致 谢

本论文及毕业设计是在我的导师王昭顺教授的悉心指导下完成的，在此表示由衷的感谢。王老师在我研究生阶段的科研工作给予了大力的指导，他为人师表的专业知识技能、敬业精神和对科学不懈的探索和追求所给予我的影响也将使我在未来的学习和工作中受益。

论文的顺利完成同时得到了赵万里同学、汪翔同学的大力支持和无私帮助，在此表示诚挚的谢意。

感谢在我攻读研究生学位过程中所有给予我帮助的老师们、同学们，你们的帮助使我受益匪浅。

最后，谨向在百忙之中抽出宝贵时间评审本论文的专家、学者致以最诚挚的感谢！

目录

插图

表格

第一章　绪论

1.1　研究背景及意义

在 1980 年，Martin Hellman[1]提出了一个“时间空间折中”的密码分析算法，使用了预先计算好并保存在内存和磁盘里面的数据，减少了密码分析需要的时间。这个算法在 1982 年被 Rivest 提出改进，减少了密码分析过程中所需要的存储空间。

2003 年 7 月瑞士洛桑联邦技术学院的 Philippe Oechslin 公布了一些实验数据，他及其所属的安全及密码学实验室（LASEC）采用了时间空间折中算法，使得密码破解的效率大大提高。他们开发的 Ophcrack 项目可以将一个操作系统的用户登录密码破解速度由 1 分 41 秒，提升到 13.6 秒 [2]。该项目提供了一个破解视窗作业系统下的 LAN Manager 散列（比如 hash 文件）的程序，作者免费提供了一些 Rainbow table，可以在短至几秒内破解最多 14 个英文字母的密码，有 99.9%的成功率。从 2.3 版开始可以破解 NT 散列，这功能对已经关闭 LAN Manager 散列的系统（Windows Vista 的预订设定）或是长于 14 个字母的密码特别有用。

同年 project-rainbowcrack 项目开始立项，该项目基于 Philippe Oechslin 提出的彩虹表，用 C++ 基本实现了对 MD5、SHA-1 算法的低位数低密钥空间的破解 [3]。接着出现了一个分布式彩虹表项目 Free Rainbow Tables[4]，这个项目的分布式系统是基于伯克利开放式网络计算平台（BOINC）。

1.2　国内外研究现状与进展

1.3　本文研究现状

1.4　论文组织结构

第二章　密码分析学

2.1　密码分析学的发展背景

静态测试技术是指不运行被测程序本身，而是通过分析源程序的语法、结构、过程、接口等来检查程序的正确性。主要包括源代码审核和二进制审核 [5] 。

源代码审核是指通过人工或相关工具对源代码进行审核，找出其中存在的漏洞。源代码分析是一项非常重要的措施，因为它可以在软件发布之前对潜在的漏洞进行发掘，提前找到可能的漏洞，以减少软件在实际运行过程中因漏洞而导致的损失。目前，世界各个大型的软件公司基本都有专门的小组负责对自身的软件进行安全测试，他们所使用的主要技术之一就是源代码审核技术。2006 年，美国政府与专业的源码审核公司 Coverity 达成了协议，由政府拨专款资助 Coverity 公司分析源代码并解决美国各级基础设施的关键代码的安全隐患。

二进制审核是指对二进制代码进行反汇编，通过分析反汇编代码来寻找安全漏洞。通常这需要借助于反汇编工具将待检测的可执行文件进行反汇编，将难以被辨认的机器码解析为某种更适合人理解的汇编代码，同时也还会用到其它一些反编译器和调试器等来分析和追踪寄存器值，从而来发现可能存在的漏洞。从这里也可以看出，二进制审核由于没有源代码，所以对漏洞检查的安全人员有着更高的要求。

目前，静态分析技术的目标一般是 C 和 C++，而且由于 C++ 的一些高级特性使得源代码审核更加困难，所以大部分的静态分析工具集中在 C 语言上。对于 Java 语言，目前也有一些审核工具，如 Findbugs[7] ，另外，国内著名安全组织安全焦点在 2007 会议上也有文献讨论基于 Java 的 Web 应用程序的安全问题。

静态代码分析技术包含比较多的部分，如词法分析、控制流分析、数据流分析、符号执行、类型推理、抽象解释、模型检测和自动定理证明等等。目前在安全漏洞挖掘方向使用最为广泛的是词法分析、控制流分析和数据流

分析 [6] 。此外，国外的研究人员在 2007 年度新提出了应用布尔可满足概念来改进源码审核技术的有效性，这也是当前源码审核技术研究的一个焦点。

但是静态测试技术也有着它的缺点，首先是源代码审核的方法要求提供源代码，但是一些涉及到重要内容的 Web 应用往往不会提供源代码，有些公司提供的都是编译好的可执行程序，因此源代码审核并不适用于大部分的 Web 应用的安全测试。其次，由于静态测试时针对静态的程序为目标，难以发现程序在动态运行过程中存在的问题。

2.2 密码分析学的定义

2.3 密码分析学研究的必要性

2.4 密码分析的方法

2.4.1 穷举法

2.4.2 查表法

2.4.3 时空折中法

发酵法阿斯顿发阿斯顿发阿斯顿发阿斯顿发阿斯顿发

法士大夫阿斯顿发阿德说发

阿斯顿发生的 f

2.4.4 分析法

第三章　时空折中算法

本章主要要介绍时空折中算法与

3.1　Martin Hellman 最初的算法

3.1.1　预运算

3.1.2　在线分析阶段

3.1.3　TMTO 曲线

3.1.4　密钥分析的成功率

3.2　Ronald Rivest 差异点 DP 方法

3.3　Philippe Oechslin 的彩虹表算法

3.3.1　非完美彩虹表

3.3.2　完美彩虹表

$$P_{table} \geq 1 - \prod_{i=1}^{t}(1 - \frac{m_i}{N}) \tag{3.1}$$

当 $m_1 = m$ 且 $m_{n+1} = N(1 - e^{-\frac{m_n}{N}})$

第四章　利用彩虹表进行密码破解

4.1　破解单向 Hash 函数

4.1.1

4.1.2

4.1.3

4.1.4

4.2　破解分组加密算法

4.3　破解其他加密算法

第五章　彩虹表算法优化设计与实现

5.1　计算能力优化

5.1.1　基于 GPU 的优化方案

5.1.2　基于云计算的优化方案

5.2　存储优化

5.3　算法结构优化

5.4　本文的实现与性能分析

第六章　本文总结与展望

6.1　总结

6.2　展望

参考文献

[1] Martin Hellman. A cryptanalytic time-memory tradeoff. *IEEE Transactions on Information Theory*, vol.26(401-406), 1980. 1

[2] P.Oechslin. Making a faster crytanalytic time-memory trade-off. *Lecture Notes in Computer Science*, vol.2729, 2003. 1

[3] 冯登国. 密码分析学. 清华大学出版社, 2000-08.

[4] 冯登国，裴定一. 密码学导引. 科学出版社, 北京, 1999.

[5] ZhuShuangLei. The Time-Memory Tradeoff Hash Cracker, 2003. http://project-rainbowcrack.com.

[6] 徐隽. 密码分析中的“时间内存替换”. *Neinfo Security*, 05 2004.

[7] Nvidia CUDA. http://developer.nvidia.com/object/cuda.html.

[8] 方海英. 基于时空折中算法的 word 文档破解研究. Master's thesis, 杭州电子科技大学, 2009.

[9] 金銓. DES 密码算法的彩虹攻击技术及其 GPU 实现. Master's thesis, 上海交通大学, 2010.

[10] 叶剑. 基于 GPU 的密码算法实现技术研究. Master's thesis, 解放军信息工程大学, 2010.

[11] 王彩霞. 密码分析中几种方法的研究及其设计与实现. Master's thesis, 西北大学, 2004.

[12] 翁捷. 带随机数 MD5 破解算法的 GPU 加速与优化. Master's thesis, 国防科技技术大学研究生院, 2010.

[13] 将秉天. 基于分布式计算的密码恢复系统研究. Master's thesis, 上海交通大学, 2010.

[14] Jin Hong. The cost of false alarms in Hellman and Rainbow Tradeoffs. http://eprint.iacr.org/2008/362.pdf.

[15] Koji Kusuda and Tsutomu Matsumoto. Achieving higher success probability in time–memory trade-off cryptanalysis without increasing memory size. *TIEICE: IEICE Transactions on Communications/Electronics/ Information and Systems*, 1999.

[16] Michael S. Distributed Pre-computation for a Cryptanalytic Time-Memory Trade-Off. http://ritdml.rit.edu/dspace/bitstream/1850/7805/1/MTaberThesis10-2008.pdf.

[17] Daegun Ma. Studies on the Cryptanalytic Time Memory Trade-Offs. http://library.snu.ac.kr.

[18] Johan Borst，Bart Preneel，Joos Vandewall. On the Time-Memory Tradeoff Between Exhaustive Key Search and table pre-computation. *Proceedings of the 19th symposium on Information Theory in the Benelux,Veldhoven(NL)*, pages 111–118, 1998.

[19] S. Che，M. Boyer，J. Meng，D. Tarjan，J. W. Sheaffer and K. Skadron. A Performance Study of General Purpose Applications on Graphics Processors using CUDA. *Journal of Parallel and Distributed Computing*, 2008.

[20] NVIDIA Corporation. NVIDIA CUDA Programming Guide, 2011.

[21] G. Avoine，P. junod and P. oechslin. Time-memory trade-offs: False alarm detection using checkpoints. In Proceedings of Progress in Cryptology – 6th International Conference on Cryptology (INDOCRYPT’05). Lecture Notes in Computer Science, vol. 3797. Cryptology Research Society of India, Springer-Verlag, Bangalore, India, pp, 183–196, 2005.

[22] A. Biryukov and A. Shamir. Cryptanalysis time/memory/data tradeoffs for stream ciphers. Proceedings of Asiacrypt' 00(T. Okamoto, ed.)no 1976 in Lecture Notes in Computer Science, pp. 1-13,Springer-Verlag,2000.

[23] V. L.L. Thing and H.M. Ying. A novel time-memory trade-off method for password recovery. *In Proceedings of the Ninth Annual DFRWS Conference*, 6(1):114–120, 2009.