一种基于样本的模拟口令集生成算法

韩伟力 袁 琅 李思斯 王晓阳

(复旦大学软件学院 上海 201203)

(上海市数据科学重点实验室 上海 201203)

摘 要 大规模的用户口令集因可用于评估口令猜测算法的效率、检测现有用户口令保护机制的缺陷等,而广受系统安全研究领域的重视.然而,尽管可以通过一些渠道,譬如网站口令泄露、用户自愿征集或者个别网站出于研究目的的共享等,获取真实的大规模用户明文口令对当前研究人员来说仍然非常困难.为应对上述问题,该文提出了一种基于样本的模拟口令集生成算法(Sample Perturbation Based Password Generation,SPPG).该算法利用较容易获得的小规模真实口令样本,通过学习生成概率模型,并产生大规模用户口令集合.为评估这一算法的效能,该文提出了一组模拟口令集质量的检测指标,包括真实口令覆盖率、Zipf 分布拟合度等.最后,论文对比了 SPPG 算法与当前常见的用户口令猜测概率模型,包括概率上下文无关文法和多种马尔科夫模型,在生成用户口令集上的效能差异.结果显示,SPPG 算法产生的模拟口令集在各指标下都有更好的表现.平均地,在真实口令覆盖率上,相对上下文无关文法和四阶马尔科夫模型分别提高了 9.58%和 72.79%,相对三阶和一阶马尔科夫模型分别提高了 10.34 倍和 13.41 倍,并且 Zipf 分布的拟合度保持在 0.9 及以上的水平.同时,其口令结构分布和特殊模式的使用也更符合真实用户生成口令的情况.

关键词 口令安全;口令集生成;样本;概率上下文无关文法;马尔科夫模型

中图法分类号 TP391 **DOI 号** 10.11897/SP.J.1016.2017.01151

An Efficient Algorithm to Generate Password Sets Based on Samples

HAN Wei-Li YUAN Lang LI Si-Si WANG Xiao-Yang

(*Software School*, *Fudan University*, *Shanghai* 201203)

(*Shanghai Key Laboratory of Data Science*, *Shanghai* 201203)

Abstract Large-scale real user password sets are well regarded important in the field of system security research, due to their usages in evaluating the efficacy of the algorithms that guess passwords, and detecting defects of existing password protection mechanisms, etc. At present, some ways of capturing real passwords are available for researchers, such as accidental or malicious passwords disclosure, voluntary user contributions, or sharing by voluntary websites for research purposes. However, there are some serious limitations involved in collecting user password sets in the above ways. For example, password sets that are captured from passwords disclosure may have been tampered, and therefore their quality cannot be guaranteed. What's more, types of these password sets are limited. As a result, it is still difficult for researches to have access to the large-scale clear-text user passwords in a systematic manner. Motivated to resolve the above issue, this paper presents a sample perturbation based password generation algorithm (SPPG for short). The algorithm is to use a given small-scale real user password sample as a training set to generate a probability model that can then be used to provide large-scale password

收稿日期:2016-10-31;在线出版日期:2017-03-01.本课题得到上海市科委"创新行动计划项目"(16DZ1100200)、国家自然科学基金(61572136,61370080)资助.**韩伟力**,男,1975 年生,博士,教授,中国计算机学会(CCF)高级会员,主要研究方向为访问控制、网络身份安全、大数据.E-mail:wlhan@fudan.edu.cn.**袁 琅**,女,1993 年生,硕士研究生,主要研究方向为信息安全.**李思斯**,女,1995 年生,硕士研究生,主要研究方向为信息安全.**王晓阳**,男,1955 年生,博士,教授,中国计算机学会(CCF)高级会员,主要研究领域为数据分析与安全.

sets. The small-scale sample is relatively easier to obtain. With the purpose of improving the authenticity of the simulation password sets, the SPPG algorithm is designed based on the idea of sample perturbation. On the one hand, the algorithm takes advantage of the Probabilistic Context-Free Grammar to parse the sample, and then generates passwords that have the same structures with passwords in the sample. On the other hand, it also utilizes rules that are frequently used for users to deform their passwords, and then generates passwords that are similar to passwords in the sample. To evaluate the efficacy of the SPPG algorithm, this paper presents a set of criteria to evaluate the quality of the simulation password sets. These criteria include the coverage rate of the real passwords, the goodness of fit to the Zipf distribution, the similarity of password structure distributions and the proportion of special patterns. In the end, this paper compares the efficacy of the SPPG algorithm with the popular probability models of password guessing, including the Probabilistic Context-Free Grammar and several variants of the Markov models. In the experiment, small-scale samples are randomly selected from real user password sets, and then are used by different models to generate the simulation password sets. The experiment results show that the SPPG algorithm has better performances. On average, the coverage of the real passwords is improved by 9.58% and 72.79% respectively compared with the Probabilistic Context-Free Grammar and the 4-order Markov model. And the coverage of the real passwords is 10.34 times more than the 3-order Markov model and 13.41 times more than the 1-order Markov model. Besides, the goodness of fit to the Zipf distribution remains at a high level that is no less than 0.9. As for the password structure distribution and the proportion of special patterns, simulation password sets generated by the SPPG algorithm are also shown to be more similar to the real password sets compared with simulation password sets generated by the other models.

Keywords password security; password set generation; sample; password context-free grammar; Markov model

1 引 言

文本口令是目前最常用的网站用户认证方式之一.它在为用户和网站管理者提供使用便利的同时也伴随着严重的安全威胁.在系统安全研究领域,大规模的用户口令集因可用于评估口令猜测算法的效率、检测现有用户口令保护机制的缺陷等,而成为十分关键的研究材料.当前,真实的用户口令集可通过一些渠道收集,如:部分口令安全研究直接利用公开可获得的大规模泄露口令作为用户口令集进行研究[1-3];也有研究者通过用户自愿征集的方式来收集口令;另外,个别网站还会出于研究目的而共享用户口令[4].然而,后两种渠道获取的口令规模有限,而网上泄露的用户口令又有着其他严重的局限,例如:

(1)当前泄露的用户真实口令,其数据质量不能保证,导致研究结果会随着数据质量的差异而不一致.例如文献[5]指出 Tianya① 和 7k7k② 口令集,由于数据污染而包含大量的相同条目,这样的口令集会给口令研究的结果带来偏差.甚至,迄今为止尚不能判断这些数据集是否还存在其他的污染.

(2)当前泄露的用户真实口令种类有限.当前泄露的大规模用户口令集通常都是基于 Web 的论坛型网站口令集,缺少包括网上银行、企业信息系统等更为敏感的信息系统口令集.这使得研究人员得到的研究结果不能直接影响到这些高度敏感系统的安全防护.

(3)当前泄露的口令集存在时效性问题.随着时间的推移,很多系统采取单向加密机制保护存储的用户口令,使得研究人员通过口令泄露获取用户明文口令变得越来越困难.这导致当前的口令安全研究很多采用的是 2012 年左右泄露的用户口令.

综上,有针对性地获取大规模的用户口令集对

① Tianya. http://help.tianya.cn/about/history/2011/06/02/166666.shtm

② 7k7k. http://www.7k7k.com/html/about.htm

于研究人员来说仍然十分困难.然而无论是真实用户口令集还是高质量的模拟用户口令集,对于评估用户口令猜测算法的效率、评估用户口令强度度量方法的有效性、构建口令字典等都同样十分重要.

为了解决这一问题,本文利用机器学习领域虚拟样本生成的思想,提出了一种利用现有的口令作为小样本来生成能反映目标网站真实用户口令选择的模拟口令集的方法.通过这一方法生成的口令集,由于用户口令的小样本相对可控,因此无论是大规模模拟数据集的质量、用户口令的种类,还是用户口令的时效性都可以得到有效保障.

在机器学习领域,虚拟样本是指在未知样本概率分布函数的情况下,利用研究领域先验知识并结合已有的训练样本,产生待研究问题的样本空间中的部分合理样本[6].在模拟口令集生成的问题中,先验知识表现为现有的口令研究中发现的一些用户口令设置规律,已有的样本即研究者能够获取的小规模口令数据.模拟口令集的目标是尽可能生成属于真实用户口令分布空间中的合理样本,即生成大规模真实的网站用户口令.用户口令的生成方法总体可以归纳为3类:(1)基于特定场景的规则(如长度限制或字符种类限制等)产生具有一定特征的口令;(2)基于字典,再结合一些变形产生口令.当前流行的口令攻击工具John the Ripper① 产生候选口令的字典模式就属于这种情况;(3)建立口令概率模型,训练口令数据,并生成概率模型空间中的其他口令.常用的用户口令模型包括Narayanan和Shmatikov[7]在2005年提出的基于马尔科夫模型(以下简写为"Markov"模型)的口令生成方法和Weir等人[8]在2009年提出的基于概率上下文无关文法(Probabilistic Context-Free Grammar,以下简写为"PCFG")的口令生成方法.在用户口令研究中,口令生成主要用于网站口令猜测攻击.基于概率模型的方法由于相对其他方法具有更好的适应性和扩展性,因此是目前最有效的口令攻击方法之一.然而,这些以攻击为目的而生成的口令在恢复网站原始口令的同时,也包含许多不属于真实用户设置的口令.本文提出的基于样本的模拟口令集生成算法(Sample Perturbation Based Password Generation,以下简写为"SPPG")充分考虑到现有口令生成模型的不足,并在它们的基础上实现了更优的模拟口令生成效能.

本文也提出了一组指标来评估口令集生成算法的有效性.评估的标准主要是所生成的模拟口令集的真实性.本文认为,口令概率模型生成的口令集真实性可以通过模拟口令是否符合真实用户设置口令的习惯来判断.这涉及两个层次的条件:首先,模拟口令集需要符合人类口令集普遍的分布规律.Wang等人对网站用户口令集的整体频率分布进行了研究②,他们指出用户口令集的口令出现频次与频次的排序之间存在Zipf分布的特点.因此,利用Zipf分布的拟合程度来评估模拟口令集应是个良好的指标;其次,模拟口令集需要描述样本对应的网站中用户口令设置习惯,例如该网站中所有口令的长度分布、字符类型分布等.本文综合考虑了这两个条件,并提出从真实口令覆盖率、Zipf分布拟合度、口令结构分布以及特殊模式的使用这4个方面对口令集真实性进行全面的统计分析.结果显示,由SPPG算法生成的模拟口令集相比PCFG模型和一阶或多阶Markov模型具有更高的真实性.

本文的主要贡献包括:

(1)提出了一种高效的基于样本的模拟口令集生成算法SPPG.该算法利用小规模用户真实口令样本,采用基于扰动的方式生成大规模模拟口令数据.SPPG算法借助了上下文无关文法学习生成概率模型,但进一步对该模型空间进行了裁剪并添加了与样本具有更高相关性的口令,增加了模拟口令集的真实性.另外,SPPG算法不必对整个口令空间进行概率排序,这明显提高了模拟口令集生成速度.平均地,PCFG模拟口令集的生成时间约为SPPG的2.43倍;而单机的一阶Markov模拟口令集生成时间约为SPPG的33.64倍.

(2)提出利用多个指标从不同角度对模拟口令集的质量进行综合评估的方法.以中英文网站泄露的大量真实口令数据为实验材料,对比了SPPG算法与现有两种主要口令模型分别生成的模拟口令集的真实性.评估结果显示SPPG的模拟口令集的真实口令覆盖率相对PCFG提高了9.58%,相对一阶Markov提高了13.41倍;在不同样本和模拟规模下Zipf拟合度始终保持在0.9以上,能较好地符合Zipf分布;口令结构分布和包含特殊模式的口令比例都比PCFG和Markov模型更接近真实用户生成口令的情况.

① John the Ripper password cracker. http://www.openwall.com/john/

② Zipf's Law in Passwords. https://eprint.iacr.org/2014/631.pdf 2014

本文第 1 节描述背景和主要的研究内容;第 2 节介绍相关工作,包括两种口令概率模型、虚拟样本生成方法和现有的口令特征研究;第 3 节提出基于小样本的模拟口令集生成算法;第 4 节利用多个评估指标对比几种模型产生的模拟口令集质量;第 5 节是实验要点和其他问题的讨论;最后,第 6 节对本文进行总结和展望.

2 相关工作

2.1 口令概率模型

利用口令概率模型生成口令的过程大致包括训练口令样本,建立概率模型,再输出概率空间中包含的所有可能口令. Ma 等人[9]对口令概率模型设计空间做了详细的总结,他们将口令模型分为基于整个字符串的模型和基于模板的模型. 基于模板的模型将一个用户口令分解为几个片段,然后分别为每个片段计算概率. 典型的代表为 PCFG 模型[8]. 而基于整个字符串的模型则没有分段的概念. 典型代表为 Castelluccia 等人[10]提出的基于整个字符串的 Markov 模型,另外还有 Melicher 等人[11]提出的基于人工神经网络的模型等.

2.1.1 基于上下文无关文法的口令生成

PCFG 是一种基于模板的口令模型. 该方法将口令用数字(D)、字母(L)和特殊字符(S)这 3 种形式表示,且连续相同类型的字符划归到同一片段. 例如口令“password123”表示为 L_8D_3,称为半终端结构. 口令概率的计算分为半终端结构的计算和半终端结构实例化为具体字符串的计算. 例如,口令“password123”的概率计算为

$$P(\text{“password123”})=P(L_8D_3)P(\text{“password”}|L_8)P(\text{“123”}|D_3).$$

概率模型建立好之后,口令生成过程通过优先队列实现概率从高到低的输出.

2.1.2 基于马尔科夫模型的口令生成

马尔科夫模型本身是自然语言处理中的方法,由 Narayanan 等人[7]首次用来描述口令的字符序列分布,从而帮助约减口令猜测空间,并实现快速的字典攻击. 文献[10]基于这一模型计算口令概率,实现了具有适应性的口令强度评估标准. 后来其他研究者[12]也使用 Markov 模型进行猜测攻击实验并验证了该模型的有效性. 文献[9]还对马尔科夫链的多种变形做了详细讨论,包括马尔科夫链不同阶数的选择以及一些标准化处理和平滑处理方法. 一个 n 阶马尔科夫链的计算公式可表示为 $P(x_i|x_{i-1},x_{i-2},\cdots,x_1)=P(x_i|x_{i-1},\cdots,x_{i-n})$. 对口令“$c_1c_2c_3\cdots c_i$”,按一阶马尔科夫模型计算其概率为 $P(\text{“}c_1c_2c_3\cdots c_i\text{”})=P(c_1|c_0)P(c_2|c_1)P(c_3|c_2)\cdots P(c_i|c_{i-1})$,其中 c_0 表示开始字符,而 $P(c_i|c_{i-1})$ 的值从训练集中统计而来. 概率模型建立之后,口令生成过程根据由马尔科夫链形成的树形结构而从根部开始搜索可能的所有口令.

一些研究试图对 PCFG 和 Markov 模型进行改进. 例如,Zou 等人[13]指出 PCFG 模型中基本的高概率口令结构能产生的口令数目较少,因此将这些高概率口令结构按用户划分习惯进行再次划分. Veras 等人[14]提出基于自然语言处理的方法. 其核心思想是在 PCFG 算法的基础上,通过分词和词性标注进一步挖掘字母片段中包含的语义信息. 自然语言处理方法目前的口令猜测攻击效果介于 PCFG 和 Markov 之间[15]. 一些研究[9,16]也用实验对比了 PCFG 和 Markov 模型口令猜测的表现. 通常,PCFG 模型在猜测初期即尝试次数较少的时候猜测效率比 Markov 模型更高;但 PCFG 模型最终的覆盖率比较有限,而 Markov 模型在猜测后期表现更好. Ur 等人[17]指出,由于不同的概率模型在口令猜测的有效性上存在系统性的差别,依赖任何一种单一的模型都难以取得理想的结果. 各模型在使用时,需要仔细地配置并与其他方法进行结合.

本文首次提出了专门用于生成模拟用户口令集的算法. 后文中讨论的 PCFG 和 Markov 模型生成口令集是一种通用的模拟口令集生成途径. 本文用它们的对比实验来展示本文提出算法的效率.

2.2 使用样本生成模拟数据集

在机器学习领域,虚拟样本生成技术已经得到了许多研究. 这些研究按照生成思想大致可以分为 3 类[18]:(1)基于研究领域具体的先验知识构造虚拟样本;(2)基于扰动的思想构造虚拟样本;(3)基于研究领域的分布函数构造虚拟样本. 在口令生成的方法中,基于规则限制和基于字典的方法类似于基于先验知识的构造方法;用概率模型生成候选口令的方法类似于基于分布构造函数的构造方法. 在生成模拟口令集的场景中,简单的基于规则的生成方法要求目标集合具有明显的规则特征;基于字典的口令生成方法适应性和口令空间的覆盖率有限;基于概率模型的方法由于并不能完整地描述小样本中包含的口令分布信息,所以合理性有限. 以 PCFG 和 Markov 模型为例,PCFG 模型重点描述了半终

端结构的计算和半终端结构实例化，而 Markov 模型则更关注相邻字符之间的链式概率关系. 它们都在一定程度上对训练口令进行了解析和概率统计，但都不够完整. 事实上，在先验知识有限且分布构造函数难以确认的情况下，生成基于样本的模拟口令集最适宜采用基于扰动的思想.

文献[19]以建立优良的口令强度标准为目的，对 PCFG 和 Markov 模型的局限性作了初步的分析. 他们指出用户设置口令时倾向于直接对现有的口令进行一些混合规则的变化，而不是产生一个全新的无关的口令. 因此这两种模型刻画的概率空间和实际情况有一定差距. 这些讨论也印证了本文采用基于扰动的思想生成模拟口令集的合理性.

2.3　用户口令特征分析

研究者利用大规模泄露的口令或通过调研收集的真实用户口令，进行了详细的口令特征分析[9,20-22]①. 这些研究使得我们对用户口令设置习惯有了比较全面的了解.

2.3.1　口令频次分布

哈佛大学语言学专家 Zipf 在语料库中发现一条统计型经验规律，称为 Zipf 分布. Zipf 分布的表述为：将单词在语料库中出现的次数由大到小排列，单词频数与单词排序数的常数次幂存在反比关系. 2012 年，文献[23]调研了能否使用 Zipf 分布来描述用户口令集中的口令频率. 实验表明用户口令集中频次与其排序基本符合 Zipf 分布. 2014 年，Wang 等人再次分析了口令集中的这一特性②，研究表明 Zipf 分布完美地存在于用户生成的口令集中. 同时，论文给出了 Zipf 公式：$f_i=C/i^s$，其中 C 和 s 为具体数据集决定的常量，f_i 为口令在集合中出现的频次，i 为口令频次由高到低排序的序号.

2.3.2　口令结构特点

许多研究对用户口令集的结构特征进行了经验分析，分析内容主要包括口令长度分布、口令中 95 个可打印字符的分布、口令中“数字/字母/特殊字符”的字符类型分布等. 例如，文献[20]以部分英文用户和西班牙用户的口令为研究对象，分析了两类用户口令在口令长度、字符类型、首尾字母的频率分布等方面存在的差别. 文献[9]对中英文网站用户的口令长度分布、字符类型分布等特征进行挖掘. 统计结果显示不同网站的用户口令集在结构特征上有一定的差异. Shen 等人[24]也从口令长度、组成和大小写选择等方面对大规模口令集进行了分析. 他们将统计结果与往年的口令特征研究进行了对比，并指出用户口令特征会以某些方式随着时间不断改变.

2.3.3　口令中的语义信息

研究发现语言、地区文化等因素也会影响口令的设置. 文献[21]调研了 4 个公司的口令构造异同并展示了中国文化对口令设置的影响. 例如中文用户的口令中会包含一些与普通话有谐音的数字(“5201314”、“7758520”等). 文献[22]利用可视化方法调研了口令中的语义信息，并且发现口令中包含明显的日期格式. 文献[5]对中英文用户口令中的特殊模式作了更详细的调研，最后发现口令中各种语义信息，包括汉语拼音、英文单词、诗词、行政区等. Li 等人[25]针对大规模中英文口令集进行了系统性分析，发现中文用户口令特有的性质. Han 等人① 分析了口令重用情况，并指出了口令重用带来的严重安全威胁. Liu 等人[26]分析了 2011 年中国部分网站泄露的大量真实口令中的结构和语义规律，并进一步发现这些口令与口令之间、口令与用户其他信息之间一些明显的站内和跨站关联规则.

3　口令集生成算法设计

3.1　算法概述

为了在生成大规模口令的同时保持口令集的真实性，本算法采用基于扰动的样本生成思想并结合 PCFG 模型来构造模拟口令集. 这意味着口令集主要围绕已有的样本进行扩展，即基于已有的元素对口令空间进行填充. 填充的方法主要是按照用户设置密码通常采用对已有密码进行变形而不是重新创建完全无关的密码的思想，基于训练集里的口令进行一些变形，从而生成更可能符合用户实际使用的口令. 最终，模拟口令集中的口令就由与样本完全相同的口令以及基于样本的相似口令组成.

基于扰动的口令生成不仅可以提高模拟口令的真实性，还可以实现比单纯的 PCFG 和 Markov 等概率模型更快速的模拟口令生成. 算法中 PCFG 模型的使用，一方面是利用它来控制模拟口令集口令结构分布的合理性，另一方面也扩大了模拟口令集的生成空间. 尽管 Markov 模型在口令猜测方面也有很好的表现，并且相对 PCFG 模型可以产生更大

① Han Wei-Li, Li Zhi-Gong, Ni Min-Yue, et al. Shadow attacks based on password reuses: A quantitative empirical view. IEEE Transactions on Dependable and Secure Computing, 2016, DOI: 10.1109/TDSC.2016.2568187

② Zipf's Law in Passwords. https://eprint.iacr.org/2014/631.pdf 2014

的猜测空间,但通常 PCFG 模型能更好地描述原始口令集的特征.因此,本算法主要借助 PCFG 模型对口令结构进行划分.

3.2 算法详细描述

如算法 1 所示,本文提出的基于样本的模拟口令集生成算法主要包括两步:

(1)样本的训练.

步骤 1 表示根据 PCFG 的方法对样本建立概率模型.首先将口令转化为以数字(D)、小写字母(L)、大写字母(U)和特殊字符(S)的形式表示的字符类型,再统计每一种子模板中不同长度的结构片段对应的具体口令子字符串概率.

(2)模拟口令的生成.

步骤 2～18 表示算法会循环地为 *SampleSet* 中每个口令生成 X 个相关的模拟口令.其中,$X=N/M$,N 为模拟口令集的目标口令数,M 为样本集的口令总数,X 是向下取整的结果.步骤 3～17 具体地根据样本口令来生成模拟口令.在文献[3]的用户口令设置习惯调研中,用户在设置新口令时按照与已有口令的相似程度可以分为三种情况:使用已有口令、设置相似的口令和设置全新的口令.本算法也利用这一规律生成三类与样本口令具有不同相似程度的模拟口令,它们分别是与样本口令完全相同的口令、与样本口令相似的口令以及与样本口令具有相同结构的任意口令.首先,在步骤 3～6 中生成与 p 完全相同的模拟口令 p',并加入模拟口令集 *SimulationSet*.相同口令的比例为模拟口令总数的 1/10;剩余部分的模拟口令由步骤 7～17 来生成.步骤 9 随机生成和口令 p 在 PCFG 模型空间中具有相同结构的新口令 p'.为了提高生成真实口令的可能性,在步骤 10 判断口令 p' 在 PCFG 模型中的概率值是否小于 $1/N$.若小于 $1/N$ 说明按照 PCFG 模型的估算,p' 在模拟口令集中出现频率小于 1,算法在步骤 11 对 p' 进行重新生成.如果二次生成的口令 p' 概率仍然小于 $1/N$,则在步骤 13 中用 *Transformation* 方法将 p' 取为与 p 相似的口令,*Transformation* 方法在后文有详细的介绍.在以上的步骤中,随机产生与口令 p 结构相同的口令,是利用 PCFG 这种比较符合常规的口令结构划分方式对口令结构分布进行控制.若直接利用 PCFG 空间随机生成新口令,只能提高模拟口令生成速度,而无法提高 PCFG 模型的模拟口令真实性.因此步骤 10～15 通过重新生成的方式使得新口令 p' 集中于 PCFG 概率空间中的高概率部分,另一部分则通过口令变形的方式生成.

考虑到步骤 2～18 生成的是样本集合整数倍的模拟口令,不一定等于模拟口令集目标总数 N.因此剩余小部分的模拟口令 *SimulationSet2* 需要再通过 *getSimulationSet*(即步骤 2～18)来生成.其中,作为基准的样本口令 *SampleSet2* 从 *SampleSet* 中随机采样,*SampleSet2* 的总数 $M_2=N_2/X$.步骤 22～24 将最后一部分模拟口令加入模拟口令集 *SimulationSet* 中.最终,算法返回 *SimulationSet*(步骤 25).

算法 1. 基于样本的模拟口令生成算法.

输入:已知的样本集合 *SampleSet*
一种口令结构划分方法 PCFG
模拟口令集目标口令数 N
中间结果:不同类型和长度的口令片段及其比例 *SegProbs*
输出:模拟口令集 *SimulationSet*

```
1.  SegProbs←PCFG(SampleSet)
2.  FOR each p in SampleSet DO
3.    X1←X/10>1? X/10:1
4.  FOR i=0; i<X1; i++ DO
5.      SimulationSet.add(p')
6.    END FOR
7.    X2=X-X1
8.    FOR i=0; i<X2; i++ DO
9.      p'←SameStruct(SegProbs, p)
10.     IF PCFG(p')<1/N THEN
11.       p'←SameStruct(SegProbs,p)
12.       IF PCFG(p')<1/N THEN
13.         p'←Transformation(p)
14.       END IF
15.     END IF
16.     SimulationSet.add(p')
17.   END FOR
18. END FOR
19. N2←N-M*X
20. SampleSet2←PasswordSampling(SampleSet,N2)
21. SimulationSet2←getSimulationSet(SampleSet2,N2)
22. FOR each p in SimulationSet2 DO
23.   SimulationSet.add(p)
24. END FOR
25. RETURN SimulationSet
```

算法 2 展示了基于变形的相似口令生成方法 *Transformation* 的具体内容.与样本口令相似的口令是通过对样本口令进行常见的规则变换而得到.一些研究[3,19]调查了广受用户欢迎的一些针对已有

口令的变换规则.不同的研究因为调研的规模大小、目标人群的差异等原因可能会对这些规则的具体使用情况给出不一致的结论.但总体而言这些变换规则被使用的频率基本类似.在实际的用户口令设置中,可能用到的相似口令变换规则不胜枚举.本算法选取了在文献[3,19]的调研中,用户使用比例最高的十条变换规则的交集部分.注意,"反转"(例如,将口令"abc123"变为"321cba")和"插入网站特定的信息"(例如,向口令中插入网站的名字)虽然也属于上述交集部分,但在本文的算法中暂不考虑.这是由于,为了便于记忆,用户选择反转的口令通常带有一定的特征,这些特征的多样性使得算法难以迅速且准确地过滤出这类口令;而插入网站特定的信息时,可用于插入的信息也是形式多样.因此,为了保证算法生成的扩展口令的真实性以及算法生成模拟口令的时间效率,我们最终针对交集部分里的六条规则进行相应的变换.算法 2 首先在步骤 3 中随机选定一条当前口令变换要使用的规则,然后相应地在步骤 5、7、9、11、13 或 15 中实现变换.以下是这几种变换的实现细节.

步骤 4～5 对应删除变换.如果口令由两种及以上的字符类型组成,算法会对口令中出现个数最少的那类字符 C 进行删除.为了增加相似口令的多样性,被删除的字符可以是口令中包含的所有的 C 类字符,也可以是由别的字符类型分隔开的一段 C 类字符.另外,一些用户习惯设置新口令时在现有口令基础上添加一个字符.因此,删除变换也可以是删除口令中的一个字符.例如,样本口令"123password123"通过删除变换可以得到相似口令"password123"、"password"或"123password12".考虑到许多用户设置新口令时会选择对现有口令进行添加部分字符的变换.同时,进入算法 2 的口令为在 PCFG 模型中概率较小的口令.这类口令往往具有长度较长或结构较为复杂的特征.因此进行删除部分字符的处理很可能得到与样本口令相似且符合用户口令设置习惯的新口令.注意,如果当前场景中明确规定了最短口令长度,那么当删除变换执行后的口令长度小于最小口令长度,则删除不成功.

步骤 6～7 对应字母大小写变换.这也是最常见的用户口令变换之一.大小写变换一般为口令的首字母变换,同时也可能有整个字母片段的大小写变换(若字母片段为全小写则将它转为全大写形式;若字母片段为全大写,则将其转为全小写形式;若有的大写有的小写,则转为全大写或全小写形式).

步骤 8～9 对应 Leet[①] 变换.主要对常见的几种形似的字符做替换,包括 a 与@的互换、s 与 $ 的互换、o 与 0 的互换、i 与 1 的互换、e 与 3 的互换和 t 与 7 的互换.例如将从样本口令"password"变换到"p@$$word";步骤 10～11 对应子字符串位置的变换.算法以特殊符号作为分隔符,将子字符串按照字符类型分类.若除去特殊符号后口令不是单一的字符类型,则将首尾的子字符串进行位置交换.例如从"gzwz0204"到"0204gzwz";步骤 12～13 对应连续字符的变换.口令中字符的连续可能有两种形式.一种是字符串的 ASCII 值连续,一种是字符串在标准键盘中同一行位置里的连续.连续字符的变换是将字符串按照相同的规律添加一个字符或删除一个字符或整个口令字符串替换为另一个同类型的字符串.例如从样本口令"12345678"变换到"123456789"、"1234567"或"abcdefgh";步骤 14～15 表示若口令本身具有回文或重复的特征,则取口令的一半作为相似口令,例如从样本口令"password-password"到相似口令"password".

最后,算法 2 里最外层的循环(步骤 1～2)表示如果当前的口令无法按照随机选中的某变换规则进行扩展,例如口令"vxfd5w5x"因为不包含 Leet 变换相关的字符,无法进行 Leet 变换,则重新选择其他规则.

算法 2.　基于规则的相似口令生成 *Transformation*.

输入:口令 p

输出:p 经过变形后的相似口令 p'

1. $matchRule \leftarrow$ FALSE
2. WHILE !$matchRule$
3. $ruleType \leftarrow RuleTypeSampling()$
4. IF $ruleType=1$ THEN
5. $p' \leftarrow Delete(p)$
6. ELSE IF $ruleType=2$ THEN
7. $p' \leftarrow CaseConversion(p)$
8. ELSE IF $ruleType=3$ THEN
9. $p' \leftarrow Leet(p)$
10. ELSE IF $ruleType=4$ THEN
11. $p' \leftarrow SubstringMovement(p)$
12. ELSE IF $ruleType=5$ THEN
13. $p' \leftarrow SequenceTransformation(p)$
14. ELSE IF $ruleType=6$ THEN

① Leet. https://simple.wikipedia.org/wiki/Leet

15. $p' \leftarrow Half(p)$
16. END IF
17. IF p'!=NULL
18. $matchRule \leftarrow$ TRUE
19. END IF
20. END WHILE
21. RETURN p'

4 评 估

4.1 评估指标

通过样本生成的具有较高真实性的模拟口令集,一方面应当符合一般用户口令集的口令频率分布规律,另一方面应当能反应该样本对应的真实用户口令设置特点.本文使用以下 4 个方面的指标对生成的模拟口令集质量进行分析:

(1)口令覆盖率.即模拟口令集里的口令属于网站真实口令的比例.这一指标直接反映了口令模型还原原始口令库的能力.

(2)Zipf 分布拟合优度.假设一个口令集包含了 n 个不同的口令,其 Zipf 分布由以 10 为底的双对数关系表示,有回归方程:$\log f_i = \log C - s * \log i$.其中,$i$ 为各口令按照出现的频次从高到低排序的序号,f_i 为第 i 个口令出现的频次.统计学中,回归直线与各观测点的接近程度称为回归直线对数据的拟合优度.拟合程度由决定系数 R^2[①] 来度量.R^2 表征了因变量的变异中有多少百分比可由自变量来解释,其计算公式为

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}.$$

其中,SS_{res} 称为残差平方和,SS_{tot} 为总平方和;y_i 为因变量观察值,在 Zipf 分布中即 $\log f_i$ 的观察值;$\hat{y}_i$ 为回归公式拟合的值,$\bar{y}$ 为观察值的平均数.R^2 的值域为[0,1],R^2 越接近 1 表示自变量对因变量的解释程度越高,相关方程式参考价值越高.

(3)口令结构分布.将口令字符串进行分解,可以发现不同口令集合中口令包含的结构也呈现一定的统计特征[5,20-21].本文选择两个基本的口令结构特征进行分析:口令长度分布和字符类型分布.字符类型是指将 95 个可打印字符分为数字(0~9)、小写字母(a~z)、大写字母(A~Z)和特殊字符 4 类.这 4 种基本字符类型可形成 15 种不同的组合方式.

(4)特殊模式的使用.为了便于记忆,用户设置口令时倾向于遵循一定的结构或语义模式.模拟口令集也应该体现这一规则,即包含特殊模式的口令在口令集中也应该占据一定比例.口令中可能包含各种形式的特殊模式,我们选取常见的一些结构模式和语义模式进行统计.其中,结构模式包括单一字符(口令中只包含一种单一字符,例如"111111")、两种单一字符拼接(例如"!!!!!@@@@@")、顺序字符(口令为 ASCII 值连续上升或下降的一串字符序列,例如"abcdefgh"、"123456")、顺序字符叠加(即由叠加的顺序字符组成,例如"aabbccdd")和键盘模式(符合标准键盘上的位置分布规律,例如属于键盘中的同一行字符"asdfgh"、z 字形分布"qawsxd"、蛇形分布"zxcfgh"等);语义模式包括常见中英文单词(例如"iloveyou")和日期格式.其中,日期格式通常代表着用户的生日、纪念日或重要节日,主要有六位短日期和八位长日期两类.一般年月日之间可以无间隔,也可以有"-"、"."、"/"这 3 种分隔符.例如用 Y 表示年,M 表示月,D 表示日,YYYYMMDD 的八位长日期也可能有 YYYY-MM-DD、YYYY.MM.DD 或 YYYY/MM/DD 的形式.

4.2 评估中使用的数据集

本文使用CSDN[②]、嘟嘟牛[③]、Rockyou[④]和Yahoo[⑤]这 4 个网站泄露的大量真实口令作为实验数据.CSDN 是中国最大的 IT 社区和服务平台,在 2011 年底被黑客曝光了六百多万个账号及明文口令.嘟嘟牛(以下用 Duduniu 表示)是中国一家主要为网吧提供管理软件平台的商业网站,同样在 2011 年泄露约一千五百万个用户口令.Rockyou 是一家美国游戏社交网站,2009 年遭受了泄露事故,导致约三千两百万纯文本口令泄露.Yahoo 是在 2012 年被黑客组织 DD3Ds Company 利用 SQL 注入攻击获取了约四十四万的用户账户登录详情.这些泄露的口令库可以从网上公开获取,我们去除用户个人信息,只收集其中的用户口令进行研究.另外,我们对这些口令做了基本的清理,去除可能包含干扰信息的口令,最终只保留由 95 个可打印 ASCII 字符组成的口令,且长度范围限定在 4 到 40 之间.数据集详细信息见表 1.

① 决定系数 Coefficient of determination. https://en.wikipedia.org/wiki/Coefficient_of_determination
② CSDN. http://www.csdn.net/
③ 嘟嘟牛. https://www.duduniu.cn/
④ Rockyou. http://rockyou.com/
⑤ Yahoo. http://www.yahoo.com/

表 1　数据集的基本情况(数量(U)表示口令集里不相同的口令的数量)

网站	原始数量	清理后的数量	数量(U)	语言
CSDN	6428629	6420426	4030918	中文
Duduniu	15131833	15049451	9491329	中文
Rockyou	32603048	32575770	14325709	英文
Yahoo	442774	442287	342205	英文

4.3　模拟口令集的评估

本节通过实验对比本文提出的 SPPG 算法与 PCFG 及马尔科夫模型生成的模拟口令集质量. 为了更加准确地刻画用户口令使用情况,我们采用了文献[9]和文献[16]中对 PCFG 的改进实验方案. 一方面,将 PCFG 方法中半终端结构的字母类型进一步划分为大写字母类型(U)和小写字母类型(L);另一方面,对大写或小写字母类型的半终端到具体口令片段的频率计算,不再使用外部字典,改为利用训练集来生成相应的字典. 由于马尔科夫模型中马尔科夫链的阶可以有一阶或任意多阶的形式,我们选择了其中最基本的一阶马尔科夫模型以及通常在口令猜测中表现较好的三阶和四阶马尔科夫模型.

实验的主要流程包括:获取原始口令库样本、用 SPPG 算法和两种概率模型分别训练口令样本并生成指定数目的模拟口令集,最后按照 4.1 节中的指标评估各模拟口令集的质量. 这些模拟口令集在后文中分别表示为 SPPG 模拟口令集、PCFG 模拟口令集、一阶 Markov 模拟口令集、三阶 Markov 模拟口令集和四阶 Markov 模拟口令集.

实验中,样本从原始口令库中随机抽取,样本规模为原始口令库的十分之一. 三种方法训练样本后,分别生成口令数目等于样本中口令数目 10 倍、100 倍、1000 倍(即原始口令库规模的 1 倍、10 倍、100 倍)的模拟口令集. 注意,Markov 模型在一些研究中[9,17]被证明在口令猜测到约十亿级的次数以上时往往比其他模型具有更高的猜测命中率. 然而在进行口令集的模拟时,一方面对原始口令库命中率的提高并不能代表整个模拟口令集真实性的提高,另一方面 Rockyou 数据集的 100 倍已经达到十亿的规模,且在现实中几乎不可能有用户口令数据集会超过这个规模. 因此本实验的模拟口令集实验数据规模设计到 100 倍较为合适.

4.3.1　口令覆盖率

图 1 显示了五个模型分别生成和原始口令库相同大小的模拟口令集时,成功落入原始口令库的口令比例. 可以看到,对 CSDN、Duduniu、Rockyou 和 Yahoo,五个模型的覆盖率呈现一致的规律. 以 CSDN 数据集为例,覆盖率最低的是一阶 Markov 模型产生的模拟口令集,其值为 0.02;尽管随着 Markov 阶数的增加覆盖率有所增加(四阶 Markov 的覆盖率为 0.15),但是仍然明显低于 PCFG 和 SPPG 模拟口令集. 对比 PCFG 和 SPPG 模拟口令集,SPPG 模拟口令集相对原始口令库的口令覆盖率(0.33)略高于 PCFG 模拟口令集(0.30). 根据四个网站数据集的平均结果,SPPG 模拟口令集相对 PCFG 模拟口令集提高了 9.58%;相对四阶 Markov 提高了 72.79%;相对三阶 Markov 提高了 10.34 倍;相对一阶 Markov 提高了 13.41 倍. 总体而言,SPPG 模拟口令集具有最高的覆盖率.

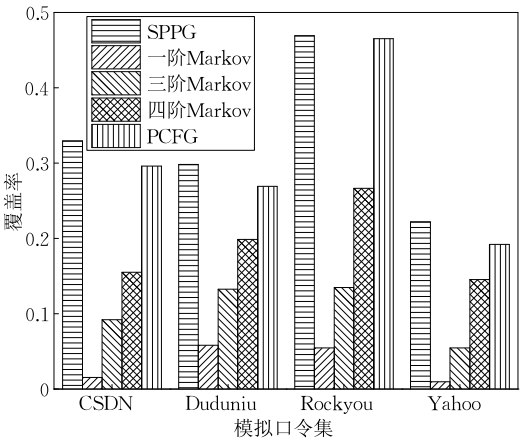

图 1　模拟口令集覆盖率(覆盖率等于模拟口令集里包含原始口令库中口令的数目,再除以原始口令库的口令总数. 图中每组数据集从左到右依次对应 SPPG、一阶 Markov、三阶 Markov、四阶 Markov 和 PCFG 生成的模拟口令集)

4.3.2　Zipf 分布拟合度

根据 4.1 节中的公式计算五个模型生成的不同规模的模拟口令集 Zipf 拟合度,得到如图 2 所示的 Zipf 决定系数. 由图 2 可以看到,大部分模拟口令集的 Zipf 决定系数都大于 0.9,说明这些模拟口令集都比较符合 Zipf 分布. 其中,各阶 Markov 模型生成的模拟口令集无论样本的网站来源以及口令集规模的大小如何,Zipf 决定系数都维持在 0.95 以上,表示非常符合 Zipf 分布. SPPG 的模拟口令集 Zipf 决定系数略小于 Markov 模型,但仍然比较稳定,维持在 0.9 左右. 而 PCFG 模拟口令集符合 Zipf 分布的能力最差,每项 Zipf 决定系数都略低于另外 4 个模型生成的模拟口令集. 其中,PCFG 模拟口令集生成的 CSDN 原始口令库 1 倍大小的模拟口令集决定系数等于 0.64,偏离 Zipf 分布的程度最大.

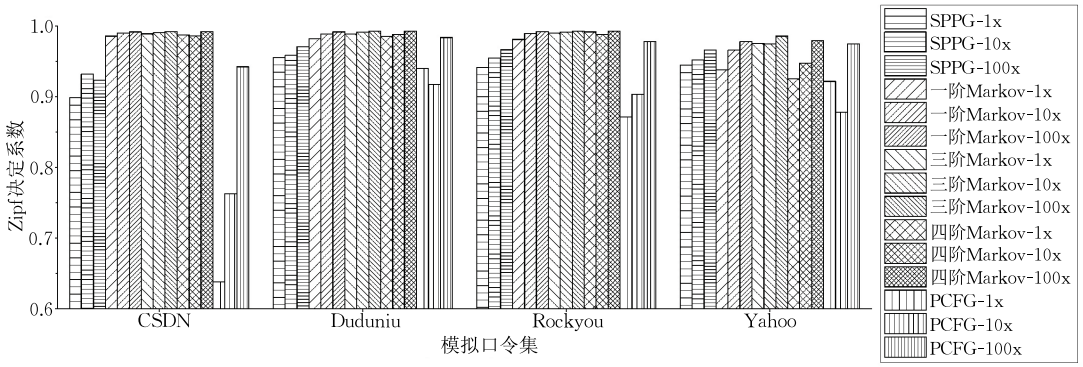

图 2 各模拟口令集 Zipf 决定系数(1 表示完全符合 Zipf 分布,0 表示完全不符合 Zipf 分布.图例中,PCFG-1x 表示用 PCFG 方法生成的规模为原始口令库 1 倍的模拟口令集;模拟口令集 1 倍到 100 倍对应的柱形填充样式依次从稀疏到密集)

4.3.3 口令结构分布

我们分别对 CSDN、Duduniu、Rockyou 和 Yahoo 原始口令库以及基于它们的样本生成的模拟口令集里口令长度和字符类型的分布做了统计.统计结果表明,不同网站的口令结构分布有所差异,这可能受网站口令创建规则、网站账户密码重要性和用户群体差异等各方面因素的影响.

图 3～图 6 分别是 CSDN、Duduniu、Rockyou 和 Yahoo 原始口令库与模拟口令集里口令长度的分布情况.观察各模拟口令集与原始口令库口令长度的相似度,SPPG 模拟口令集和 PCFG 模拟口令集的分布与原始口令集的分布非常相似,在图形中线条几乎重合;而 Markov 模拟口令集的分布偏差较为明显.从四阶 Markov 到一阶 Markov 模型,马尔科夫链的阶数越小,生成的模拟口令集口令长度分布与原始口令库偏差越大.Markov 模拟口令集长度分布偏差主要表现为,模拟口令长度向更短的区间集中,而长度大于 10 的口令比例明显偏小.

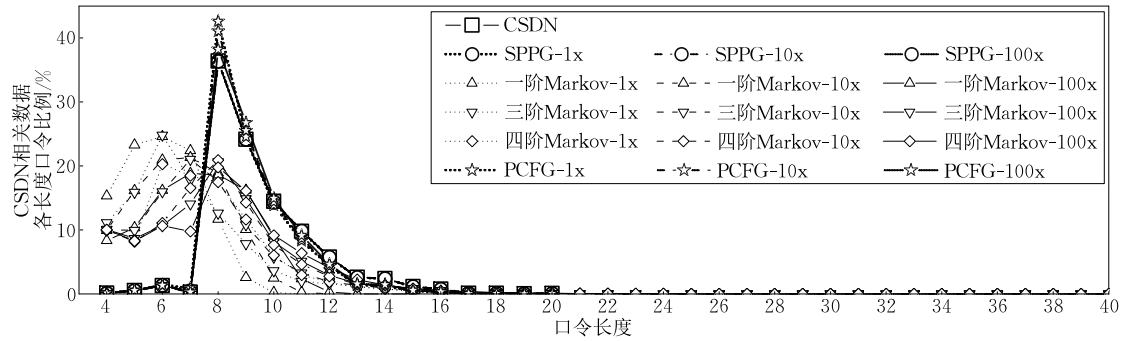

图 3 CSDN 原始口令库以及各模拟口令集里口令长度分布(百分比表示口令数据集中指定长度的口令占整个数据集口令总数的比例.方形标记点加实线线条代表原始口令库;圆形标记点代表 SPPG 模拟口令集,上三角、下三角和菱形标记点分别代表一阶 Markov、三阶 Markov 和四阶 Markov 模拟口令集,五角星标记点代表 PCFG 模拟口令集;短点线、虚线和实线分别代表模拟口令集为原始口令集 1 倍、10 倍和 100 倍的规模大小.后面类似的图形采用一样的标记规则)

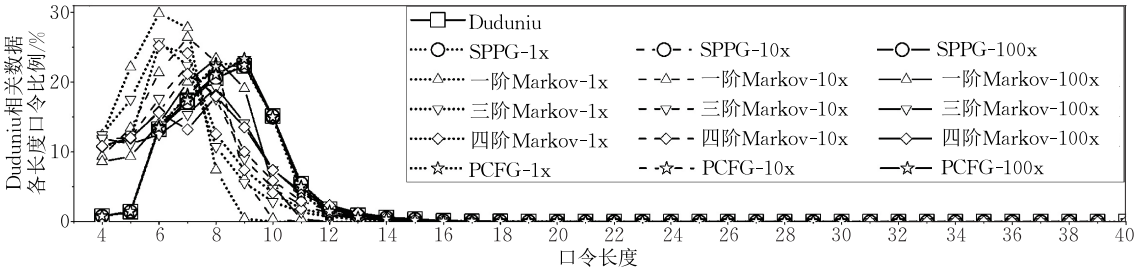

图 4 Duduniu 原始口令库以及各模拟口令集里口令长度分布(SPPG 模拟口令集和 PCFG 模拟口令集的分布与原始口令集的分布非常接近;而 Markov 模拟口令集的分布偏差较为明显,长度集中于更短的区间)

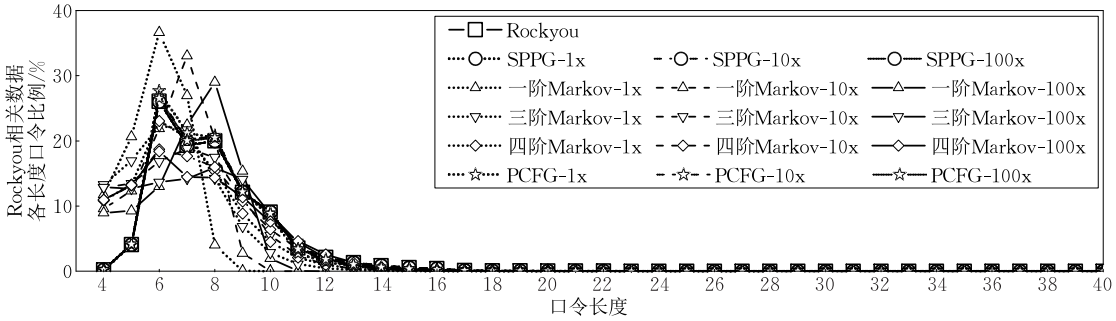

图 5　Rockyou 原始口令库以及各模拟口令集里口令长度分布(SPPG 模拟口令集和 PCFG 模拟口令集的分布与原始口令集的分布非常接近;而 Markov 模拟口令集的分布偏差较为明显,长度集中在小于等于 10 的区间)

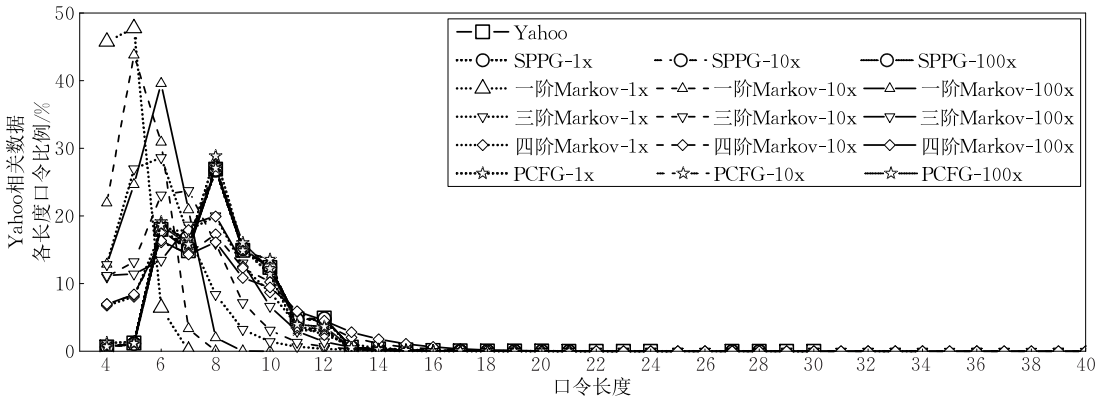

图 6　Yahoo 原始口令库以及各模拟口令集里口令长度分布(SPPG 模拟口令集和 PCFG 模拟口令集的分布与原始口令集的分布非常接近;而 Markov 模拟口令集的分布偏差较为明显,长度集中于更短的区间)

以图 3 中模拟口令集规模为原始口令集 1 倍的数据为例进行具体观察:CSDN 原始口令库中长度为 8 的口令最多,比例为 36.40%;SPPG 和 PCFG 模拟口令集里也是长度为 8 的口令最多,比例分别为 36.48%和 42.59%;一阶 Markov、三阶 Markov 和四阶 Markov 模拟口令集里则是长度为 6 的口令最多(比例分别为 24.58%、24.80%和 20.29%),而对应的长度为 8 的口令比例仅 11.65%、12.64%和 17.46%.对于长度大于 10 的口令,原始口令库、SPPG 模拟口令集和 PCFG 模拟口令集里的比例分别为 22.75%、22.59%和 16.03%;而在三阶 Markov 和四阶 Markov 模拟口令集里的比例分别为 6.31%和 7.86%,在一阶 Markov 模拟口令集里比例不足 0.01%.

对比长度分布,五个模型的模拟口令集在字符类型分布方面相对原始口令库的偏差并不明显.对 CSDN 和 Duduniu 相关数据集,模拟口令集都是以“纯数字”或“数字+小写”类型为主,“纯小写”类型的比例次之;对 Rockyou 和 Yahoo 相关数据集,模拟口令集都是以“纯小写”或“数字+小写”类型为主,“纯数字”类型的比例次之.表 2 显示了各数据集原始口令以及规模为原始口令集 1 倍的模拟口令集里占比最多的这几种字符类型的具体比例.观察表 2,仍然可以发现 SPPG 模拟口令集的字符类型比例与原始口令库的比例最为接近.

事实上,口令的字符类型和口令长度可能存在一定关联,不同长度口令的字符类型往往有显著区别.口令概率模型可能改变特定长度口令的字符类型组合情况.因此,我们进一步将口令长度按照[4,7]、[8,10]、[11,40]三个区间进行分类,并统计在指定长度区间内口令的字符类型分布.注意,由于这几个口令集里字符类型 90%以上都集中在“纯数字”、“纯小写”、“数字+小写”、“数字+大写”等几种类型.包含大写字母和特殊字符的口令比例太小,在图表中难以观察.因此,这里将大写和小写统一归为字母类型,同时只显示“纯数字”、“纯字母”和“数字+字母”的分布情况.最后得到图 7 至图 10 的结果,由此可以看到:

(1)对原始口令集.3 种长度区间的口令集相对整个口令库的字符类型分布略有变化.其中,长度为 4~7 的短口令集里,由单一类型字符组成的口令比例明显增多(中文网站中比例最高的为“纯数字”类型,英文网站中比例最高的为“纯小写”类型);长度为 8~10 的口令集里,字符类型分布与整体分布最

表 2 CSDN、Duduniu、Rockyou 和 Yahoo 原始口令库以及规模为原始口令库 1 倍的模拟口令集里，口令字符组合类型的分布情况 （单位：%）

网站	字符组合类型	原始口令库	SPPG-1x	一阶 Markov-1x	三阶 Markov-1x	四阶 Markov-1x	PCFG-1x
CSDN	纯数字	**45.08**	**45.21**	51.75	49.83	48.64	43.67
	数字＋小写	**35.62**	**31.75**	25.31	26.37	28.62	41.98
	纯小写	**11.67**	**11.72**	18.15	16.42	15.31	11.42
	其他类型	7.63	11.32	4.80	7.38	7.43	2.92
Duduniu	数字＋小写	**51.20**	**49.00**	42.57	45.98	45.51	57.00
	纯数字	**32.75**	**32.93**	37.02	34.45	35.16	30.58
	纯小写	**11.01**	**11.02**	16.54	14.11	14.08	10.37
	其他类型	5.04	7.05	3.87	5.47	5.25	2.06
Rockyou	纯小写	**41.70**	**41.71**	43.21	43.63	43.97	40.74
	数字＋小写	**33.20**	**32.59**	34.11	31.23	30.22	38.17
	纯数字	**15.94**	**15.96**	14.32	15.53	16.14	15.53
	其他类型	9.17	9.74	8.37	9.60	9.67	5.56
Yahoo	数字＋小写	**50.66**	**45.53**	43.92	43.23	47.40	61.63
	纯小写	**33.05**	**33.13**	41.67	37.02	34.08	31.45
	纯数字	**5.86**	**5.77**	9.93	7.88	7.43	5.69
	其他类型	10.43	15.57	4.48	11.88	11.09	1.23

注：表中“纯数字”表示口令仅由数字组成，“纯小写”表示口令仅由小写字母组成，“数字＋小写”表示口令为数字和小写字母的混合.“其他类型”表示以下类型的总和，包括“纯大写”、“纯符号”、“数字＋大写”等两种字符的组合、“数字＋小写＋大写”等 3 种字符的组合以及“数字＋小写＋大写＋符号”.总体而言，各模拟口令集字符组合类型的分布与原始口令库差别不大，但 SPPG 模拟口令集的分布与原始口令库最为相似.

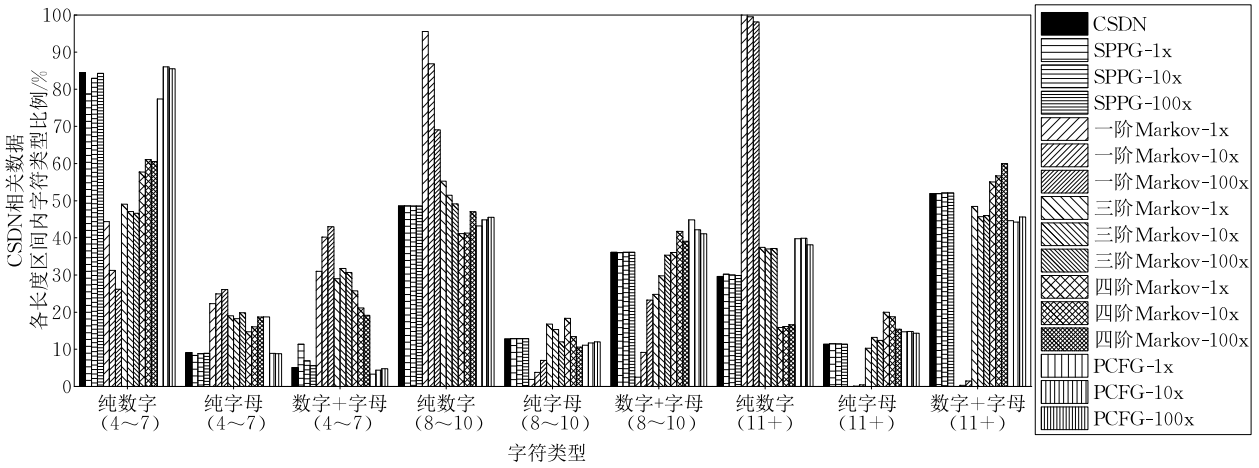

图 7 CSDN 原始口令数据集和模拟口令集按照口令长度分类后，每个区间里的 3 种主要口令字符类型分布（例如，“纯数字(4～7)”表示在原始口令集或模拟口令集里长度范围属于 4 到 7 的所有口令中，由纯数字组成的口令比例。横坐标每组数据对应的柱形从左到右分别为 CSDN 原始口令集、SPPG 1 倍到 100 倍大小的模拟口令集、一阶 Markov 1 倍到 100 倍大小的模拟口令集、三阶 Markov 1 倍到 100 倍大小的模拟口令集、四阶 Markov 1 倍到 100 倍大小的模拟口令集以及 PCFG 1 倍到 100 倍大小的模拟口令集）

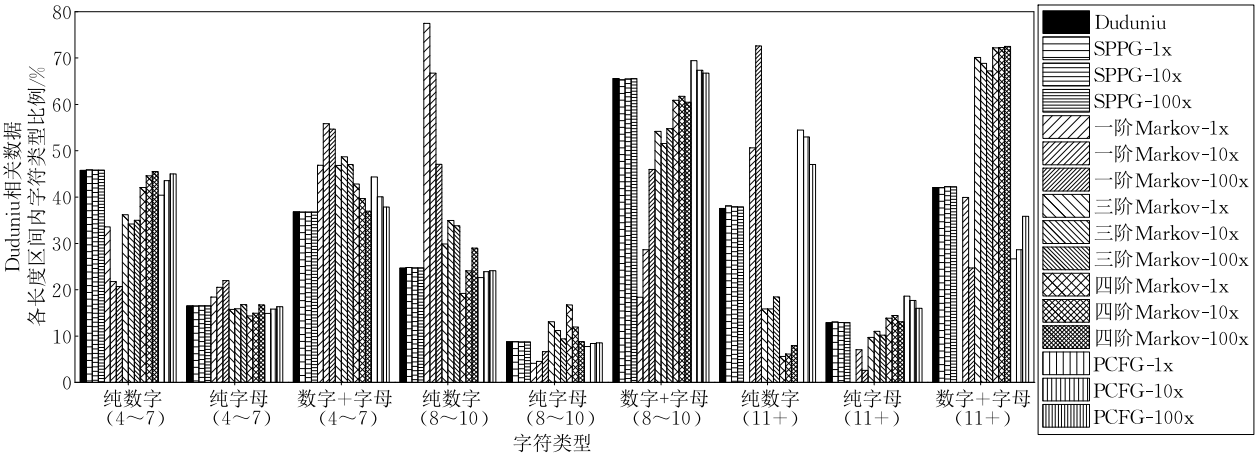

图 8 Duduniu 原始口令数据集和模拟口令集按照口令长度分类后，每个区间里的 3 种主要口令字符类型分布

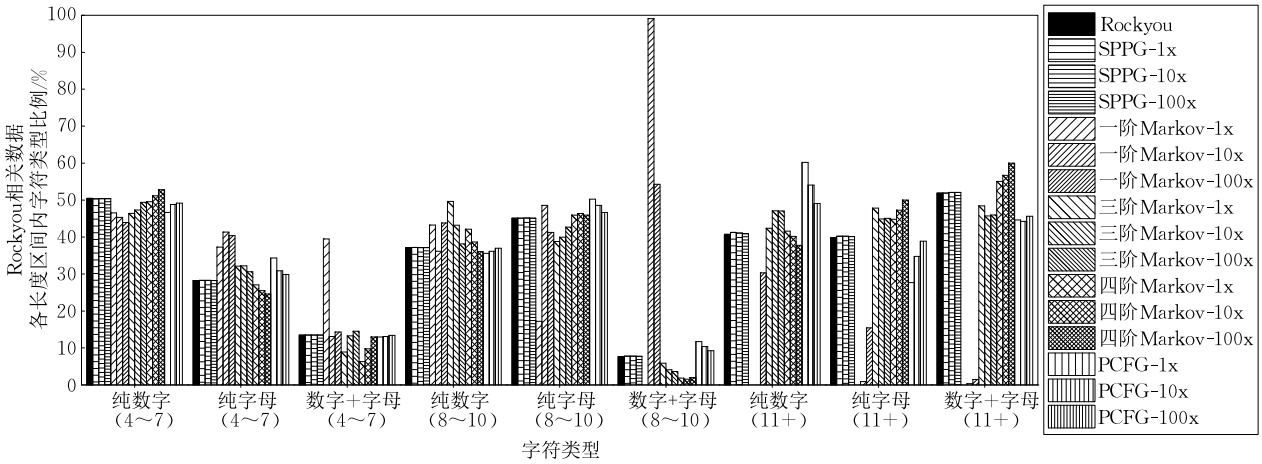

图 9　Rockyou 原始口令数据集和模拟口令集按照口令长度分类后，每个区间里的 3 种主要口令字符类型分布

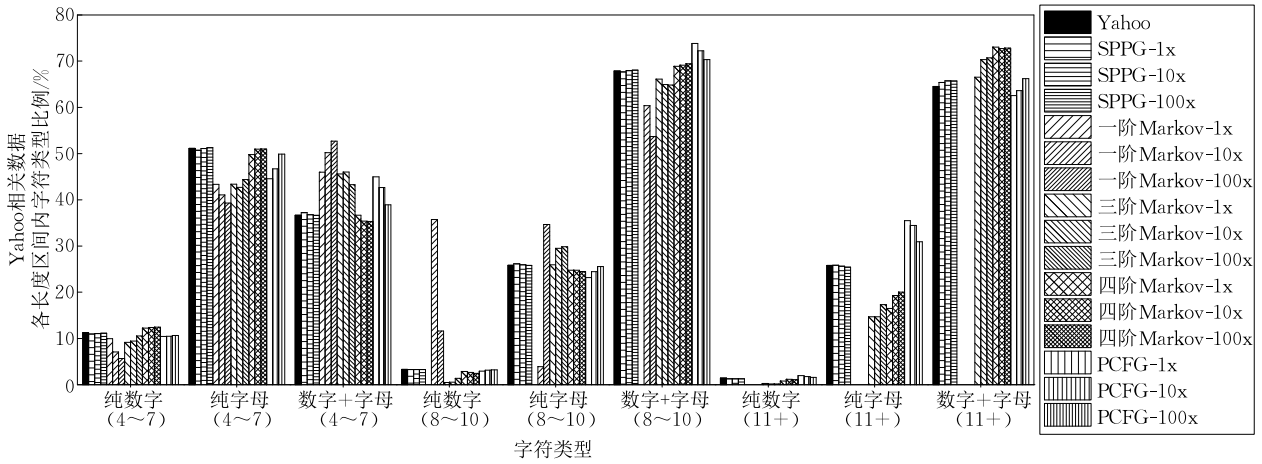

图 10　Yahoo 原始口令数据集和模拟口令集按照口令长度分类后，每个区间里的 3 种主要口令字符类型分布

相似；长度为 11 及以上的口令集里，混合字符类型“数字＋字母”的口令比例增多.

(2) 模拟口令集在特定长度区间内的字符组合类型分布出现了比较明显的差异. 其中，SPPG 模拟口令集在任何数据集和口令规模下都和原始口令集的字符类型分布几乎保持一致，说明在划分长度区间的情况下，SPPG 模拟口令集也能很好地模拟出原始口令库的字符类型分布；PCFG 模拟口令集在特定口令长度区间内的字符类型分布也和原始口令集的比较接近，但是略有差异. 以图 7 中的数据为例，1 倍大小的 PCFG 模拟口令集在长度为 4 到 7 的范围内的口令相对原始口令集而言，纯字母类型偏多而纯数字类型偏少. 随着模拟口令集的规模的增大，这种偏差越来越小，100 倍的 PCFG 模拟口令集和原始口令集的字符类型分布类似；Markov 模拟口令集分口令长度观察时，与原始口令库的字符类型分布差别明显，其中一阶 Markov 模拟口令集的偏差最为明显. 例如在图 7 中长度为 4～7 的口令集里，CSDN 原始口令集的纯数字比例高达 84.46%，而在一阶 Markov 中对应的值只有 44.36%；长度为 8～10 的口令集里，原始口令集的纯数字比例为 48.59%，而在一阶 Markov 中对应的值增长到 95.54%. 另外，一阶和多阶 Markov 模型随着模拟口令集 1 倍、10 倍到 100 倍的规模变化，其字符类型分布也明显变化，不及 SPPG 模拟口令集和 PCFG 模型口令集的字符类型分布稳定.

综上，在模拟原始口令集的口令结构方面，SPPG 模拟口令集、PCFG 模拟口令集、四阶 Markov 模拟口令集、三阶 Markov 模拟口令集和一阶 Markov 模拟口令集的有效性依次降低.

4.3.4　特殊模式的使用

用户设置的口令中可能包含多种特殊模式. 本文对结构模式中常见的单一字符、两种单一字符拼接、顺序字符、顺序字符叠加和键盘模式以及语义模式中的中英文和日期的包含情况进行统计分析. 图 11 显示了原始口令库以及模拟口令集使用特殊

模式的口令比例.以 CSDN 规模为原始口令集 1 倍的相关数据为例:CSDN 原始口令库中使用特殊模式的口令比例为 46.53%;SPPG 和 PCFG 模拟口令集里对应的比例分别为 46.84%和 46.06%;一阶 Markov、三阶 Markov 和四阶 Markov 模拟口令集里对应的比例分别为 12.38%、27.28%和 36.02%.从图 11 可以看出,与原始口令库中特殊模式的使用比例最接近的是 PCFG 模拟口令集和 SPPG 模拟口令集,且包含特殊模式的口令比例随着模拟口令集规模的增大基本保持不变.而三种 Markov 模拟口令集里符合特殊结构模式或特殊语义模式的口令比例明显偏低,并且随着 Markov 模型阶数的降低,口令中包含特殊模式的比例越来越低.

本节从四个方面对模拟口令集的真实性进行了评估.在口令覆盖率方面,SPPG 模拟口令集优于 PCFG 模拟口令集和 Markov 模拟口令集.Markov 模型阶数越低,口令覆盖率越低;在 Zipf 分布拟合程度上,Markov 模拟口令集表现最优,SPPG 的决定系数略低于 Markov 模拟口令集,但高于 PCFG 模拟口令集的决定系数.特别地,当 PCFG 模型生成的 CSDN 数据库模拟口令集 Zipf 决定系数明显偏低时(0.64),SPPG 模拟口令集仍然保持在 0.9 的水平;在口令结构分布方面,SPPG 模拟口令集相对其他模拟口令集表现最好;在口令特殊模式的使用上,SPPG 和 PCFG 模型和原始口令集中的比例基本相等,而 Markov 模拟口令集里这类口令的比例明显减少.综上所述,SPPG 模拟口令集在这四个指标上具有最好的综合表现.

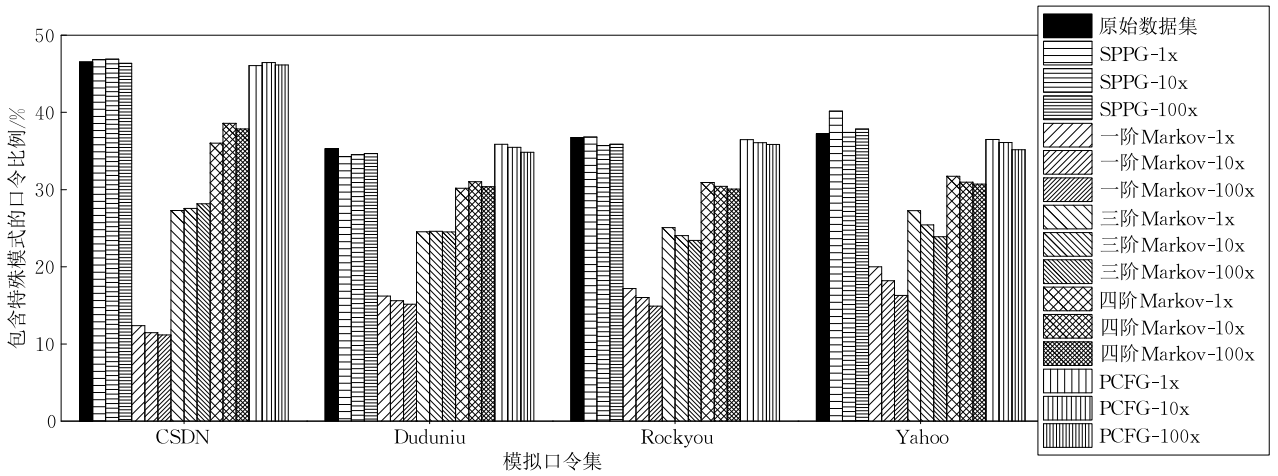

图 11 各原始口令库以及模拟口令集里使用特殊模式的口令比例

5 讨 论

5.1 实验环境配置

本文使用 Java 语言实现模拟口令集的质量分析,实验环境数据如表 3 所示.

表 3 实验环境

属性	内容
CPU	Intel Core i5-2400
内存	14GB
操作系统	Windows 7 旗舰版
开发语言	Java

5.2 模拟口令生成速度

各阶 Markov 模型、PCFG 模型和本文提出的 SPPG 算法生成模拟口令集的速度有较大差异.表 4 是 5 种模型生成不同规模口令集的耗时对比.通常,PCFG 模型的口令生成速度略大于 Markov 模型.而表 4 中 PCFG 模拟口令集的生成速度明显大于 Markov 模拟口令集的生成.这主要是由于,在本文的实验中,PCFG 和 Markov 模型采用了不同的实现方式.考虑到 Markov 模型在生成大规模口令时需要消耗大量的内存并可能造成程序内存不足的情况,Markov 模型按照文献[9]的思路,通过阈值试探的方式来生成指定数目的口令;而 PCFG 模型仍然直接在内存中使用优先队列从高概率到低概率输出候选口令的方式[8].作为对比,本文提出的算法使用了一些随机方法,减少了在整个口令空间搜索和排序的过程.因而在内存消耗和口令生成速度方面有明显的优势.尽管也有研究对 PCFG 和 Markov 模型的实现进行速度优化.例如研究者[27]将 PCFG 猜测过程中具有相同概率的口令片段合并处理,减少比较过程.这些优化实现相对本文算法仍然比较耗时.另外,当 PCFG 和 Markov 模型的口令生成速度随着模拟口令集的规模不断降低时,本文的算法

可以持续保持常量的速度进行.

表 4　各模型生成模拟口令集的耗时对比

算法	倍数	CSDN	Duduniu	Rockyou	Yahoo
SPPG	1	10s 198ms	21s 14ms	36s630ms	889ms
	10	53s 72ms	1min 55s	3min 41s	3s 448ms
	100	5min 53s	14min 20s	28min 47s	22s 573ms
一阶 Markov	1	2min 41s	4min 31s	14min 51s	9s 552ms
	10	24min 34s	1h 17min	2h 35min	1min 44s
	100	4h 34min	13h 34min	23h 16min	18min 1s
三阶 Markov	1	1min 21s	4min 39s	6min 45s	2s 828ms
	10	17min 4s	29min 39s	1h 14min	43s 187ms
	100	3h 23min	9h 53min	19h 9min	7min 30s
四阶 Markov	1	1min 16s	2min 43s	4min 55s	2s 188ms
	10	11min 4s	21min 36s	47min 51s	14s 687ms
	100	2h 41min	4h 48min	7h 14min	1min 50s
PCFG	1	16s 319ms	44s178ms	1min 5s	1s 507ms
	10	2min 38s	4min 26s	8min 30s	15s 741ms
	100	14min 57s	32min 28s	53min 2s	1min 11s

注：表中倍数 1、10 和 100 分别表示生成口令数目等于 CSDN、Duduniu、Rockyou 和 Yahoo 原始数据集中口令数目 1 倍、10 倍和 100 倍的模拟口令集. SPPG、一阶 Markov、三阶 Markov、四阶 Markov 和 PCFG 分别表示使用 SPPG 算法、一到四阶的 Markov 模型以及 PCFG 模型生成模拟集. 表格中 h 表示小时，min 表示分，s 表示秒，ms 表示毫秒.

5.3　小规模真实口令样本的获取

有针对性地获取大规模真实用户口令非常困难，但获取小规模真实用户口令相对容易. 例如，可通过以下几种方式实现：

(1) 收集网上泄露的口令. 随着许多著名网站明文口令泄露事件的发生，目前可以从网上收集到一些国内外的真实用户口令集. 这些口令集的规模从几万、几十万至几千万大小不等，研究者可根据具体的实验需求，用其中的某些口令集作为样本，生成更大规模的模拟口令集.

(2) 进行用户口令调研. 网上泄露的口令种类有限，研究者也可以通过调研来获取特定类别的用户口令. 例如，希望收集全中国普通高等学校两千多万在校学生的常用口令，获取每位学生的口令信息并不现实. 研究者可以选取其中具有代表性的几十个学校，并从这些学校中分别随机调研数千名学生的常用口令，最后形成数十万的口令数据作为样本.

(3) 网站管理员收集本网站的用户口令. 为了掌握本网站用户口令使用情况，以便在未来制定更有效的口令强度评估策略，管理员可主动收集本网站一段时间内注册的用户口令. 这些口令构成了小规模的真实口令样本.

6　结论与展望

本文首次提出了一种基于样本的模拟口令集生成算法 SPPG. 该算法利用较容易获得的小规模用户真实口令样本，产生大规模用户口令集合. 本文进一步选取 4 个指标综合评估了 SPPG 算法以及现有的口令模型生成模拟口令集的质量. 评估结果表明 SPPG 的模拟口令集可以达到比其他模型生成的模拟口令集更高的真实口令覆盖率. 平均地，SPPG 相对 PCFG 模拟口令集口令覆盖率提高了 9.58%；相对四阶 Markov 提高了 72.79%；相对于三阶 Markov 提高了 10.34 倍；相对于一阶 Markov 提高了 13.41 倍；在不同样本和模拟规模下 Zipf 始终保持在 0.9 以上，能较好地符合 Zipf 分布；口令结构分布和包含特殊模式的口令比例都非常接近真实用户口令集中的各种分布和比例. 综上，本文 SPPG 算法生成的模拟口令集具有更高的真实性.

在接下来的工作中，我们将进一步通过模式识别等方法对口令数据的特征属性进行更加深入的挖掘，寻找更精准的用户相似口令特征. 例如在进行相似口令变换时考虑按照口令语义或词性分类[14]进行单词的替换. 此外，我们还将研究调整参数优化模拟口令集生成算法.

参　考　文　献

[1] Bonneau J, Preibusch S, Anderson R. A birthday present every eleven wallets? the security of customer-chosen banking pins//Proceedings of the 16th International Conference on Financial Cryptography and Data Security. Kralendijk, Bonaire, 2012: 25-40

[2] Mazurek M L, Komanduri S, Vidas T, et al. Measuring password guessability for an entire university//Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security. New York, USA, 2013: 173-186

[3] Das A, Bonneau J, Caesar M, et al. The tangled web of password reuse//Proceedings of the 21st Network and Distributed System Security Symposium. San Diego, USA, 2014: 1-15

[4] Bonneau J. The science of guessing: Analyzing an anonymized corpus of 70 million passwords//Proceedings of the IEEE Symposium on Security and Privacy. San Francisco, USA, 2012: 538-552

[5] Han Wei-Li, Li Zhi-Gong, Yuan Lang, Xu Wen-Yuan. Regional patterns and vulnerability analysis of Chinese web passwords. IEEE Transactions on Information Forensics and Security, 2016, 11(2): 258-272

[6] Poggio T, Vetter T. Recognition and structure from one 2D model view: Observation on prototypes, object classes, and symmetries. Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Boston, USA: Technical Report Artificial Intelligence Memo: 1347, 1992

[7] Narayanan A, Shmatikov V. Fast dictionary attacks on passwords using time-space tradeoff//Proceedings of the 12th ACM Conference on Computer and Communications Security. Alexandria, USA, 2005: 364-372

[8] Weir M, Aggarwal S, Medeiros B D, Glodek B. Password cracking using probabilistic context-free grammars//Proceedings of the IEEE Symposium on Security and Privacy. Oakland, USA, 2009: 391-405

[9] Ma J, Yang W N, Luo M, Li N H. A study of probabilistic password models//Proceedings of the IEEE Symposium on Security and Privacy. Oakland, USA, 2014: 689-704

[10] Castelluccia C, Dürmuth M, Perito D. Adaptive password-strength meters from markov models//Proceedings of the 19th Network and Distributed System Security Symposium. San Diego, USA, 2012: 1-14

[11] Melicher W, Ur B, Segreti S M, et al. Fast, lean, and accurate: Modeling password guessability using neural networks//Proceedings of the 25th USENIX Security Symposium. Austin, USA, 2016: 175-191

[12] Dell'Amico M, Michiardi P, Roudier Y. Password strength: An empirical analysis//Proceedings of the 29th Conference on Information Communication. San Diego, USA, 2010: 983-991

[13] Zou Jing, Lin Dong-Dai, Hao Chun-Hui. A password cracking method based on structure division probability. Chinese Journal of Computers, 2014, 37(5): 1206-1215 (in Chinese)
(邹静, 林东岱, 郝春辉. 一种基于结构划分概率的口令攻击方法. 计算机学报, 2014, 37(5): 1206-1215)

[14] Veras R, Collins C, Thorpe J. On the semantic patterns of passwords and their security impact//Proceedings of the 21st Network and Distributed System Security Symposium. San Diego, USA, 2014: 1-16

[15] Wang Ping, Wang Ding, Huang Xin-Yi. Advances in password security. Journal of Computer Research and Development, 2016, 53(10): 2173-2188(in Chinese)
(王平, 汪定, 黄欣沂. 口令安全研究进展. 计算机研究与发展, 2016, 53(10): 2173-2188)

[16] Dell'Amico M, Filppone M. Monte carlo strength evaluation: Fast and reliable password checking//Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. Denver, USA, 2015: 158-169

[17] Ur B, Segreti S M, Bauer L, et al. Measuring real-world accuracies and biases in modeling password guessability//Proceedings of the 24th USENIX Security Symposium. Washington, USA, 2015: 463-481

[18] Yu Xu, Yang Jing, Xie Zhi-Qiang. Research on virtual sample generation technology. Computer Science, 2011, 38(3): 16-19(in Chinese)
(于旭, 杨静, 谢志强. 虚拟样本生成技术研究. 计算机科学, 2011, 38(3): 16-19)

[19] Wang Ding, He De-Biao, Cheng Hai-Bo, Wang Ping. FuzzyPSM: A new password strength method using fuzzy probabilistic context-free grammar//Proceedings of the 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. Toulouse, France, 2016: 1-12

[20] Abbott J, Garcia V M. Password differences based on language and testing of memory recall. International Journals of N&N Global Technology on Information Security, 2015, 2: 1-6

[21] Yang Cheng, Hung Jui-Long, Lin Zhang-Xi. An analysis view on password patterns of Chinese internet users. Nankai Business Review International, 2013, 4(1): 66-77

[22] Veras R, Thorpe J, Collins C. Visualizing semantics in passwords: The role of dates//Proceedings of the 9th International Symposium on Visualization for Cyber Security. New York, USA, 2012: 88-95

[23] Malone D, Maher K. Investigating the distribution of password choices//Proceedings of the 21st International Conference on World Wide Web. New York, USA, 2012: 301-310

[24] Shen Chao, Yu Tian-Wen, Xu Hao-Di, et al. User practice in password security: An empirical study of real-life passwords in the wild. Computers and Security. 2016, 61: 130-141

[25] Li Zhi-Gong, Han Wei-Li, Xu Wen-Yuan. A large-scale empirical analysis of Chinese web passwords//Proceedings of the 23rd Usenix Security Symposium. San Diego, USA, 2014: 559-574

[26] Liu Gong-Shen, Qiu Wei-Dong, Meng Kui, Li Jian-Hua. Password vulnerability assessment and recovery based rules mined from large-scale real data. Chinese Journal of Computers, 2016, 39(3): 454-467(in Chinese)
(刘功申, 邱卫东, 孟魁, 李建华. 基于真实数据挖掘的口令脆弱性评估及恢复. 计算机学报, 2016, 39(3): 454-467)

[27] Komanduri S. Modeling the Adversary to Evaluate Password Strength with Limited Samples[Ph. D. dissertation]. Institute for Software Research, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA, 2016

HAN Wei-Li, born in 1975, Ph. D., professor. His research interests include access control, digital identity, internet identity security, big data.

YUAN Lang, born in 1993, M. S. candidate. Her main research focuses on information security.

LI Si-Si, born in 1995, M. S. candidate. Her main research focuses on information security.

WANG Xiao-Yang, born in 1955, Ph. D., professor. His research interests include data analysis and security.

Background

The security of textual passwords is an important topic in the field of system security research. Many efforts have been devoted to password related researches, including password guessing, password strength evaluation, etc. Large-scale user passwords sets are very useful for these researches. However, it is not easy for researchers to have access to large-scale user passwords in a systematic manner. This paper aims to utilize a given small-scale and true sample set that is relatively easier to obtain and generate a large-scale simulation password set.

In the aspect of password cracking, researchers have introduced several practical password guessing algorithms and software, such as PCFG (Probabilistic Context-Free Grammar), Markov models, John the Ripper, etc. However, these methods could generate a large number of invalid passwords that do not belong to a real user password set. This paper proposes a new method that generates simulation passwords by perturbing the sample with the purpose of making simulation passwords sets more similar to the real user passwords. Furthermore, this paper presents four evaluation criteria, including the coverage rate of the real passwords, the goodness of fit of the Zipf distribution, the similarity of password structure distributions and the proportion of special patterns on structures or semantics. A series of experiments show the advantages of the new algorithm compared with PCFG and Markov models.

This paper is supported by the Shanghai Innovation Action Project under Grant No. 16DZ1100200 (Data Trade Supporting Big Data Testbed: Key technologies and Toolkit), which focuses on constructing an infrastructure for big data oriented computing, and the National Natural Science Foundation of China under Grant No. 61572136 (Research for the Mechanism of Mobile Sensing Security and Its Optimization), which studies the security issues caused by sensing abilities of mobile devices, and then proposes a defense mechanism. The two projects are valued for the research of big data and mobile device security.

Our team has been involved in the research of password security for over four years. We published (including online publish) four research papers in USENIX Security 2014, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security (ref. [5], page 5 footnote ①, [25]. We have another paper in this field online published at IEEE TDSC).

The method proposed by this paper will help the big data testbed (Grant No. 16DZ110020) to create simulated data based on a small and true sample, when a test of big data requires massive data to evaluate the efficiency and effectiveness of proposed algorithms/methods, but the real data are absent. In addition, the generated data can be used to study the improved techniques for mobile device security.