文章编号:1001 - 9081(2011)06 - 1521 - 04
doi:10.3724/SP.J.1087.2011.01521

应用非迭代 Apriori 算法检测分布式拒绝服务攻击

高　琰[1,2],王台华[1],郭　帆[1],余　敏[1]
(1. 江西师范大学 计算机信息工程学院,南昌 330022;　2. 江西省气象信息中心,南昌 330046)
(gaoyan1011@sina.com)

摘　要:提出了一种非迭代 Apriori 算法,无需多次扫描事务数据库,使用一步交集操作处理同一时间段的网络数据包,通过挖掘各数据包之间的强关联规则,可较快检测分布式拒绝服务(DDoS)攻击。与现有算法相比,检测 DDoS 攻击的时间和空间性能较优。在 DARPA 数据集上的实验结果表明应用该算法能有效检测 DDoS 攻击。

关键词:数据挖掘;Apriori 算法;分布式拒绝服务攻击;入侵检测
中图分类号: TP393.08　　**文献标志码**:A

DDoS detection with non-iterative Apriori algorithm

GAO Yan[1,2], WANG Tai-hua[1], GUO Fan[1], YU Min[1]
(1. *School of Computer and Information Engineering, Jiangxi Normal University, Nanchang Jiangxi* 330022, *China*;
2. *Jiangxi Provincial Meteorological Information Center, Nanchang Jiangxi* 330046, *China*)

Abstract: An improved non-iterative Apriori algorithm was proposed to detect Distributed Denail of Service (DDoS) attacks. An one-step intersection operation was used to process network packets within the specific time range, and the strong correlation rules of the packets were studied so as to achieve the quick detection of DDoS atttacks. In comparison with current algorithms, it shows better performance in efficiency and storage space in detection of DDoS attacks. Experimental results on DARPA data-sets show the algorithm is able to detect DDoS effectively.

Key words: data mining; Apriori algorithm; Distributed Denial of Service (DDoS) attack; intrusion detection

0　引言

拒绝服务(Denial of Service, DoS)攻击是目前较常见的网络攻击行为,这类攻击主要导致网络或系统因过载而停止提供正常的网络服务,其中最常见的攻击是对目标发送大量的攻击数据包来消耗或阻塞目标主机或网络的资源,它们统称为数据包洪泛攻击,例如 SYN 洪泛、UDP 洪泛、Smurf 攻击等。

随着计算机性能和网络带宽的提高,目标系统有能力抵御 DoS 攻击,使得一般的 DoS 攻击成效变得微乎其微。分布式拒绝服务(Distributed Denail of Service, DDoS)攻击利用成千上万分布在 Internet 上的计算机,同时产生大规模的数据包,把这些数据包送往同一目标系统并导致其瘫痪。DDoS 攻击已经成为近年来对 Internet 具有极大影响的恶意攻击方式。

如图 1 所示,一个完整的 DDoS 攻击体系通常分成四个部分。①真正的攻击者,它们操纵整个攻击过程。控制一台或多台傀儡机作为发送控制指令的机器;②控制主机,黑客用来给攻击机发送指令的中转站,控制主机上运行着一种可控的代理程序控制大量的攻击机进行攻击;③攻击后台代理或傀儡攻击机,它们运行着一种特定程序产生数据流发送到受害者,这些傀儡主机往往位于受害者所属网络之外以逃避受害者的响应,同时也处在攻击者网络之外因此难以被追踪;④目标机,即攻击的受害者。

本文提出一种改进的数据挖掘算法,可在目标系统受到 DDoS 攻击时,根据当时数据流量的特征挖掘出相应的规则形成报警。

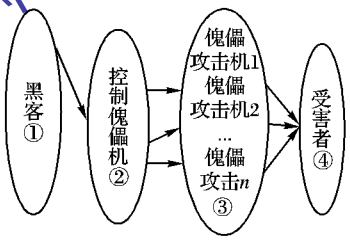

图 1　DDoS 攻击原理

1　相关工作

自 1999 年第一次爆发大规模 DDoS 攻击后,研究人员提出了众多针对 DDoS 的防御和分析方法。文献[1]提出基于源端网络防御 DDoS 攻击,在源端网络配置防御系统,通过监控网络的双向业务流并周期性地与正常业务模型相比较来检测攻击,并对可疑包进行速率限制,使得攻击发生时也能为正常业务提供服务,但是该方法只能发现 DDoS 攻击存在,无法实际检测和分析追踪。文献[2]提出一种消除 DDoS 攻击的综合方案,包括 IDS、IP 标记和 IP 包过滤等功能,但是该方案只能作用在局部范围内。文献[3]对 DDoS 攻击进行了形式化描述,对 DDoS 攻击的原理和特征给出了完整的分析。文献[4]对基本型概率包标记方案进行了改进,使得计算量大大减少,能有效追踪控制主机。文献[5]综合叙述了多种 DDoS 攻击预防、检测、响应的方法,设计了一种 ICMP 回溯攻击源的防范方法,采用合作过滤机制使产生的 ICMP 回溯包更有效并在尽可能靠近 DDoS 攻击源的地方过滤攻击包和保护合法包。

收稿日期:2010 - 11 - 22;**修回日期**:2011 - 01 - 19。　**基金项目**:国家 973 计划项目(2007CB316505)。
作者简介:高琰(1973 -),女,江苏阜宁人,硕士,主要研究方向:数据处理、信息安全;　王台华(1987 -),男,江西吉安人,硕士研究生,主要研究方向:信息安全、软件体系结构;　郭帆(1977 -),男,江西于都人,副教授,博士,主要研究方向:信息安全、软件体系结构;　余敏(1964 -),女,江西南昌人,教授,博士,主要研究方向:分布式计算、信息安全。

研究人员也提出了一些应用数据挖掘技术检测 DDoS 的方法，如文献[6]提出了基于数据挖掘的 DDoS 攻击入侵检测系统，首先将网络原始数据包转化成连接记录，再使用 *K*-means 方法，将训练数据进行聚类划分，最后使用 Apriori 算法将连接记录转化成流量特征形成关联规则。文献[7]列出多种改进的 Apriori 算法融合了多种技术，如散列项集计数、事务压缩、划分、选样和动态项集等，主要在控制候选集的规模和减少数据库扫描次数等几个方面提高算法的效率。并且文中还提出了对一种 DHP 的改进算法 DHPP，该算法使用一个数组来替代哈希树进行候选 2-项集的出现次数统计。

本文采用了与文献[6]类似的检测框架，但是与传统 Apriori 算法对海量数据库进行多次扫描和连接运算来产生频繁项集不同，本文设计的改进算法不需要多次扫描数据库而通过交集次数来获得候选集的支持度，有效减少了时间消耗，有助于快速发现 DDoS 攻击。

2 应用改进 Apriori 算法检测 DDoS

2.1 改进算法的提出

文献[8]提出了一种非迭代 Apriori 算法用于报警关联，该方法避免了传统 Apriori 算法的一些缺陷如需要多次数据库扫描、生成大量候选集以及迭代求解频繁项目集等，仅需一步交集操作即可得到最大频繁项目集。支持度可由交集的次数得到而无需再次扫描事务数据库，并且将报警中的一些属性进行编号能减少存储空间且方便搜索候选集列表，从而提高算法效率。但是，该方法仅用于在海量报警信息中挖掘各报警之间的关联规则，并没有对具体报警属性之间的关联进行深入分析，特别是该算法没有挖掘报警的时间属性，因此难于检测那些将一条完整的攻击链分成多个攻击步骤来实现的攻击。

本文在文献[8]的基础上对非迭代 Apriori 算法做出进一步改进和调整，使用一步交集操作处理同一时间段内的网络数据包，同时增加对各报警属性的分析，特别是时间和连接标记等属性，寻找各数据包属性之间的关联特征并有效检测 DDoS 攻击。

2.2 理论基础

洪泛式 DDoS 攻击，无论是采用源 IP 伪造技术，还是采用 IP 反射技术来隐藏攻击者的真实地址，以避免检测系统的追踪，攻击发生时往往会出现以下一种或几种以下现象：1）攻击发生前域名服务器接收到大量的反向解析目标 IP 主机名的 PTR 查询请求；2）攻击发生时数据包的流量会明显超出正常工作时的极限；3）出现特大型的 TCP 和 UDP 数据包；4）不属于正常连接通信的 TCP 和 UDP 数据包；5）在短时间内产生大量的数据包，且数据包的目标 IP 和端口相同，在一段时间内重复出现类似数据包。

本文方法检测 DDoS 攻击主要依据：1）许多数据包具有发生 DDoS 攻击时的异常现象；2）这些数据包的数量呈现周期性的增加，通过挖掘上述攻击特征形成关联规则。通过将一步交集操作加入时间判断，对所有数据包转换成流量记录并按照时间阈值分段，只有处在同一时间段的记录才做交集操作，这不仅减少了交集个数提高了算法效率，还可以体现 DDoS 攻击的时间特性，使得同一段时间内产生的规则特征更集中。

2.3 算法设计

针对 DDoS 的原理和攻击特点对非迭代 Apriori 算法做出的相应改进包括：1）对交集操作作剪枝处理，两条记录发生的时间差在一定阈值之内才做交集，否则将两条记录看成不相关记录，或者其中一条是下一轮攻击中的记录；2）对项集组的时间阈值做动态调整，如果新产生的项集和已产生的候选集合并，则更新该候选集的时间上下限值，否则新项集被视为一个新的候选集；3）产生的候选集中必须含有目标 IP、端口、协议、发生时间、连接状态等属性。

在预处理过程中，将会对一些属性分别进行数字编号，在减少存储空间同时也方便了查找候选项集列表。

改进后的算法步骤包括：

步骤 1 解析数据包流量。提取出每个包的属性（时间、源和目标 IP、端口、包类型、攻击方式、连接状态、TTL 值等等），并对其中一些属性编码，形成一个个记录，作为算法的输入。

步骤 2 对记录按照时间分段做交集。若两个记录时间差值在给定的时间阈值内，则做交集操作形成候选集，如果这个候选集已经存在，则它的支持度计数，同时修改这个时间段的界值。否则形成一个新的候选项，时间段即两个记录的差值。

步骤 3 生成频繁项集。任一个支持度超过给定的支持度阈值的候选集，就是一个频繁项集。

步骤 4 形成关联规则。算法只生成形如这样的规则 $X \rightarrow Y$，其中 Y 是攻击部分，而 X 就是一些特征属性的集合。若一条规则的置信度超过给定的阈值，即认为这是一条强关联规则。

算法的形式化描述如下：

输入：事务数据库 $D = \{T_1, T_2, \cdots, T_n\}$，最小事务支持阈度 *min_sup*、*min_conf*、*interval*。

输出：最大频繁项目集 L。

```
//预处理部分
for( i = 1;i < = n;i++ ){
  for( j =1; j < = m;j++ ){
  If ( item_j ∈ T_i )
    Convert( item_j );                          //将属性编码
  }
}
//算法核心
Count =0;                                       //项集列表元素个数
For each T_i ∈ D,T_j ∈ D(i ≠ j) {
  //两事务时间差值在阈值内,做交集
  If( abs( T_i . time - T_j. time) < = interval ) {
  temp = T_i ∩ T_j;
  If ( temp ! =∅) {                             //若交集不为空
    //搜索候选项集列表
    index =Serchitem( temp );
    If ( index ! = -1) {
      //若已经存在,则自动计数且修改时间上下值,否则将此
      //候选集加入候选列表
      UpdateTime( itemsets,temp );
      hash[ index ]++;
    else{
      itemsets. add( temp );
      hash[ Count++ ] =1;
    }
  }
}
For each itemsets. get( k ) ( k ≤ Count ) {
  If ( hash[ k ] ≥ min_sup - 1 ) {
    //候选集 k 就是一个频繁候选集
    Add itemsets. get( k ) to L
  }
```

```
}
//关联部分,针对入侵检测系统中的报警规则来形成关联规则
For each L_i ∈ L {
  Y = Attacks;                                    //攻击部分
  X = L_i - Y;
     //如果这条规则是强关联规则,那么产生一条有用的规则
  If( conf( X → Y) ≥ min_conf ) Associate( X → Y)
}
```

2.4　算法比较

本节通过对一个样本数据库的挖掘分析,将 2.3 节算法与传统 Apriori 以及 DHPP 算法进行性能比较。给定事务数据库 D[7],其中有四个事务,假设最小支持度 $minsup = 2$,且项集中的项都是有序的。如表 1 所示。

表 1　事务数据库 D

TID	Items
1	{ACD}
2	{BCE}
3	{ABCE}
4	{BE}

Apriori 算法:

1)1-候选集无需做连接,扫描 5 次事务数据库,得到 4 个频繁项集{A2, B3, C3, E3};

2)由第一步的 4 个频繁项集做 6 次连接得到 6 个候选集,扫描 6 次事务数据库,得到 4 个频繁项集{AC1, BC2, BE3, CE2};

3)迭代生成 3-候选集需要做 6 次连接得到 4 个候选集,扫描 4 次事务数据库得到一个频繁项集{BCE2},算法结束。

Apriori 算法需要 12 次连接操作和 15 次扫描事务数据库,一共得到 9 个频繁项集,其中项集{AC, BCE}是最大频繁项集,算法除了保存每一步中的候选集之外,没有额外开辟空间。

DHPP 算法如下所示。

1)由 Apriori 算法生成 1-频繁项集,且生成一个 4 个空间的一维数组 remap 用于保存 1-频繁项集中的项,且下标对应各项的 ID,同时生成 4×3 个空间的一维数组 fre2support 来存放 2-项集的支持度。初始化为 0。

2)扫描事务数据库的每一个事务。先生成一个临时长度为 4 的数组 list 来记录 T1 中含有频繁项集的 ID,接下来每扫描一个事务的 ID 来生成候选项集的全部组合,然后在 fre2support 中对应的位置支持度加 1。这一步只扫描了 1 次事务数据库。

3)重复 2)的过程。这一步也只扫描了一次事务数据库。算法结束。

DHPP 算法依靠开辟额外数组来保存每个候选集中的项的变化和支持度,因此没有做自连接运算,但在第 2)步开始每一步都需要开辟 4 个空间的临时数组。算法一共扫描了 6 次事务数据库,使用了 22 个额外的空间。本文算法如下所示。

对 4 个事务分别做 C42 次交集操作得到 4 个候选集,再遍历一次候选集列表统计计算其支持度,得到 4 个频繁项集{C2, AC2, BCE2, BE2},且项集{AC, BCE}为最大频繁项集。算法只用了 6 次交集操作 4 次比较操作和 4 个额外空间就得到了最大频繁项集。

从上述分析可知,随着事务数据库的项数增多,扫描数据库的时间开销也相应增多,Apriori 算法需要迭代的次数就越多。而 2.3 节的非迭代 Apriori 改进算法不需迭代求解候选集,只需扫描一次事务数据库。

DHPP 仅在生成 1-频繁项集时需要多次扫描事务数据库,扫描数据库次数与本文算法相当。但是它所需要的额外存储空间比本文算法大很多,而且 DHPP 需要产生所有的频繁项集,而在实际的 DDoS 检测中,往往只需要寻找最大频繁项集即可。本文算法不需产生所有频繁项集,而是通过一次交集操作即可得到所有的最大频繁项集,其余的频繁项集在需要时可通过对最大频繁项集的分解得到。

本文算法对于 DDoS 攻击发生时流量特征的挖掘,是根据最大频繁项集的流量特征来产生关联规则,从而降低误警率。

3　实验分析

为了验证算法的有效性,采用 Snort 2.8.3.2 的 Win32 版本在 alert_full 模式下输出报警。网络数据采用 MIT Lincoln 实验室提供的 DARPA2000[9] 数据集,本文选择 2000 年场景 1 LLDOS1.0 和场景 2 LLDOS2.0.2。两个场景都是林肯实验收集一天的数据流量。实验在离线状态下实现。

原始网络数据首先经过 SNORT 处理生成文本信息;然后将数据包按照一定的格式(源 IP,源端口,目标 IP,目标端口,协议,包描述信息以及攻击类型的信息,攻击发生时间等作为挖掘的属性)经过预处理后成为一条条记录作为事务数据库中的事务,作为 2.3 节中算法的输入;整个算法使用 Java 语言在 Eclipse3.2 平台上实现,算法对经过预处理后的数据包进行挖掘得到发生 DDoS 时数据包的特征,形成规则。

表 2 列出了 DARPA2000 原始数据的信息,分别是场景 1 LLDOS1.0 和场景 2 LLDOS2.0.2 内网和 DMZ 中的数据流量包经过包的解析过程,提取相应的属性形成的记录数。表 3 列出了 3 组不同支持度时的实验效果,取置信度为 0.1。每一组分别对两个场景进行分析得到异常数据包流量、误报率、虚报率的实验结果比较。

表 2　DARPA2000 原始数据表

原始数据	场景			
	LLDoS1.0		LLDoS2.0.2	
来源	内网	DMZ	内网	DMZ
数据包流量	649 787	394 089	347 987	236 753

实验表明应用场景 1 LLDoS1.0 中在阶段 1 到阶段 5 收集

表 3　不同阈值实验结果的比较

数据项	第一组		第二组		第三组	
	LLDoS1.0	LLDoS2.0.2	LLDoS1.0	LLDoS2.0.2	LLDoS1.0	LLDoS2.0.2
支持度	0.01	0.01	0.001	0.001	0.1	0.1
异常数据包流量	542 756	276 878	876 524	468 242	156 846	89 768
误报率	0.004	0.003	0.052	0.067	0.000 7	0.000 9
虚报率	0.001 3	0.000 9	0.004	0.005	0.000 3	0.000 2

的数据来说明关联出的数据包特征是有价值的。实验关联出阶段 5 发起的一次 DDoS 攻击,实验挖掘出有两类明显特征的数据包:由源 IP 131.84.1.31 通过大量的非常用端口发送的数据包,只有 R 标记且 ACK 值为 0;还有就是对应的回应包,目标 IP 为 131.84.1.31 通过非常用端口接收到的数据包,只有 A 标记且 ACK 值为 0,这两类数据包还具有 IpLen:20、Seq:0x0、Win:0x0、TcpLen:20 的特征。基本上每隔 0.01 s 有 130 多个带有这两条攻击特征的数据包。这正还原了 LLDoS1.0 数据集的一个完整的攻击链。如图 2 所示,攻击者 131.84.1.31 首先寻找是否存在 rpc sadmind 漏洞,接下来对 RPC 请求解码,然后发起 exploit 攻击获得管理员权限,最后调用 rsh 得到远程 Shell,攻击机 131.84.1.31 通过远程登录控制那些正在运行攻击程序的傀儡机,最后由三台主机通过随机端口同时发送大量的连接请求,形成一次分布式拒绝服务攻击。实验还关联出攻击机 131.84.1.31 通过 80 端口大量的与其他主机通信的数据包,这些数据包可以看成是主机正在试探目标机的运行状态,为发起 DDoS 做好准备,在算法的预处理部分可以通过过滤 80 端口的数据包来避免这类虚警。

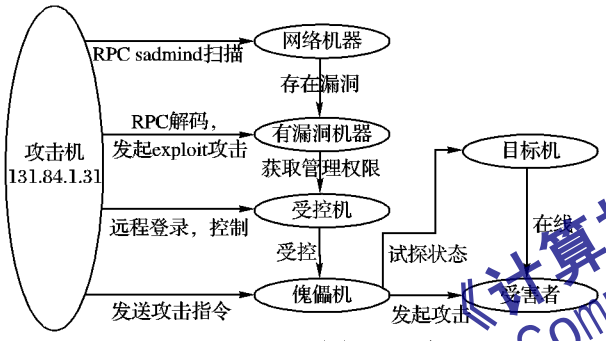

图 2 LLDoS1.0 一个完整的攻击链

在对场景 2 DARPA2000 LLDDoS2.0.2 挖掘中,也可以发现实验关联出来的结果类似于场景 1 的报警关联,这是因为林肯实验室场景 2 的实验数据的五个攻击步骤和场景 1 是一样的,我们从表 2 可以看出它们的区别在于场景 1 的实验数据量更大,而且它加入的正常数据包的数量也更多。

DARPA2000 建立了两个包含多个攻击方法和步骤的攻击过程数据集 LLDoS1.0、LLDoS2.0.2,其中 LLDoS1.0 包含了利用 IPSweep 探测被攻击系统的 sadmind 服务,利用该服务的缓冲区溢出漏洞获得三台被攻击系统的 ROOT 权限,然后分别在这三台机器上安装 DDoS 攻击工具的攻击程序,并在其中的 1 台机器上安装控制程序,然后通过控制程序控制攻击程序发动 DDoS 攻击;最后进一步分析 DARPA1999 测试的数据,提供检侧各种攻击的特征和模型,LLDoS2.0.2 采用比 LLDoS1.0 更隐秘的方法实现攻击。

从表 3 中可以看出在支持度阈值为 0.1 的实验效果是很好的;如果设置较小可以保证异常的数据包会被检测出来,但也有很多正常的通信连接包会当成虚警,例如在场景 1 中攻击机 172.16.113.84 通过 80 端口对目标机 209.67.29.11 提出通信请求的数据包,这个可以看成是攻击机在试探目标机的状态,属于正常的 TCP 连接请求,而不属于 DDoS 攻击;而如果支持度设置较大的时候则导致含有某些特征的异常包由于支持度较小而被当成正常包过滤掉,从而形成漏检。

3.1 一些说明

DARPA2000 评估数据集是入侵检测领域使用最为广泛的测试数据集,但是它自身也有一些不足,比如攻击过程过于标准化。文献[10]中指出 DARPA1999 数据集的几个缺点:被攻击的目标网络拓扑结构过于平坦;被攻击主机分布不平均;收集的网络数据与典型的美国空军网络流量缺乏统计学上的相似性。而这些问题在 DARPA2000 数据集中依然存在。另外,本文使用 DARPA2000 数据集作为唯一的测试源可能会使结果具有一定的片面性。但是,DARPA2000 中含有的 DDoS 攻击种类有很多,而且攻击的顺序成一定的规律,比较容易检测,对于检测是否发生 DDoS 攻击来说还是具有很高的实验指导意义。

实验结果表明我们可以从大量的数据包流量中找出是否发生 DDoS 攻击,并且可以挖掘出发生 DDoS 攻击时,这些数据包流量记录的特征属性,可以帮助管理员提前处理,减少 DDoS 发生的危害性和解决问题的工作量。

4 结语

传统的 Apriori 算法使用迭代自连接会产生大量的重复候选集和不必要地扫描事务数据库,效率很低。本文提出一种通过移除迭代而使用一步交集操作来生成频繁项集的改进算法。在加入对时间属性的挖掘之后,实验表明,改进算法能够挖掘到 DDoS 攻击所持有的特征属性形成强关联规则,这些规则表示了攻击目标和攻击者的意图,以及数据包的固有特征,为管理员处理 DDoS 攻击提供方便。

该算法可以快速得到由最大频繁项集所生成的强关联规则,很可能会遗漏那些支持度不高却带有明显攻击行为的报警,但是针对 DDoS 攻击的检测来说,那些支持度不高的报警不具备 DDoS 洪泛性的特点,因此可以忽略。实验是在固定的场景中检测攻击的,具有一定的局限性,检测的手段也比较简单。在今后的工作中,将研究如何自动检测分析由多条报警组成的攻击场景。

参考文献:

[1] 卢建芝. 基于源端的防 DDoS 攻击的实现[J]. 计算机应用, 2004, 24(12):201 - 202.

[2] 胡小新. 一种 DDoS 攻击的防御方案[J]. 计算机工程与应用, 2004, 40(12):160 - 163.

[3] 杜彦辉. 分布式拒绝服务攻击的形式化描述[J]. 计算机应用研究, 2004, 21(3):214 - 216.

[4] 李德全, 苏璞睿, 冯登国. 用于 IP 追踪的包标记的注记[J]. 软件学报, 2004, 15(12):250 - 258.

[5] 罗淇方. 分布式拒绝服务 DDoS 攻击检测与防范方法研究[D]. 南宁: 广西大学, 2006.

[6] 杨长春, 倪彤光, 薛恒新. 一种基于数据挖掘的 DDoS 攻击入侵检测系统[J]. 计算机工程, 2007, 33(23):167 - 169.

[7] 沈鑫. 基于数据挖掘的 DDoS 攻击检测方法的研究与实现[D]. 郑州: 信息工程大学, 2006.

[8] 王台华, 万宇文, 郭帆, 等. 应用于入侵检测系统的报警关联的改进 Apriori 算法[J]. 计算机应用, 2010, 30(7):1785 - 1788.

[9] MIT Lincoln Labs. 1999 DARPA intrusion detection evaluation [EB/OL]. [2007 - 03 - 15]. http://www.ll.mit.edu/IST/ideval/index.html.

[10] SANDHU R, McLEAN J. Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln laboratory[J]. ACM Transactions on Information and System Security, 2000, 3(4): 262 - 294.