海平[2]

(南通大学附属医院信息科 南通 226001)[1] (南京邮电大学计算机学院 南京 210003)[2]

摘 要 提出了一种基于信任机制的密钥管理方案,通过评测发起节点的信任值,判断节点是否可以作为密钥更新的簇头节点。方案中采用了双向信任数据包转发机制,不仅可以获取节点的信任值,同时也验证信任数据包的正确性,进一步提高了密钥管理实施过程中的安全性。同时对所阐述的方案,从安全性、通信复杂度和存储消耗等方面进行了理论研究和分析,并进行了实验仿真,具有较高的安全性。

关键词 信任机制,密钥管理,无线传感器网络

中图法分类号 TP303.01 **文献标识码** A

Key Management Scheme of Wireless Sensor Network Based on Trust Evaluation Algorithm between Nodes

CHEN Hao[1,2] HUANG Hai-ping[2]

(Information Department, Afflicted Hospital of Nantong University, Nantong 226001, China)[1]

(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)[2]

Abstract This paper presented a key management scheme based on trust mechanism. Through evaluating the trust value of the originating node, it determines whether the node can be the head node of a cluster for key updating. This scenario with a two-way trust packet forwarding mechanism, can not only get the trust value of the node, but also verify the correctness of the trust packet. It further improves the security of the key management implementation process. Meanwhile, this paper did theoretical research and analysis from the aspect of security, communication complexity and memory consumption, and experimental simulation.

Keywords Trust mechanism, Key management, Wireless sensor networks

1 引言

随着无线传感器网络的广泛应用,其安全性愈发受到人们的关注。传统的密钥管理方案虽然安全性较高,但相对于能耗敏感、部署环境的传感器网络而言,算法协议要求网络负载大、节点计算能力高等特点,传统方案不能应用于传感器网络。因此很多学者致力于研究适用于无线传感器网络的密钥管理方案。现有的随机密钥主要分为:随机密钥分配和确定型密钥预分配算法。随机方案中,节点能够根据预存的密钥进行连接,也可以依靠中间节点建立安全连接。例如文献[1]首先建立一个密钥池,任意节点从密钥池中存储到节点中。因此任意两个节点间在某种概率上具有连通性。文献[2-4]也属于随机密钥算法,其部署简便,但连通率低,存储开销大。确定型密钥管理方案中,节点预存储了全局密钥或是根据节点某些确定信息,节点间可以相互确定通信密钥,例如采用多项式函数结构[5]进行密钥分发。而文献[6]利用事前部署的密钥,建立新的密钥对,保证网络具有良好的扩展性。之后依据密钥池和随机想法的方案,例如阶层式 WSNs 密钥管理 IKDM 密钥分配方案[7]、EPKM 密钥预先分配方案[8],仍然存在着网络全连通问题和节点相互认证问题。

信任是实体间一种特定行为评价机制,强化了传感器网络节点的管理,提高了网络传输成功率,充分利用了有限资源。文献[9]提出了基于信任机制的对偶密钥建立模型,根据预存的全局初始密钥来生成一条对偶密钥,而节点的合法性则需要网络中的信任评价机制完成。该方案具有能耗低的特点,但其抗俘获性有待提高。TDKM 方案[10]是结合了 EBS 动态密钥管理方案[11]和信任管理方案。信任机制负责判断节点是否被俘获,而在密钥更新时,剔除不安全节点,显著增强了网络抵抗性。

本文是基于多项式函数的密钥管理方案,并引入信任机制,判断节点是否可以担任重构节点。只有网络中的大多数节点认为其是可信节点,则发送随机密钥,从而构造新的多项式函数密钥,提高了网络的安全性,且一定程度地降低了网络能耗。

2 密钥管理方案描述

2.1 信任模型假设

在此方案中,信任模型可以评价节点之间的相互关系,其满足以下几个特点:

(1)初始状态下,即无线传感器刚部署在指定区域时,所

本文受国家自然科学基金(61373018)资助。

陈 昊(1985—),男,主要研究方向为无线传感器网络信息安全;黄海平(1981—),男,副教授,硕士生导师,主要研究方向为无线传感器网络、计算机软件在通信中的应用和信息安全。

有的节点相互之间都相互信任，且信任度最高。

(2)两个节点间的信任函数与网络运行时间 t 相关，运行时间越长，相互信任度随之降低。因此需要通过相互的通信来维护相互的信任值，根据节点间的安全质量、通信质量、服务质量等因素来对信任进行奖励和惩罚。

(3)传感器节点的信任值范围在 $[0,1]$ 之间，信任值的产生不仅根据自身节点的数据维护，也可以通过信任节点间接获取。

网络在实际的运行过程中，节点主要与相邻的节点进行相互通信，因此其信任值得到不断地维护，而相对较远的节点，因为一定时间没有进行通信，信任值会逐渐降低。而经常通信的节点，由于可靠的信任模型算法，相互间的信任度较高。利用信任机制可以较大可能地剔除非法节点，保证网络的安全性。

2.2 密钥分配算法

(1)密钥预分配阶段

在节点部署前，每个节点内需要预存储某些安全数据和算法函数，具体包括以下几点：

1)对每个节点进行唯一编号 ID，标识每个节点。

2)初始密钥 E，以及对称加解密算法 f_E 和 f_E^{-1}。

3)哈希函数 $H(x)$。

当节点部署在指定区域后，则开始与周围节点进行相互通信，其目标是获取周围节点环境信息，并开始分簇。其数据包格式如下：

$$p(ID,f(H(ID,T,n,C))$$

其中，T 为时间信息，n 是随机数，C 是节点已分簇信息。

每个节点通过这样的方式，了解周围节点已经分簇的信息，节点通过比较通信距离后，加入较近的簇，或是与未分簇的邻居节点组成新的簇。当簇内节点大于某个值 $count_{\max}$ 后，则不再扩充节点。为了避免存在某些节点成为孤立节点，当该节点发现周围的簇内节点都已经大于 $count_{\max}$ 时，则可能将该节点标注为孤立节点，加入与簇内节点平均跳数最小的簇。如果某些簇的节点数量小于 $count_{\min}$，且无法合并或加入相邻的簇，则需要取消该簇，将这些节点都标注为孤立节点，并进行重新选择。

经过密钥预分配和分簇后，整个无线传感器网络被划分成多个相对独立的安全区域，且每个簇内的节点平等，共同管理维护簇中密钥。

(2)密钥产生和分配

网络分簇后，每个簇独立开始产生密钥，为了保证参与密钥构造的节点具有较高的可靠性，需要利用簇内节点共同参与，具体过程如下：

1)假设某节点 S 请求簇内密钥更新，则首先需要请求其余节点对节点 S 的信任度值数据，S 节点从信任表中选择多个较为信任的节点，其数量 n 等于多项式函数的次数。按照随机顺序和其逆序发送两个数据包，数据包的格式如下：

$$ID,E_{en}(ID_s,nouce,p_{1,2,\cdots,n},Trust)$$

E_{en} 为发送节点的通信密钥，初始阶段使用预存储，之后使用多项式所构造的密钥。$p_{1,2,\cdots,n}$ 为数据包的转发顺序。$Trust$ 是范围在 $[0,1]$ 的信任值，初始值为 1。同时反向顺序也发送信任请求值数据包。

2)其余节点收到该数据包后，根据发送节点的 ID 值，利用多项式函数计算出密钥 $F(H(ID))$，F 为多项式函数。

3)解密后得到 ID_S、$p_{1,2,\cdots,n}$、$Trust$，参与信任评价的节点信任表中找到对 ID_S 节点的信任值 $Trust^S$，并计算 $Trust^S * Trust$ 值后，替换原来的信任值，重新封装数据包，然后按照 $p_{1,2,\cdots,n}$ 的顺序转发数据包。

4)每个节点都对 S 节点评价后，数据包最终回到 S 节点，得到两个信任值 Tr^{S+}、Tr^{S-}，如果两个值相等，则将信任值 Tr^S 传递给各个节点，并验证 Tr^S 是否等于 $\prod_{i=1}^{m-1} Trust_i^s * \prod_{i=n}^{m+1} Trust_i^s * Trust_m^s$。可进行第 7 步操作。若不等，则正向或逆向传递信任值时，同一节点信任评价出现数据不一致，或是传递过程中被篡改。S 节点将 Tr^{S+}、Tr^{S-} 按照其原来的顺序再次传递给每个节点。

5)节点收到前一个节点传递的信任数据后，除以对 S 节点的信任值 $Trust^S$，再传递给下一个节点。完成这个步骤后，每个节点都获得 $\prod_{i=1}^{m-1} Trust_i^{s+}$、$\prod_{i=n}^{m+1} Trust_i^{s+}$、$\prod_{i=1}^{m-1} Trust_i^{s-}$、$\prod_{i=n}^{m+1} Trust_i^{s-}$，$m$ 是参与评价过程的任意节点，分别是正向和逆向传递时，前后信任值的乘积。假设 p 点正向传递数据发生错误，则前 p 个节点一定存在 $\prod_{i=1}^{m-1} Trust_i^{s+} = \prod_{i=1}^{m-1} Trust_i^{s-}$，$\prod_{i=n}^{m+1} Trust_i^{s+} \neq \prod_{i=n}^{m+1} Trust_i^{s-}$。后 p 个节点信任值关系 $\prod_{i=1}^{m-1} Trust_i^{s+} \neq \prod_{i=1}^{m-1} Trust_i^{s-}$，$\prod_{i=n}^{m+1} Trust_i^{s+} \neq \prod_{i=n}^{m+1} Trust_i^{s-}$。逆向传递发生错误时，后 p 个节点信任值关系 $\prod_{i=1}^{m-1} Trust_i^{s+} \neq \prod_{i=1}^{m-1} Trust_i^{s-}$，$\prod_{i=n}^{m+1} Trust_i^{s+} = \prod_{i=n}^{m+1} Trust_i^{s-}$，而前 p 个点信任关系都不相等。

6)经过步骤 5)后，可以确定信任评价发生在哪个方向。假设 p 点正向发送错误，则前 p 个节点要求下个节点重新计算信任评价值 $\prod_{i=1}^{m} Trust_i^{s+}$，并传递给下个节点，保持 $\prod_{i=1}^{m-1} Trust_i^{s+} = \prod_{i=1}^{m-1} Trust_i^{s-}$ 等式成立。并将正确结果传递给 S 节点。

7)经过以上的步骤，每个节点计算 $\prod_{i=1}^{m-1} Trust_i^s * \prod_{i=n}^{m+1} Trust_i^s * Trust_m^s$，得到 S 点的信任值。同时每个节点根据 S 点的信任程度 Tr^S，考虑是否生成一个密钥发送给 S 节点。

8)如果 S 节点的 Tr^S 值较高，则可收到其余节点所发送过来的随机密钥，采用多项式函数构造密钥的方法：

$$P(x)=\prod_{i=1}^{n}(x-a_i)=\sum_{i=1}^{n} b_i x^i$$

a_i 是各节点发送过来的随机密钥，通过计算后，得到表达式 $\sum_{i=1}^{n} b_i x^i$。最终将多项式的系数发送给簇内各个节点。

(3)节点加入和退出

当有新节点加入时，簇内各个节点对其的信任度还不够，因此在信任评价过程中，该节点不在转发顺序中，直至新节点在某些节点的信任度达到一定值之后才有可能参与密钥更新过程。而当密钥生成后，需要将多项式的系数告诉给新节点，以便可以与其它节点进行通信。而当节点需要退出簇时，由于各个节点相互独立，节点退出后对其它节点并无安全影响。

3 密钥方案分析

3.1 安全性分析

无线传感器网络通常被部署在环境恶劣的区域，抗击敌方的恶意攻击。该基于信任模型的密钥管理方案具有较好的安全性。其主要体现在以下几个方面：

1. 信任模型的应用。信任机制的维护是基于通信状况以及安全监测评价，通过这样的反馈，每个节点内部形成独立的信任表。表中存储经常相互通信的节点的信任值，而未发生通信关系的节点都会随着时间的流逝，其信任值不断地降低。在密钥产生过程中，发起节点仅请求较为信任的节点参与过程，保证了通信质量和安全。而参与的节点经过对发起节点的信任评价后，根据其网络对该节点的信任值，判断节点是否是可靠，从而决定了是否要进行密钥生成过程。

2. 防止信任数据篡改。方案中信任评价数据包双向发送给各个节点，取得信任结果后，将信任值数据再按原来的顺序传递给各节点。如果某节点传递过程中，修改信任乘积值，按照步骤 5 的过程，判断 $\prod_{i=1}^{m-1} Trust_i^{s+}$、$\prod_{i=n}^{m+1} Trust_i^{s+}$、$\prod_{i=1}^{m-1} Trust_i^{s-}$、$\prod_{i=n}^{m+1} Trust_i^{s-}$ 的等价关系，可确定传递数据出现错误的节点。如果 S 有欺骗行为，传递给各个节点虚假的 Tr^s，使得 $Tr^s \neq \prod_{i=1}^{m-1} Trust_i^s * \prod_{i=n}^{m+1} Trust_i^s * Trust_m^s$，从而判断 Tr^s 是非法数据。

3. 密钥的安全性。多项式共享密钥的特点是必须在 n 个或 n 个以上的节点被俘获后，敌方才能破解整个簇，具有较高的抗俘获性。对于 S 而言，必须有 n 个或 n 个以上节点认为 S 是可信节点，才会发送新的密钥，从而构造新的多项式密钥函数，否则 S 点发起的密钥更新过程失败。

3.2 通信复杂度分析

对于能耗敏感的无线传感器网络，通信复杂度是衡量密钥管理方案。本方案需要获取发起节点的网络信任值，其过程是双向转发信任数据包。假设网络簇的数量为 n^c，簇中节点间通信平均条数为 n^t。双向转发过后，网络端到端通信次数为 $2n^c \cdot n^t \cdot n$，如果双向信任值相同，则网络需要再发送 $n^c \cdot n^t \cdot n$ 跳通信，如果不等，则需要发送 $2n^c \cdot n^t \cdot n$。当满足信任后，每个节点发送发起节点的密钥信息值，发起节点向整个网络发送多项式函数系数，需要再次通信 $n^c \cdot n^t \cdot n + n^c \cdot n^t \cdot n^a$，其中 n^a 是簇节点平均数量，$n^c \cdot n^a$ 是整个网络的节点数据，$n^c \cdot n$ 小于 $n^c \cdot n^a$。由于网络簇的大小不随节点数量变化而变化，因此簇内通信平均条数 n 相对较小且是基本固定值。综上所述整个方案的通信复杂度为 $O(N)$，N 为网络节点的数量。

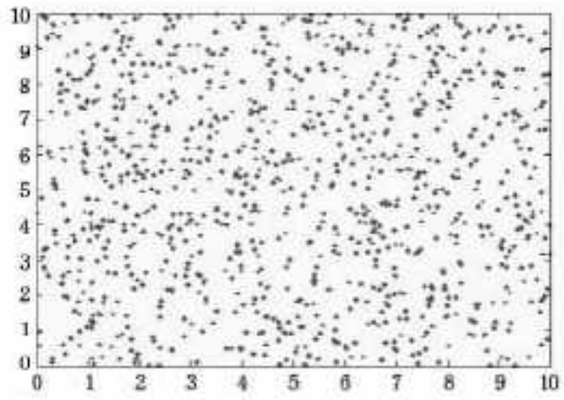

图 1 节点分布图

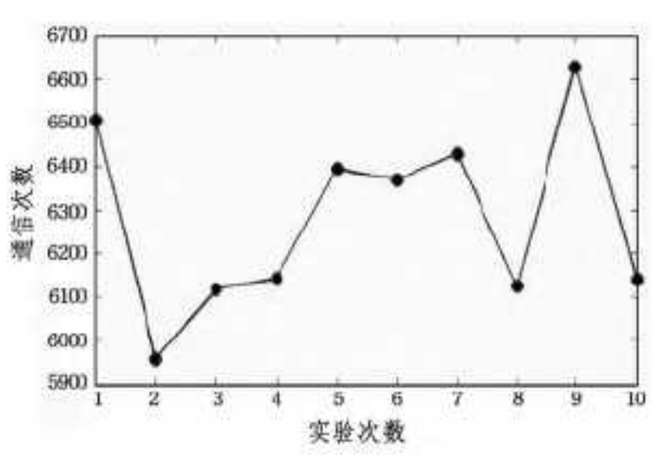

图 2 网络通信次数

假设 1000 个节点均匀部署在 $10 * 10$ 的区域内，通信半径为 0.5，控制每个簇的大小在 20—30 之间，多项式的次数设定为 15，即每个簇有 15 个随机节点参与密钥重构过程。每个簇完成一次更新时，重复实验多次的结果如图 1、图 2 所示。

从图 2 可以看出，整个网络进行一次密钥更新，每个节点平均需要发送或转发 6.3—6.4 个数据包。

3.3 存储开销

本文提出的密钥管理方案存储，包括密钥更新信息和信任数据。在节点预部署前，在每个节点中存储节点 ID、初始全局密钥、加解密算法以及哈希函数，其存储复杂度为 $O(1)$。在网络部署后，每个节点需要存储信任表，信任表中仅存储本簇内其他节点的信任关系。网络分簇节点数量最大为 $count_{max}$，与多项式函数的次数同一数量级，且不随网络的扩大而增加存储，存储复杂度为 $O(1)$。综上所述，网络分簇后，节点所存储的数据信息量基本固定，复杂度为 $O(1)$。

在图 1 网络仿真环境中，网络被划分成 38 个簇，各个簇的节点数量分布如图 3 所示。

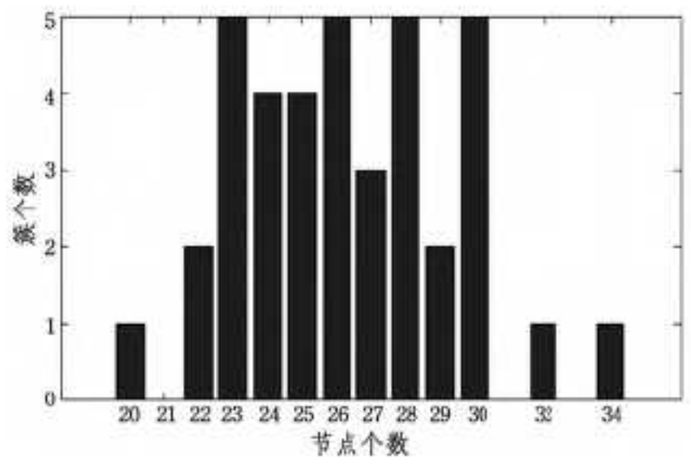

图 3 簇节点数量分布图

由于每个节点需要存储簇内其他节点的信任值，所有节点需要存储的信任记录共 26666 条，大约每个节点存储 26.6。现假设同一区域内，部署节点的数量为 400，500，…，1000，图 4 记录了节点平均存储信任记录数量。

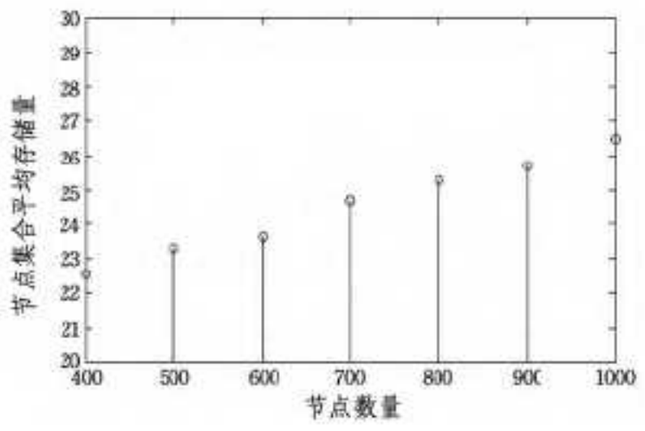

图 4 节点平均存储信任值数量

在图中可以看到，节点数量越多节点存储信任值数据越多，这是由于节点数量越多，节点部署越密集，因此在簇中节点的平均数量也上升，从而导致节点存储开销随节点数量缓慢增加。但是在分簇时，节点数量控制在 20～30 之间，所以节点的存储开销并不大。

结束语 无线传感器网络的密钥管理是传感器网络安全的重要方面，为能耗敏感、资源受限的节点提供可靠的密钥建立、维护、更新等。本文侧重于利用无线传感器网络节点所存储的密钥信息来寻找可靠节点，实现节点的多项式密钥更新过程。经对比，提出的密钥管理方案具有较好的安全性，并且通信与存储能耗较低。下一步需要优化网络的分簇算法，并将方案应用于实际传感器网络中，以检验协议是否有效地运行。

参 考 文 献

[1] Eschenauer L, Gligor V D. A Key-management Scheme for Distributed Sensor Networks [C]// Proc. of the 9th ACM Conference on Computer and Communications Security. [S. l.]. ACM Press, 2002: 41-47

[2] Wenliang Du, Jing Deng, Yunghsiang S. Han, Pramod K. Varshney. A pair-wise key pre-distribution scheme for wireless sensor networks [C]// The 10th ACM Conference on Computer and Communication Security(CCS'03). Washington DC, USA, ACM Press, Oct. 2003: 12-21

[3] Liu D, Ning P. Location-based pair-wise key establishments for static sensor networks [C]// Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Security of Ad Hoc and Sensor Networks. Fairfax, Virginia, USA, ACM Press, 2003: 72-82

[4] Eschenauer L, Gligor V D. A key-management scheme for distributed sensor networks [C]// The 9th ACM Conference on Computer and Communication Security(CCS'02). Washington DC, USA, ACM Press, Nov. 2002: 41-47

[5] Blundo C, Santis A D, Herzberg A, et al. Perfectly secure key distribution for dynamic conferences [J]. Information and Computation, 1998, 146(1): 1-23

[6] Wen Mi, Zheng Yan-fei, Ye Wen-jun, et al. A key management protocol with robust continuity for sensor networks [J]. Computer Standards & Interfaces, 2012, 31(4): 642-647

[7] Cheng Y, Agrawal D P. An improved Key Disstribution mechanism for Large-Scale Hierarchical Wire-less Sensor Networks [J]. Ad hoc Networks, 2007: 35-48

[8] Cheng Y, Angrawal D P. Efficient pairwise key Es-tablishmentand manangement in StaticWireless Sensor Network [C]// Proceedings of Moblile Ad-hoc and Sensor Systems Conference. 2005

[9] 成奋华. 传感器网络中基于信誉模型的对偶密钥建立算法 [J]. 计算机应用, 2011, 7(31): 1876-1879

[10] 程伟, 程良伦. 基于信任的无线传感器网络动态密钥管理方案 [J]. 计算机测量与控制, 2011, 19(9): 2315-2318

[11] 程芳权, 彭智勇. 可信云存储环境下支持访问控制的密钥管理 [J]. 计算机研究与发展, 2013, 50(8): 1612-1627

(上接第 394 页)

(3)如果对应的文件有 crypt—file 标志,将用户要写的内容加密后再写入页缓冲。

3.3 文件的密钥存储保护

与 eCryptfs 类似,本方法生成的加密文件在文件中有一页头来描述这个被加密的文件,主要包括这一类型的加密文件标志、被加密的文件加解密密钥,以及使用的加解密算法等。由于多了一页的文件偏移,在 open 文件和 lseek 系统调用中,对文件的偏移需要自动修正为真正的文件偏移,确保对用户使用透明。

文件加解密密钥的保护密钥可以设计为每个用户拥有自己独立的密钥或者整个系统使用一个保护密钥。

4 应用

采用这种面向特定应用的内核级加密文件技术对 office 文件进行加密,配置只有 office 程序能够访问 office 文件的明文,就能确保其他程序无法正常访问 office 文件。office 文件无论在本地还是网络的流转都只能以密文方式进行,只有本地的 office 进程能够得到明文。即使在一个被植入木马程序的系统上,office 文件还是受到很好的保护。

大多数特洛伊木马在被入侵的目标主机上运行服务器端,开启端口等待连接。黑客通过客户端控制木马程序执行各种操作,一般木马在进驻目标机器后通过网络与外界通信,发回所搜集到的各种敏感信息,并接受黑客的指令完成其他各种操作。比如 Linux. Backdoor. Kaiten[7] 和 Linux. Backdoor. Rexob[8] 就是这种类型的木马。但是这些木马只能访问 office 文件的密文,无法获取 office 文件的明文,office 文件就不会真正被泄露。

结束语 本文提出了一种面向特定应用的内核级文件加密技术,改进了 eCryptfs、dm-crypt 等内核级加密文件系统可能由于木马程序造成信息泄密的问题。下一步将对该技术进一步扩展,将可信技术、利用硬件可信根的度量及硬件的加密能力用于用户信任的可执行文件度量和文件的加密,进一步提高信息系统的安全性。

参 考 文 献

[1] Wolfgang Mauerer. 深入 Linux 内核架构[M]. 北京: 人民邮电出版社, 2010

[2] Halcrow M A. ecryptfs: An enterprise-class encrypted filesystem for linux[C]// In Proceedings of the Linux Symposium. Ottawa, Canada, July 2005: 201-218

[3] 唐晓东, 付松龄, 何连跃. 基于 eCryptfs 的多用户加密文件系统设计和实现[J]. 计算机应用, 2010, 30(5): 1236-1238

[4] 陈忠贵, 舒远仲, 吴文俊. 加密文件系统中缓冲技术的研究[J]. 南昌航空大学学报, 2010, 24(2): 67-71

[5] Peters M. Encrypting partitions using dm-crypt and the 2.6 series kernel[OL][2004-6-6] http://archive09.linux.com/feature/36596

[6] Red hat. Logical Volume Manager Administration. Appendix A. The Device Mapper[OL]. [2013-09-29]https://access.redhat.com/documenation/en-US/Red—Hat—Enterprise—Linux/6/html/Logical—Volume—Manager—Administration/device—mapper.html/

[7] Symantec. Linux. Backdoor. Kaiten[OL]. http://symantec.com/security—response/writeup.jsp?docid=2006-021417-0144-99&tabid=2

[8] Symantec. Linux. Backdoor. Rexob. [OL]. http://symantec.com/security—response/writeup.jsp?docid=2007-072612-1704-99&tabid=2