华中科技大学

硕士学位论文

计算机病毒特征码自动提取技术的研究

姓名：金雄斌

申请学位级别：硕士

专业：信息安全

指导教师：徐兰芳

2011-01-19

摘 要

随着信息化程度，Internet 开放性以及信息高速公路运营能力的提高，计算机病毒的传播能力和破坏能力也是以惊人的速度增长。主流计算机病毒检测方法由于特征码的提取需要一定的时间而具有防毒滞后性。特征码的自动提取必将是未来一段时间内反病毒工作的发展趋势。

针对 Honeycomb，Polygraph 等系统只能自动提取蠕虫病毒特征码的这个缺点，在充分分析研究其基本原理、设计特征等基础上，总结出了新的计算机病毒特征码自动提取算法的设计方法。基于这个设计方法以存储特点为突破口去设计一个新的基于存储特点的计算机病毒特征码自动提取算法。设计算法之前首先确定新算法的两个必要条件——计算机系统中的存储特点和计算机病毒存储特点。对这两个条件进行充分的分析与研究，最终完成算法设计。算法抓住病毒在感染目标文件时植入病毒代码存储上的特点，快速定位出被植入的病毒代码并从中提取出特征码。算法包含特征码设计，特征码自动提取和扫描引擎检测三大模块。算法采用可变长度的特征码集形式来增加扫描引擎的检查效率；特征码自动提取模块自动定位病毒代码的位置 ，然后以扇区为单位提取其中某些字节作为特征码；扫描引擎检测模块首先针对每个目标文件都进行一次准特征码提取操作，将获得到准特征码放到特征码库中去匹配，匹配成功后再进行二次准特征码提取和二次特征码匹配，最终确定目标文件是否被特征码库中的病毒感染过。

依据基于存储特点的计算机病毒特征码自动提取算法设计了一个计算机病毒特征码自动提取仿真系统并进行仿真测试。仿真测试结果表明了系统能实现计算机病毒特征码的自动提取工作，同时也能针对可以目标文件进行准特征码提取，然后与特征码库中的特征码匹配比较，从而具有计算机病毒检测的功能，同时也表明设计出的计算机病毒特征码自动提取算法是正确的，扫描检测效率也是非常高的。

关键字：计算机病毒，特征码，自动化提取，基于存储特点

Abstract

Along with the enhancement of the informationization level, Internet openness and information highway operation ability, the propagation ability and destructive power of computer virus are also growing at an alarming rate. The mainstream method of computer virus detection has some anti-virus lag because it needs time to extract signature from virus. So the automatic extraction of computer virus's signature will be the trend of the development of antivirus work in future period.

Conclude a method of designing a new computer virus signature automatic extraction algorithm after a fully analysis and research on the basic principle and design features of the Honeycomb System, Polygraph System and some others, because they only automatically extracting signature from worms. According to this method, design a new computer virus signature automatic extraction algorithm from the breakthrough - storage features. Define two requirements of the new algorithm – computer files' storage feature and computer virus's storage feature before designing the algorithm. And then analysis and research the two requirements and finish the design of the new algorithm. The new algorithm seizes the storage feature of the part of program, which is inserted into a normal file by the virus when the virus infects it. And then locate this part of program quickly and extract the signature from it. The new algorithm contains signature designing, signature extracting and signature detecting three models. The signature designing model uses the variable length signature set to improve the signature detecting efficiency; the signature expression model can locate the computer virus program's part in a infected file quickly, and extract some bytes binary code by a sector unit from it as its signature; the signature detecting model does signature extraction operation once on every object file to get many prospective signature sets, and the match them in signature library, if any signature set match successfully, then extracting signature set from the object file twice, and match it twice. At last determine the object file is infected or not.

Design a computer virus automatic extraction simulation system According to the computer virus automatic extraction algorithm based on storage features. The system can automatically extract the signature from computer virus, and also can does computer virus detection jobs by extracting quasi signature set from the object file, and then match it with every signature of the signature libratory. The results of the simulation system's test show that the computer virus automatic extraction algorithm is correct, and its efficiency of the detection scanning is very high.

Keywords：Computer Virus，Signature，Automated Extracting，Base On Storage Features

独创性声明

本人声明：所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本课题的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期：　　年　　月　　日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本论文属于

保密?，在_____年解密后适用本授权书。

不保密?。

（请在以上方框内打“√”）

学位论文作者签名：　　指导教师签名：

日期：　　年　　月　　日　　日期：　　年　　月　　日

1 绪 论

1.1 课题背景及研究意义

计算机技术的迅猛发展，给人们的工作和生活带来了前所未有的便利和高效率。随着计算机走进社会的各个领域，走进千家万户，计算机系统已能实现对工作、生活、管理、办公的自动化，成为社会不可缺少的一部分，人们对计算机的依赖程度越来越大。然而，计算机系统并不安全，人们在享受计算机带来的各种优越性的同时，也正面临着计算机系统由于各种恶意代码或者黑客攻击与侵害而无法正常工作的威胁。互联网的普及应用，更加剧了计算机病毒的泛滥。因此，计算机病毒的研究成为了信息安全领域的一个重要内容。

计算机病毒是一个广义的病毒概念，包括狭义的需要宿主的计算机病毒、无需宿主就可以独立存在的计算机蠕虫等各种能让计算机脱离正常运行控制，难以根治的恶意代码。计算机病毒种类很多，且以相当惊人的速度递增，另外计算机病毒编写也越来越容易，很多事例都表明：一名普通的网民通过简单的培训就可以写出破坏性相当大的计算机病毒程序来，如“熊猫烧香”病毒案件，这一切令人们谈病毒而色变。据金山软件公司发布的《2009年4月份中国电脑病毒疫情及互联网安全报告》显示，2009年4月份新增计算机病毒1,773,892个，病毒感染计算机数量为20,084,015台次。现在，各种计算机病毒的产生和全球性蔓延已经给计算机系统的安全造成了巨大的威胁和损害，其造成的计算机资源的损失和破坏，不但会造成资源和财富的巨大浪费，而且有可能造成社会性的灾难。美国一市场研究公司Radicati Group日前发表的一项调查报告表明，2003年计算机病毒造成的全球经济损失超过280亿美元，2008年则已超过850亿美元，预计2009年将超过1000亿美元，计算机病毒造成的经济损失几乎成线性增长。人们将计算机病毒称之为“21世纪最大的隐患”、“不流血的致命武器”。计算机病毒越来越能影响人类生活的发展，但是自从计算机病毒这个潘多拉的魔盒被打开后，计算机反病毒工作者们一直在努力地跟计算机病毒作斗争，从计算机反病毒技术的产生和发展的整个过程，可以看出计算机病毒技术的发展，也推动了反病毒技术的发展，新的反病毒技术的出现，又迫使计算机病毒再更新其技术。目前，计算机反病毒技术仍然滞后于病毒技术。因此，对计算机病毒检测技术

的研究是今后反病毒技术领域内长期研究和发展的热点和难点。所以反病毒工作者的目标就是尽最大的可能限制病毒及其发展，或者在病毒爆发的初期就能发现，从而做好防护措施，减少损失[1~2]。

计算机病毒最早起源于美国电话电报公司（AT&T）的贝尔（Bell）实验室中的一项游戏——磁芯大战（core war）。core war 体现出计算机病毒的一些基本特点，如自我复制、感染主机，甚至有了破坏功能。但由于游戏的参与者彼此约定不向公众透漏游戏的内容，所以这种游戏的危险性还没有表现出来。直到 1983 年，计算机病毒这个魔盒最终还是被打开，科恩·汤普逊在其获奖典礼上透漏计算机病毒的存在，此后"计算机病毒"一词诞生，同时计算机反病毒工作也应运而生。计算机病毒的查杀方法主要有以下几种[3]：

1. 特征码法。特征码法主要通过采集已知病毒的特征码来判断目标程序文件是否被病毒感染过，此方法是检测已知病毒的最为简单、有效、开销最小的方法。其特点是检测快速准确、能识别病毒的名称、误报率低等，并可根据检测结果，对被感染的文件做解毒处理。其不足是不能检测未知病毒（特征码库中不含该病毒的特征码）；提取已知病毒特征代码的费用开销大、效率低；在网络上效率低，即在网络服务器上，因长时间检索会使整个网络性能变坏，而且反病毒软件必须随着新病毒的不断出现而频繁更新版本和特征码库[4]。

2. 校验和法。校验和法是利用保存的原文件的校验和与目标被检测文件的校验和相比较来判断文件是否内容被修改，从而判断是否被病毒感染。此方法不仅能发现已知病毒，也能发现未知病毒，却不能识别病毒的名称，不能做解毒处理，主要用在病毒爆发的初期，病毒的特征码还没有被提取出来时的病毒检测[5]。

3. 行为监测法。行为监测法是利用病毒的特有的行为特征，如对 EXE 文件写操作、打开端口、删除系统服务等，来监测病毒的方法。此方法的特点是可发现已知病毒和未知病毒、可比较准确地预报多数未知的病毒。不足是可能误报警、不能识别病毒名称、实现时难度非常大[6]。

4. 软件虚拟法。软件虚拟法是用软件来模拟和分析程序的运行过程，根据病毒的行为特征来检测和识别病毒。此方法是检测多态病毒和未知病毒的最新、最有效的方法，但虚拟软件的实现目前还有很大难度[7]。

当前的计算机病毒检测技术主要是基于特征码检测法。其基本思想是从被感染的目标文件中提取病毒样本的特征码，并命名。然后将此特征码添加到扫描引擎的特

征码库中，在病毒检测时通过搜寻病毒特征码库查找是否存在相匹配的特征码来发现病毒。这种检测方法相对于其他几种方法优点多，实现容易。目前常见的计算机病毒防杀软件对已知计算机病毒的检测大多采用这种方法[8]。

基于特征码的计算机病毒的查杀方法是计算机反病毒最有力的武器，该方法基于一个前提，即必须掌握计算机病毒最合适的特征码。目前几乎所有反病毒引擎使用的病毒特征码库中的病毒特征码都是依靠人工手动从病毒中提取出来，因此，该方法有以下几点不足[9]：

1. 人员素质要求高。要想提取准确实用的特征码，必须有经验丰富的专业计算机病毒分析师经过分析病毒，并从中提取出一段二进制数据；

2. 花费时间较长。计算机病毒分析师需要时间分析病毒。另外，提取出来的特征码要经过定量的测试才能正式加入到特征码库中，否则会造成误报、错杀的可能。时间上的滞后性导致一种新病毒在爆发的初期不能被反病毒引擎及时检测出来，遏制其大规模爆发；

3. 人工提取出的特征码存在很大的随意性，反病毒引擎检测时只能采用效率低下的串匹配方式，虽然串匹配的研究工作较有进展，但是反病毒引擎检测效率以及资源消耗仍然是影响该方法发展的瓶颈。

经过对各种病毒特征码提取状况的调查和分析，发现人工分析和处理计算机病毒的工作效率很低，如一名熟练的病毒分析师按一天工作 8 小时计算，平均每天只能提取 12.8 个病毒的特征码。在病毒“井喷式”爆发的今天，一天就可能产生上百个病毒。12.8 个/天的处理速度显然是不够的，为此研究病毒特征码的自动化提取方法和技术显得犹为重要[10]。

1.2 国内外研究概况

在计算机反病毒工作的早期，基于特征码的计算机病毒检测技术就被提出。由于它误报率低，检测病毒非常有效而一直作为主流的查杀方法。但由于它的不足又催生了一系列替代它的方法，如行为监测法、软件虚拟法等。在计算机病毒特征码自动提取算法上的研究不多，也鲜有成果。

计算机病毒特征码自动提取算法（CVSAE，Computer Virus Signature Automatic Extraction）是在特征码提取的过程就引入自动化技术，代替计算机病毒分析师从被

感染了病毒的目标文件中，定位出病毒体，最终提取出一些数据按照设计好的特征码格式组织成该病毒的特征码，并且该特征码能够发挥类似人工提取出的特征码的实现病毒检测功能。另外算法还应包含与提取特征码相应的扫描引擎检测模块，该模块依据前一模块提取的特征码，完成计算机病毒扫描检测的工作。概算法不仅在应对新病毒时有很好的查杀作用，很高的扫描检测效率，同时还不用依赖反病毒厂商发布的特征码库，而自己完成特征码库的维护工作。

目前关于 CVSAE 算法研究较多的，有比较显著效果的大多数是针对蠕虫类病毒的。而针对于传统寄生型病毒几乎没有这方面的研究成果。

1.2.1 针对蠕虫病方面的研究

蠕虫病毒是一种新型的计算机病毒，蠕虫病的特点表现在以下两个方面：(1) 依赖于系统漏洞，通过网络发动攻击；(2) 可以独立存在不需要寄存在目标程序文件中。

Honeycomb 系统是蠕虫病毒特征码自动化提取领域的最早尝试，该系统是 Christian，Kreibich 在 2003 年设计出来用于蠕虫病毒的特征码自动化提取与防范，是自开源项目 honeypot 中 honeyd 的一种改进与扩展，在 honeyd 的基础上增加了插件技术和回调钩子（HOOK）。插件技术是为了能够独立 honeyd 而编写的插件程序；HOOK 是为了能够在收到或者发送数据包时通知插件程序处理数据。Honeycomb 系统有两大模块——数据流搜集模块和特征码提取模块，数据流搜集模块负责搜集足够多的数据流提供给特征码提取模块以提取特征码，特征码提取模块就是在搜集到足够多的数据流的基础上，从中提取出特征码。该系统可以看成是一个原型系统，在实际应用方面的价值不大[11~12]。

EarlyBird 系统是 Singh S 和 Estan C 在 2004 年针对 Honeycomb 系统设计出来的改进系统。该系统也是基于开源项目 honeypot 中的 honeyd。由于 Honeycomb 系统在特征码提取时没有给出详细的步骤、过程，EarlyBird 系统在这个方面进行了细化。系统的基本思想是：定义蠕虫病毒数据流中的一个定长字符串标识一个蠕虫病毒，通过检测可疑数据流中的该字符串的出现的次数来确定蠕虫病毒是否存在于网络[13]。

2004 年，Hyang-Ah Kim 和 Brad Karp 对 Honeycomb 系统进行改进，设计出了针对蠕虫病毒的 CVSAE 系统——Autograph 系统。，Autograph 系统改进突出在两方

面：(1) 在特征码提取时，重点关注可疑数据流中的字符串，为了更加准确地找到蠕虫病毒的特征字符串，采用了子串划分算法帮助找到蠕虫病毒的特征字符串作为特征码提取出来；(2) Autograph 系统是一个分布式系统，在网络环境下对蠕虫病毒的检测，分布式系统更具有优势 [14]。

2005 年，随着一大批变形病毒的出现，之前的 Honeycomb 系统，EarlyBird 系统和 Autograph 系统都已经不能发挥作用了，因为变形蠕虫病毒的编写者为了躲避上述三个系统的检测，在蠕虫病毒的数据流中故意掺杂了很多无效的随机内容使得蠕虫病毒的行为具有相当的隐蔽性。为此 James Newsome 设计了一个功能更为强大的 Polygraph 系统，其创新点就是能自动提取变形蠕虫的特征码。变形蠕虫病毒的出现使得传统的单一连续字符串无法作为特征码使用，这就要求对特征码的格式和特征码提取方式方法上有新的突破[15]。

1.2.2 针对寄生病毒方面的研究

Honeycomb 系统之后的连续几年内有多个系统被设计出来应对包括变形病毒在内的各种蠕虫病毒。在传统的寄生病毒的特征码自动化提取方面也有成果，2007 年，有学者在研究大量的 VB 病毒后设计了一种针对 VB 应用程序病毒的 CVSAE 算法。研究发现 VB 病毒在某个固定的位置含有该程序的编译信息，这部分信息对于某个 VB 程序而言是独一无二的，在此基础上经过大量的实验验证，最后提出了一个针对 VB 病毒的 CVSAE 算法 [10]。

1.3 课题任务及论文结构

由于其他几种反病毒方法存在各种自身致命缺点，基于特征码的反病毒方法是目前各大反病毒厂商最主要采用的一种方法。本文致力于 CVSAE 算法的研究，主要研究计算机病毒特征码的提取问题及与之关联甚大的特征码匹配。目前国内外关于 CVSAE 算法的研究不是很多，研究内容局限于基于某一种病毒或者某一个分支，不具有普遍性。本文将根据计算机存储特点去研究一种计算机病毒特征码提取的新方法。全文重点解决三个问题：

1. 深入剖析 window 系统下计算机病毒的运行机理和特点，以及目前基于特征

码的计算机病毒检测方法中特征码提取的详细过程，并对其优缺点进行分析，从中探索一个新 CVSAE 算法；

2. 总结目前国内外已有的 CVSAE 算法，分析比较它们的特点和不足，从中寻找改进方法或者新的提取方法；

3. 总结 window 下文件的存储特点，针对病毒在硬盘中存储特点寻找计算机反病毒的突破口。

本文的篇章结构为：

第一章绪论主要系统性的叙述课题背景，选题意义和课题任务以及与课题研究内容相关的国内外研究简况。

第二章，将详细分析基于特征码的计算机病毒检测技术和针对蠕虫病毒等的 CVSAE 系统，着重分析计算机病毒特征码提取的原则和操作步骤，分析针对蠕虫病毒和 VB 病毒的 CVSAE 系统、算法的工作原理、设计特点。进一步分析研究 CVSAE 算法的可行性，尝试寻找计算机病毒自动化提取方法。

第三章，将首先详细分析总结国内外现有的 CVSAE 系统、算法的特点与不足，从中总结出设计 CVSAE 算法的方法。其次根据设计方法确定设计新的基于存储特点 CVSAE 算法。在设计新算法之前确定其两个必要条件。

第四章，根据第四章的算法设计一个 CVSAE 系统，该系统包括特征码提取模块和扫描引擎检测模块。另外在特征码设计上也有创新，采用特征码集的形式。

第五章，主要进行新的 CVSAE 系统测试与测试结果的分析。

第六章，总结研究结果。包括工作总结和工作展望两个部分。

2 基于特征码的计算机病毒检测技术

自1987年10月第一例计算机病毒Brain诞生以来，计算机反病毒工作也随之开始了，发展到今天计算机反病毒工作取得了显著的成效。虽然没有完全灭绝所有的计算机病毒，但是能够遏制住其疯狂的发展势头。计算机反病毒技术有，特征码检测、行为检测、软件虚拟等，其中被各大反病毒厂商主要使用的是基于特征码法[16]。

2.1 基于特征码的计算机病毒检测技术

2.1.1 计算机病毒特征码

计算机病毒感染计算机系统中的普通程序时，会将自身的代码片段分成几段或者直接插入到普通程序，即宿主程序中。但是同一个病毒或者同一种病毒在感染普通程序的方式是一样的，计算机病毒会以固定的方式将自身代码片段插到宿主程序中，这给发现病毒后再清除病毒留下可能，但给病毒检测带来了麻烦。计算机病毒的检测需要在宿主程序中找到存在病毒代码的依据，从而报告发现了计算机病毒。严格来说只有在宿主程序中检测到了某个病毒的完整代码才能很确定的说发现了计算机病毒。然而这样的工作量是相当大的，检测效率很低。后来发展成以计算机病毒的“指纹”来代替计算机病毒的全部代码，大大缩小工作量，提高效率。这就是现在用于计算机反病毒重要的手段——基于特征码的计算机病毒检测技术。这个“指纹”就是计算机病毒的特征码[22]。

计算机病毒的特征码要能够把计算机病毒程序与其他一般程序区分开来，即能够唯一标识一个或者一类多态病毒，能够成为计算机病毒程序的“指纹”、“DNA”。“指纹”的提取是非常重要的。计算机病毒大小不一，差别也很大，从短小的只有一百多个字节，到长的有上百KB 字节。如果只是随意从病毒体内选取一段二进制数据串作为代表该病毒的特征码，很可能在不同的环境中，该二进制数据串因为有了变化，不能将被该病毒感染的文件检测出来而造成漏报，或者因为不真正具有代表性在很多普通没有被感染的程序文件中也能匹配成功，造成很多的误报。选二进制数据串作为病毒特征码显然不合适。计算机病毒的特征码需要满足以下几个条件：

1. 不能取自于数据区里的数据，因为数据区里的内容很容易改变；

2. 在保持唯一性的前提条件下，尽量使特征码短小，减少检测过程中时间和空间上不必要的开销；

3.必须是经过仔细分析病毒程序之后才选出最具代表性的特征码，才足以将该病毒区别于其他病毒以及该病毒的其他变种的二进制数据串；

4. 特征码必须能将病毒与普通非病毒程序区分开。不然将非病毒程序当成病毒报告，这就是误报。如果这种误报太多了就会使用户放松警惕，如果真的病毒一来，破坏就会很严重。另外，如果将这引起误报送给病毒清除程序，则会带来更为严重的后果——杀毒程序会将好程序变成坏程序而无法使用；

5. 长度一般不宜超过64个字节。

选定好的特征码是很不容易的，这也是病毒扫描程序的基础所在，目前提取特征码都是依靠有经验的计算机病毒分析师人工分析，提取出来。一般情况下特征码是连续的若干个字节组成的串，但有时候却需要是一个可变的长串，即在特征码中包含一个到几个"模糊"字节。需要使用通配符来表示某一个特征码。检测程序遇到这种特征码时，只要除"模糊"字节之外的字节串都能完全匹配，也能判别出病毒。如若给定特征码为"E9 7C 0F 10 ?? 37 EB"，则"E9 7C 0F 10 B7 37 EB"和"E9 7C 0F 10 93 37 EB"都能被识别出来。

计算机病毒程序一般都有明显的特征码，经过有经验的计算机病毒分析师的分析，还是可以提取出有效的特征码的。特征码可能是感染标记，也可能是若干计算机指令组成的一段计算机程序。可从下面几种角度来具体提取计算机病毒的特征码[8]：

1. 当计算机病毒发作时，把病毒在计算机屏幕上出现的提示信息作为病毒的特征码，如大麻病毒会提示："Your PC is now stoned"等。另外其他一些很独特的信息也可以用于特征码中，如爱虫病毒，它是以主题为"I LOVE YOU"的邮件形式存在于outlook 等邮件客户端软件中。这种方式广泛存在于早期的计算机病毒中，当时计算机病毒编写者大多数都是以炫耀计算机技术为目的，病毒发作时会清楚提示你。现在的计算机病毒编写者都以窃取用户数据，控制主机来获取利益等为目的，病毒都会采取各种隐藏技术、措施来隐藏自身。

2. 利用计算机病毒的感染标志。很多计算机病毒为提高传播效率而使用感染标志。计算机病毒的编写者为了达到区分其他程序与自己的病毒的目的，感染标志选

取也具有相当的特殊性，这在一定程度上恰好满足特征码的条件。即可以使用病毒感染标志作为病毒的特征码值。如熊猫烧香病毒的“WHBoy”等等。

3. 当前两种方法无法提取出特征码时，就只有使用第三种。从计算机病毒代码段的任何非起始地方开始取出的连续的、不超过 64 字节且不含空格字符的一串字节都可以作为计算机病毒的特征码。例如 CIH 病毒的特征码为“55 8D 44 24 F8 33 DB 64 87 03”，如图 2.1 所示。

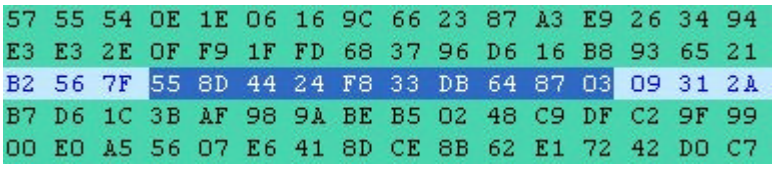


图 2.1 CIH 的特征码

从理论上讲，随机从病毒的代码段中取出的连续 64 字节内容，则可分辨的病毒数可达 2^{8*64} 种。为了提高扫描检测效率，特征码长度一般取 7~15 字节之间。但在实际上，由于病毒种类个数繁多，且病毒制造者有可能故意针对一定的病毒特征值修改病毒，另外病毒中的一些基本指令的也有可能存在于普通的程序代码中。因此，连续的 64 字节不一定就能完全区分两种不同的病毒，也不一定能区分出被病毒感染过的程序与普通程序。所以病毒的特征码提取出之后，还得经过定量的测试验证。验证通过之后方可作为计算机病毒的特征码，放入特征码库中。

根据计算机病毒特征码所需完成作为计算机病毒的“指纹”的使命，分析了其应该满足的条件，也从有经验的计算机病毒分析师那里得到了提取计算机病毒特征码的方法步骤。总体来说，计算机病毒的特征码跟计算机病毒功能结构关系很大，大多数情况下都是基于计算机病毒分析师对病毒功能分析结果，如计算机病毒发作特征，感染特点等信息。如果要实现计算机病毒特征码的自动化提取，应该从两个方向着手，一是采用智能算法，该算法在一定程度上担任计算机病毒分析师的角色，能够分析出病毒的发作特征或者感染标志等特点，另一个就是让计算机病毒特征码能够尽量脱离其功能性的束缚，即特征码不与病毒的功能结构有关系。

2.1.2 基于特征码的计算机病毒检测方法

虽然基于特征码的计算机病毒检测方法总体来说效率不是很高，消耗资源也较多，但是相对于其他几种计算机病毒检测方法而言，它在检测效率、消耗资源、误报率和可行性等方面还是有决定性的优势。基于特征码的计算机病毒检测方法实现步骤如下[8]：

1. 准备工作。获得各种计算机病毒的特征码，丰富特征码库。

(1) 采集已知计算机病毒样本。一种病毒如果既能感染 COM 文件，又能感染 EXE 文件和引导区，就要同时采集 COM 型、EXE 型和引导区域的三种病毒样本。

(2) 在每个病毒样本中，抽取特征码。在提取既能感染 COM 文件又能感染 EXE 文件的病毒的特征码时，要尽量从两种样本共有的代码中提取。

(3) 将特征码以设计好的格式放到病毒特征码库。由于计算机病毒很多，提取出的特征码也少不了，因此特征码库就会是一个很大的数据文件，在检测病毒时需要反复多次访问，所以良好，高效的特征码库的数据结构设计也是一个关键。

2. 扫描检测病毒。扫描引擎会逐个打开计算机系统中的目标文件，将病毒特征码库中的病毒特征码逐个与之匹配。如果匹配成功，由于特征码与病毒一一对应，便可以断定被查文件中患有何种病毒，报告检测到病毒的名称等信息，然后匹配下一个特征码或者关闭当面目标文件打开下一个目标文件，重新逐个匹配特征码库的每一个特征码。

在扫描引擎方面方面，虽然它的改进发展很快，但是它们都是应着特征码在提取时的侧重点不同而发展改变。而特征码提取的侧重点还是依赖于某个计算机病毒或者说某一种变形计算机病毒的一方面的特性，而不具有普遍性。总体来说扫描引擎的改进还是依赖于特征码提取方式方法上的改进，一定程度上来说，良好的扫描引擎的设计又能影响特征码的提取思想。

2.2 针对蠕虫病毒的 CVSAE 系统

蠕虫病毒是计算机病毒具有其独特的特点——爆发时依赖网络。针对蠕虫病毒特点的 CVSAE 系统可实现蠕虫病毒特征码的自动提取以及对蠕虫病毒的检测防范[17-19]。

2.2.1 Honeycomb 系统

Honeycomb 系统有两大模块——数据流搜集模块和特征码提取模块。

数据流搜集模块使用 Honeypot 捕获那些尝试连接某些指定 IP 的可疑连接请求，并可以保存一定数量的 TCP 或 UDP 的会话状态。搜集到的数据流越多就越有利于提取出有效的特征码，为了防止连接请求过多而造成系统资源耗尽，需要设定一个阈值，以限制最多监测多少个连接请求。数据搜集模块搜集可疑连接上的所有数据包，系统都会进行流重组，获得一系列可疑数据流。

特征码提取模块在可疑数据流中提取特征码。其工作流程分为三步：第一步，数据流中网络层和运输层的检查。如 IP 检查，TCP 标志位检查等。如果发现异常信息则提取出来；第二步，对具有相同目标端口的连接上的数据流进行模式字符串提取。系统使用 LCS 算法，对所有可疑数据流进行最大相同字串匹配，长度超过设定值的字串作为入侵特征输出。第三步，将前两步提取出的特征码组成一个特征码池，也就是一个有长度有限的队列，定期对特征码池进行整理并以伪代码的形式输出 Bro 规则或者 Snort 规则 [20-22]。

2.2.2 EarlyBird 系统

EarlyBird 系统的核心是一个侧重于线速实时的提取特征码算法，可以看做是 honeycomb 系统的改进。EarlyBird 认为一个一定长度的字符串足以标识一种蠕虫病毒。当蠕虫病毒爆发时这个字符串将会在某台主机对外的数据流中大量出现，同理当在某台主机的网络数据流中发现了这个字符串，而且出现的次数达也到了某一阈值，也就可以断定这台主机必定感染了这种病毒[13,23]。

EarlyBird 中使用了两个表结构——高频字符串表和地址分布表来检测当前网络中是否存在蠕虫病毒活动的情况，并在检测到蠕虫病毒的同时就可以提取出其特征码。高频字符串表中记录某网络的数据流中出现的字符串的频率高低程度，共有 4 个属性项，每个字符串有 4 个属性组成，当某字符串的 4 个属性项中的计数均超过阈值时，在地址分布表中建立一条记录，表示含有该字符串的数据包的源 IP 和目标 IP 的分布情况。捕获的数据包经过系统整理成一条数据流，逐字节滑动，连同本数据包使用的协议、端口信息，即头部字段，一起作为 Hash 函数的输入，计算字符串

的 Hash 值，插入到高频字符串表中，如果表中存在则其对应的次数自增 1，同时更新地址分布表。如果两个表中的次数均超过阈值，则输出该字符串为特征码。在高频字符串表和地址分布表进行插入或者查找操作时都使用 Rabin fingerprint 算法作为 Hash 函数。EarlyBird 系统的工作流程如图 2.2 所示[24-25]。

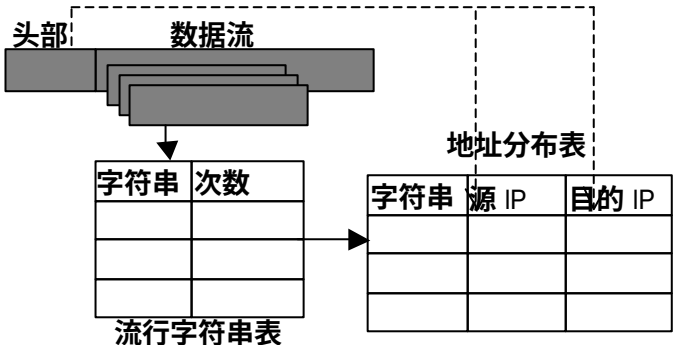


图 2.2 EarlyBird 系统流程

2.2.3 Autograph 系统

Autograph 系统也可以看做是 Honeycomb 系统的改进，也分为两大模块——可疑数据流选择模块和基于内容的特征码提取模块。其系统的输入为通过网络边缘隔离区的所有网络数据包，输出为蠕虫病毒的特征码列表[26]。

可疑数据流选择模块主要负责从捕获的 TCP 数据包中选择符合条件的数据包，将这些数据包存储下来并定期重组数据流，重组后数据流输出供特征码提取模块操作使用。Autograph 系统认为蠕虫病毒大多数情况下会通过大规模扫描 IP 地址来寻找可攻击对象的，如果网络中某个外部 IP 对 Autograph 系统监控的多个内部 IP 都尝试 TCP 连接请求超过 s 次，且连接都没有成功，则可认为外部 IP 是一个传染源，有必要也必须对其发送给内部网络的所有数据包进行搜集、分析、监控并保存[27]。

基于内容的特征码提取模块的工作是将前面搜集的数据流进行特征码提取。Autograph 系统调用 COPP 算法对可疑数据流进行子串划分，然后分别统计各个子串出现的次数、频率，接着根据频率从高到低排列这些子串进行组成高频字符串集，然后继续按频率从高到低从高频字符串集中删除字符串，并将包含该字符串的数据流从可疑数据流中移除，多次重复这个过程，当被移除数据流的数量占所有数据流

总数的比率到达某个设定的阈值时，特征码提取完毕[28-29]。

2.2.4 Polygraph 系统

Polygraph 系统的特点在于可以自动提取变形蠕虫的特征码。变形蠕虫病毒的出现使得传统的单个连续字符串无法作为特征码使用，这就要求对特征码的格式和特征码提取方式方法上有新的突破。Polygraph 系统首先就需要解决变形病毒的特征码问题[15]。

变形蠕虫病毒会故意在 TCP 数据流中增加随机无效的字符阻挠系统通过统计各个子串的出现频率来自动提取特征码。Polygraph 系统中给出了三种类型特征码来应对变形蠕虫病毒。即：关联字符串组、有序字符串组和带贝叶斯权值的字符串组[30]。

关联字符串组是由多个字符串组成的集合，每个字符串组标识一种蠕虫病毒。由于变形蠕虫会在自身的数据流中故意插入无效信息，原来可以提取出的完整特征码被拆成特征码碎片，需要收集这些碎片组成一个字符串集合才能识别某个蠕虫病毒。有序字符串组与关联字符串组一样，也是由多个字符串组成的集合，但区别在于，该集合里的字符串间有严格的先后顺序。带贝叶斯权值的字符串组也是一个多字符串集合。集合中每个元素，也就是每个字符串都被赋予一个权值。当进行可疑数据流监控时，如果网络数据流中出现字符串的权值之和大于某一阈值，就可以报告检测到蠕虫病毒。

Polygraph 系统分为两大模块——数据流搜集模块和特征码提取模块。其系统框架结构如图 2.3 所示。

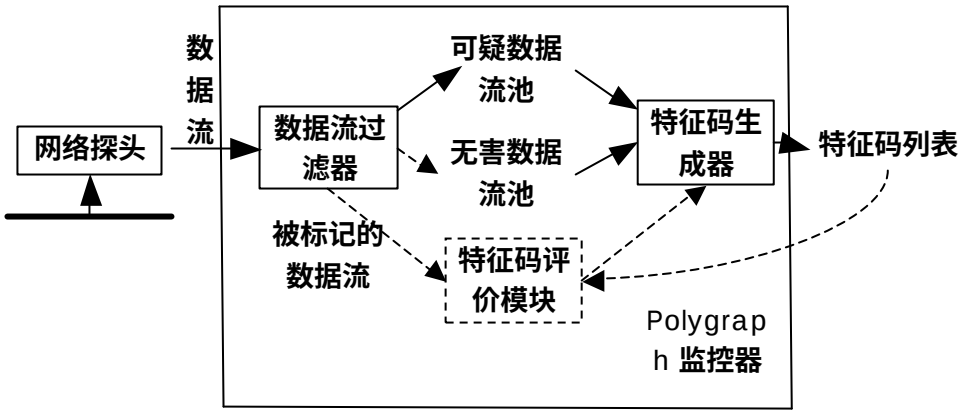


图 2.3 Polygraph 系统的框架结构

Polygraph 系统通过网络探头捕获网络中所有的数据包，经过系统重组后得到多个数据流。数据流过滤器只能将可疑数据流从所有数据流分离出来，在实际应用中过滤器提取出来的可疑数据流中可能还会包含少量的正常数据流，可疑数据流池中也会存在多个不同类蠕虫病毒产生的数据流量的可能。为了提高特征码的质量，Polygraph 系统在对可疑数据流池进行提取特征码操作前，首先对可疑数据流池中的可疑数据流进行一项聚类分析。能够把不同的蠕虫产生的流量聚类到不同的集合中，然后对这些集合逐个进行特征码提取。在 Polygraph 系统中，使用层次式聚类算法来实现可疑数据流分类 [31]。

Polygraph 系统给出三种类型的变形蠕虫病毒特征码的提取过程都可以概括为以下两步：

第一步，使用在 Smith-Waterman 算法基础上改进过的 L.Hui 算法，将经过层次聚类算法处理可疑数据流，获得很多子字符串，字符串依据出现频率可以删除掉被掺杂进去的随机数据获得特征码碎片。第二步，依据三种类型特征码的要求，对这些特征码碎片再次处理，就能获得对应类型的特征码[32,33]。

2.3 针对 VB 病毒的 CVSAE 算法

2007 年，有学者对 VB 病毒作了充分的研究，发现 VB 程序代码中编译信息、项目信息和窗口信息位置比较固定，并且反映了程序实现的功能。这部分信息是可以作为特征码的一部分。在此基础上，他们提出了一种针对 VB 病毒的 CVSAE 算法[10,34-36]。

算法就是针对使用 VB 语言编译生成的动态链接库或者可执行程序。算法从固定的位置提取出程序入口点后的编译信息、项目信息和窗口信息，特别是启动窗口的信息，作为特征码的初步数据。 算法思想是：从程序入口点后第 19 个字节开始取出 7 个字节，再取出包括编译文件相关信息的 64 个字节，然后利用二段校验和的方法，得到病毒特征码。由于动态链接库文件和可执行文件都是 PE 格式的文件，虽然格式有差异，但是该算法都可以处理[37-38]。VB 病毒扫描检测的过程是首先依据特征码的步骤提取目标文件的准特征码，然后将准特征码依次与特征码库中的特征码逐个匹配。

2.4 小结

本章主要叙述了传统的基于特征码的计算机病毒检测方法，针对蠕虫病毒的CVSAE系统和针对VB病毒的CVSAE算法的工作原理，详细过程等内容。

基于特征码的计算机病毒检测方法作为使用最为广泛的计算机病毒检测方法，它的优势是依赖于准确合适的特征码，同时它的扫描检测效率方面的弊端也日益凸现。

针对蠕虫病的四个CVSAE系统，在结构上都很相似。它们抓住了蠕虫病毒发作时的行为特点，从其入侵系统的数据流中自动提取出对应的特征码来。这个与传统的基于特征码的病毒检测方式或者理念有很大的不同，正是如此能够拓宽计算机病毒特征码的提取思路。

针对VB病毒的CVSAE算法虽然还是停留在传统意义上的特征码提取，严格来说它也只是针对蠕虫病毒起作用，而且还只是使用VB编译器生成的VB病毒。

综上所述，CVSAE算法的研究还是应该从特征码提取上入手，然后再设计与之匹配的扫描引擎。

3 新 CVSAE 算法设计

Honeycomb 系统、EarlyBird 系统、Autograph 系统、Polygraph 系统等较有实用价值的蠕虫病毒特征码自动化提取系统以及 VB 病毒特征码自动化提取算法，这些特征码自动化提取系统、算法各自具有很大的局限性。本章将研究 CVSAE 算法的设计方法，并依据该方法设计一个新的更具有通用性的基于存储特点的计算机病毒的 CVSAE 算法。

3.1 现有系统特点分析

传统的基于特征码的计算机病毒检测方法，针对蠕虫病毒的四个 CVASE 系统以及针对 VB 病的 CVSAE 算法，在各自领域能够担负起对应的反病毒任务。虽然不能针对所有病毒具有查杀防范作用，但是现有系统的思想还是有助于更具有通用性的 CVSAE 算法的设计。

3.1.1 传统基于特征码检测方法分析

传统基于特征码的病毒检测法是被使用最为广泛的计算机反病毒技术，使用时间最长，其优缺点也是最容易暴露出来。它的特点是查杀病毒准确率高、误报率低、实现简单。缺点的关键就是在特征码。特征码提取困难，而且需要花费时间，不能在病毒爆发初期就遏制住病毒，导致不必要的损失。特征码是人工手动从病毒体重提取出来的，具有很强的随意性，所以在检测时只有逐字节匹配。传统的扫描引擎采用的是字符串匹配的思想，字符串匹配算法的最好的复杂度在 $O(n+m)$，其中 n 表示目标字符串的长度，即目标程序的长度，m 表示匹配字符串的长度，即特征码的长度。总体来说效率较低。

另外，传统基于特征码检测方法还使得用户对于反病毒厂商有很强的依赖性，用户必须经常从反病毒厂商那里更新病毒特征码库，否则不能查杀出近期才发现的病毒。

3.1.2 针对于蠕虫病毒的 CVSAE 系统特点分析

Honeycomb 系统可以说是开创了计算机病毒，严格来说是针对蠕虫病毒的 CVSAE 算法的先河，从它的设计初衷，设计思想等可以看出，Honeycomb 系统最重要的贡献还是思想上的突破，而并不具有非常大实用价值。Honeycomb 系统的一个特点，就是从开源项目 Honeypot 中学习网络数据流监控的思想，然后利用蠕虫病毒在入侵系统的行为上寻找提取特征码的机会，从而达到检测蠕虫病毒的目的。这一技术相对于之在计算机病毒文件自身代码上提取特征码的技术是一个较大的突破，给计算机反病毒工作提出了一个新的借鉴和思路。

EarlyBird 系统的原型来源于 Honeypot，相比 Honeycomb 系统，EarlyBird 在特征码提取方面作了更为详尽、细致的改进而更具有实用性。在特征码提取方面，EarlyBird 关注数据流中的高频字符串。另外它还突出了大网络流量、高效高速、实时监控的特点，并在算法上做了很多优化工作，如使用高效快速的 Hash 函数用于表的查询等。在提取特征码整个框架上，不但兼顾考虑字符串出现的频率，而且也考虑包含该字符串的源目的 IP 的分布情况，进一步提高了特征码的准确性，可靠程度。

Autograph 系统的一个重要特点是分布式提取特征码，单个 Autograph 系统在利用 TCP 连接不成功次数来发现可疑数据流时灵敏度不高。如果采取分布式部署，各个监控点间可疑相互通报各自监控结果，共享数据信息，就可以加速发现可疑源 IP 的进度。同时当一个 Autograph 提取特征码完毕之后，也可以与其他各监控点共享此特征码，加快对新的蠕虫病毒进行封堵的速度。在特征码提取方面，Autograph 系统与 EarlyBird 系统相似，也是关注可疑数据流中高频字符串上，区别是 Autograph 系统使用 COPP 算法对可疑数据流进行子串划分，然后再进行各子字符串的频率统计，从而更加准确、高效找出高频字符串。另外 Autograph 系统还是一个分布式系统，在可疑数据流快速发现，以及在提取出特征码后对蠕虫病毒的快速响应，积极封堵上也有较大的改进。

Polygraph 系统是针对变形蠕虫病毒特征码提取而加以改进。变形蠕虫独特的隐藏技术可以轻易绕过 EarlyBird 系统和 Autograph 系统，因为它在自身数据流中插入大量无效随机数据，从而使得 EarlyBird 系统和 Autograph 系统在提取特征码时有相当大的困难，以致检测成功率很低。Polygraph 系统改进的方向有两个：一是可疑数据流聚类分离，将不同蠕虫病毒生成的可疑数据流分离开分别提取特征码，这样提

取出的特征码更加准确、有效；二是使用改进的 Smith-Waterman 算法从多个数据流从提取出相同高频的字符串，忽略蠕虫病毒故意插入的无效字符串。

3.1.3 针对 VB 病毒的 CVSAE 算法特点分析

针对 VB 应用程序病毒的 CVSAE 算法在检测 VB 病毒方面非常实用，检测时只需在固定的位置提取特征码信息然后加以匹配，检测效率相对于其他检测方法有非常大的提高。算法的不足是无法识别寄生型病毒，因为寄生型病毒往往把代码段插入到宿主程序中，不会将自己的项目信息、编译信息等特征码也带入宿主程序中，即使带入了宿主程序中，也很可能不在某一固定的位置。该算法的特征码提取思想是针对计算机病毒文件自身。虽然在计算机反病毒工作方面有一定的局限性，但 CVSAE 算法方面是一次全新的尝试。

3.2 CVSAE 算法的设计方法

Honeycomb 系统、EarlyBird 系统等是针对蠕虫病毒的特征码提取系统，这些系统都是在蠕虫病毒进行系统入侵时，通过提取蠕虫病毒入侵系统的数据流上的特征信息，从而判定蠕虫病毒的爆发，继而阻挡蠕虫病毒对系统的入侵。这与传统的计算机病毒检测理念不一样，传统的计算机病毒检测主要是通过扫描本地的计算机文件，判定该文件是否为病毒文件或者是否被病毒感染过。这些系统不但拓宽了计算机反病毒工作的思路，且其中自动提取蠕虫病毒特征码的方法还是值得借鉴的。

在前面第二章曾有一个小结论：要想实现特征码的自动化提取，就必须使特征码脱离计算机病毒功能结构的束缚，即自动化提取出的特征码不能要求其表示某个特殊的角色，如感染标志等。因为在自动化提取特征码的过程中，病毒的功能对算法是透明的。这一点在蠕虫病毒特征码自动化提取的四个系统中都有体现，而且越来越明显。由最开始的字符串到高频子字符串就能说明这一点，即特征码的粒度越来越小，特征码所表示的功能就越来越小。蠕虫病毒特征码自动提取是基于蠕虫病毒爆发时入侵系统的数据流，但是由其变化发展的过程可以得到关于设计 CVSAE 算法方面的启示：

1. 搜集数据流时应该做到不同的病毒有不同的样本，防止样本中有多种病毒的

存在。在计算机病毒大爆发的今天，同一个文件被感染了多种计算机病毒也是很有可能的，在做特征码提取时，样本应当选择准确，最好是只被一个病毒感染，甚至是只被感染一次。这对于能够自动提取出准确合适的特征码是非常关键的。

2. 特征码的形式有所改进，传统的单一字符串形式虽然简单，但是要求很高。特征码应朝特征码集形式改进，特征码集就是说特征码应该是多个短小字符串的集合，而且在特征码匹配时，工作量不需要重复。例如在 Polygraph 系统中有有序字符串组型的特征码，匹配到第一个字符串后，只需要接着继续匹配第二个，而无需从头开始匹配第二个，以此类推。

3. 寻找合理，合适的阈值。在蠕虫特征码自动化提取的过程中，常常涉及到阈值数据。阈值选取是一个关键，阈值太小，很容易将正常的数据流（或者程序文件）当成计算机病毒报告出来，这样就会有很大的误报率；阈值太大，就会在可以确定某数据流（或者程序文件）是计算机病毒的情况下还继续分析、确认，会导致检测效率低下。如果能够将阈值与某项参数关联起来，让其随着扫描进程的进行而动态改变将是理想的结果。

针对 VB 病毒的 CVSAE 算法在实际应用上有一定的局限性，算法提取的特征码与计算机病毒功能特征有很大的关联。前面曾有一个小结论——特征码自动化提取时，特征码应该不具有一定的结构功能特性。这两者间似乎存在一定的矛盾。其实，针对 VB 病毒的 CVSAE 算法在检测 VB 病毒上的实用性可以说明，特征码可以具有某一个大类所具有的公共功能特性。针对 VB 病毒的 CVSAE 算法提取出的特征码是与 VB 这一类应用程序公共的特性相关，就是在入口地址后的固定地方含有编译信息、项目信息等。要寻找适用于所有的计算机病毒的 CVSAE 算法，可以考虑所有计算机病毒的公共特性，而不是某个病毒所有的特性，如感染标志等。针对 VB 病毒的 CVSAE 算法给出的启示是：CVSAE 算法中特征码应该是所有计算机病毒所共有的特性，至少是某一类病毒的公共特性，而不是某个病毒所具有的特性。

3.3 新算法的两个必要条件

大多数计算机病毒是作为一个文件存储在计算机系统中，其存储特点涉及到计算机系统文件在硬盘上的存储特点。计算机病毒检测时是逐个扫描计算机系统中的文件。前一节总结的设计 CVSAE 算法表明，特征码取自于病毒的公共特点，这是设计

CVSAE 算法的关键。计算机病毒是作为一个计算机系统文件存储在硬盘上的，因此计算病毒的存储特点就是计算机病毒的共同特点。从存储特点这个角度就可以去设计一个新的 CVSAE 算法。而且更具有通用性。而研究计算机系统文件的存储特点和计算机病毒的存储特点就是研究基于存储特点的 CVSAE 算法的两个必要先决条件。

3.3.1 计算机文件存储特点分析

计算机系统中所有的文件都是存储在硬盘上，包括各种计算机病毒。文件都是以某种格式存放，由文件管理系统负责将各种文件以二进制的形式存放在硬盘中的某个磁片、某个扇区、某个磁道上。windows 系统的文件系统有 FAT16、FAT32、NTFS 和 NTFS5 四种，都以链式存储，即一个文件可能在硬盘上不是连续的，由多个最小存储单元链接而成。这样就能方便文件的增加、删减。本文将以 FAT32 为例分析计算机文件存储特点[39]。

1. MBR 扇区

在按下电源按钮之后，计算机会开始执行主板上的 BIOS 程序，BIOS 程序的功能主要有常用硬件（内存、鼠标等）检测和启动硬盘的配置。BIOS 程序执行完之前，BIOS 程序会把启动盘（前面配置过）上的第一个扇区加载到指定内存中，此时计算机系统自动跳到那块内存上去执行上面的程序。这样 BIOS 程序就把系统的控制权交给了第一个扇区。这个扇区就是 MBR（Master Boot Record）扇区，有时也称作主引导扇区。扇区是硬盘的最小存储单位，一般容量为 512 字节。如果 BIOS 程序执行完后就会跳到 MBR 扇区的开头开始执行，所以 MBR 必须是一段程序代码。在 512 字节的 MBR 扇区中，MBR 的引导程序占了其中的前 446 字节（偏移在 0000H-01BDH），在它后面有 64 字节的 DPT（硬盘分区表，Disk Partition Table），最后还有两字节“55 AA”是分区有效地结束标志。

MBR 不随操作系统的不同而不同的，即不同的操作系统可能会存在相同的 MBR，即使不同也不会夹带操作系统的性质。具有公共引导的特性。首先来分析一段 MBR，使用 winhex 查看一块 80G 的硬盘的 MBR 扇区如图 3.1 所示。

其中前面大方框中的就是 MBR 引导代码，后面小方框中的就是 DPT 分区表。为

了方便管理，硬盘都支持分区管理也就是常说的 C 盘、D 盘等。在 DPT 的 64 字节中，以 16 字节为分区表单位来描述一个分区属性。第一个分区表项描述一个分区属性，一般为基本分区。第二个分区表项描述除基本分区外的其余空间，称作扩展分区。图 3.1 中所示的第一个分区属性为“80 01 01 00 0C FE FF FF 3F 00 00 00 FC 8A 38 01”，其中各字段的含义如表 3.1 所示，详见附录 A。

表 3.1 中字节位移是指相对 MBR 扇区而言，即该字段在 MBR 扇区的位置。引导指示符中 0x80 表示活动分区，0x00 表示非活动分区，其他值对 windows 系

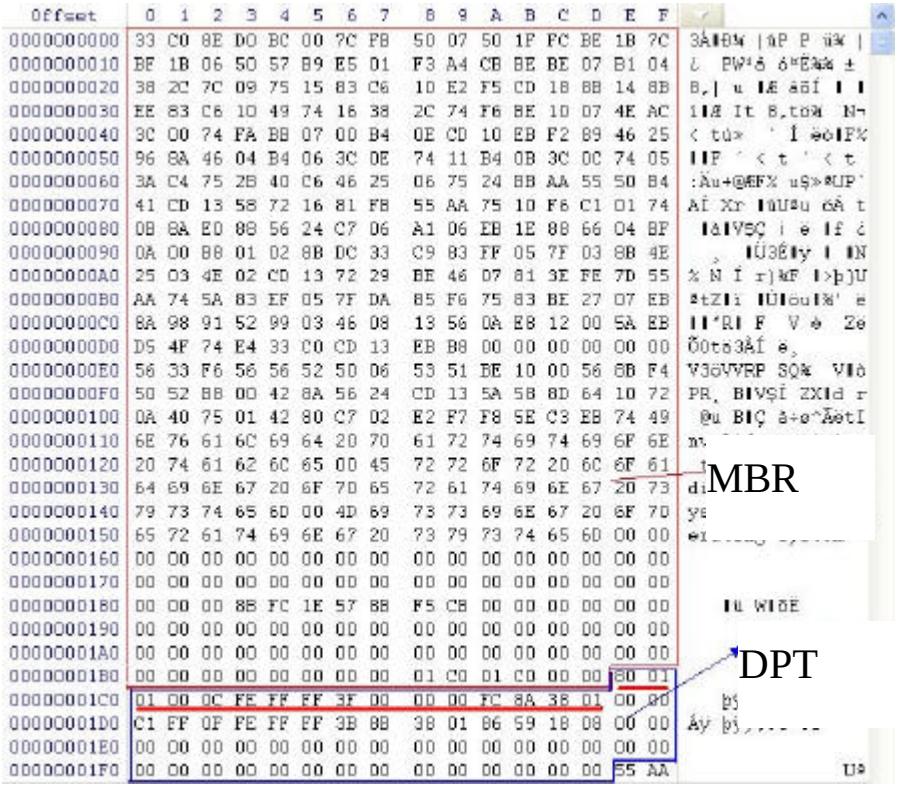


图 3.1 FAT32 系统下一个 MBR 扇区

统而言都是非法的。为了访问方便，系统不允许跨柱面分区，所以分区的粒度都是柱面，这样开始柱面都是 0x00。相对扇区数是指在该分区前面的分区数，也就是该分区的起始扇区号。总的扇区数表示该分区一共有多少个扇区，总的扇区数乘以扇区大小 512 字节将得到该分区的空间。

2. FAT32 分区

MBR 扇区中仅用 MBR 引导代码以及硬盘的分区信息，而此处的分区信息仅仅是主分区和逻辑分区的信息，并没有细分到硬盘上的各个小分区（如 C 盘，D 盘）的信息。下面通过图 3.2 来了解下硬盘的 FAT32 分区结构。

从图 3.2 可以看出，硬盘的第一扇区是 MBR 扇区，紧接着就是 62 扇区的保留扇区，由于 MBR 扇区很重要，所以系统会把 MBR 扇区在保留扇区中备份一份。接下来就是第一个分区，分区的第一个扇区是 DBR 扇区，接着就是一定数量的保留扇区，保留扇区的个数在 DBR 扇区中有一个字段来说明。接着就是文件分配表（FAT，File Allocation Table），FAT 有两份，即 FAT1、FAT2，FAT1 和 FAT2 内容是完全一致的，由于 FAT 的内容非常重要，所以硬盘上有了两个完全相同的 FAT——FAT1 和 FAT2，每次修改时，都会同时对 FAT1 和 FAT2 做同样的操作。FAT2 后面开始就是数据区，在数据区里面，硬盘的分配与回收以簇为单位，簇的大小在 DBR 扇区中有相应的字段指明。数据区的开始是 root directory，里面存放该分区的文件名字等属性。

<table>
<tr><td>分区间隔</td><td>分区间隔（gap）63 扇区</td></tr>
<tr><td rowspan="6">分区二</td><td>数据区</td></tr>
<tr><td>Root Directory</td></tr>
<tr><td>FAT2</td></tr>
<tr><td>FAT1</td></tr>
<tr><td>保留扇区</td></tr>
<tr><td>DBR 扇区</td></tr>
<tr><td>分区间隔</td><td>分区间隔（gap）63 扇区</td></tr>
<tr><td rowspan="6">分区一</td><td>数据区</td></tr>
<tr><td>Root Directory</td></tr>
<tr><td>FAT2</td></tr>
<tr><td>FAT1</td></tr>
<tr><td>保留扇区</td></tr>
<tr><td>DBR 扇区</td></tr>
<tr><td rowspan="2">分区间隔</td><td>保留扇区（62 个）</td></tr>
<tr><td>MBR 扇区</td></tr>
</table>

图 3.2 FAT32 系统下分区结构图

(1) DBR

DBR 扇区（DOS Boot Record）是操作系统引导记录扇区。通常在分区的第一个扇区，由于其比较重要，在后面的保留扇区中同样有备份。在这 512 字节的 DBR 扇区中，包括跳转指令、厂商标志、操作系统版本号、BPB（BIOS Parameter Block）、扩展 BPB、OS 引导程序和结束标志等几部分。图 3.3 就是 winhex 查看的一个 DBR 扇区。

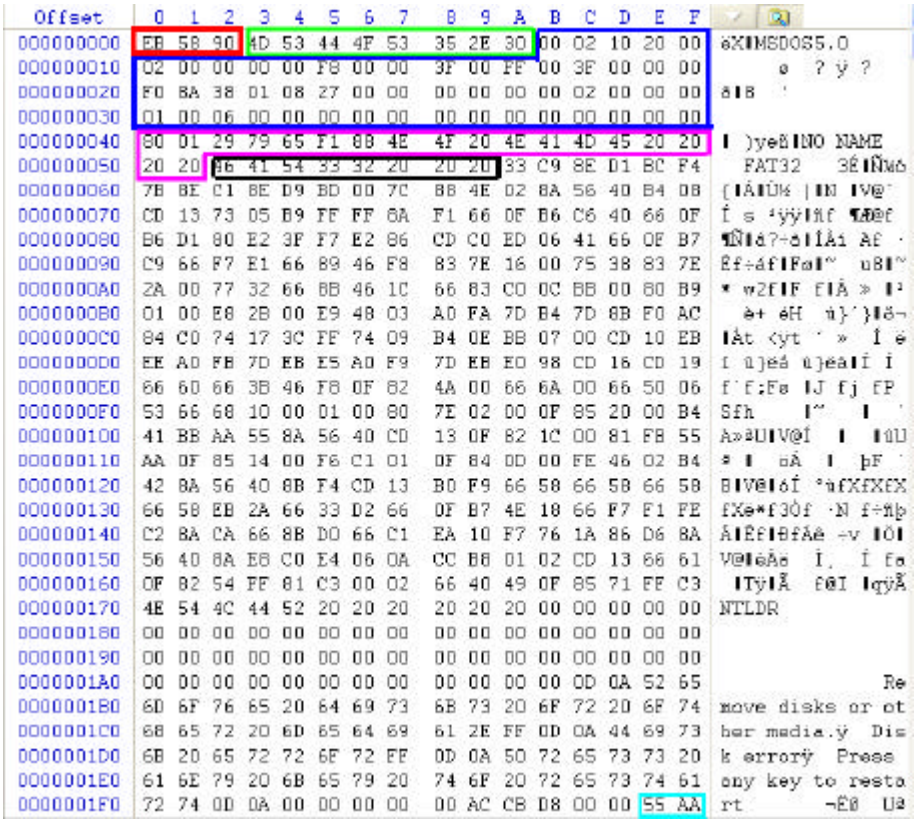


图 3.3 FAT32 系统下一个 DBR 扇区

图 3.3 中使用多个方框标出各个字段，其中各字段的含义如表 3.2 所示，详见附录 A。

当 MBR 中的引导程序被执行完了之后，会将控制权交给 DBR 扇区，DBR 扇区的开始是一个跳转指令，“EB 58 90”表示向后跳转 90 字节，这样就进入了 OS 引导程序区。跳转指令后是一个 8 字节长的字符串，叫做 OEM ID，OEM ID 表示了格式化该分区的操作系统的名称和版本号，这是为了保持与低版本操作系统的兼容性。接下来的一段 53 字节的内容是为了引导程序的入口参数 BPB 字段，分区的大小，

起始位置等信息都封装在该字段里面。同样是为了兼容各版本的原因，BPB 字段后还有一个 26 字节的扩展 BPB 字段，相对高版本的系统才会使用到里面的相关参数。下面通过表 3.3 和 3.4 分别说明 BPB 字段和扩展 BPB 字段里面各个子字段的含义，表 3.3 和表 3.4 详见附录 A。

在扩展 BPB 字段后面就是引导 420 字节的引导程序代码。起始位置是 0x5A，即 90。DBR 扇区的前三个字节是一条跳转指令“EB 58 90”，CPU 执行完后这条指令后就会向后跳过 90 字节开始执行，正好就是引导程序代码。在引导程序代码后面还有最后的两字节“55 AA”，是表示结束有效的标志。

(2) FAT 表和数据的存储原则

FAT 表是 Microsoft 在 FAT 文件系统中用于磁盘数据（文件）索引和定位的一种链式结构。如果将磁盘比作一本书，那么 FAT 表就是这本书的目录，而数据文件就是各个目录对应的内容。但 FAT 表的表示方法与目录还是有很大的不同。

在 FAT 文件系统中，文件的存储依照 FAT 表制定的簇链式数据结构来进行的，同时 FAT 文件系统将组织数据时使用的目录也抽象为文件，以简化对数据的管理。

计算机系统的很多文件经常增加或删除内容，有时候甚至要删除整个文件，在管理这些文件时，如果采取顺序存储，类似数据结构中的数组，虽然可以进行随机访问，但是对于数据内容经常变化，就要经常移动后面数据以适应前面数据的增加与删除时，移动数据要消耗大量的时间和资源，这样顺序存储的效率就非常低。而且当系统频繁存储或者删除多个文件时，一段时间后会产生很多存储碎片，当需要存储一个比最大碎片大而比碎片之和小的文件时，系统几乎无法处理，除非转移文件，合并碎片，但是碎片整理是非常消耗时间和资源的，这样很不划算的。这样就需要一个比较灵活的存储方式，类似于数据结构中的链表。在 FAT 文件系统中，就是采用了这种链式存储结构。

在 FAT32 文件系统格式化硬盘时会将整个数据区以固定数目的扇区为一簇划分很多个簇，簇就是数据区中分配，使用的最小单位。簇的大小一般为 2^n 个扇区的大小，这个在 BPB 字段中每簇扇区数定义。每个簇都有对应的序号，当存储某个文件时，只需要 FAT 表中存放着该簇的下一簇的簇序号。通过这些序号可以找到哪些簇是应该连成一块存放某个文件。FAT32 存储三个文件 A.txt、B.txt、C.txt 的示意图如图 3.4 所示。

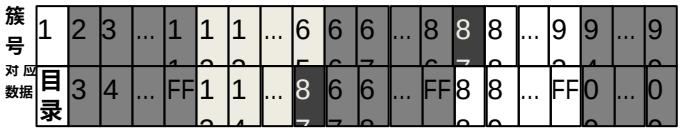


图 3.4 FAT 32 **系统下存储文件示意图**

图 3.4 **中所示第一个文件占用了第** 2、3、……、11 **簇。第二簇对应的存储内容为** 3，**表示某一文件在写完第二簇后接着写第三簇，依次类推，当写到** 11 **簇时，文件已经结束，在该簇对应的位置上填上** FF **作为文件结束标志。第二个文件占用了第** 12、13、……、65、87、88、……、93 **簇。第** 65 **簇对应的内容是** 87，**写完第** 65 **簇后，因为第** 66 **簇已被占用，所以从第** 87 **簇开始继续写，直至写到第** 93 **簇文件结束。第三个文件占用了第** 66、67、……、86 **簇。而第** 94、95、……、99 **簇内容为** 0，**表明这些簇未被使用，属于空闲空间。由于存储空间可以同** FAT **表连接起来，在目录里面只需要记下文件的起始簇号，和簇的数目就可以。如图** 3.5 **所示。**

A.TXT	2	10	2004.3.22 10:41	2004.3.22 11:30	只读
B.TXT	66	20.5	2004.3.18 12:31	2004.3.20 18:50	系统
C.TXT	12	60.3	2004.2.25 16:51	2004.3.12 19:37	存档

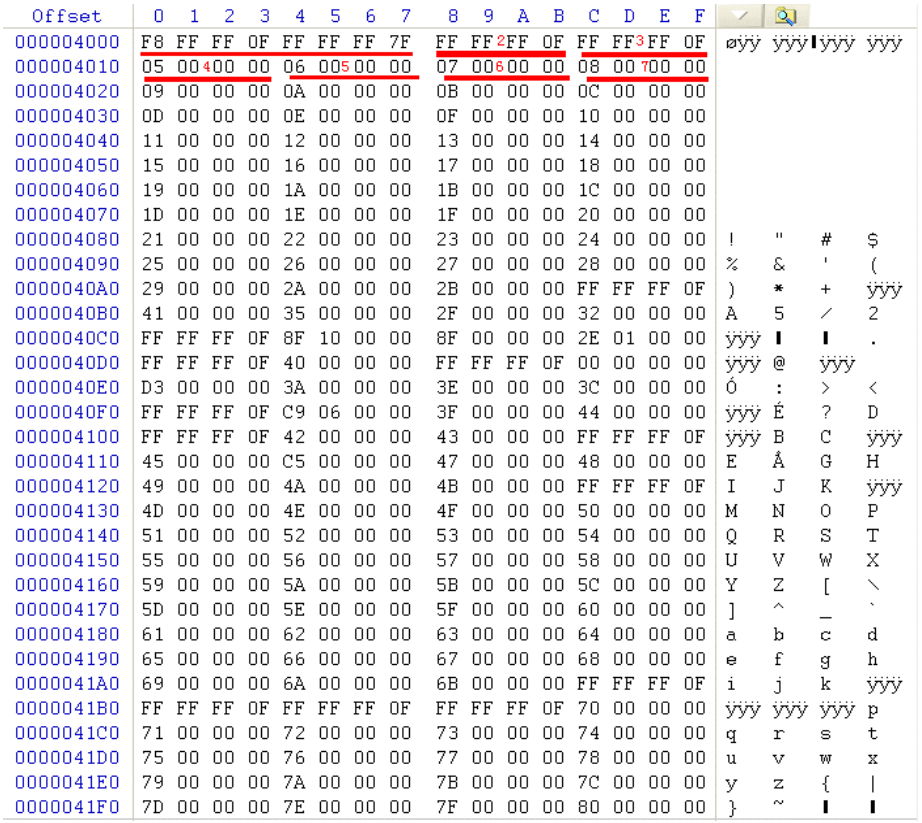
图 3.5 **目录结构示意图**

通过 winhex **可以查看一个** FAT32 **系统中的一个** FAT **扇区，如图** 3.6 **所示。**

图 3.6 **中八个字节**“F8 FF FF 0F FF FF FF 7F”**位介质描述单元，并不参与** FAT **表簇链关系。后面有下横线标出的，每四字节表示一簇。如** 0x08-0x0B **就表示第** 2 **簇（实际上是第一簇），在前面** BPB **字段中** 0x2C **根目录簇号就说明了这个。第** 2 **簇对应的内容是**“FF FF FF 0F”**表示这个文件只使用了第** 2 **簇就结束，实际上这个文件就是根目录。图** 3.5 **中第** 4 **簇的内容是**“00 00 00 05”**表示第** 4 **簇的下一簇就是第** 5 **簇。依次类推到第** 0x2B **簇时文件结束。**

(3) **目录文件**

分区的数据区在 FAT **表的帮助下被划分成了一个个数据块，每个数据块就是一个文件，数据块由一个或者多个簇组成，而各个簇间的联系就是依靠** FAT **表来指示。文件存放好了之后，可以通过两种方式进行访问，一种方式就是从数据区开始逐个访问，另一种方式就是将各个文件的起始位置记录下来，存放在某个固定的地方，**



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000004000	F8	FF	FF	0F	FF	FF	FF	7F	FF	FF	FF	0F	FF	FF	FF	0F
000004010	05	00	00	00	06	00	00	00	07	00	00	00	08	00	00	00
000004020	09	00	00	00	0A	00	00	00	0B	00	00	00	0C	00	00	00
000004030	0D	00	00	00	0E	00	00	00	0F	00	00	00	10	00	00	00
000004040	11	00	00	00	12	00	00	00	13	00	00	00	14	00	00	00
000004050	15	00	00	00	16	00	00	00	17	00	00	00	18	00	00	00
000004060	19	00	00	00	1A	00	00	00	1B	00	00	00	1C	00	00	00
000004070	1D	00	00	00	1E	00	00	00	1F	00	00	00	20	00	00	00
000004080	21	00	00	00	22	00	00	00	23	00	00	00	24	00	00	00
000004090	25	00	00	00	26	00	00	00	27	00	00	00	28	00	00	00
0000040A0	29	00	00	00	2A	00	00	00	2B	00	00	00	FF	FF	FF	0F
0000040B0	41	00	00	00	35	00	00	00	2F	00	00	00	32	00	00	00
0000040C0	FF	FF	FF	0F	8F	10	00	00	8F	00	00	00	2E	01	00	00
0000040D0	FF	FF	FF	0F	40	00	00	00	FF	FF	FF	0F	00	00	00	00
0000040E0	D3	00	00	00	3A	00	00	00	3E	00	00	00	3C	00	00	00
0000040F0	FF	FF	FF	0F	C9	06	00	00	3F	00	00	00	44	00	00	00
000004100	FF	FF	FF	0F	42	00	00	00	43	00	00	00	FF	FF	FF	0F
000004110	45	00	00	00	C5	00	00	00	47	00	00	00	48	00	00	00
000004120	49	00	00	00	4A	00	00	00	4B	00	00	00	FF	FF	FF	0F
000004130	4D	00	00	00	4E	00	00	00	4F	00	00	00	50	00	00	00
000004140	51	00	00	00	52	00	00	00	53	00	00	00	54	00	00	00
000004150	55	00	00	00	56	00	00	00	57	00	00	00	58	00	00	00
000004160	59	00	00	00	5A	00	00	00	5B	00	00	00	5C	00	00	00
000004170	5D	00	00	00	5E	00	00	00	5F	00	00	00	60	00	00	00
000004180	61	00	00	00	62	00	00	00	63	00	00	00	64	00	00	00
000004190	65	00	00	00	66	00	00	00	67	00	00	00	68	00	00	00
0000041A0	69	00	00	00	6A	00	00	00	6B	00	00	00	FF	FF	FF	0F
0000041B0	FF	FF	FF	0F	FF	FF	FF	0F	FF	FF	FF	0F	70	00	00	00
0000041C0	71	00	00	00	72	00	00	00	73	00	00	00	74	00	00	00
0000041D0	75	00	00	00	76	00	00	00	77	00	00	00	78	00	00	00
0000041E0	79	00	00	00	7A	00	00	00	7B	00	00	00	7C	00	00	00
0000041F0	7D	00	00	00	7E	00	00	00	7F	00	00	00	80	00	00	00

图 3.6 FAT32 系统下 FAT1 扇区

即建立一个索引。显然逐个访问的效率太低。其实在 FAT32 中就有这样一个索引——目录文件。上一节提到过根目录，根目录一般就放在数据区的第一个簇中，簇号为 2。FAT32 文件系统的目录结构其实就是一棵有向的从跟到叶子的树，也就是说要想访问文件系统的任何一个文件，都必须从根目录开始，逐层向下查找。根目录就是这个目录结构的入口。

在 FAT32 中，目录也是作为一个文件来看待的，如果一个目录中的文件足够多，1 簇没有存储，也可以在动态增加一簇。此时需要将该簇的序号替换原来的“FF FF FF 0F”。在目录文件中，每 32 字节表示一个目录项，即表示一个文件（暂时不考虑长文件名的情况）。32 字节的目录项中各字节的含义如表 3.5 所示，详见附录 A。

对于短文件名的文件，系统将文件名分为文件名+扩展文件名两部分来存储。文件名为 8 字节，扩展文件名 3 字节，不够部分使用空白字符（0x20H）填充，如果在 0x00 处的取值为 0x00H 时，表明该目录项为空，没有被使用。如果为“E5H”，表明该目录项曾被使用过，但是对应的文件或者文件夹已被删除。如果文件名为“.”或者

“..”，则表示该记录是一个目录文件。其中“.”表示当前目录，“..”表示上一级目录。0x0B 的属性字段表示该文件的属性。该字段在短文件名的目录项中不允许为 0x0F，如果取值为 0F，则表明该目录是一个长文件名文件的目录段。而长文件名的目录项的各字节含义如表 3.6 所示，详见附录 A。

下面通过 winhex 查看一下 FAT32 系统下一个根目录的扇区，如图 3.7 所示。

图 3.7 中，该根目录一共有 8 个文件，在图 3.6 的左侧已经标出。第一个文件的属性字节取值为 0x08，表示该目录项是一个卷标，即指示当前盘符为 TEST

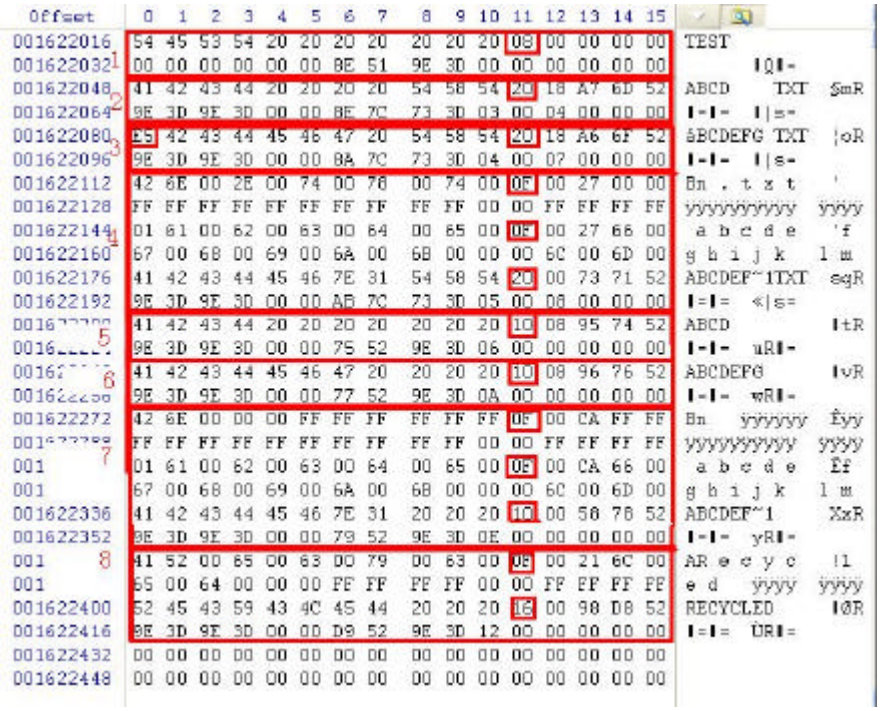


图 3.7 FAT32 系统下根目录扇区

盘，不占磁盘空间，起始簇号为 0，大小亦为 0；第二个文件的属性字节为 0x20，表示该目录项指示的是一个存档文件，文件名为 abcd.txt，起始簇号是 00000003，文件大小 00000004 字节；第三个文件的属性字节同样为 0x20，也是指示一个存档文件，但是它的 0x00 处的取值为 0xE5，表示这是一个已经被删除了的存档文件，起始簇号为 00000004，文件大小为 00000007 字节；第四个文件中，前两个目录项的属性字节取值 0x0F，表示这两个目录项只是一个长文件名目录的一部分，到第三个目录项属性字节取值 0x20，表示这个长文件名的文件是一个存档文件，按照长文件名目录项格式，可以取出这个长文件名的文件是 abcdefghijklmn.txt，起始字节为 00000005，

文件大小为 00000008 字节；第五个文件的属性字节是 0x10，表示该文件是一个文件夹目录文件，文件夹名为 abcd，其初始簇号是 00000006，大小没有指示。同理可以看出第六、七、八个文件分别是 abcd 的文件夹、abcdefg 文件夹、长文件名 abcdefghijklmn 的文件夹，以及 RECYCLED 的系统隐藏文件夹。

3. 计算机文件存储特点总结

计算机系统文件的存储方案的特点就是分块分配，链式使用。

计算机的存储介质是一个连续的块状空间，计算机系统中的文件很多，而且很多文件经常会改变，只有分块分配，链式使用式存储方案才能满足性能上的要求。如果采用线性顺序存储，系统就会在不停转移文件上花费过多的资源与时间。这与内存的段页式管理方案很类似，也与数据结构中的链表结构有异曲同工之妙。分块分配，链式使用要求分配使用或者释放空间时以固定大小的存储单元为单位，在分配使用时各个被分配的存储单元不要求是连续空间，它们间的链接关系由第三方负责，最为高效的是可以随便在文件结构增加内容而不用考虑文件结尾后面是否有可用的空闲空间。

当把一个文件的部分内容增加到另一个文件中时，这在存储单元上就很可能会出现这样的情况，这两个文件会有若干最小存储单元一模一样。这在计算机病毒检测方面将会是很大的帮助，当病毒感染一个目标文件时，会把自身代码的复制到目标文件中，如果在目标文件中找到一个存储单元与某病毒文件的某一个存储单元一模一样，而这个存储单元还是这个病毒文件特有的，那么就确定该目标文件被该病毒感染过。而实际中判定两个存储单元是否一模一样要比在这个文件中搜索某个特征串要高效的多。

3.3.2 计算机病毒文件存储特点分析

计算机病毒要想通过自己的代码被执行而获得系统控制权，其自身或者感染了的文件必须是可执行文件，如 EXE 文件或者 COM 文件。它们都是 PE（Portable Executable）格式文件。

1. PE 文件分析

PE(Portable Executable 可移植的执行体)格式是 Win32 环境自身所带的可执行文件的格式，Windows XP 也算是 Win32 的系统，其下面的可执行文件和动态链接库均采用 PE 文件格式[40]。

PE 文件是一种线性数据流式文件，主要由四部分组成：DOS 部分、PE 文件头(PE Header)、节表(Section Table)、节(Section)。每个部分中又是由几个小部分组成的，其详细结构如图 3.8 所示。

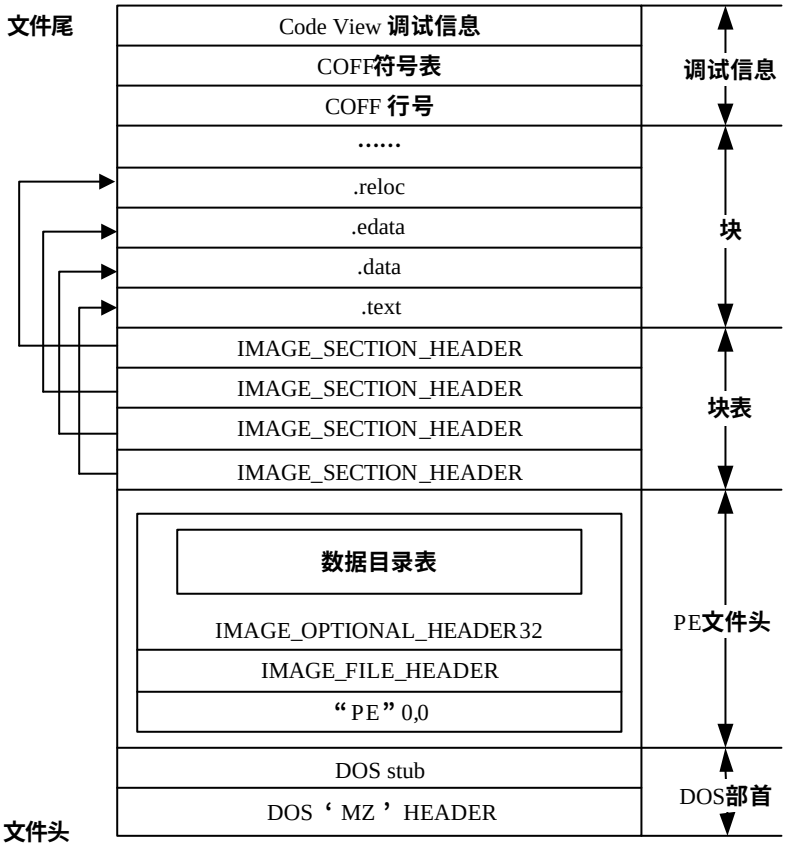


图 3.8 PE 文件结构图

(1) DOS 部分

为了兼容低版本的系统而保留的 DOS 部分。所有的 PE 文件的都必须以一个简单的 DOS MZ header 开始。在这个结构中，标志字段 e_magic 通常被赋值为“MZ”，表明结构是 DOS MZ header。计算机病毒在感染时通常就会首先去判断目标文件的

该字段是否有 DOS MZ header 的标志。如果没有标志，则说明目标文件不是一个 PE 文件，不用感染，否则就进一步判定后面的内容。在 DOS MZ header 中还有一个非常重要的 e_lfanew 字段，该字段指示 PE 头结构的开始位置。紧跟着 MZ header 之后的是 DOS Stub，在不支持 PE 文件格式的操作系统中，简单地显示一个错误提示信息。这部分的大小不是固定的，需要借助前面的 e_lfanew 字段来定位 PE 文件头。

(2) PE 文件头

PE header 是 PE 相关结构 IMAGE_NT_HEADERS 的简称，其中存放 PE 文件的重要信息，这些信息是被操作系统或者应用程序来识别该 PE 文件的核心数据的，如代码、数据等。PE header 主要由三个部分组成：PE 文件的标志(Signature)，映像文件头(IMAGE_FILE_HEADER)和映像可选头(IMAGE_OPTIONAL_HEADER)。PE 文件的标志(Signature)为一个 4 字节的字符串，被定义为 0x00004550H，即字符串“PE/0/0”。若目标文件对应位置没有这个标志，说明该目标文件不是 PE 文件；映像文件头(IMAGE_FILE_HEADER)存放着 PE 文件一共有多少节，字符表的指针，字符表的数目等等信息；映像可选头(IMAGE_OPTIONAL_HEADER)包含 PE 文件的物理分布信息、逻辑分布信息，程序的入口地址等等重要信息，虽然名字说是可选，却是必须的，大小为 224 字节。在映像可选头的最后是一个 IMAGE_DATA_DIRECTORY 型数据目录数组。

(3) 节表

PE 文件头后面的就是节表，因为后面的每个节都有一个对应的节头结构，节表实际上就是一个存放所有节的节头结构的数组，即节表是一个 IMAGE_SECTION_HEADER 型的结构数组。至于该结构中元素的个数则是由前面 PE 文件头中 IMAGE_FILE_HEADER 中 NumberOfSections 字段指示。所以节表就依次顺序存放各个节的头部结构，大小不是固定的。

(4) 节

一般 PE 文件都会有几个节，各个节依照前面节表中各个节头部的顺序存放。PE 文件有很多节，与计算机病毒有密切关系的节：代码节(.text)含有程序的可执行代码；初始数据节(.data)存放在编译时已经确定的数据；资源节(.rsrc)存放如图表、对话框等程序要用到的资源；重定位节(.reloc)存放一个重定位表，若装载器不是把程序装到程序编译时默认的基地址时，需要用这个重定位表来做一些调整。各节按照各自的固定格式存放相应数据。

2. 计算机病毒文件存储特点总结

计算机病毒文件都是 PE 文件。PE 文件是一个线性数据流文件，PE 文件会将任一个程序文件按照固定的格式存储起来，当程序需要运行时就从相应的对方获取该程序的各方面信息（如代码、数据等等）加载到内存。所以 PE 文件逻辑上就是一个线性的数据流。

(1) 计算机病毒文件在硬盘中的存储细节

观察计算机病毒文件头部，即 PE 文件头部，在硬盘的存储详情，如图 3.9 所示。

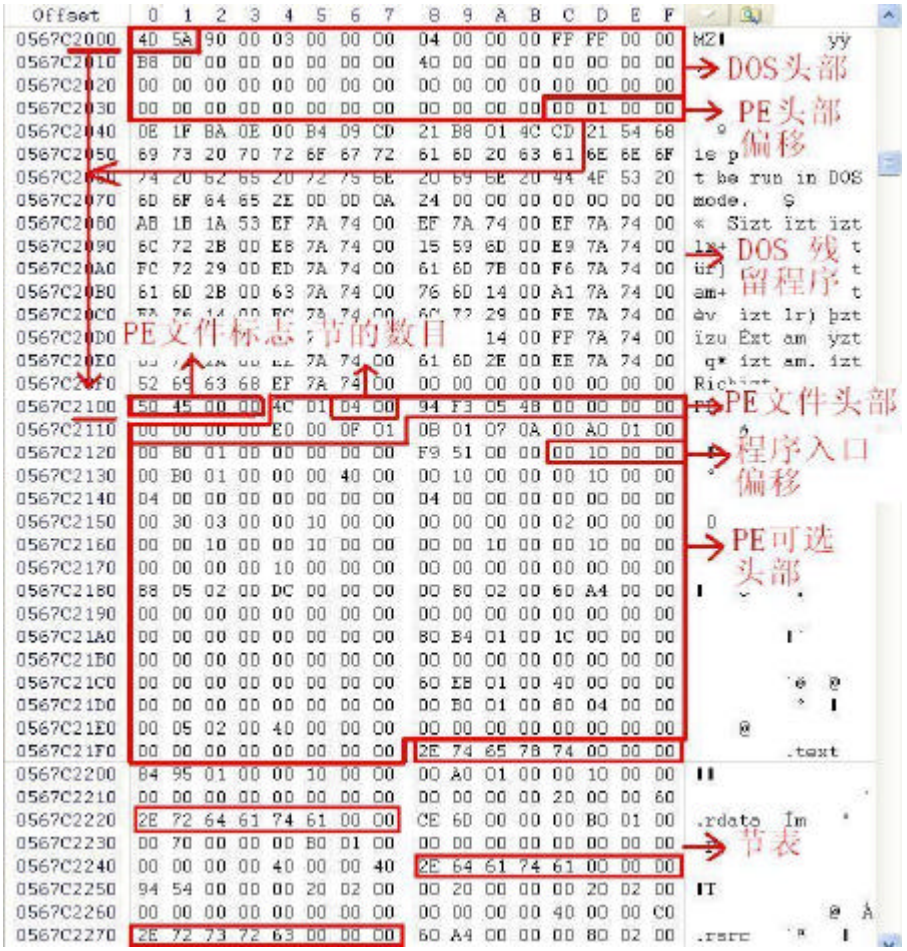


图 3.9 计算机病毒文件头部存储信息

在图 3.9 中，很容易看出计算机病毒文件除了节外的各个部分，DOS 头部结构、DOS 残留程序、PE 文件标志、PE 文件头部结构、PE 可选头部结构和节表。在 DOS 头部结构中 e_lfanew 字段取值为 0x00000100，即 PE 文件头的偏移为 0x00000100，

结合文件的起始位 0x0567C2000，所以 PE 文件头的位置为 0x0567C2000 + 0x00000100 = 0x0567C2100。恰好在 0x567C2100 处找到 PE 的文件标志 0x50450000。在紧跟着的 20 字节是 PE 文件头部结构，其中的 NumberOfSections 字段取值为 0x0004，表示该病毒文件有 4 个节表头部结构，4 个节表。PE 文件头部结构后面跟着的 224 字节内容为 PE 文件可选头部结构，其中的 AdressOfEntryPoint 字段取值 0x00001000，即该病毒文件的代码节偏移为 0x00001000，所以代码节的起始位置为 0x0567C2000 + 0x00001000 = 0x0567C3000。在 PE 可选头部结构后面的是节表，前面 PE 文件头部结构的 NumberOfSections 字段说明该病毒文件有 4 个节表头部结构，4 个节。从图 4.8 中可以看出该病毒文件有.text、.rdata、.data 和.rsrc 四个节。下面通过 winhex 尝试着在 0x0567C3000 处开始寻找代码节和其他三个节，如图 3.10 所示。

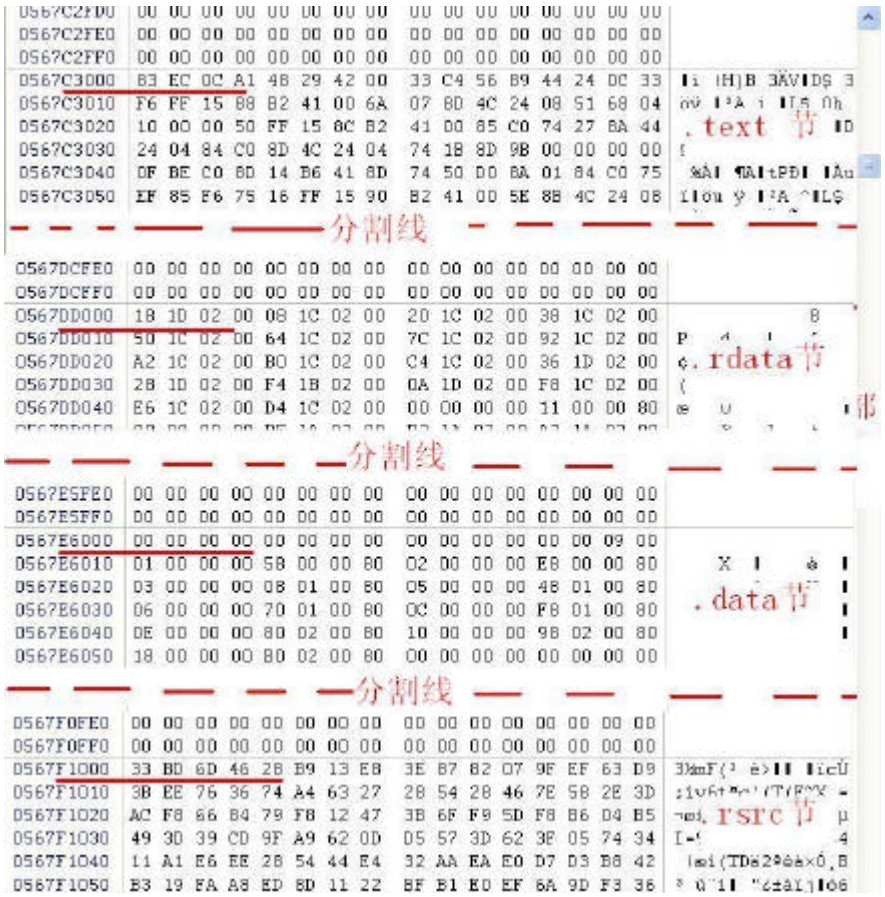


图 3.10 计算机病毒文件 4 个节的存储信息

根据 PE 头部结构中的信息，代码节的开始位置在 0x0567C3000，通过 winhex，在 0x0567C3000 处发现了.text 节，并在后面的 0x0567DD000，0x0567E6000，0x0567F1000 处分别找到了.rdata 节，.data 节和.rsrc 节。

(2) 计算机病毒文件的存储特点

计算机病毒文件是一个线性数据流文件，里面按照固定格式存放这执行改程序所需要的各种信息。通过前面的分析可知，计算机病毒文件在存储时并没有严格按照顺序依次存放的，在节表与节之间，节与节之间一般都有填充内容。通过分析图 3.10 中各节的起始地址时可以发现，节的起始地址都是 0x1000 的整数倍，在图 4.9 中，每个扇区是 512 字节，即 0x200，一簇是 8 个扇区，即有一簇的容量是 0x200 * 8 = 0x1000。由此可以知道节都是另起一个新簇的。当一个计算机病毒感染一个目标文件时，如果通过增加一节的方法来将自身的病毒代码植入目标文件时，目标文件就会跟计算机病毒文件有相同的一个或者多个簇。如果该计算机病毒的特征码取自该簇，只要在目标文件中找到该簇就可以确定该目标文件被该计算机病毒感染过。

3.4 基于存储特点的 CVSAE 算法

设计 CVSAE 算法的方法要求特征码取自于计算机病毒的共同的特征。计算机病毒的存储点就是计算机病毒共同特征之一，从这个突破口下去就可以设计一个新的 CVSAE 算法。由于现有所有的计算机病毒在不发作，即处于潜伏期的病毒都是以文件的形式存储于计算机系统中，所以这个新的基于存储特点的 CVSAE 算法将会有很广泛的适用性。

3.4.1 算法思想

在计算机病毒感染目标程序文件时必定会将自身的某一部分代码植入到目标程序文件中去，由于计算机系统文件都是分块分配，链式使用的，计算机病毒文件若增加一个新的节时，需要开辟一个新的簇，在计算机病毒文件和被感染的目标程序文件中就必定存在一个或者多个内容完全一致的簇，或者至少是存在一个内容完全一致的扇区。如果把这一簇或者多个簇选作计算机病毒的特征码时，用它来在目标程序文件中匹配，也能起到检测计算机病毒的作用，而且如果以簇为单位对目标程

序文件进行一个扫描匹配，检测效率将比串匹配效率要高得多。

3.4.2 特征码设计

计算机病毒与被感染的目标程序文件共有的部分是被计算机病毒植入到目标程序文件中的部分内容。根据计算机系统文件的存储特点以及计算机病毒的存储特点，分析知这部分内容可能就是在一个，可能是多个全新的簇或者扇区上。计算机病毒检测时就是要发现这个，可能有多个的共有簇或者扇区，一种极端情况就是目标程序文件的所有簇或者扇区与计算机病毒都是相同的，这就是蠕虫病毒，蠕虫病毒能够独立存在，不需要将自身代码植入任何文件，或者说是它用自身代码替换了目标程序文件的所有内容。现在的特征码就是为了帮助扫描引擎来发现这个共有的簇或者扇区。这样一来，问题就转化为了怎样判断两个簇（簇块）或者两个扇区是完全相同的。一簇有 8 个扇区，每个扇区是 512 字节。如果对该簇的所有字节内容逐一匹配，在效率上就会很低。特征码应该采用特征码集的形式出现，这里也是采用选取该簇的某些扇区，或者该扇区的某些字节来组成一个特征码集。又因为在存储时都是靠前存储的，所以不应该均匀选取特征码，即在判断两个簇（簇块）是否完全一致时，选择确认该簇（簇块）的第 1、2、4、7 扇区（簇）；在判断两个扇区是否完全一致时，选择确认该扇区的第 1、3、7、15、31、63、127、255、511 字节的内容。所以扇区的特征码就是该扇区的第 1、3、7、15、31、63、127、255、511 字节的内容，簇的特征码就是该簇的第 1、2、4、7 扇区的特征码的集合。

3.4.3 特征码提取

计算机病毒检测时需要在目标程序文件中找到与计算机病毒共有的簇或者扇区，所以特征码提取时就应该从这些共有的簇或者扇区中提取，在特征码设计时，已经设计好了特征码，也知道了如何从目标簇或者目标扇区中提取出特征码组成一个特征码集。现在的问题转化为在计算机病毒文件中找到目标簇或者目标扇区。此处有两种方案：

方案一，在开始时，准备两套完全一样包含很多程序文件的计算机系统，即两套计算机系统中的所有文件都是一样的，让一套计算机系统隔离起来，不受任何病毒

的侵扰。另一套系统则开放给病毒，一段时间后，某种计算机病毒感染了一些程序文件，搜集这些文件并将它们分别对第一套系统中对应文件进行比较，找到被该病毒植入的部分，从而确定目标簇（簇块）或者目标扇区。

方案二，搜集很多个被同一种计算机病毒感染过的目标程序文件，在一些可疑的地方，如代码节的开始，或者最后一节等常见的被病毒植入代码的地方检查，尽量找到被计算机病毒植入代码地方，从而确定目标簇（簇块）或者目标扇区。有些病毒采用分段植入式，即在目标程序文件中寻找填充区来插入病毒代码，所以这些填充区域也是检查的重点。

上述两个方案是针对狭义的寄生式计算机病毒而言，对于能够独立存在的计算机病毒而言，如蠕虫病毒，就更加简单，代码节的所有簇都是目标簇块。

3.3.4 扫描引擎的特征码匹配

扫描引擎一般都是与特征码的组成，提取方式有关的，针对手动提取的计算机病毒特征码，就只有采取串匹配的方式进行扫描检测，因为手动提取特征码时，提取的地方，内容都是相当随意的，毫无章法而言，所以检测时需要将代码节等等逐一进行串匹配。对于基于存储特点的 CVSAE 算法，特征码提取都是来自于一个簇或者一个扇区内的某些字节，而且病毒植入代码时要么选择增加一节，要么选在填充区间内。这样被感染文件要么是从一个新簇开始增加了一节，要么大小不变。针对此类型的特征码而言，首先检查填充区间是否被写入数据，如有有则从对应写入的扇区开始，针对每个扇区都提取准首扇区特征码，拿到特征码库中去匹配，命中就以对应扇区开始，提取准特征码，拿到特征码库中作二次匹配。从而确定目标文件是否被感染。填充区间检查完毕后针对每节都提取准首簇的特征码，拿到特征码库中作一次匹配，同理匹配成功再做二次匹配。最后确定目标文件是否被感染病毒，如果感染则获取对应病毒的名称。

3.5 算法分析

3.5.1 算法适用对象

从算法的工作流程来看，该算法主要是针对文件型病毒而言的，对于引导型病毒等提取形式存在的病毒不具有效用。严格来说算法只是针对静态病毒有检测效果，算法检测病毒时时扫描硬盘上的各个文件，很多病毒爆发时就常驻内存，将自己都清除掉了。所以算法要想取得更好的查杀效果，应该在扫描引擎模块加以改进。

3.5.2 时间复杂度

该算法主要包含两大模块，特征码的提取模块和扫描引擎的检测模块。对于一个计算机病毒而言，特征码提取模块只会被执行一次或者几次，而扫描引擎的检测模块就会被执行无数次，也就是说相对于检测模块而言，特征码提取模块只是被执行有限次。所以算法的效率就由检测模块来决定。

在特征码提取模块里面，重点在目标簇定位过程，提取特征码只是一个读文件的过程，常数时间内可以完成。在目标簇定位方面，很显然方案一的性能要高于方案二，方案一的缺点就是需要额外的资源。

在扫描引擎检测模块，算法是以节或者扇区为单位对目标程序文件的代码节或者其他节进行一个准特征码提取工作，然后就是特征码的匹配工作。算法中的特征码是采用特征码集的形式存在，该集合的最小元素是扇区特征码，即 9 字节。特征码集的匹配就是逐个匹配特征码集中的各个元素。扇区特征码匹配只是两个定长字符串的匹配，可以事先将这个字符串组成一个数值，通过比较数值是否相等来判断匹配成功与否，时间复杂度为 $O(1)$。特征码集匹配的时间复杂度为 $O(n)$，其中 n 为特征码集中元素个数，最大值为 4(簇)*4(扇区)。准特征码的个数有两种，一种是填充扇区的个数，最大值为 16(节)*7(扇区)，另一种就直接节的数目，甚至更少，一般数据区不做特征码提取，最大值为 16(节)。所以总的时间复杂度为 $O(n*m)$，其中 n 为特征码集中扇区个数，最大值为 144，而 m 为准特征码的个数，最大值为 112。另外这种扫描检测是一次提取，多次匹配（每种病毒的特征码都需要匹配一次），而传统

的串匹配式扫描引擎是多次匹配，最终扫描引擎的时间复杂度为 $O(d*n*m)$。d 为特征码库中病毒的种类，也就是特征码的个数，n,m 跟前面定义一样。而传统的串匹配扫描引擎的时间复杂度为 $O(d*(p+q))$，d 的定义与前面一样，特征码的个数。p,q 则分别表示计算机内所有 PE 文件的大小之和，所有的特征码长度之和。这个两个数据一般都是在 G，M 和 M，K 级别的了。这样基于存储特点的 CVSAE 算法的扫描引擎的时间复杂度将远远低于传统的扫描引擎了。

3.6 小结

本章主要通过分析传统基于特征码的计算机病毒检测方法，针对蠕虫病毒的 CVSAE 系统和针对 VB 病毒的 CVSAE 算法的优缺点，总结了设计新的 CVSAE 算法的方法。该方法的关键是：要求特征码来自于计算机病毒的公共特性，而不是某个病毒独自的功能特性。计算机病毒都是计算机系统中的文件，或者说寄生在计算机系统中某个文件中。这样计算机系统中文件的存储特点就是所有计算机病毒的公共特性，计算机病毒的存储特点就是所有计算机病毒的公共特点。所以计算机系统文件的存储特点和计算机病毒的存储特点就是新算法的两个必要条件。

在分析计算机系统文件的存储细节和计算机病毒的存储细节之后，可以总结计算机系统文件的存储特点为分块分配，链式使用。计算机病毒文件存储特点为——节间有空隙，节起始于新的一簇。最后结合这两个特点和 CVSAE 算法的设计方法设计一个基于存储特点的 CVSAE 算法，并对新的算法进行各方面分析。本章的重点就是设计出一个基于存储特点的 CVSAE 算法原型。

4 系统设计

前面几章讨论了 CVSAE 算法的设计方法，并设计出了一个基于存储特点的 CVSAE 算法原型。本章的重点是根据算法原型设计出这个基于存储特点的 CVSAE 系统。

4.1 模块设计

根据第 3 章的原型算法，基于存储特点的 CVSAE 系统应该包含两大模块：特征码提取模块和计算机病毒扫描引擎模块。在特征码提取模块中又有目标簇（簇块）或者扇区定位子模块和特征码提取子模块；在扫描引擎模块又包含快速扫描，全盘扫描和自定义扫描等三个子模块。在特征码提取模块里，目标簇（簇块）或者扇区定位子模块找到计算机病毒中被植入到目标文件中的部分，特征码就是从该部分中提取出来的。所以特征码提取之前必须找到目标簇（簇块）或者扇区，可以手动设置目标簇（簇块）或者目标扇区。也可以根据方案一所示的方法。特征码提取子模块就是在目标簇（簇块）或者扇区确定之后，依据特征码设计原则来提取出特征码集。扫描引擎模块里，快速扫描就是针对启动项等关键区域进行扫描，范围小速度快；全盘扫描就是扫描硬盘上所有文件，范围大速度慢；自定义扫描就是由人工设定扫描范围。该系统的总体设计模块图如图 4.1 所示。

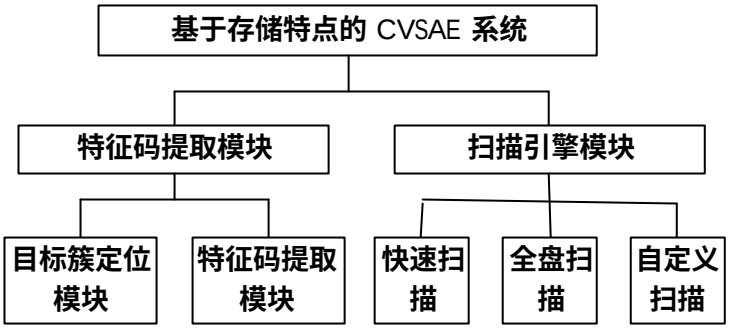


图 4.1 系统总体设计模块图

4.2 特征码提取模块

特征码提取模块主要是从被计算机病毒感染的目标程序文件中找到被病毒植入的部分，然后以这部分内容作为目标簇（簇块）或者目标扇区，从中提取出特征码集出来。目标簇（簇块）或者目标扇区的寻找采用原型算法中的方案一，即对同一文件在被计算机病毒感染前后两个版本，找到被植入的部分。

4.2.1 寻找目标簇

寻找目标簇（簇块）或者目标扇区（findObjectClustersOrSectors）的工作流程为：

(1) 首先检测两个文件是否都是 PE 文件，是则继续否则退出；

(2) 比较两个文件的大小，将小文件作为未被感染的版本，将大的文件作为被感染的版本，分别获取两个文件的节数，看看被感染的文件是否是通过增加了节的方式而被植入了病毒代码的，是则转(3)，否则转(4)；

(3) 逐节比较，找到被感染了文件中多出的一节或者多节，返回该节的偏移位置和大小（簇的数目 * 簇大小）。

(4) 在填充空间（节表之后和每节之后不足一簇的区域内）中寻找本来属于空白区域而被填充了内容的扇区，返回扇区的偏移和大小（扇区的数目 * 扇区大小）。

4.2.2 提取特征码

提取特征码（extractSignature）的工作流程为：

(1) 根据目标空间的大小来判断目标空间是簇级别还是扇区级别的，如果目标空间大小大于或者等于一簇的字节数，则是簇级别的，否则就是扇区级别。如果是簇级别的转（2），否则转（3）；

(2) 依次从第 1、2、4、7、…等每隔 i 个簇，就提取一个簇的特征码，i 每次自增 1，在提取各个簇的特征码时，依次提取该簇里面的第 1、2、4、7 扇区的特征码，在提取各个扇区的的特征码时，依次提取该扇区里面的第 1、3、7、15、31、63、127、255、511 字节的内容，然后拼成一个特征码集返回，同时也返回该特征码集的大小。

(3) 依次从第 1、2、4、7（如果存在）扇区，就提取一个扇区的特征码，在提取

各个扇区的的特征码时，依次提取该扇区里面的第 1、3、7、15、31、63、127、255、511 字节的内容，然后拼成一个特征码集返回，同时也返回该特征码集的大小。

4.2.3 特征码库设计

当前模块负责将提取的特征码集放在特征码库中，提取的特征码是少量的，因此对特征码库的要求不是很高。但是在扫描引擎模块中，设计特征码库的重要性就会得到真正体现，因为每扫描一个目标程序文件，就会对特征码库查找一遍，而计算机系统中的目标程序文件数量是很大，特征码库会被无数次查询，因此良好的特征码库设计也是扫描引擎效率高的一个关键。由于提取出的特征码是分了类，所以特征码存储也应该分类，即目标簇为簇级别的特征码集放在一块，而目标簇为扇区级别的特征码集又单独放在一块。特征码库的设计图如图 4.2 所示。

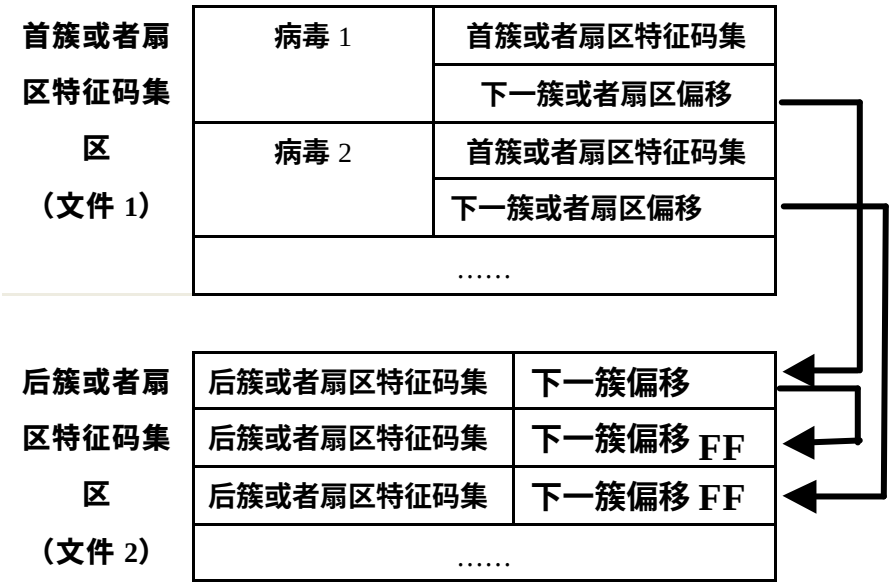


图 4.2 特征码库设计图

特征码库分为两部分，即簇特征码库和扇区特征码库。特征码库分为两个部分：首簇或者扇区特征码集区和后簇特征码集区，分别存放在两个文件中。首簇或者扇区特征码集区存放计算机病毒的首簇或者扇区的特征码集，后簇或者扇区特征码集区存放该病毒剩下的簇或者扇区的特征码集。首簇或者扇区特征码集区中以固定的结构存放着病毒名称，首簇或者扇区特征码集，和下一簇或者扇区的偏移。如果只有一簇或者扇区，该字段使用 0xFF 填充。否则填上后簇或者扇区

中该病毒特征码集的偏移。后簇或者扇区特征码集区中也固定的结构存放一簇或者扇区的特征码集和下一簇或者扇区偏移。如若没有下一簇或者扇区则填充0xFF。

4.3 扫描引擎模块

扫描引擎模块主要负责依据前面提取出的特征码集组成的特征码库来扫描计算机系统中的文件，检测是否有文件被特征码库中的计算机病毒感染过。依据扫描范围又划分了快速扫描，全盘扫描和自定义扫描三个子模块。虽然是三个子模块，但是核心部分都是一样。核心部分内容又包含两个重要步骤：递归查找目标文件和对该目标文件进行病毒扫描检测。

递归查找目标文件（searchFile）模块的工作流程为：

(1) 指定找到的目录为当前工作目录；

(2) 开始搜索所有文件(*.*)；

(3) 判断当前目录是否搜索完毕，是则返回，否则继续；

(4) 判断找到是否为目录，是目录则进入该目录递归调用自身searchFile，否则继续；

(5) 是文件，则调用病毒查找模块；

(6) 搜索下一个文件，转 (3)。

计算机病毒检测（scanVirus）模块的工作流程为：

(1) 检查目标文件是否为 PE 文件，不是 PE 文件则退出，否则继续；

(2) 读取目标文件的头部信息，获取节的数目，代码节的偏移地址。

(3) 检查填充空间是否有非零内容，如果有则提取目标扇区的特征码集。否则转（5）；

(4) 针对每个扇区提取出的扇区特征码集，然后逐个分别与扇区特征码库中每个病毒的首扇区特征码集进行匹配，如果匹配不成功则返回；如果匹配成功，以目标程序文件的该扇区为目标扇区提取出准特征码集，然后取出匹配成功的那个病毒的所有扇区特征码集与准特征码集进行匹配，匹配成功则检测出被感染，向上提交检测出来的病毒信息，感染目标文件的名字，返回；否则没有检测出来，直接返回；

(5) 从代码节中的每个簇提取出簇的簇特征码集，然后逐个分别与簇特征码库中

每个病毒的首簇特征码集进行匹配，如果匹配不成功则返回；如果匹配成功，以目标文件中的该簇为目标簇提取准特征码集，然后取出匹配成功的那个病毒的所有簇特征码集与准特征码集进行匹配，匹配成功则检测出病毒，向上提交检测出来的病毒信息，感染目标文件的名字，返回；否则就没有检测出病毒感染，继续；

(6) 除了.data 节，针对每节执行(5)。

4.4 小结

本章在第一部分详细描述了该仿真实验系统的设计过程及原理，第二部分是对仿真实验系统进行有实验数据的测试验证。

需要注意的是：本章设计的仿真实验系统仅仅只是一个仿真系统，只是为了证明该算法的可行性。也在一定程度上说明，要想原型算法更具有实用价值还需要进一步的深入研究和解决一些技术层面上的问题。

5 仿真实验与结果分析

在第三章中，在总结了 CVSAE 算法的设计方法的基础上设计了一个新的 CVSAE 算法——基于存储特点的 CVSAE 算法，在第四章中，依据这个算法设计了一个系统。本章将进行这个系统的仿真实验与验证工作。

5.1 实验目的

本实验是针对基于存储特点的 CVSAE 系统的实验测试，为了检验系统的自动提取计算机病毒特征码的可用性，以及在病毒检测方面的性能效率，最终说明基于存储特点的 CVSAE 算法的正确性等。

5.2 实验环境

1．硬件环境：PC机（奔腾IVCPU 2.66GHz，1.5GB 的内存），操作系统为Windows XP系统；

2．软件环境：Visual C++ 6.0 编程软件，VMware 虚拟机，一些毒样本等。

5.3 关键技术与数据结构设计

在设计 CVSAE 仿真系统时需要解决的关键问题有几个：

1. 文件存储信息的获取。分析计算机文件系统存储特点和计算机病毒存储特点时经常用到一些存储参数，如扇区大小，每簇扇区数等。因此这些参数并不是一直都是固定不变的。这些参数需要从 DBR 扇区中的 BPB 字段中获取。通过函数 CreateFile()和 ReadFile()来实现。

```
hDriver=CreateFile("\\\\.\\:F", GENERIC_READ, FILE_SHARE_READ, NULL,
        OPEN_EXISTING, 0, NULL);//创建 F 盘的句柄
ReadFile(*hDriver, bufferSector, 512, &bytesRead, NULL);//DBR 在第一扇区，直接读
```

2. 特征码数据结构设计。最终供扫描引擎使用的是一个特征码集，在提取特征

码时以扇区为最小单位，多个扇区就依次存放。

```
struct Signature
{       long size; //特征码的长度，10 的整数倍
        char *signatureContent;//存储特征码的各个字节内容
};
```

3. **特征码库数据结构设计。特征码被分成了扇区特征码和簇特征码，存储时也是分类存储，方便快速匹配。**

```
struct SignatureLibFirst_sector
{       char virusName[21];//病毒名称
        char signatureFirst[10];//首扇区病毒特征码
        long signatureLastOffset;//后扇区偏移
};
struct SignatureLibLast_sector
{       char signatureLast[10];//一个扇区病毒特征码
        long signatureLastOffset;//下一扇区偏移
};
struct SignatureLibFirst_cluster
{       char virusName[21];//病毒名称
        char signatureFirst[40];//首簇病毒特征码
        long signatureLastOffset;//后簇偏移
};
struct SignatureLibLast_cluster
{       char signatureLast[40];//一个簇病毒特征码
        long signatureLastOffset;//下一簇偏移
};
```

5.4 实验设计

1. 特征码自动提取

在目标簇定位子模块中，系统采用的是方案一，即对比某个文件被病毒感染前后的内容，定位出被病毒植入的部分。另外系统还增加了一个手动定位的功能，即有用户输入该部分的偏移及大小。此处设置两个测试用例。

测试用例 1，输入同一个文件感染病毒前后的两个版本，获得特征码；

测试用例 2，输入一个被病毒感染了的目标程序文件，手动设置被感染部位的偏移及大小。

提取出的特征码集按照设计的特征码库格式存放到特征码库中，以便扫描引擎在扫描检测计算机病毒时使用。

测试用例 3，输入一个被病毒感染了的目标程序文件，手动设置被感染部位的偏移及大小，选择保存到特征码库的功能。

2. 病毒扫描

逐个扫描目标文件，如果是 PE 文件则提取出其准特征码，将准特征码放到特征码库去匹配，最终完成扫描引擎的测试。由于系统有快速扫描，自定义扫描和全盘扫描三个子模块。因此需要三个测试用例：

测试用例 4，使用快速扫描模块；

测试用例 5，使用自定义扫描模块；

测试用例 6，使用全盘扫描模块。

5.5 实验结果

由于计算机病毒比较危险，测试在虚拟机里进行。这对扫描速度可能会有些影响。

测试用例 1，输入同一个文件感染病毒前后的两个版本，获得特征码。测试结果如图 5.1 所示。

测试用例 2，输入一个被病毒感染了的目标程序文件，手动设置被感染部位的偏移及大小。测试结果如图 5.2 所示。

图 5.1 图

5.2

测试用例 3，输入一个被病毒感染了的目标程序文件，手动设置被感染部位的偏移及大小，选择保存到特征码库的功能。测试结果如图 5.3 所示。

测试用例 4，使用快速扫描模块。测试结果如图 5.4 和 5.5 所示。

测试用例 5，使用自定义扫描模块。测试结果如图 5.6 所示。

测试用例 6，使用全盘扫描模块。测试结果如图 5.7 所示。

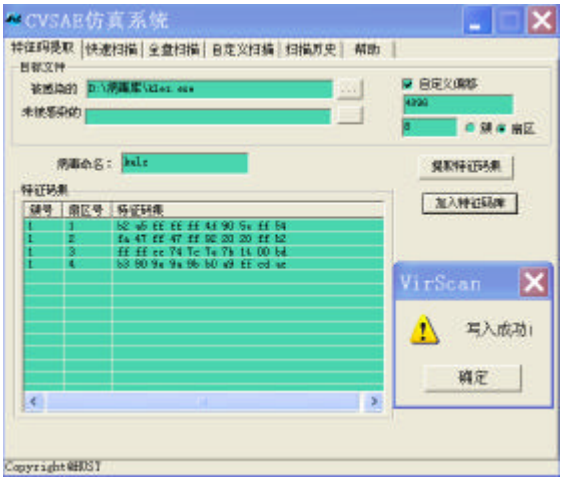


图 5.3

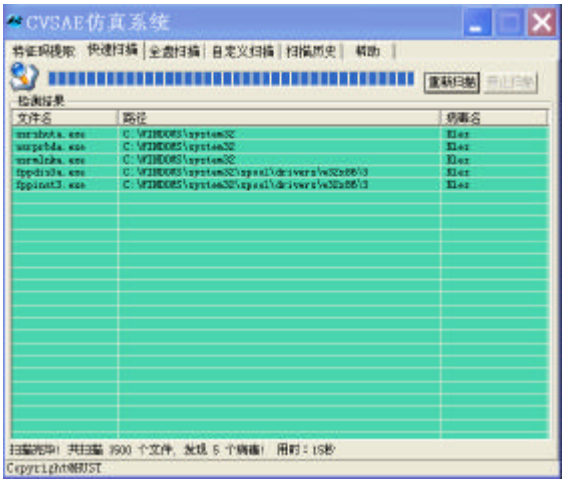


图 5.4

```
          0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00000000h: 6B 65 6C 7A 00 CD CD CD CD CD CD CD CD CD CD CD ; kelz.屯屯屯屯屯?
00000010h: CD CD CD CD CD CD B2 A5 FF FF FF 4F 90 5E FF 54 FA ; 屯屯崩?   O悸  T?
00000020h: 47 FF 47 FF 92 20 20 FF B2 FF FF CC 74 7C 7E 7B ; G  G  ?   ?  鄄|~{
00000030h: 14 00 BD B3 90 9E 9A 9B B0 A9 FF CD AC CD CD CD ; ..匠閎殯瘥  同屯?
00000040h: FF FF FF FF                                     ;
```

图 5.5

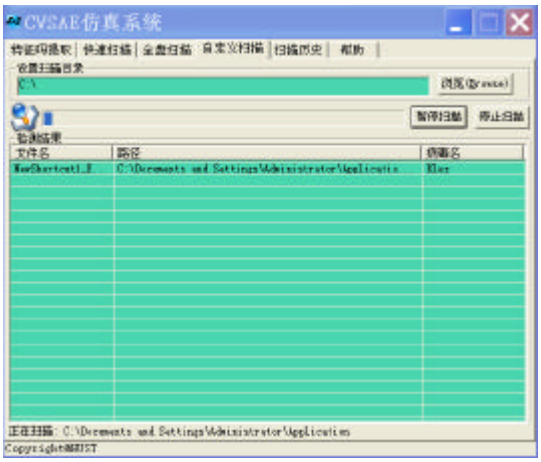


图 5.6

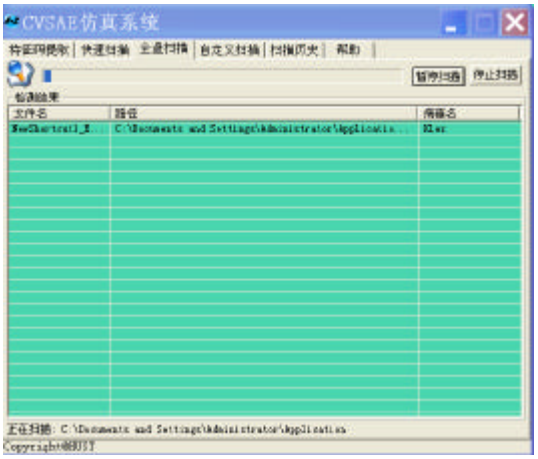


图 5.7

5.6 结果分析

经过以上测试，表明仿真系统达到了预期目标。仿真系统能够针对目标文件自动提取出特征码，并依靠此特征码能够在文件系统中把所有感染了此病毒的被感染文件找出来。

6 总结

6.1 工作总结

近几年来，计算机病毒的爆发有愈演愈烈的趋势，2007 年肆虐全中国的“熊猫烧香”病毒至今还让不少人心有余悸。虽然行为监测、软件虚拟等新一代计算机反病毒软件起了不少作用，但是它们实用价值比较低，很难担负目前情况下的计算机反病毒任务。基于特征码的计算机病毒检测方法是计算机反病毒工作的有效的武器。分析基于特征码的计算机病毒检测方法，也存在不少问题，如提取病毒的特征码需要相当的时间而导致反病毒工作的严重滞后性，另外就是病毒检测时采用效率低下的串匹配模式从而使得整个反病毒工作的效率不高。所以寻求 CVSAE 算法的研究工作迫在眉睫。

本文从选题到研究工作告一段落过程中，主要完成了以下工作：

1. 搜集关于计算机病毒方面的国内外资料，了解目前最新的计算机反病毒工作面临的形势与新的挑战。

2. 结合病毒结构对病毒的特征码作了详细的分析。基于特征码的病毒检测方法的关键就是特征码。因为特征码可以很好地区分被病毒感染了的文件和正常未被感染的文件，使得基于它的检测方法在实际中有很强的实用性，准确率很高，误报率很低。这也是它能够担任起主流的计算机反病毒任务。然而其不足也是源自于特征码。只有准确掌握了病毒的特征码才能准确查杀该病毒。特征码是整个检测方法的基础，也是该检测方法发展的瓶颈。

3. 由于关于 CVSAE 算法的研究很少，论文只是分析了现有四个蠕虫病毒的特征码自动提取与检测系统，和一个 VB 应用程序病毒的特征码自动提取算法。通过分析比较几个系统或算法的异同点，尝试找出设计 CVSAE 算法的思路或者方法。从特征码选取入手，结合特征码结构形式变化去实现计算机病毒特征码的自动提取。

4. 针对文件的存储特点，探究基于存储特点的 CVSAE 算法的可行性以及相关实现方法。根据计算机系统的存储特点和计算机病毒的特点设计了一个基于存储特点的 CVSAE 算法原型。

5. 根据基于存储特点的 CVSAE 算法原型设计了一个仿真系统，仿真系统从存储

特点的角度从感染了病毒的目标程序文件中提取出病毒的特征码集，然后系统依据特征码集扫描计算机系统内部的文件，检测是否被该病毒感染。

6.2 工作展望

论文设计的基于存储特点 CVSAE 算法是一个原型算法。由于水平和认识有限，没有进一步细化。它的意义主要在于开阔思路。该原型算法需要进一步研究的方向有：

1. 对应多态病毒或者变形病毒的处理能力。其实也是特征码法的共有的通病，在检测这两类病毒时，特征码法有相当的窘迫。多态病毒或者变种病毒的一个很明显的特征码就是变化，而特征码的精髓就是不能变。所以需要寻求一种方法来调和二者间的对立矛盾。如研究出多态病毒或者变形病毒的变与不变，研究它们如何变，变化间的联系等等。

2. 提取特征码的过程中，需要找到计算机病毒植入到目标程序文件中的部分区域。原型算法虽然给出了两种方案。但需要做进一步的研究。

3. 在提取特征码时，在选取簇，扇区，字节等方面应该做更多的测试实验，确定更加合理的阈值。

致 谢

论文写到这里也是一个让人激动的时刻，展望即将踏入色彩缤纷，充满挑战的社会，不由让人心潮澎湃；回首将近三年在美丽华中科技大学校园里的点点滴滴，不由让人感慨万千。在研究生期间，我结识了很多可敬可爱的老师，同学和朋友。在与他们一起度过的学习生活中，我学到了很多东西，不仅有书本上的知识，更多是做人做学问的真知。同时他们还给的关心，帮助是我快速成长。在这让人感慨的时刻，我要想所有跟我共同经历过着美好的三年校园生活的老师，同学和朋友们表达深深的谢意。

首先要感谢的是徐兰芳老师，她是我的导师。徐老师诲人不倦的高尚师德、严谨活泼的治学态度让我受益良多。徐老师不但亲自给我们上课，帮助我们学习理论知识，还在我参与课题项目组的课余时间跟我讨论学术研究上的问题，帮助我在做学问上面有所提高。很可惜我到最后都没有完成一篇学术论文，每每想到这里我对徐老师都是心存愧疚。徐老师对我生活上的关心，帮助也让我感动万分。本篇论文也是在徐老师的悉心指导下完成的。再次衷心感谢徐老师！

感谢实验室的彭老师，付老师对我在研究生期间项目上的指导，帮助。让我不仅在实验室项目上作出了一定的贡献，同时也锻炼了我的项目攻关能力，遇事寻求解决问题的能力。感谢韩老师，韩老师不仅在我的论文选题时给我出了很多主意，在论文写作时也提供了不少的帮助。

感谢同实验室的同学以及师弟师妹们。跟他们一起营造了一个良好的实验室氛围，共同学习，共同进步。

感谢养育我，教育我的长大成人的父母。我明白，无论什么时候你们都会在背后默默的关心我，支持我，这一切都是我不断前进的动力。谢谢你们！

最后衷心感谢在百忙中抽出宝贵时间审阅我论文的各位评审老师和答辩委员会的各位老师，感谢你们的无私指导和宝贵意见，感谢给我一个检验我自己的机会。谢谢你们！

参考文献

[1] 金山软件. 2009 年 4 月份中国电脑病毒疫情及互联网安全报告

[2] Global Information, Inc. 2006-2010 年企业反间谍软件市场调查报告

[3] Peter Szor.计算机病毒防范艺术. （第 1 版）.段海新，杨波，王德强.北京:机械工业出版社.2007.6～8, 31～42,79～118,281～325

[4] 曾鸣，赵荣彩，姚京松等. 基于特征提取的二进制代码比较技术.计算机工程与应用，2006: 7～11

[5] Fred Cohen. A Cryptographic Checksum for Integrity Protection in Untrusted Computer Systems. Computers and Security,1987:357～360

[6] 徐明，陈纯，应晶. 基于系统调用分类的异常检测. 软件学报，2004，15(3):391～403

[7] 曾宪伟，张智军，张 志. 基于虚拟机的启发式扫描反病毒技术. 计算机应用与软件，2005,22(9):125～126

[8] 傅建明，彭国军，张焕国. 计算机病毒分析与对抗.（第 1 版）.武汉:武汉大学出版社. 2003.77～90,173～200,341～388

[9] 忍云韬.基于二进制多态变形的恶意代码反检测技术研究:[硕士学位论文].成都:电子科技大学图书馆.2007

[10] 金庆，吴国新，李丹. 反病毒引擎及特征码自动提取算法的研究. 计算机工程与设计.2007,24(12):5863～5866

[11] Kreibich C，Crowcroft J. Honeycomb-Creating Intrusion Detection Signatures Using Honeypots. In:Proc. of the 2nd Workshop on Hot Topics in Networks (HotNets-II).Cambridge, MA USA.2003:51～56

[12] N. Provos. Honeyd-A Virtual Honeypot Daemon. In 10th DFN-CERT Workshop. Hamburg. Germany. 2003:89～92

[13] Singh S, Estan C, etc. Automated Worm Fingerprinting. In:Proc. of the 6th ACM/USENIX Symposium on Operating System Design and Implementation (OSDI 2004).San Francisco, CA. 2004:45～60

[14] Kim H-A,Karp B. Autograph:Toward Automated Distributed Worm Signature

Detection. 13th USENIX Security Symposium. San Diego. CA USA. 2004:271～286

[15] Newsome J,Karp B,Song D. Polygraph:Automatically Generating Signatures for Polymorphic Worms. In:Proc. of the 2005 IEEE Symposium on Security and Privacy(IEEE S&P 2005).Oakland, California, USA.2005:226～241

[16] Jeremy Z.Kolter, Marcus A.Maloof. Learning to Detect Malicious Executables in the Wild. Industry/Government Track Paper. 2004:470～478

[17] 郑辉，李冠一，涂生. 蠕虫的行为特征描述和工作原理分析. 见：第三届中国信息和通信安全学术会议. 2003,3(1):168～172

[18] 曾贵华.特洛依木马攻击下的量子密码安全性.软件学报.2004,15(8):1259-1264

[19] 刘元勋. 变形蠕虫自动检测技术研究:[硕士学业论文].济南:山东大学.2007

[20] 刘国柱，尚衍筠. 基于主动防御模式下蠕虫病毒特征码的提取模型及算法设计. 计算机应用与软件. 2009,6(26):362～368

[21] 郑辉.Internet 蠕虫研究:[博士学位论文].天津:南开大学.2003

[22] P. B. Gibbons and Y. Matias.New sampling-based summary statistics for improving approximate query answers. in :Proceedings of the ACM SIGMOD.1998:331–342

[23] U. Manber. Finding similar files in a large file system. in :Proceedings of the USENIX Winter 1994 Technical Conference, San Fransisco, CA, USA. 1994:1～10

[24] D.Gusfield.Algorithms on Strings,Trees and Sequences. Cambridge University Press. 1997:16～29

[25] MOORE, D.SHANNON, C. Code-Red: A Case Study on the Spread and Victims of an Internet Worm. In Proceedings of the 2002 ACM SIGCOMM Internet Measurement Workshop (IMW 2002).2002.3(1):326～337

[26] SINGH,S. ESTAN, C.VARGHESE, G. AND SAVAGE, S. The Early-Bird System for Real-time Detection of Unknown Worms. Tech. Rep.CS2003-0761, UCSD.2003:26～31

[27] STANIFORD, S.PAXSON, V.AND WEAVER, N. How to own the Internet in Your Spare Time. In Proceedings of the 11th USENIX Security Symposium .2002:149～167

[28] MUTHITACHAROEN, A. CHEN, B. AND MAZI`E RES, D. A Low bandwidth Network File System. In Proceedings of the 18th ACM Symposium on Operating

Systems Principles (SOSP 2001) .2001:589～603

[29] M.O.Rabin. Fingerprinting by Random Polynomials. Technical Report. Center for Research in Computing Technology. Harvard University. 1981:15～81

[30] **王双成，苑森淼，王 辉**. **基于类约束的贝叶斯网络分类器学习**. **小型微型计算机系统**. 2004,25(6): 968～971

[31] D. Gusfield. Algorithms on Strings, Trees and Sequences. Cambridge University Press. 1997:70～77

[32] L.Hui. Color Set Size Problem with Applications to String Matching. In Proceedings of the 3rd Symposium on Combinatorial Patttern Matching. 1992,64(4):230～243

[33] T.Smith,M.Waterman. Identification of Common Molecular Subsequences. Journal of Molecular Biology. 1981,17(147):195～197

[34] **张波云,殷建平, 蔺敬波等.基于多重朴素贝叶斯算法的未知病毒检测.计算机工程**.2006,32(10):18～21

[35] **禽敬波,殷建平,张波云.基于通用图灵机模型的病毒判定性定理证明.计算机科学**.2005,32(8):243～245

[36] **叶翔.主机安全防护系统研究与实现**:[**博士学位论文**].**武汉：华中科技大学**.2004

[37] Tessa Lau,Oren Etzioni,Daniel Sweld.Privacy interfaces for information management.Communications of the ACM.1999,42(10):89～94

[38] Roger Clarke.Internet privacy concerns confirm the case for intervention.Communication of the ACM.1999,42(10):60～67

[39] **尤晋元，史美林，陈向群等**. windows **操作系统原理**.（第 1 版）.**北京**:**机械工业出版社**.2001.353～372

[40] **汤志忠,李文龙,苏伯珙.一种软件流水的反流水算法.软件学报**.2004,15(7):987-993

附录 A 论文用到的表格

表 3.1 FAT 系统下 DPT 表项字段含义

字节偏移	字段长度	值	字段名和定义
0x01BE	BYTE	0x80	引导指示符，指明该分区是否为活动分区
0x01BF	BYTE	0x01	开始磁头
0x01C0	6 bits	0x01	开始扇区（只取低高 6bits）
0x01C1	10 bits	0x00	开始柱面（加前一字节的低 2bits 一共 10bits）
0x01C2	BYTE	0x0C	系统 ID，定义分区类型，0C 表示 FAT32
0x01C3	BYTE	0xFE	结束磁头
0x01C4	6 bits	0xFF	结束扇区（只取高 6bits）
0x01C5	10 bits	0xFF	结束柱面（加前一字节的低 2bits 一共 10bits）
0x01C6	DWORD	0x0000003F	相对扇区数（本分区开始之前的扇区数）
0x01CA	DWORD	0x01388AFC	总的扇区数

表 3.2 FAT32 系统下 DBR 各字段划分及含义

字节偏移	字段长度	字段名及含义
0x0000	3 BYTES	跳转指令
0x0003	8 BYTES	厂商标志和 OS 版本号
0x000B	53 BYTES	BPB
0x0040	26 BYTES	扩展 BPB
0x005A	420 BYTES	引导程序代码
0x01FE	2 BYTES	有效结束标志

表 3.3 FAT32 系统下 BPB 字段的含义

字节偏移	长度 BYTE	值（图 4.3）	名称和含义
0x0B	2	0x0200	扇区字节数，每个扇区的字节数，此处 512
0x0D	1	0x10	每簇扇区数，每簇的扇区数，此处 16
0x0E	2	0x0020	保留扇区数，此处为 32
0x10	1	0x20	FAT 数，该分区 FAT 个数，本字段一般为 2
0x11	2	0x0000	根目录数，低于 FAT32 版本有效，FAT32 只为 0
0x13	3	0x0000	小扇区数，低于 FAT32 版本有效，FAT32 只为 0
0x15	1	0xF8	媒体指示符，0xF8 表示硬盘
0x16	2	0x0000	FAT 扇区数，低于 FAT32 版本有效，FAT32 只为 0
0x18	2	0x003F	每道扇区数，每条磁道包含扇区数
0x1A	2	0x00FF	磁头数，磁盘中磁头的数目
0x1C	4	0x0000003F	隐藏扇区数，也就是分区间隔的扇区数，63
0x20	4	0x01388AF0	总扇区数，该分区中的扇区总数
0x24	4	0x00000827	每 FAT 扇区数，FAT1 占的扇区数
0x28	2	0x0000	扩展标识
0x2A	2	0x0000	文件系统版本号，FAT32 专用
0x2C	4	0x00000002	根目录簇号，一般为 2
0x30	2	0x0001	文件系统信息扇区号，FAT32 专用
0x32	2	0x0006	备份引导扇区号，DBR 的备份扇区号，一般为 6
0x34	12	12 字节 0x00	保留字段

表 3.4 FAT32 系统下扩展 BPB 字段含义

字节偏移	长度 BYTE	值（图 4.3）	名称和含义
0x40	1	0x80	物理驱动器号
0x41	1	0x01	保留字段

续表 3.4 FAT32 系统下扩展 BPB 字段含义

0x42	1	0x29	扩展引导标签，本字段必须为 0x28 或 0x29
0x43	4	0x88F16579	分区序号，格式化时获得的一个序号
0x47	11	“NO NAME”	卷标
0x52	8	“FAT32”	系统 ID

表 3.5 FAT32 系统中目录项各字节含义

字节偏移	字节数	含义	
0x00-0x07	8	文件名	
0x08-0x0A	3	扩展名	
0x0B	1	属性字段	0x00000000（读写）
			0x00000001（只读）
			0x00000010（隐藏）
			0x00000100（系统）
			0x00001000（卷标）
			0x00010000（子目录）
			0x00100000（存档）
0x0C	1	系统保留	
0x0D	1	创建时间的 10 毫秒位	
0x0E-0x0F	2	文件创建时间	
0x10-0x11	2	文件创建日期	
0x12-0x13	2	文件最后访问日期	
0x14-0x15	2	文件起始簇号的高 16 位	
0x16-0x17	2	文件的最近修改时间	
0x18-0x19	2	文件的最近修改日期	
0x1A-0x1B	2	文件起始簇号的低 16 位	
0x1C-0x1F	4	文件的长度	

表 3.6 FAT32 系统中长文件名目录项各字节含义

字节偏移	字节数	含义		
0x00	1	属性字段	7	保留
			6	1 表示长文件名目录最后一项
			5	保留
			4	顺序号数值
			3	
			2	
			1	
			0	
0x01-0x0A	10	长文件名 unicode 码（1）		
0x0B	1	长文件名目录项标志，取值 0FH		
0x0C	1	系统保留		
0x0D	1	检验值		
0x0E-0x19	12	长文件名 unicode 码（2）		
0x1A-0x1B	2	文件起始簇号（目前常置为 0）		
0x1C-0x1F	4	长文件名 unicode 码（3）		

计算机病毒特征码自动提取技术的研究

作者： 金雄斌
学位授予单位： 华中科技大学