

基于GPU的彩虹表弱口令旁路检查工具

随着信息数据的快速增长，信息数据安全性变得愈加重要，根据国家电网信息运维的特殊情况，需采用旁路检查的方法对弱口令进行检查和监测。设计了一套基于GPU的彩虹表旁路弱口令检查工具，扩大了彩虹链长度，利用CUDA架构对彩虹表进行并行查找，在节省了空间的同时，极大地提升了查找速度。本工具已经完全可以用于弱口令的实时监测，极大提升了公司信息安全的水平。

马思硕¹ 汪明² 陈连栋² 耿海洋² / 1. 国网北京市电力公司信息通信分公司
2. 国家电网公司信息通信分公司

1 研究背景



马思硕/工程师

随着信息数据的快速增长，特别是随着云计算和大数据的高速发展，信息数据的安全性已经成为关系国计民生的问题。密码学是信息安全中最为重要的学科，随着计算机系统的不断发展，计算性能的不断提升，特别是密码学的破译方法不断改进，使得相对简单的算法变得越来越不可靠。也使得对加密算法的研究也不断加深，尽量保证算法加密效率情况下提高被破解的复杂度，从而带动了加密和破解技术双方面的提高。

国家电网公司从SG186项目开始，进行了大规模的信息系统建设，信息安全的相关政策和制度也愈加完善，特别是针对账号权限方面，制定了详细的管理措施。但由于技术和历史原因，信息系统在建设之初并没有采用严格的账号安全性检查机制，造成目前在各信息系统中存在大量弱口令，成为安全隐患。国网公司目前共有信息系统100多套，虽然可以通过对系统进行升级来弥补缺

陷，但由于国网公司的大部分的系统需要7×24 h不间断运行，所以急切需要以旁路的形式对多套系统进行检查和实时监测，不影响系统的运行状态。

2 研究方案

由于不能影响系统的稳定运行，只能采用旁路的检查方案，即通过读取数据库中的用户账号，来判断是否为弱口令。数据库中的账号信息为加密信息，故需要还原为明文，才能判断。目前的还原技术有两类，一类是暴力破解，一类是查表法。但暴力破解法需要海量运算时间，而查表法需要海量存储空间。

2003年，Oechslin 提出了彩虹表密码分析算法，彩虹表对前两种方法进行时空折中，将所需时间和空间控制在可接受范围内，使得破解Hash函数加密密码成为现实。但是，由于受CPU计算能力限制，彩虹表算法针对多比特数密文的分析时间和空间依旧难以满足实际需求。近年来，由于GPU性能的快速提高及其并行执行线程特性，特别是CUDA的广泛应用，使得将GPU应用于彩虹表构建和查找成为了研究的热点。因此，本文研究量化了彩虹表构造查找过程与时间空间的关系，建立了参数模型，并将分析结果应用于GPU架构设计、查找彩虹表过程，成功实现了基于GPU的彩虹表弱口令旁路检查工具。该工具提高了密码破解效率，克服了传统方法在时空上的局限性，增加了彩虹表可破解的密码长度，并为实际的密码解析系统参数选择提供了理论依据。

关键词/Keywords

彩虹表 ·
GPU ·
弱口令 ·
旁路 ·

3 彩虹表

3.1 内存时间权衡Hellman表

彩虹表源自Hellman提出的内存时间权衡(time-memory trade-off) Hellman表,其核心思想是将明文计算得到的Hash值由一个还原函数(R函数)映射到明文空间,以这样一次运算为单位,共做K(K≥1)次运算后,得到最终值,只储存初始值和最终值,从而节省了1/K的存储空间,而K值即称为链长。通过查找索引,可以获得该Hash链中的所有明文。若干个彩虹链组成一个表单,所有的表单集合在一个文件里面,构成Hash链表。

原始的内存时间权衡方法存在明显的缺陷,在生成Hash链的过程中可能出现碰撞,即不同的链中得到相同的K值,产生了合并,降低了破解效率。原因是R运算和Hash运算并非一一映射,而是大空间(明文空间)到小空间(Hash值空间)的映射。表越大,即链的数目越多,出现链碰撞的可能性越大。

3.2 彩虹表

针对Hellman方法的不足,Oechslin提出一种新的链结构:使用一系列的还原函数,在链的不同位置使用不同的还原函数R,这样只有在同一个位置出现冲突的链才会合并,虽没有完全消除合并,但已将冲突时出现合并的概率降到了1/t。虽然只是一个简单的改进,其效果却是显著的。

1) 合并的数目大大减少,并且合并的链必有相同的终点,因此能很容易地侦测出合并的出现,从而进行剔除,使单张表内不存在合并。

2) 不会出现环,同一条链上即使出现冲突,但冲突点之后的还原函数必定是不一样的,因此不会出现环。

3) 因为合并数目减少,同时不用做环侦测,故破解时间大幅缩短,提高了破解的效率。

4) 链长是固定的,这便于破解的进行,更利于硬件的实现。Oechslin把这种链结构叫作彩虹链,生成的表就称为彩虹表(Rainbow Table)。

彩虹表参数分析从下列方面:

1) 彩虹表法通过3个参数的选取来进行内存与时间的权衡:链的长度t、每张表中链的数目m及表的数目l。所有参数的选取首先要达到希望的

破解成功率。

$$P_{\text{success}} = 1 - \prod_{i=1}^l (1 - \frac{m_i}{N})^l \tag{1}$$

式中,m₁=m,m_{n+1}=N(1-e^{-m_n/N}),N是输入集总数。确定参数后估算内存及时间。

2) 占用内存:M=lmm₀,m₀是存储一条链的起点与终点所需的空间,目前用于存储一个点的大小为8 B,因此m₀=16 B。

3) 预计算时间:T_{generat}=lmtt₀,t₀是链上得到一个节点的运算时间,在个人电脑上操作时这个时间约为Hash时间的3倍。

4) 最长破解时间:T=ltt₀,最长破解时间其实是破解失败的情况下花费的时间,每张表中都进行t次匹配。

4 弱口令检查工具设计

4.1 CUDA

目前针对GPU的研究已经被广泛应用,GPU已经作为一种并行计算结构被应用于各行业的大规模并行计算中,统一计算设备架构(Compute Unified Device Architecture, CUDA)是用于GPU计算的开发环境,它是一个全新的软硬件架构,可以将GPU视为一个并行数据计算的设备,对所进行的计算进行分配和管理。由于GPU的特点是处理密集型数据和并行数据计算,因此CUDA非常适合需要大规模并行计算的领域。目前CUDA被广泛的应用在图形动画、科学计算、地质、生物、物理模拟等领域。

4.2 并行GPU设计

为了提高计算效率和工具性能,本文采用分布式GPU来构造和查找彩虹表,并结合前述分析选取参数,设计实现密码破解系统。如下图所示,本工具采用的GPU是NVIDIA GTX 980,其具有2 048个流处理器,每个流处理器都可以单独进行运算。本工具包括密文分析模块、GPU组处理模块和彩虹表模块。密文分析模块通过模式匹配技术对加密协议数据进行预处理分析和冗余过滤,选择性分发给10个GPU,每个GPU模块独立并行工作,彩虹表模块通过GPU完成彩虹表初始化、构造和查找。成功解析的消息密钥会由GPU组处理模块自动上传到前段密文分析模块形成历史密钥。

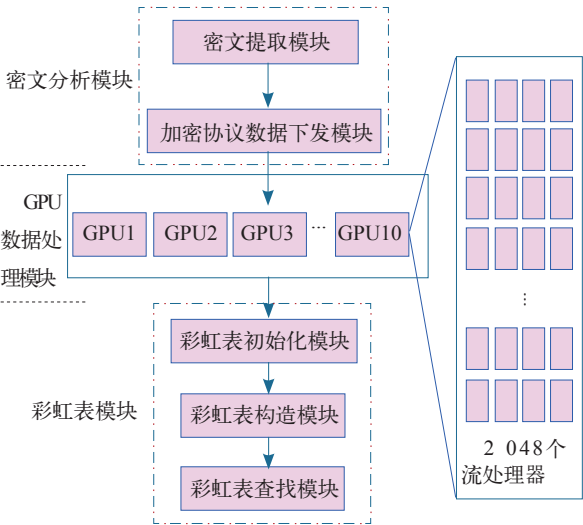


图 基于GPU的弱口令检查工具架构

GPU加速可以解决链长过长导致查找时间长的的问题，引入10个并行GPU线程，每个GPU线程加速2 048倍，总共加速20 480倍。GPU的使用可以解决彩虹表构造时间瓶颈和查找时间瓶颈，要进一步解决彩虹表占用空间瓶颈，系统参数的选取至关重要。由上述知，扩大彩虹表链长来缩短彩虹表链数量，可以大大减少彩虹表对磁盘的占用。

5 实验结果

本文最终选取的链长 $K=800\ 000$ ，本工具对常用的密码位数的数字+字母密文进行解密，其所占空间如表1所示。

表1 彩虹表占用空间

样本	链长为800 000所占空间/GB
1~8位数字+字母	0.054
1~9位数字+字母	1.944
1~10位数字+字母	69.984
1~11位数字+字母	2 519.424
1~12位数字+字母	90 699.264

工具性能如表2所示。

目前商用密码破译成功率为80%左右，而本工具对于11位以下的数字+密码的破译成功率已经超过94%，成功率大大提升，且11位以下的数字+密码破译时间仅为毫秒级，完全可应用于大规模密码查找和监测。而12位以上的密码+数字，目前

表2 工具性能

样本	破解时间/s	破解成功率(%)
1~8位数字+字母	0.004 8	98.5
1~9位数字+字母	0.004 9	97.6
1~10位数字+字母	0.009 3	94.2
1~11位数字+字母	0.165 2	94
1~12位数字+字母	5.781 0	89.3

商用的破译时间大都需十几天甚至几十天，而本工具所需时间仅为秒级，获得了突破性的提升。

6 结束语

本文针对系统数量庞大、无法对系统进行停机升级的部门和企业，设计并实现了一种基于GPU的彩虹表弱口令旁路检查工具，该工具利用GPU的CUDA结构，利用超长链长的彩虹表，并行计算查找彩虹表，在保证占用空间可以接受的前提下，大大加速的密码破译速度，从而可以使得彩虹表应用于旁路的密码实施监测，提高了工作效率，提升了企业信息数据的安全性。

参考文献

[1] 荣凯, 邱卫东, 李萍. 基于彩虹表的Hash攻击研究[J]. 信息安全与通信保密, 2011(4): 76-79.

[2] 金铨, 谷大武, 赵建杰. 彩虹表密码分析算法的图形处理器优化设计与实现[J]. 上海交通大学学报, 2011, 45(7): 1006-1010.

[3] 李昕, 曹天杰, 米国粹. 基于分布式环境下的彩虹表密码攻击[J]. 计算机应用与软件, 2011, 28(2): 290-293.

[4] Philippe Oechslin. Making a faster cryptanalytic time-memory trade-off[C]//Advances in Cryptology: CRYPTO'03, California, USA: Springer-Verlag, 2003: 617-630.

[5] 钱睿智. 基于GPU加速的MD5哈希函数加密算法研究[D]. 武汉:华中科技大学, 2009.

作者简介

马思硕 (1986-), 男, 工程师, 主要研究方向为电力信息通信、信息系统运维和信息安全。

(收稿日期: 2015-12-10) **EA**
(编辑: 鞠佳)