MADIDS:基于移动代理的分布式入侵检测*)

陈 波 罗光春 卢显良
(电子科技大学计算机科学与工程学院 成才 610054)

摘 要 传统的IDS在WAN上配置时,通常会出现计算瓶颈和维护更新不易等问题。本文提出了一种基于移动代理的新型分布式入侵检测系统(Mobile Agent Distributed IDS)。MADIDS是针对WAN环境专门设计的,数据的处理通过各节点所设置的代理来进行分布式计算,不仅能实现全网络范围内的入侵检测功能,具有良好的可移植性;而且对网络系统和主机的资源占用较低,减少出现网络瓶颈的可能。文中建立了MADIDS的体系结构和理论分析模型,并讨论了MADIDS的维护更新机制。

关键词 移动代理,分布式入侵检测系统,性能

A Distributed IDS Based on Mobile Agent

CHEN Bo LUO Guang-Chun LU Xian-Liang
(College of Computer Science & Engineering, UESTC of China, Chengdu 610054)

Abstract When traditional Intrusion Detection System (IDS) is used to detect and analyze in WAN, it usually causes the computation bottleneck. This paper presents a new Mobile Agent Distributed IDS (MADIDS) system based on the mobile agents. This system is specifically designed for WAN. In MADIDS, the agents that are set at each node process the data transfer by distributed computation architecture. It has the ability of intrusion detection within the entire network and has good portability. The consumption of the network and servers' resources is not high, which means the possibility of network bottleneck is decreased. In this paper, we construct the infrastructure and theoretical model of MADIDS, and the deficiencies of MADIDS and future research work are also indicated.

Keywords Mobile agent, Distributed IDS, Performance

1 引言

现代入侵检测技术系统的发展趋势是宽带高速实时的检测技术、大规模分布式的检测技术、数据挖掘技术、更先进的检测算法和入侵响应技术。移动代理使得分布式协同入侵检测更为灵活,尽管基于移动代理技术的IDS产品目前还没有,但相关的研究和实验室样品已出现较多[1]。移动代理作为一种新兴技术,在大规模、分布式、跨平台的应用中拥有独特优势。引入代理技术主要有如下原因:(1)代理具有独立性,它就像软件中所讲的组件技术一样,解决了传统IDS的不易配置的问题;(2)代理技术可以集HIDS和NIDS于一身,这样可以使得代理技术兼有二者的优势,尤其是在数据采集上的优势;(3)代理易于升级和扩展。但移动代理也存在一些缺陷,比如代理的安全性,尤其是MA安全问题一直都是头等问题;其次,Agent的执行效率以及大代码量都是值得关注的问题。对比于静态代理,移动代理技术在IDS中的应用早在96年就有人提出[2]。移动代理具有移动性、自我藏匿性、受损后的自我恢复等优秀的特点,非常适合于入侵检测,目前最为典型的实例是Iowa State University的MAIDS系统[3],它具有以下特点:(1)移动计算能力。移动Agent携带的是计算而非数据,所以无需数据采集模块回送大量的事件信息给数据分析模块,只需分发携带分析算法的移动Agent至数据采集地即可,大大减轻了网络负荷;(2)自我藏匿和躲避。Agent能在网络中随机的迁移,使得攻击者很难对一个Agent定位,对其发动攻击就更为困难了。当一台主机被攻陷时,Agent会自动作出响应;(3)自我恢复。当Agent被毁坏后,备份Agent会根据有关状态信息自动恢复到原来的状态,重新加入入侵检测系统中;该系统主要将移动Agent用于对入侵行为的检测,而本文的研究重点与之不同,是将该技术运用在入侵检测系统的大规模配置,以及维护系统完整性和一致性的工作上[4]。

2 基于移动代理的入侵检测

2.1 移动代理的优点

移动Agent技术给分布式系统的设计、实现和维护都带来了新的活力,它有着多方面的优点:(1)减轻网络负载;(2)克服网络隐患;(3)封装协议;(4)移动Agent异步自主运行;(5)移动Agent的应变能力;(6)具有自然异构性由于移动Agent通常是独立于计算机和传输层,所以能提供了无缝系统集成的最优条件。

2.2 MADIDS体系结构

本文提出了一种基于移动Agent[5~7]的新型入侵检测系统—MADIDS。MADIDS使用分层体系结构,将部署在整个WAN上的入侵检测系统划分为若干域。每个域由域服务器管理,所有域服务器由主服务器管理。这种体系结构实现了IDS系统在WAN上的统一部署、管理和维护,大大增加了系统的可扩展性;MADIDS使用移动Agent来实现入侵证据收集、数据分析与判断、通信和同步机制,比传统的入侵检测系

*)本文受信息产业部电子信息产业发展基金支持。陈 波 博士研究生,教授;罗光春 博士,副教授;卢显良 教授,博士生导师。

统具有更好性能。MADIDS 还采用了高效的 Agent 代码加载机制和"分层汇聚"特征分析机制来降低在 WAN 中的网络开销，并通过 Agent 自我复制的功能来增加系统的抗毁和自我修复能力。MADIDS 的体系结构如图 1 所示。

在 MADIDS 中，Agent 担负着数据传输、通信、管理协作等任务。如图 2 所示，在 MADIDS 中 Agent 可以划分为以下几类：(1)IA(Interface Agent)：IA 运行在客户机上，接受事件产生器(Event generators)的功能调用，然后通过派遣、控制或和其他 Agent 协作来进行工作；(2)WA(Working Agent)：WA 接收来自 IA 的命令或者申请，在网络中移动并通过事件分析器(Event analyzers)执行入侵特征分析，然后返回执行结果信息；(3)DMA(Domain Manage Agent)：负责 MADIDS 一个域中名字管理、分层事件分析管理、属性管理、空间管理和访问控制管理。DMA 可以自身复制和移动以靠近使用的数据来获得更好的效率和性能；(4)MMA(Main Manage Agent)：MMA 担负着协调与管理 MADIDS 中所有 DMA 运行的任务。MMA 和 DMA 相互协作，完成整个 MADIDS 中的各项管理工作。

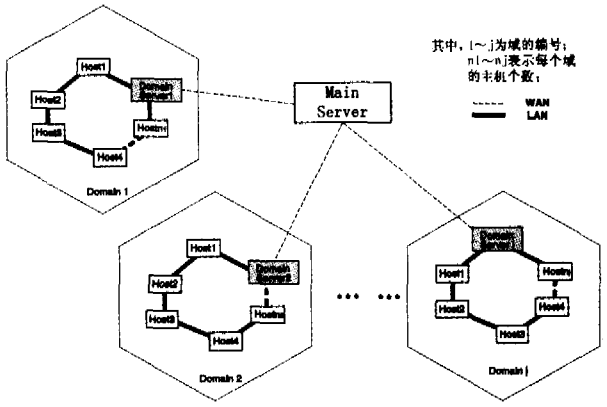

图 1　MADIDS 体系结构

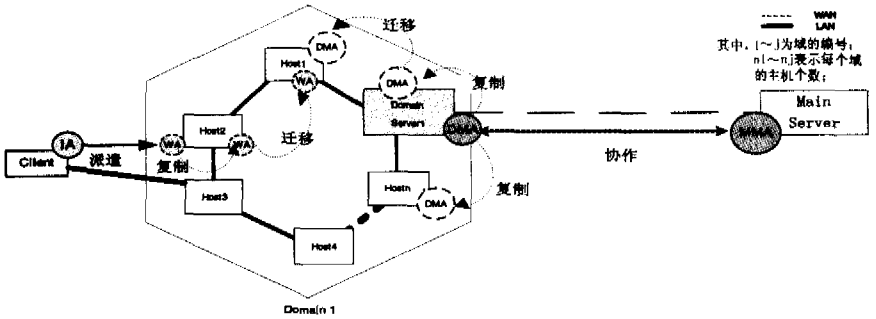

图 2　MADIDS 中的 Agent 分工协作图

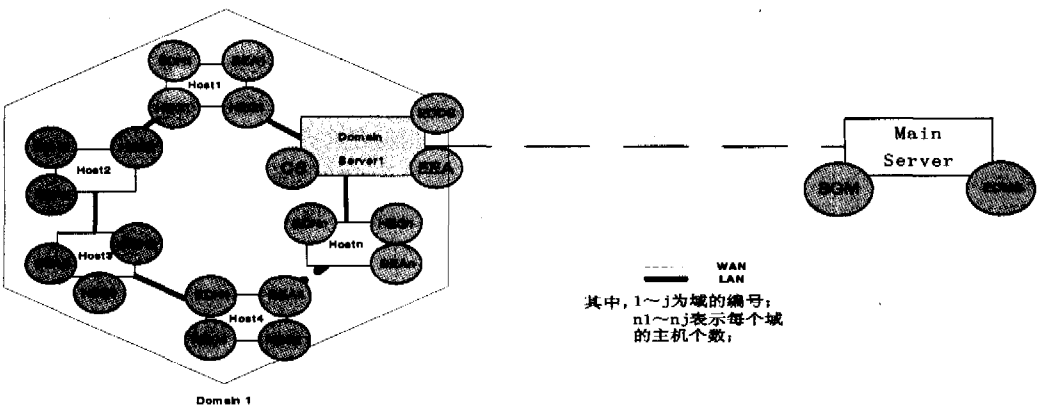

图 3　MADIDS 中各入侵检测模块的部署情况

MADIDS 中各入侵检测模块的部署情况如图 3 所示：(1)基于主机的事件产生器 HEG(Host based Event generators)：在 MADIDS 中，事件产生器分为基于主机的事件产生器 HEG 和基于网络的事件产生器 NEG(Network based Event generators)两种。对处于 MADIDS 中的每台主机上，都部署一个 HEG，用于获取本地主机上的入侵原始数据，例如进程的报告、监控和禁止，数据收集、日志记录和报告等；(2)基于网络的事件产生器 NEG(Network based Event generators)：NEG 并不需要在每台主机部署，在一个域的所有主机中仅部署两台即可，其一作为主 NEG，用于获取网络上的入侵数据，另外一个作为备份 NEG，可用于系统功能的备份和修复、抗毁；(3)基本事件分析器 BEA(Basic Event analyzers)：事件分析器 EA 从逻辑上分为基本的事件分析器 BEA 和扩展的事件分析器 EEA(Extend Event analyzers)两种。BEA 配置于每台主机，包含基本的功能，可实现常见攻击行为的分析检测；(4)扩展的事件分析器 EEA(Extend Event analyzers)：EEA 在每个域中，不配置于每台主机(Host)，而是放置于域服务器(Domain Server)中，具有更复杂和完善的分析功能，例如基于神经网络的检测算法，基于数据挖掘的匹配算法等；(5)控制台 CS(Console)：在传统 IDS 中，控制台通过接受事件分析器的分析结果，来判定是否存在入侵行为，并做出相应对策，在 MADIDS 中 CS 仍起相同作用；(6)响应单元 RU(Response units)：响应单元 RU 同传统的 IDS 中一样，存在于整个系统需要的位置，并对入侵行为进行相应响应和处理；(7)事件数据库 ED(Event databases)：事件数据库分 3 个层次存在于主服务器(Main Server)、域服务器(Domain serv-

er)和所有主机上(Host),对应以上分层,分别为存放于主服务器的 EDMS(Event databases saved in Main Server),存放于域服务器的 EDDS(Event databases saved in Domain server)和存放于主机的 EDH(Event databases saved in Host);(8)系统管理平台 SGM(System General Manager):SGM 存在于主服务器(Main server)上,负责 MADIDS 系统的整体管理、维护、升级等工作,它不直接参与入侵检测的具体事务。

MADIDS 分层体系结构的优点是:(1)DMA 负责管理一个域,MMA 负责管理所有 DMA。这种体系结构避免了 MADIDS 中所有管理工作集中在少数服务器上,而使这些服务器成为系统扩展的瓶颈;(2)在 MADIDS 中域内通过 LAN 连接,带宽大、延迟小。因此 MADIDS 在域内通信时可以使用针对 LAN 设计的协议来获得较高的性能;同时在域内还可以认为主机之间是相互信任的,可以适当地降低安全标准来获得更好的性能。当跨域通信时,可以使用针对 WAN 环境设计的通信协议和安全协议,以获得较好的性能和安全性;(3)在一个局域网中针对一个主机的很多入侵行为,也可能同时发生在 LAN 内的其他主机上,在一个局域网中多个主机的入侵检测协同可以有效地提高检测效率。

2.3 MADIDS 的维护更新机制

MADIDS 是在 WAN 上使用的分布式入侵检测系统,因此 MADIDS 的维护、升级和管理应该尽量减小网络通信量,特别是在 WAN 上的通信量。为此设计了分层更新机制。这种机制可以有效降低服务器负荷,在保证系统整体更新高效、准确、同步的同时,网络通信量也较小,较好地适应 WAN 的运行环境。在 MADIDS 中,系统更新的主要内容是含有检测规则的事件数据库,只有这样才能保证整个系统对入侵行为进行检测和处理的有效性,这一点类似于病毒处理软件。

事件数据库 ED(Event databases)分为存放于主服务器的 EDMS(Event databases saved in Main Server),存放于域服务器的 EDDS(Event databases saved in Domain server)和存放于主机的 EDH(Event databases saved in Host)。同时,三个层次的所有事件数据库都设计为带有时间标记的增量式数据结构,即在任一数据库初始化后,所有时间段内更新的数据都可以方便地查找到。在以上层次结构的基础上,MADIDS 中存在着两个系统更新流程:系统整体规则生成流程和系统整体规则更新流程。限于篇幅,仅介绍整体规则生成流程。

从系统的角度看,更新的来源有两种,即系统外和系统内,系统外的更新来源可能是引入了其它系统的经验和规则,例如其它系统所发现并经证实的新入侵行为特征,或者引入了新的智能检测算法,等等;从系统内生成新规则,则是在本系统的局部发现了新的可能存在的入侵行为特征后,通过"分层核对"方式提交给系统并完成更新,按以下步骤进行,如图 4 所示。

(1)在 $Host_1$ 上,发现了新的可能入侵行为标记 NMLSI(New Marks Left by Suspected Intruders);(2)Host 生成一个 Agent,发送到 DUS_1 申请核对是否该入侵特征已存在,只是本地主机没有;(3)Agent 到达 DUS_1 后,通过 DUS_1 查找 $EDDS_1$,若该行为特征已存在,表示整个系统已存在该规则,只是 $Host_1$ 还未得到及时更新,于是 $Agent_1$ 返回 Host,并结束运行;若该行为特征标记 NMLSI 通过 DUS 未得到验证,则 Agent 自我复制为 $Agent_1$、$Agent_2$、…、$Agent_{n-1}$(n=MADIDS 中域的总数);(4)$Agent_1$ 移动到 DUS_2,$Agent_2$ 移动到 DUS_3,相应的其它 Agent 也分别移动到其它域的 DUS 上。一般地,$Agent_{n-1}$ 移动到 DUS_n 上,分别通过 DUS 查找相关 EDDS,核对是否存在该入侵行为特征标记;(5)查找结束后,所有 Agent 返回到 MUS 上并互相交换信息,若其中任一 Agent 发现该行为特征已存在,表示整个系统已存在该规则,只是局部还未得到及时更新,于是除 $Agent_1$ 外的所有 Agent 结束运行,$Agent_1$ 返回 $Host_1$,并结束运行;若所有 Agent 均为核实该 NMLSI,则其它所有 Agent 结束运行,$Agent_1$ 向 MUS 发出申请;(6)Agent 在 MUS 上,对 EDMS 进行核实,如果发现该行为特征已存在,表示整个系统已存在该规则,只是较低层次还未得到及时更新,于是 $Agent_1$ 返回 $Host_1$,并结束运行;若 $Agent_1$ 未核实到该 NMLSI,则 $Agent_1$ 向 MUS 发出更新 EDMS 的申请,在得到 MUS 已更新 EDMS 的确认后,$Agent_1$ 返回 $Host_1$ 并结束运行。

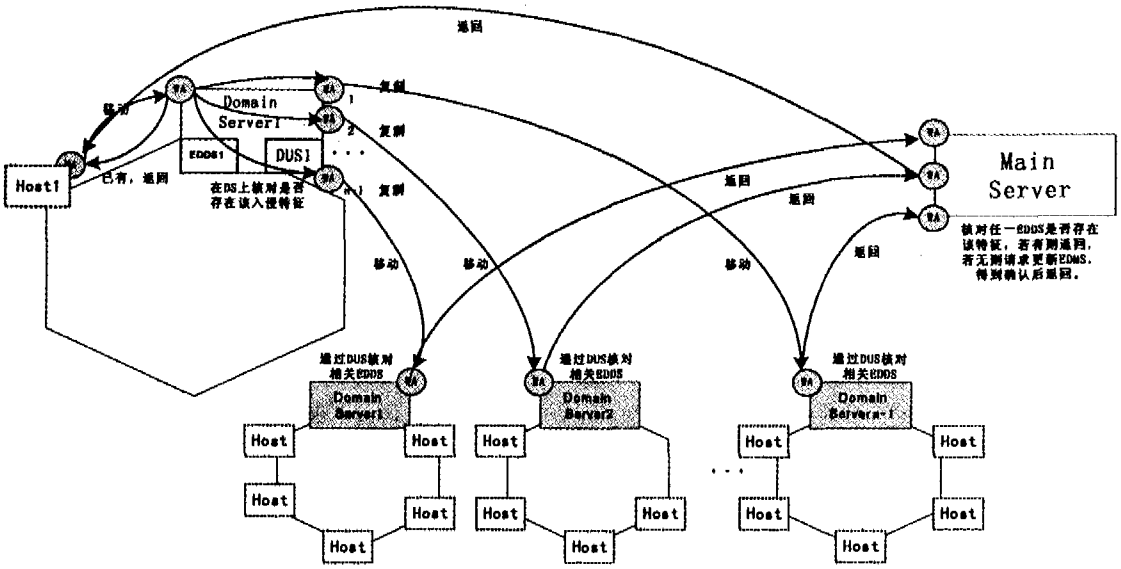

图 4 MADIDS 的分层核对机制

(下转第 110 页)

4 实例

为了证实这个思想，作者用 C++ 编程语言开发了产生这类椭圆曲线的程序。选择素数长度，比方说，192 位，这个程序生成一个素数 p = 1928952416777687917470659164182613737573049894109870667458 9。这个素数能被表示为 W^2+4V^2，W=55002053592931705449171458117，V=637657789655571204607292216 65。这个椭圆曲线的阶 $\#E(F_p)$ 是 192895241677687917470659164171613326854463553020036375 8356。并且 $\#E(F_p)=h*r$，h=4，r = 4822381041944219793676647910429033317136158882550090939589。数字 r 是素数。执行程序耗时 15 秒。椭圆曲线 $E(F_p)$是密码系统安全的。

结论 本文提出了一个产生一类椭圆曲线的方法。当 $p=6k+1$，并且 p 是素数时，那么 p 能够被分解为 W^2+4V^2。本文也证明了在有限域 F_p 具有 j 不变量为 1728 的椭圆曲线 E 的阶是 $p+1\pm 2W$（当 $W=4L+1$ 时，$\#E(F_p)=p+1+2W$；当 $W=4L-1$ 时，$\#E(F_p)=p+1-2W$），同时给出了构建这类椭圆曲线的快速算法。

参 考 文 献

1 Muller V, Paulus S. On the Generation of Cryptographically Strong Elliptic Curves:[Technical Report]. Technical University of Darmstadt, 1997

2 Schoof R. Elliptic curves over finite fields and the computation of square roots mod p. Math. Comp. ,1985,44:483～494

3 Atkin A O L, Morain F. Elliptic curves and primality proving. Mathematics of Computation,1993,61(203):29～68

4 Kurotani K, Matsuo K, Chao J, et al. Consideration of security of hyperelliptic cryptosystems. IEICE, Symposium on Cryptography and information Security, SCIS'98, 4.1-D, Jan. ,1983

5 Johnson D, Menezes A. The Elliptic Curve Digital Signature Algorithm (ECDSA):[Technical report CORR 99-34]. Dept. of C&O, University of Waterloo, Canada. Available at: http://www.cacr.math.uwaterloo.ca

6 Ireland K, Rosen M. A Classical Introduction to Modern Number Theory. Spring-Verlag, 1982

7 Morain F. Building cyclic elliptic curves modulo large primes. In: D. Davies, editor, Advances in Cryptology - EUROCRYPT'91, volume 547 of Lecture Notes in Comput. Sci. , Springer-Verg, 1991. Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, Brighton, United Kingdom, 1991. 328～336

(上接第 105 页)

3 实验环境、结果与性能分析

为验证 MADIDS 的性能，针对其规则更新进行了实验。由于条件限制，实验环境仅在电子科技大学校园网内选取了三处位置进行，如图 5 所示。其中，主服务器 MS 放置于子网 202.115.14.1/24 内；第一个域的各个节点包括 DS1、Host1、Host 2、Host 3，放置于子网 202.112.10.1/24 内；第二个域的各个节点包括 DS2、Host4、Host 5、Host 6，放置于子网 202.115.1.1/24 内。其中主服务器为一台 Intel 服务器，配置为 1.2G CPU，512M 内存、100Mbps 以太网卡，其它 8 个节点均采用 PC 赛扬 666。

数据库平台为 mysql4.0，实验中管理的 IDS 为 snort。移动 Agent 平台采用 Windows NT4+Aglet1.3+JDK1.0 来搭建，通过 Aglet Workbench 及相应软件包实现。实验中首先配准所有主机的时钟，在每个节点上都由一个读系统时间的程序进行计时，并与 MADIDS 的运行配合进行。对规则更新进行了 500 次实验，从 Main Server 上的 EDMS 更新开始计时，直到最后一个 Host 上的 EDH 得到更新为止，作为一次完整的实验过程。由于已对所有节点的时钟进行了配准，系统一次规则更新的周期时间，可由最后一个完成移动 Agent 返回动作的 Host 上的时间标记，减去 MS 上的初始时间标记得到。从实验情况来看，每次实验都能完成 MADIDS 的设计目标，一个新规则在两个域的 8 个节点得到全面配置，最短耗时 0.8 秒，最长耗时为 3.3 秒。周期长短取决于当前网络拥塞状况、移动 Agent 执行效率、数据库执行效率等因素。

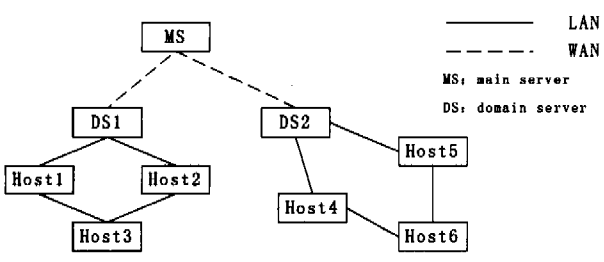

图 5 MADIDS 实验环境示意图

由实验与分析可知，这种"分层核对"机制具有如下优点：(1)有效地减少 MADIDS 中对新的入侵行为特征标记的管理和复制入侵行为特征标记内容的通信量，特别是在 WAN 上的通信量；(2)Main Server 不需维护每一个主机的新入侵特征，而仅仅需要维护 Domain Server 即可，这样就大大降低了维护的整体开销。限于篇幅，规则生成机制外，其它内容不再赘述。

结论 本文提出了一种基于移动 Agent 的新型入侵检测系统 — MADIDS。MADIDS 使用分层体系结构，整个系统由若干域组成，域内通过 LAN 连接，域间通过 WAN 连接。每个域由域服务器管理，所有域服务器由主服务器管理。这种体系结构保证了很好的伸缩性和扩展性，MADIDS 在系统管理中使用了"分层生成"机制，这种机制降低了维护系统整体性和一致性的负荷，网络通信较小，非常适合在 WAN 中使用。

参 考 文 献

1 Mell P, McLarnon M. Mobile agent attack resistant distributed hierarchical intrusion detection system. In: Proceedings of RAID' 99, CERIAS, Purdue University, 1999

2 Gregory M, White B, Fisch E A, Pooch U W. Cooperating security managers: A peer based intrusion detection system. IEEE Network, 1996.14

3 Slagell M. The Design and Implementation of MAIDS (Mobile Agents for Intrusion Detection System). Masters Creative Component paper, Mark Slagell, Iowa State University, May 2001

4 张云勇. 移动 Agent 及其应用. 北京：清华大学出版社，2002

5 Baumann J, Hohl F, Rothermel K, Straβer M. Mole - Concepts of a Mobile Agent System. WWW Journal, Special issue on Applications and Techniques of Web Agents, 1998

6 Karnik N M, Tripathi A R. Design Issues in Mobile-Agent Programming Systems, IEEE Concurrency, 1998, 6(3):52～61

7 Johansen D. Mobile Agent Applicability. In: Rothermel K, et al. eds, Mobile Agents: MA'98: proceedings, Berlin[etc.]: Springer, 1998, Lecture notes in computer science; ISBN 3-540-64959-X, 1998, 1477