北京大学

博士研究生学位论文

题目：口令安全关键问题研究

姓　　名：汪　定

学　　号：1301111295

院　　系：信息科学技术学院

专　　业：计算机科学与技术（软件服务工程）

研究方向：网络信息安全

导　　师：王 平 教授

二〇一七年六月

版权声明

摘 要

身份认证是确保信息系统安全的第一道防线，口令是应用最为广泛的身份认证方法。尽管口令存在众多的安全性和可用性缺陷，大量的新型认证技术陆续被提出，但由于口令具有简单易用、成本低廉、容易更改等特性，在可预见的未来仍将是最主要的认证方法。即使在一些高安全需求环境中，口令也是构造多因子身份认证方案的基石。

虽然口令认证技术在过去 50 年里一直受到计算机安全界的高度关注，但口令学术界没有能够引领工业界，人们面临的口令安全问题越来越严重。产生这一现状的主要原因有三：(1)口令安全研究涉及多学科交叉知识，如密码学、自然语言处理、统计学和社会学；(2)以往研究主要集中于如何设计更安全高效的口令密码协议，缺少对人的因素的关注，而现实中用户往往是最薄弱的环节；(3)以往口令研究成果多局限于实验实证或对现象的简单统计，缺乏从理论层面进行规律性、原则性和系统性的研究。针对以上问题和挑战，本文采用数据驱动的方法，基于可证明安全理论、统计学习理论和自然语言处理技术，从理论的视角对口令安全所涉及的 6 个关键问题进行研究，主要完成了以下工作：

1. **发现口令服从Zipf分布，并给出精确分布函数。**“口令服从什么分布”是身份认证领域最为基础性的问题之一，40 多年来一直悬而未决。通过引入自然语言处理技术和统计计算理论，利用 14 个大规模真实口令集，提出了两个口令分布理论模型：PDF-Zipf 和 CDF-Zipf。其中，PDF-Zipf 模型可较好地刻画高频口令的分布，而CDF-Zipf模型可精确地刻画整个口令分布。实验结果显示，PDF-Zipf 模型的决定系数(R^2)在97%以上，CDF-Zipf 模型得到的理论分布与实际分布的最大累积分布误差在0.48%~4.59%(平均1.84%)。两个模型的结果都表明，人类生成的口令服从 Zipf 分布。进一步揭示该理论的三项重要应用，并提出一个口令分布强度评价标准。
2. **提出一套定向在线口令猜测攻击理论模型。**“在未获取网站服务器口令存储文件的情况下，攻击者如何利用个人信息来加速在线口令猜测”鲜见公开研究。针对这一问题，提出一个定向在线口令猜测攻击框架。该框架包含 7 个理论模型，以刻画 7 种不同现实能力的定向在线猜测攻击者。大规模真实数据测试结果显示，在允许猜测100次的情况下，如果仅知道目标用户的一些常见个人信息，成功率可达20%；如果知道用户在其它网站曾泄露的一个口令，成功率可达73%。这一结果远超出此前人们的预期。
3. **提出一个Honeywords攻击和设计理论体系。** Honeywords 技术是检测口令文件泄露的一种十分有前景的技术，由图灵奖得主 Rivest 和 Juels 在

ACM CCS'13 上首次提出。本文指出，他们给出的4个主要 honeywords 生成方法均存在严重安全缺陷，且此类启发式方法无法简单修补；进一步提出一个 honeywords 攻击理论体系，成功解决“给定攻击能力，攻击者如何进行最优攻击”这一公开问题；反过来，攻击者的最优攻击方法可被用来设计最优 honeywords 生成方法，成功摆脱启发式设计。本文研究使 honeywords 生成方法的设计和评估从艺术走向科学。

4. **设计一个口令强度评价算法。**现有口令强度评价算法隐含的一个基本假设是，用户生成口令时从无到有构造口令。本文通过对442名用户的调查发现，用户在向一个新网站注册口令时，77.4%的用户重用或修改一个现有口令，而仅有14.5%从无到有构造一个全新的口令。基于这一发现，设计了一个基于模糊概率上下文无关文法的口令强度评价算法fuzzyPSM。大规模实验表明，fuzzyPSM优于工业界和学术界现有的 5 个主要算法。
5. **严格证明基于口令的双因子认证协议实现匿名性的本质计算复杂度。**由于智能卡是资源受限设备，学者们尝试各种巧妙的办法，仅基于对称密码技术来设计匿名双因子认证协议。遗憾的是，其中绝大多数陆续被指出存在匿名性缺陷，无法实现用户行为不可跟踪性。本文采用反证法，基于Halevi–Krawczyk(1999)和Impagliazzo–Rudich (1989)这两个经典结果，在抗碰撞Hash函数存在的假设下，成功证明：公钥密码技术是实现匿名性的基本组件。这意味着，数以百计的仅采用对称密码技术的协议都无法实现所宣称的匿名性，无论协议设计如何“巧妙”，证明多么“严格”。
6. **构建一个基于口令的双因子认证协议评价指标体系。**系统研究了当前广泛采用的双因子认证协议评价指标体系各元素间关系，指出并证明三个关键指标间存在本质冲突，无法同时实现。这意味着二十年来，数以百计的双因子认证协议无法实现所宣称的目标。此外，发现一些指标间存在微妙的冗余和歧义。为此，本文构建了一个新的协议评价指标体系，并基于 67 个典型协议的评估结果显示新指标体系的可行性和先进性。评估结果不仅有利于学界更好地理解现有方案，而且揭示了设计实用协议的挑战所在。
7. **设计一个可证明安全的口令认证协议。**基于上述理论成果，设计了一个“口令+智能卡”双因子协议，实现了随机预言机模型下可证明安全性。首次提出将“模糊验证因子”与“Honeywords”技术相结合的思想，使双因子协议可及时检测到攻击者的在线猜测行为，在满足可用性指标的同时，实现超越传统上限的安全性。这一思想为本领域一个公开问题提供了肯定性的答案：能否在实现口令本地安全更新的同时，实现双因子安全性？基于 11 个大规模真实口令集，验证了所提思想的有效性。

关键词：口令认证，Zipf定律，定向在线猜测，匿名性，可证明安全

Research on Key Issues in Password Security

Ding Wang (Computer Science and Technology)
Directed by Prof. Ping Wang

ABSTRACT

Identity authentication is the first line of defense for information systems, and passwords are the most widely used authentication method. Though there are a number of password issues regarding security and usability, and various alternative authentication methods have also been successively proposed, passwords will remain the dominant method in the foreseeable future due to its simplicity to use, low cost to deploy and easiness to change. Even in security-critical applications, passwords constitute the corner-stone of multi-factor authentication schemes.

While password-based authentication has received intensive attention in the past 50 years (since it first appeared in the 1960s), the academic community fails to lead the industry world, and password-related issues are becoming more and more serious. This unsatisfactory situation is attributed to three reasons: (1) Password security research involves multi-disciplinary knowledge such as cryptography, natural language processing (NLP) and computational statistics; (2) Existing research mainly focuses on how to design secure and efficient password-based cryptographic protocols, yet little attention has been paid to the human factors, while users are generally the weakest link in the security chain; (3) Existing research mainly performs empirical experiments or measures simple statistics, lacking an exploration of the underlying theories in a principled and systematical way. To address these challenges, taking the data-driven approach and employing the theories of provable security and computational statistics as well as NLP techniques, we focus on several important theoretical problems and make the following key contributions:

- **Uncover the Zipf's law in passwords and reveal the precise distribution function.** Despite four decades of intensive research, it remains an open question as to what is the underlying distribution of user-generated passwords. This dissertation makes a substantial step forward towards understanding this foundational question. By introducing a number of

NLP techniques and statistic-based computational theories, and based on fourteen large-scale datasets, which consist of 113.3 million real-world passwords, we propose two Zipf-like models to characterize the distribution of passwords: *PDF-Zipf* and *CDF-Zipf*. More specially, our PDF-Zipf model can well fit the popular passwords and obtain a coefficient of determination (R^2) larger than 0.97; Our CDF-Zipf model can well fit the entire password dataset, with the maximum discrepancy of the empirical distribution and the fitted (theoretical) model being 0.48%∼4.59% (avg. 1.84%). Armed with the concrete knowledge of password distribution, we suggest a new metric for measuring the strength of password creation policies. We further show three important applications of our Zipf theory.

- **Propose a suite of mathematical models for targeted online password guessing.** Little attention has been given to the question of "how to online guess a user's password with the least attempts, when the attacker is without the password file but knows some personally identifiable information (PII) about the victim". We propose TarGuess, a framework that systematically characterizes typical targeted guessing scenarios with 7 sound *mathematical* models, each of which is based on varied kinds of data available to an attacker. These models allow us to design novel, efficient and automated guessing algorithms. Extensive experiments on 10 large real-world password datasets show the effectiveness of TarGuess. Particularly, TarGuess I∼IV capture the four most representative attacking scenarios and *within* 100 guesses: (1) With some PII of the victim, TarGuess-I can gain 20% success rates; and (2) With the victim's one sister password and some PII, TarGuess-IV can reach a success rates over 73%. Such high success rates are much greater than expected.
- **Develop a theory system for attacking and generating honeywords.** Proposed by Juels and Rivest at ACM CCS'13, honeywords are decoy passwords associated with each user account, and they contribute a promising approach to detecting password leakage. We first show that the four primary algorithms introduced by Juels and Rivest for honeyword generation are vulnerable to attacks, and such heuristic algorithms cannot be easily amended. Then, we develop a theory system for attacking and generating honeywords: Given the capabilities of the attacker, how *best* to attack a honeyword generation algorithm? How *best* to devise a honeyword

generation algorithm? Our theories put an end to the heuristic design and evaluation of honywords, bringing honeyword research from art to science.

- **Suggest a novel password strength meter.** A basic assumption underlying existing password strength meters (PSM) is that, users build passwords from scratch when registering a new web account. However, our survey based on 442 users reveals that, most users (77.38%) simply retrieve one of their existing passwords from memory and then reuse (or slightly modify) it. This is in vast contrast to the seemingly intuitive yet unrealistic assumption (often implicitly) made in most of the existing PSMs. Based on this new observation, we suggest a new password strength meter, named fuzzyPSM, that employs fuzzy probabilistic context-free grammar (PCFG). Extensive experiments demonstrate that our fuzzyPSM outperforms the existing five leading PSMs from the academic community and the industry world.
- **Formally prove the intrinsic computational complexity for preserving user anonymity in password-based two-factor authentication schemes.** Since smart-cards are typical resource-constrained devices, hundreds of studies have attempted to design two-factor schemes with user anonymity by only using lightweight symmetric-key primitives such as hash functions and block ciphers. Unfortunately, most of these schemes have been found problematic in achieving user anonymity. With the technique of proof by contradiction and on the basis of the work of Halevi-Krawczyk (1999) and Impagliazzo-Rudich (1989), we put forward *a general principle*: Public-key techniques are intrinsically indispensable to construct a two-factor authentication scheme that can provide user anonymity. This indicates that these hundreds of design attempts all fail in vain, no matter how "ingenious" the design is and how "rigorous" the proof is.
- **Build a new evaluation metric for password-based two-factor authentication schemes.** We systematically explore the inherent conflicts and unavoidable trade-offs among the design criteria in the widely used evaluation metric. Our results indicate that, under the current widely accepted adversarial model, three design goals are confronted with an inherent conflict. This indicates that hundreds of existing two-factor authentication schemes cannot achieve all of their claimed goals. In addition, we also find there are some subtle redundancies and ambiguities among these design criteria. To alleviate the situation, we build a new evaluation metric for

measuring password-based two-factor authentication schemes. Finally, we provide a large-scale comparative evaluation of 67 representative two-factor schemes by using our new evaluation set. Our measurement results not only provide the missing measurements and thus present a better understanding of existing schemes, but also highlight the difficulties in designing a practical two-factor scheme.

- **Propose a provably secure password-based authentication protocol.** Based on the above six theories, we present a password-based two-factor authentication scheme that is provably secure in the random oracle model. Particularly, in our two-factor scheme, we for the first time integrate "honeywords", traditionally the purview of system security, with a "fuzzy-verifier", making our scheme hit "two birds": it not only eliminates the long-standing security-usability conflict, but also achieves security guarantees beyond the conventional optimal security bound by enabling the server to timely detect online guessing. This well addresses the seemingly intractable security-usability problem: whether or not there exist secure smart-card-based password authentication schemes and the password-changing phase does not need any interaction with the server? By employing 14 large-scale real-world password datasets, we establish the effectiveness of our novel integration.

KEYWORDS: Password authentication, Zipf's law, Targeted online password guessing, Anonymity, Provable security

目 录

第一章 引言 **1**
1.1 研究背景 1
1.2 研究现状 4
1.2.1 口令安全研究概况 4
1.2.2 用户的脆弱口令行为 6
1.2.3 口令猜测攻击算法 9
1.2.4 口令组成规则 14
1.2.5 口令强度评价方法 16
1.2.6 口令分布及其评价指标 22
1.2.7 基于口令的身份认证协议 25
1.3 有待解决的关键问题 27
1.4 本文工作 29
1.5 论文结构 31

第二章 口令分布函数 **33**
2.1 研究背景 33
2.1.1 研究动机 34
2.1.2 本章贡献 35
2.2 相关工作 36
2.2.1 口令生成策略 36
2.2.2 口令攻击 37
2.3 预备知识 39
2.3.1 口令数据集的介绍 39
2.3.2 用户口令的基本信息 41
2.3.3 线性回归 41
2.3.4 Kolmogorov–Smirnov检验 42
2.4 用户口令中的Zipf定律 43
2.4.1 PDF-Zipf模型 43
2.4.2 PDF-Zipf模型的合理性 46
2.4.3 CDF-Zipf模型 49
2.4.4 模型的对比 51

2.5 口令分布的强度指标 54
2.5.1 我们的指标 54
2.5.2 指标有效性评估 54
2.6 三项启示 56
2.6.1 对基于口令的安全协议的启示 56
2.6.2 对口令生成策略的启示 62
2.6.3 对口令分布强度评价指标的启示 64
2.7 本章小结 67

第三章 定向在线口令猜测模型 69
3.1 研究背景 69
3.1.1 相关研究 71
3.1.2 本章贡献 71
3.2 预备知识 72
3.2.1 个人信息分类 72
3.2.2 安全模型 73
3.3 用户构造口令的行为 74
3.3.1 数据集 74
3.3.2 使用流行口令 75
3.3.3 重用口令 76
3.3.4 利用个人信息构造口令 77
3.4 TarGuess: 一个定向在线猜测框架 79
3.4.1 TarGuess-I 80
3.4.2 TarGuess-II 84
3.4.3 TarGuess-III 87
3.4.4 TarGuess-IV 88
3.4.5 解决其它攻击场景 91
3.5 实验 93
3.5.1 实验方法 93
3.5.2 实验结果 94
3.5.3 进一步实验 97
3.6 本章小结 97

第四章 Honeywords攻击和设计理论体系 99
4.1 研究背景 99
4.1.1 动机和挑战 100

4.1.2 本章贡献 102
4.2 数据集、安全模型和评价指标 103
4.2.1 数据集 103
4.2.2 安全模型 107
4.2.3 评价指标 108
4.3 攻击现有方法 109
4.3.1 Juels-Rivest方法回顾 110
4.3.2 对Juels-Rivest方法的现实攻击 111
4.3.3 进一步评估Juels-Rivest方法 115
4.3.4 Honeyindex的两个主要缺陷 117
4.4 Honeywords攻击理论 118
4.4.1 理论攻击模型 118
4.4.2 其它三种攻击者类型 122
4.4.3 攻击理论的实现 123
4.5 新的honeyword生成方法 127
4.5.1 新方法概述 127
4.5.2 现有口令概率模型对比 127
4.5.3 一系列探索性实验 130
4.6 评估结果 133
4.6.1 参数 k 的扩展性 133
4.6.2 实证实验评估结果 134
4.6.3 人工实验评估结果 135
4.7 本章小结 138

第五章 基于重用行为的口令强度评价方法 139
5.1 研究背景 139
5.1.1 研究动机 141
5.1.2 本章贡献 142
5.2 预备知识 143
5.2.1 安全模型 143
5.2.2 口令强度评价方法 143
5.2.3 两个秩相关检验指标 145
5.3 用户调研 145
5.4 新方法fuzzyPSM 148
5.4.1 对比现有PSM 148

5.4.2 理解PCFG的优越性 149
5.4.3 fuzzyPSM算法 150
5.4.4 实验设置和结果 153
5.5 本章小结 155

第六章 口令认证协议中匿名性的本质计算复杂度 157
6.1 研究背景 157
6.1.1 相关工作 159
6.1.2 研究动机 161
6.1.3 本章贡献 162
6.2 对 Fan 等协议的分析 162
6.2.1 Fan 等协议回顾 162
6.2.2 Fan等协议匿名性缺陷 164
6.3 对 Xue 等协议的分析 165
6.3.1 Xue等协议回顾 165
6.3.2 Xue等协议匿名性缺陷 166
6.3.3 攻击效率分析 167
6.4 匿名双因子认证协议的公钥技术原则 167
6.4.1 公钥技术原则 168
6.4.2 原则的普适性 174
6.5 潜在的实现方案 176
6.5.1 预装载伪IDs池 176
6.5.2 群签名和环签名 176
6.5.3 IND-CCA2公钥加密技术 177
6.6 本章小结 179

第七章 基于口令的双因子认证协议评价指标体系 181
7.1 研究背景 181
7.1.1 研究动机 183
7.1.2 本章贡献 184
7.2 系统架构、攻击者模型和评价标准 185
7.2.1 系统架构 185
7.2.2 攻击者模型 186
7.2.3 Madhusudhan-Mittal 评估标准 188
7.3 对Tsai等协议的分析 189
7.3.1 回顾Tsai等协议 189

7.3.2 对Tsai等协议的分析 191
7.4 对Li协议的分析 195
7.4.1 回顾Li协议 196
7.4.2 对Li协议的分析 197
7.5 “可证明安全”方法的角色 200
7.6 Madhusudhan-Mittal评价标准的不足 201
7.6.1 实现用户匿名的方式 201
7.6.2 指标内涵存在歧义 202
7.6.3 指标间存在冗余 203
7.6.4 指标间存在内在冲突 203
7.7 我们的评价标准 206
7.7.1 具体的12项评价指标 206
7.7.2 评价指标体系有效性的讨论 207
7.8 本章小结 210

第八章 基于口令的双因子认证协议设计与分析 211
8.1 新协议描述 211
8.1.1 注册阶段 211
8.1.2 登录阶段 212
8.1.3 认证阶段 212
8.1.4 口令更新阶段 213
8.2 协议设计的基本原理 214
8.2.1 基本思想 214
8.2.2 “Fuzzy-verifier”有效性分析 215
8.2.3 “Fuzzy-verifier+Honeywords”有效性分析 217
8.2.4 “Fuzzy-verifier+Honeywords”普适性分析 219
8.3 形式化安全分析 219
8.3.1 形式化安全模型 219
8.3.2 安全性证明 223
8.4 性能评估 230
8.5 本章小结 231

第九章 结束语和展望 233
9.1 论文研究成果 233
9.2 未来工作展望 235

参考文献 236

博士期间取得的学术成果 275

致　谢 279

第一章　引言

1.1　研究背景

当今以互联网为代表的信息通信技术日新月异，深刻改变了人们的生产生活方式，网络空间已成为国家安全的“第五疆域”。特别是云计算、物联网、大数据等新型计算技术, 加速了人类信息化进程，使人们的日常生活不断网络化，资产不断数字化。与此同时，各种安全威胁和攻击手段层出不穷，网络空间面临的安全威胁日益严峻。身份认证是保障信息系统安全的第一道防线，在很多信息系统中甚至是唯一的一道防线[1, 2]。基于口令的认证技术伴随着大型机的诞生而诞生，上世纪 70 年代起被广泛用于 IBM 370 大型机的访问控制[3]，避免分时操作系统的时间片被滥用。上世纪90年代互联网进入千家万户以来，互联网服务(如电子邮件、电子商务、社交网络)蓬勃发展，口令成为互联网世界里保护用户信息安全的最主要手段之一。

随着互联网的发展，一方面越来越多的服务需要口令保护，另一方面人类大脑能力有限，只能记忆 5∼7 个口令[4]，导致用户不可避免地使用低信息熵的弱口令[5]，在口令中使用个人相关信息[6]，在多个网站中重用同一口令[7, 8]，带来严重的安全危胁。2004 年，比尔· 盖茨对外宣告口令将消亡，微软公司将使用多因子认证替代纯口令认证[9]。后续一系列学术研究也分别从口令无法抵抗离线猜测攻击[10]、口令过期策略(Password expiration policy)无法保证更新后的口令的不可预测性[11]等方面论证了口令认证技术的不可持续性。与此同时，大量替代口令的认证技术不断被提出，如图形口令[12]，生物认证[13]，行为认证[14]，多因子认证[15] 等等。总的来说，现有身份认证技术基本可分为 4 类：1) 基于用户所知，比如口令、PIN 码等；2) 基于用户所有，比如智能卡、U盾、加密卡等；3) 基于用户所是，包括人体生理特征(如指纹、虹膜、脸型、声音)和用户行为特征(如笔迹、鼠标移动速度、键盘敲击频率等)；4) 双因子(或多因子)认证技术，即前述 3 种基本认证技术的两两(或多个)组合。

出乎意料的是，时至今日，口令的地位在工业界不仅丝毫没有被撼动，并且在越来越多的信息系统应用中得到加强。这一现象吸引了越来越多的学者，开始引起学术界的反思。研究发现，虽然这些替代型方案有的在安全性方面优于口令，有的在可用性方面胜过口令，但几乎都在可部署性(Deployability)方面劣于口令，并且各自存在一些固有的缺陷[16]。比如，基于硬件的认证技术成本高昂，使用不方便；基于生物的认证技术不具有可撤销性，且存在隐私泄露问题[17]。因此，学术界逐渐开始形成一个共识[16, 18, 19]：在可预见的未来，基于口

图 1.1 基于口令的身份认证技术

令的身份认证技术仍将是最主要的用户认证方式。具体来说，在常规安全需求网络环境中，如绝大多数Internet站点，基于口令的单因素认证技术无可替代；在电子商务、电子政务、电子医疗、电子军务等若干高安全需求领域，基于口令的双因子认证技术保持主流地位。

当前关于口令认证理论与技术的研究，主要关注的是如何设计适于各种场合的安全高效的口令认证密钥交换协议(Password authenticated key exchange, PAKE)，即图1.1中用户到服务器间的通信交互部分，对应口令全生命周期9个阶段中的“传输”阶段(见图1.2)。数以百计的PAKE协议先后被提出，知名的如SRP[20]，KOY[21]，和J-PAKE[22]。相对而言，研究图 1.1 两端关于人的环节(即用户和服务器管理员)的成果却较少，然而人却往往是信息安全链路中最薄弱的环节，涉及到口令全生命周期的8个阶段。现实中，绝大多数系统为了可用性，允许用户选择口令，而用户为了减轻负担，常倾向于选择弱口令；在服务器端，管理员需要对威胁进行预防、感知和响应，这就涉及到对各类口令安全策略、机制和方法的合理选择(见图1.2右半部分)，而实践证明这不是一件容易的事。

这种研究力量的错配导致口令安全学术界没有能够引领工业界，工业界的绝大多数现行做法依然沿用上世纪80年代的一些过时指南和严重缺乏依据的“事实标准”[23–25]。这就解释了为什么2015年发生的数以千计的已经确认数据泄露事件中，63%与弱口令、缺省口令、失窃口令相关，并且这一数字呈逐年上升趋势[26, 27]。当前，基于口令的身份认证系统面临的最大安全威胁是口令猜测攻击，包括离线猜测和在线猜测两类。前者需要认证服务器存储的用户帐户口令文件，然后离线在本地进行口令猜测，可尝试的猜测次数仅受攻击者$\mathcal{A}$ 本身计算能力的限制；后者不需要口令文件，只需连网即可，但可尝试的猜测次数往往受服务器安全策略的限制，比如在一天内猜测失败 10 次则锁定帐户。

一方面，数以百计的知名网站的口令文件正在被不断的泄露，给离线猜测攻击者改进算法提供了源源不断的素材。2009 年，社交网站Rockyou遭受攻击，导致3200万用户帐号信息泄露；2011 年，数十家知名国内网站(如CSDN，

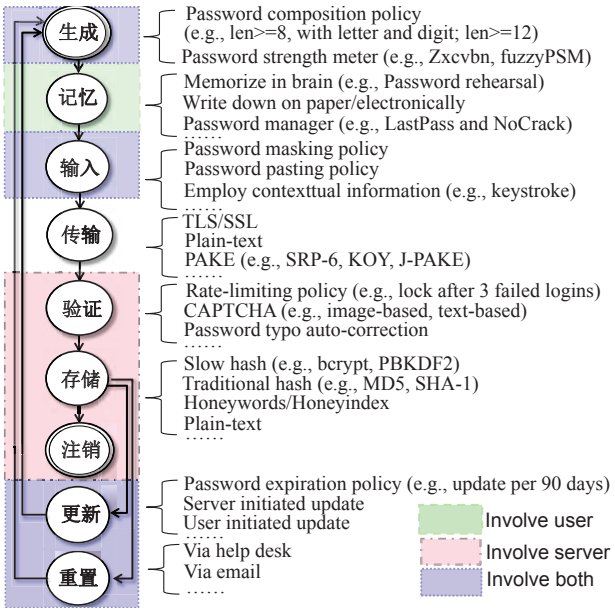


图 1.2　口令全生命周期

天涯，多玩网，人人网)发生用户口令文件泄露[28]；2012 年，最大云存储服务商Dropbox的6800万帐户信息被窃取，直到2016年8月才被发现，然后通知用户更新口令[29]； 2013 年，Yahoo 10 亿用户帐户信息，包括姓名、生日、邮件地址和口令等，被黑客窃取，2016 年12月才发现被攻击[30]；2014年，国内知名手机制造公司小米的800万用户口令帐号信息被泄露[31]；2015 年Twitch 5500万用户口令文件被泄露[32]；2016年，包括Linkedin、Tumblr、Myspace、Twitter、Vk.com、京东等知名网站的数十亿用户口令被泄露[33]。一旦口令文件被攻击者$\mathcal{A}$获得[①]，利用现有泄露的明文口令集和机器学习算法[34]，采用常见的 GPU 并行设备[35]，$\mathcal{A}$ 往往会离线猜测出泄露文件中 80% 以上口令帐户(见[36])。即使采用了强健的Hash函数(如bcrypt) 并加盐运算后存储，也只是一定程度上延缓了$\mathcal{A}$ 恢复口令的速度，并未有效消除离线猜测的危害。口令的离线猜测威胁已经引起学术界和工业界的广泛关注。

另一方面，对在线口令猜测攻击者 $\mathcal{A}$ 来说，由于 $\mathcal{A}$ 没有用户口令文件，允许的猜测次数也受限，学术界和工业界普遍认为在线猜测攻击不是一个大的威胁，容易被防范(“online guessing can be readily addressed”)[37, 38]。然而，如本文第三章将要证实的，攻击者的在线猜测成功率当前被严重低估。现实中，因这种低估，用户和服务商正饱受在线口令猜测的困扰。2013年，攻击者使用一个4万IP的僵尸网络，在线猜测出1698个Github 帐户的口令；2014 年，iCloud 的“找回手机”功能存在在线口令猜测漏洞，导致大量名人照片泄露[39]；2016年，Github再次成为在线口令猜测的受害者，攻击者使用Github 用户在

[①]本文假设网站服务器采取了正确的口令存储措施：使用Hash函数对口令进行加盐存储。

其它网站泄露的口令，成功在线猜测出一大批用户的口令[40]。近几年的新闻里，常见到名人成为定向在线口令猜测的受害者，如Barack Obama[41]、Mark Zuckerbergs[42]和Katy Perry[43]，名誉、财产遭受损失。

从国际趋势看，随着物联网、云计算、大数据技术的兴起，攻击者的口令猜测能力将不断增强，新型攻击手段将不断涌现，口令安全面临的威胁将更为严峻。一些新型口令猜测攻击手段已初现端倪，比如，基于物联网设备的大规模僵尸网络在线口令猜测攻击[44]，云计算增强的离线口令猜测攻击[45]，基于机器学习技术的口令猜测算法[34]。在国内，我国《十三五国家规划》提出了“全面保障重要信息系统安全”的目标[46]，2015 年3月1日起实施的《互联网用户账号名称管理规定》要求“互联网信息服务使用者通过真实身份信息认证后注册账号”[47], 即全面实施网络实名制，这都对保障口令安全提出了更高要求。然而，一方面如前面所提到的，国内外目前对口令缺乏系统性的研究，多关注口令传输安全等个别环节；另一方面，即使少数研究关注了其它环节，却多停留在对现象的简单统计和依赖启发式奇思妙想的层面，缺乏理论性、原则性的研究，一些基础理论问题亟待解决(详见节1.2研究现状部分)。因此，研究口令安全理论及其应用具有重要的学术价值和迫切现实需求，本文工作的开展将为基于口令身份认证问题的根本解决提供重要的理论依据和技术参考。

1.2 研究现状

1.2.1 口令安全研究概况

口令相关研究最早可追溯到1968年Wilkes[48]为分时计算机系统设计的基于口令的身份认证系统。随后几十年里，虽然计算技术几经演进，口令始终是最主要的身份认证方法。不断变化的计算环境，不断出现的新的口令攻击手段，使基于口令的身份认证技术长期成为计算机安全领域一个重要研究方向。目前，众多国际IT 巨头、著名大学和研究机构都积极开展对口令认证的研究，如微软、IBM、麻省理工学院、斯坦福大学、普林斯顿大学、剑桥大学、北京大学、复旦大学、武汉大学、西安交通大学和中科院信工所等。近年来，口令认证相关研究成果频频亮相国际重要学术会议如 ACM CCS、IEEE S&P、USENIX SEC、EUROCRYPT、CRYPTO 等，国内外期刊如ACM/IEEE Trans. 系列、软件学报、计算机学报等也反映了这一研究领域的活跃情况。

目前对口令认证技术的研究分类还没有一个统一的标准，但大体上可分为：(1)口令安全性研究；(2)口令可用性研究。本文主要关注口令全生命周期中安全性方面的研究，包括口令自身安全性(如口令分布和抗猜测攻击的强度)及其辅助系统安全性(如传输阶段的口令认证协议、存储阶段的口令Hash函数和口令文

件泄露检测方法)。关于口令的可用性，读者可关注人机交互方面的刊物，它已成为该领域一个重要研究分支[49]。需指出的是，与口令强度无关的攻击(如社会工程学[50]、口令恶意捕获软件[51])也不是本文关注点。

如节1.1所指出的，口令安全性研究投入极不平衡。由于口令传输阶段不涉及人的因素，研究的问题主要是基于口令的身份认证协议，问题容易刻画，研究目标明确，可以采用密码学工具进行数学建模。1981年Laport的基于口令身份认证协议设计工作[52]吸引了大批学者跟踪，仅2000年以前就有数以百计的协议被提出(如[53–56])。2000年以后，随着互联网技术的发展，各种网络应用环境下的口令传输安全问题也已经得到充分的研究(如三方PAKE[57]、two-server PAKE[58]、multi-server PAKE[59])，甚至最复杂的基于口令的三因素认证协议也实现了模块化设计(见[15, 60])，趋于成熟。

与之相对的是，口令全生命周期中剩余8个阶段的相关研究因为涉及到人的因素，待研究的问题常看似简单但难以入手，往往涉及多学科交叉知识并且需要大规模真实数据，容易令想进入此领域的学者们望而生畏，导致一些基本问题(如“口令服从什么分布”)长期悬而未决。自1968年Wilkes[48]的开创性工作以来，口令安全研究根据其研究方法大致可分为三个阶段：

- **阶段1**：1999年以前。主要采用启发式方式，没有系统性和理论性的探索，口令安全研究更多是一门艺术，欧美少数几个研究机构零星地发表一些成果，如Purdy (1974)关于口令存储安全[61]、Saltzer (1975) 关于口令认证系统整体设计思想[62]、Morris-Thompson (1979)对 3289 个经 DES 加密真实口令的攻击实验[1]、Klein (1990)对15000个经 DES 加密真实口令的攻击实验[6]、Davies-Ganesan (1993)提出的口令强度评价算法[63]、Wu (1999)对2500个经 DES 加密真实口令的攻击实验[64]。
- **阶段2**：2000∼2008年。主要采用启发式方式，但有一些系统性和理论性的探索，如Yan等2004年对288名学生关于口令可记忆性与安全性的调研[65]、Ives等2004年指出用户在不同网站重用口令会带来多米诺危害效应[8]、Narayanan-Shmatikov 2005年提出基于马尔科夫链理论的口令猜测攻击算法[66]、2006年Clair等指出口令空间有限从根本上无法抵抗离线字典猜测攻击[10]、Florencio等2007年质疑当前复杂口令设置策略的合理性[67]。这一阶段主基调与微软的“口令替代计划”类似，研究大多集中于揭示口令的弱点，表明口令在身份认证领域将无法担当主要角色。
- **阶段3**：2009年以来。口令安全理论体系初现端倪，口令安全研究逐渐摆脱传统的依赖简单统计方法和启发式“奇思妙想”，开始进入以严密理论体系为支撑的科学轨道。这一阶段形成了以PCFG[68]、Markov[34]和NLP[69]为代表的概率攻击理论模型，以概率攻击理论模型为基础的口令强度评价

方法(如PCFG-based PSM[70]和Markov-based PSM[71])，和以 α-guesswork[5]为代表的口令分布强度评价理论模型等标志性成果。

总的来说，2009 年以前口令安全研究中使用的真实口令集都较小，研究方法多依赖启发式奇思妙想，缺乏系统性、原则性、理论性的成果。始于 2009 年Rockyou网站泄露3200万真实口令的一系列数据泄露事件，为学术界深入研究口令提供了客观条件。2012 年初，Herley-Oorschot[18]和Bonneau等[16]分别在杂志IEEE Security&Privacy和会议IEEE S&P公开撰文指出：口令在可预见未来无可替代，并且当前口令安全研究远未成熟。这有效地消除了笼罩在信息安全领域上空十余年的“口令可能会被替代，研究口令无意义”的疑云。既然口令不可替代，只有深入理解它，人类才能更好地与之共存(“living with passwords”)[25, 72]，这形成了研究口令的内部动因。自此，极其迫切的现实需求开始吸引一大批相关领域学者投入口令安全研究，集中涌现出了一系列研究成果。下面我们主要从用户脆弱口令行为、口令构造策略、口令猜测攻击算法、口令强度评价方法、口令分布及其评价指标、基于口令的身份认证协议六个方面，对国内外口令安全研究现状进行综述。

1.2.2 用户的脆弱口令行为

口令安全性研究的难点在于，口令是人生成的，与人的行为直接相关，而每个人的行为因内在或外在环境的不同而千差万别。比如说，同样是注册一个163邮箱帐户，有的人觉得这个帐户不重要，会使用`123456`作口令。有的人后面会经常使用这一邮箱，因此采用精心构造的一个字符串(比如`brysjhhrhl`，一句诗的首字母[①])作口令。众所周知，`123456` 是弱口令；但是，如果很多用户也使用诗词的首字母作口令，那么攻击者很可能了解这一用户行为，进而`brysjhhrhl` 也可能变成弱口令。至于这两个口令谁更安全，需要具体考察它们针对四种口令猜测攻击(见节1.2.3)的抵抗能力。再比如，给定一个口令`Wang.123`，该口令如果是由“Li”姓用户产生，那么该口令可很好抵御定向在线口令猜测攻击(Targeted online password guessing attack)。如果该口令是由“Wang”姓用户产生，显然它是一个弱口令[73]，无法抵御定向在线口令猜测攻击；对于任何用户，这一口令都无法抵抗离线口令猜测攻击[34]。

用户的不安全行为是造成口令无法达到理想强度的直接原因[74]，因此理解用户的脆弱口令行为成为研究口令安全性的基础前提。一方面用户往往需要管理几十乃至上百个口令帐户[75, 76]，并且这一数字在不断增长，此外各个网站的口令设置要求往往差异很大[77]；另一方面，用户用于处理安全事务的精力十分有限且保持稳定[78, 79]，并不会随着时间的推移而有较大幅度提高。这一根本矛

[①] “brysjhhrhl”由著名诗句“白日依山尽，黄河入海流”中各汉字的拼音首字母组成。

盾导致了用户的一系列脆弱行为。近期研究表明，现实中用户的口令行为往往是理智的[80, 81]，并且只有这样，用户才能可持续地管理不断增多的帐户[82]。

当前，广泛采用的口令脆弱行为挖掘方法既有实证分析(如[5, 34, 83–86])，也有用户调查(如[7, 80, 87–89])；实证分析的数据既有来自于黑客泄露(如[34, 83–85])，也有来自于企业合作(如[5, 86])；用户调查既有小规模的传统调查(如[7, 80, 88, 90])，也有基于外包服务的新型大规模网络调查(如[87, 89])。表 1.1 列出了文献中常使用的一些真实口令集。总的来说，已发现的用户的脆弱口令行为主要可以归为以下三类。

表 1.1 文献中常用的一些真实口令集

Password dataset	Service type	Language Language	Leaked time	Total passwords	Unique password	Personal information	Typical reference
Dodonew	Gaming, Ecommrce	Chinese	2011.12	16,258,891	10,135,260		[34, 84, 91]
CSDN	Programmer forum	Chinese	2011.12	6,428,277	4,037,605		[34, 84, 91]
126	Email	Chinese	2011.12	6,392,568	3,778,168		[91]
12306*	Train ticketing	Chinese	2014.12	129,303	117,808	✓	[29]
Rockyou	Social networks	English	2009.12	32,581,870	14,326,970		[34, 71, 87, 92, 93]
Yahoo	Web portal	English	2012.07	442,834	342,510		[34, 84, 94]

*12306 includes five types of personal info: Name, Birthday, Email, Phone number and National identity number.

(1)口令构造的偏好性选择

国民口令。1979年，Morris和Thompson在他们的开创性论文里分析了3289个真实用户口令，发现86%落入普通字典，33%可以在5分钟内搜索出来。后续大量研究(如[7, 34, 83–85])表明，除了选择单词作口令，用户常常将单词进行简单变换，以满足网站口令设置策略的要求。比如“123456a”可以满足“字母+数字”的策略要求。这些最流行的单词及其变换就形成了国民口令。中文国民口令多为纯数字[83]，而英文国民口令多含字母[71]，这体现了语言对口令行为的影响。有趣的是，爱情这一主题在中英文国民口令中都占据了重要地位[84]。对表1.1中6 个网站来说，高达1.01%∼10.44%的用户选择最流行的10个口令[83]，这意味着只要尝试10个最流行的口令，其成功率就会达到1.01%∼10.44%。

字符组成结构。当网站设置口令生成策略时，口令的字符组成很大程度上由口令策略所决定。否则，用户口令的结构直接体现了用户的偏好。根据文献[34, 84]，最突出的现象是，绝大多数中文口令包含数字，并且32%∼65%仅由数字构成；英文口令喜欢包含字母，34%∼44%仅由字母构成，低于16%的口令仅由数字构成。Castelluccia等[71]发现，Myspace 的最流行10个口令中有9个是由字母和数字构成，这意味着该网站在运行不久后就执行了“字母+数字”的口令策略。在这种情况下，用户也显示了偏好：在最流行的10个口令中，有8个由“一串小写字母+1”组成。用户的这些偏好正是攻击者所努力挖掘的对象。

口令长度。用户口令长度直接受网站策略影响。比如，CSDN执行“口令长度不少于8位”的策略，该网站22.8%的口令长度不低于11，这一比例是表1.1中任意其它网站的2倍以上。文献[34]显示，对于普通网站来说，如果未设置长度

表 1.2 口令重用统计数据†

Typical references	Research method	Direct reuse	In-direct reuse
Gross et al. 2009[99]	Empirical	28%	N/A
Das et al. 2014[7]	User survey	51%	26%
Das et al. 2014[7]	Empirical	43%	30%
Bailey et al. 2014[85]	Empirical	21%	26%
Wise 2015[100]	User survey	59%	N/A
Jaeger et al. 2016[101]	Empirical	20%	N/A

† "Direct reuse" means reuse exactly the same password, while "In-direct reuse" means reuse similar but not the same password.

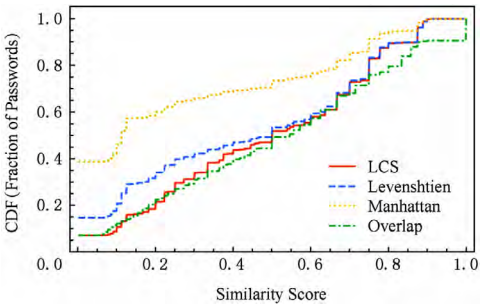


图 1.3 12306与CSDN口令间接重用相似度

限制，90%左右口令的长度介于6-11之间，这一信息对攻击者减少猜测空间具有重要作用。当网站未设置长度限制时，口令的长度分布主要受网站服务类型(重要程度)的影响，但是到底影响如何，由于当前重要应用网站的口令数据集(如E-banking、管理员口令)极少，鲜有相关研究。

(2)口令重用

一直以来，用户的口令重用行为被认为是不安全的，所以用户应当避免重用[24, 95, 96]。但是，近期研究发现，面对如此多的需要管理的帐号，重用口令是用户理智的做法[80, 97]，关键在于如何重用[82]。比如，对于所有新闻订阅帐户，重用同一口令并无不妥；但跨不同重要程度帐户(如新闻订阅帐户和工作邮箱)重用口令，是应尽力避免的[98]。表1.2 中总结了文献中用户口令直接重用和间接(修改后)重用的一些数据。

2014年，Das等[7]进一步研究了口令间接重用时，新口令和原口令间的相似度，并使用了一系列文本相似度算法(如最大共同长度LCS, 文氏距离Levenshtien, 曼哈顿距离Manhattan, 重合度Overlap)进行测量。结果表明，只有约30%的用户重用口令时简单修改，即新旧口令相似度在区间[0.8, 1]内；绝大多数用户的新旧口令相似度小于0.8，表明修改幅度较大。图1.3基于四种相似度算法，我们对12306与CSDN间接口令重用相似度进行了测量，结果与Das等[7]的发现基本一致。2014年，Haque等对80个学生的调查实验发现，用户在不同安全重要程度的帐户中重用口令：19.1% 的参与者在高安全级帐户中重用低级别帐户的口令，33%的高安全级帐户可由低级别帐户的口令在73.4秒内使用常用口令猜测软件恢复出来。

(3)基于个人信息构造口令

早在1979年，Morris和Thompson[1]就发现用户构造口令时喜欢使用姓名等个人可标识信息(Personally identifiable information, PII)，在猜测字典中加入姓名库可以显著提高口令猜测成功率。除了姓名，生日、用户名、Email前缀、身

表 1.3 口令猜测攻击的分类†

Guessing types		Exploit PII	Interact with S	Necessities for attacks	Main counter-measures	Guess num allowed	Main constraints	Typical reference
Trawling	Online		✓	Network connection	Detection, Lockout	$\leq 10^4$	Guess num allowed	[7, 93]
	Offline			With hashed PWs	Iterished hash, salt	$\geq 10^6$	Computation power	[34, 68]
Targeted	Online	✓	✓	Network connection	Detection, Lockout	$\leq 10^4$	Guess num allowed	[103]
	Offline	✓		With hashed PWs	Iterished hash, salt	$\geq 10^6$	Computation power	None

†To save space, PWs stands for passwords; num stands for number.

份证号、电话号码[83, 84, 94]，甚至地名这些个人相关信息都可能被用户使用[102]。但是，这些研究只能表明用户口令中含有PII，至于这些PII是否是属于用户本人的，无从得知，因为这些研究的口令数据集中单独附带有PII。首个公开的附带有PII的口令集来自12306.cn，中国铁路售票官网。Li等[103]测量了六类常用显式PII在129.3K 12306 口令中的含有率。所谓显式PII，是指可以直接作为口令组成部分的PII，如姓名。其中，生日的含有率最高(24.10%)，其次为用户名(23.60%)、姓名(22.35%)、 Email前缀(12.66%)，也有少量用户使用身份证号(2.996%)和手机号构造口令(2.726%)。

小结。当前，实证分析使用的数据集主要来自一些普通应用网站，这些网站极少执行严格的口令生成策略(如字符构成和长度限制)，缺乏对重要应用网站的口令行为研究；用户调查主要关注欧美英文用户，缺少对其它区域其它语言用户的研究；对个人信息使用行为的研究局限于显式的PII且不够深入，隐式PII和其它类型的个人信息(如用户在其它网站使用的身份凭证)也亟待研究。

1.2.3　口令猜测攻击算法

如节1.2.1所述，口令安全性包括口令自身安全性及其辅助系统安全性。一般来说，口令自身的安全性可分为两类[34]：一是整个口令集的安全性(即口令分布的安全性)，二是单个口令的安全性。第一类安全性的评价既可以采用攻击算法进行实际攻击，也可以采用基于统计学的评价指标；第二类安全性的评价只能采用攻击算法进行实际攻击，然后根据攻击结果来衡量，当前广泛采用的衡量指标是成功攻击该口令所需要的猜测次数[104, 105]。本节关注攻击算法，基于统计学的评价指标见节1.2.5。如表1.3，根据攻击过程中是否利用用户个人信息，口令猜测算法可分为：漫步攻击和定向攻击；依据攻击是否需要与服务器交互可分为：在线攻击和离线攻击。

基于对用户脆弱口令行为的更深入理解，攻击者会不断改进其口令猜测算法。近年来一个突出变化是，口令攻击算法逐渐摆脱了依靠“奇思妙想”的启发式方法，进入了依赖可靠的数学概率模型的科学化阶段，诸如基于概率上下文无关方法(Probabilistic context-free grammars, PCFG)[68]，基于马尔科夫链(Markov-Chain)的方法[34]，基于自然语言处理技术(Natural Language

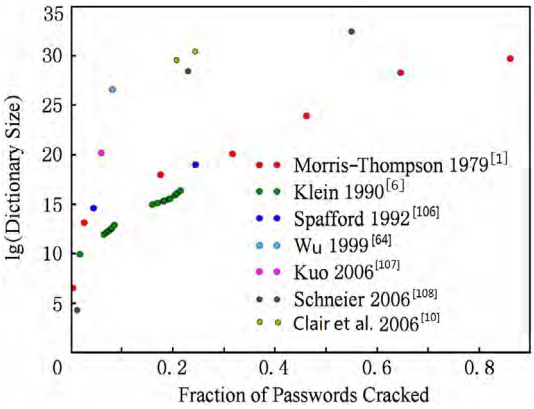


图 1.4 启发式口令猜测方法的攻击效果

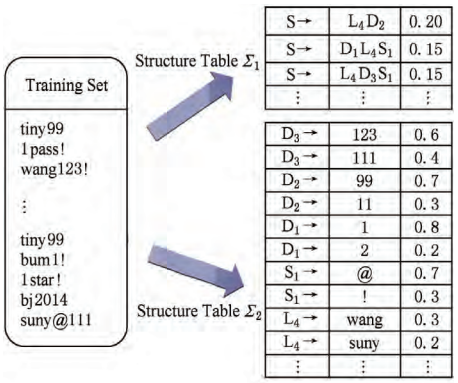


图 1.5 PCFG算法[68]的训练过程

Processing, NLP)的方法[69]。其中，基于PCFG的算法[68]和Markov的算法[34]是当前主流的漫步攻击算法，是其它概率算法的基础，也是本文后续若干理论和算法的基础。因此，本节详细介绍这两个算法，其它攻击算法简要概述。

(1)漫步口令猜测攻击算法

所谓漫步攻击(Trawling attacking)，是指攻击者不关心具体的攻击对象是谁，其唯一目标是在允许的猜测次数下，猜出越多的口令越好。这意味着，最优的攻击者会首先猜测概率排名r=1的口令，接着猜r=2的口令，依次类推。

*启发式算法。*早期的口令猜测算法基本都是漫步攻击算法，且没有严密的理论体系，很大程度上依靠零散的“奇思妙想”。比如，构造独特的猜测字典[1, 64]，采用“精心设计”的猜测顺序[6]，基于开源软件(如John the Ripper)[10, 99, 106, 107]。如Bonneau[5]所指出的，这些启发式算法难以重现，难以相互公平对比。因此，这里我们仅概述它们的攻击效果。如图1.4，绝大多数(如[6, 106, 107])启发式算法的攻破率在30%以下，猜测字典大小为$2^{12} \sim 2^{20}$。虽然少数算法(如[1, 108])能够达到60%以上，但其测试集都较小，都小于10万，并且往往仅在一个测试集上进行评估。

*基于PCFG的算法。*2009年Weir等[68]提出了第一个完全自动化的、建立在PCFG基础之上的漫步口令猜测算法。该算法的核心假设是，口令的字母段L、数字段D和特殊字符段S是相互独立的。它首先将口令根据前述三种字符类型进行切分，比如wang123!被切分为L_4: wang，D_3: 123和S_1: !，$L_4D_3S_1$被称为该口令的结构(模式)。该算法主要分为训练和猜测集生成两个阶段。在训练阶段，最关键的是统计出结构频率表Σ_1和字段频率表Σ_2。基于CFG方法，对泄漏口令进行统计，得到各结构的频率和结构中相应字母段、数字段和特殊字符段的频率，形成表Σ_1和Σ_2。比如针对$L_4D_3S_1$，统计在全部口令中以$L_4D_3S_1$为结构的口令频率，以及“wang”在长为4 的字母段的频率，“123”在长为3的数字段

中的频率和“!”在长为1的特殊字符段中的频率。整个过程如图1.5所示。

在猜测集生成阶段，依据上面获得的模式频率表Σ_1和字段频率表Σ_2，生成一个带频率的猜测集合，以模拟现实中口令的概率分布。比如，猜测wang123!的概率计算为 Pr(wang123!)=Pr(S $\rightarrow$ $L_4D_3S_1$)*Pr(L_4 $\rightarrow$ wang)* Pr(D_3 $\rightarrow$ 123) * Pr(S_1 $\rightarrow$!)= 0.15*0.3*0.6*0.3=0.0081。这表明，口令 wang123! 的可猜测度为0.0081。整个过程如图1.6所示。这样，就能获得每个字符串(猜测)的概率，将这些字符串按照概率递减排序即可获得一个猜测集。

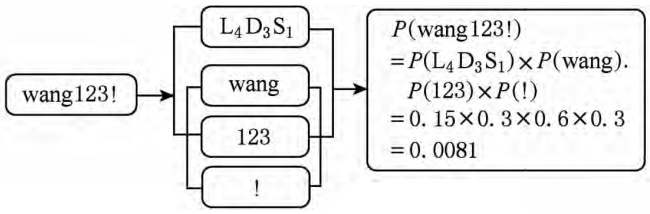


图 1.6 PCFG算法[68]的猜测集生成过程

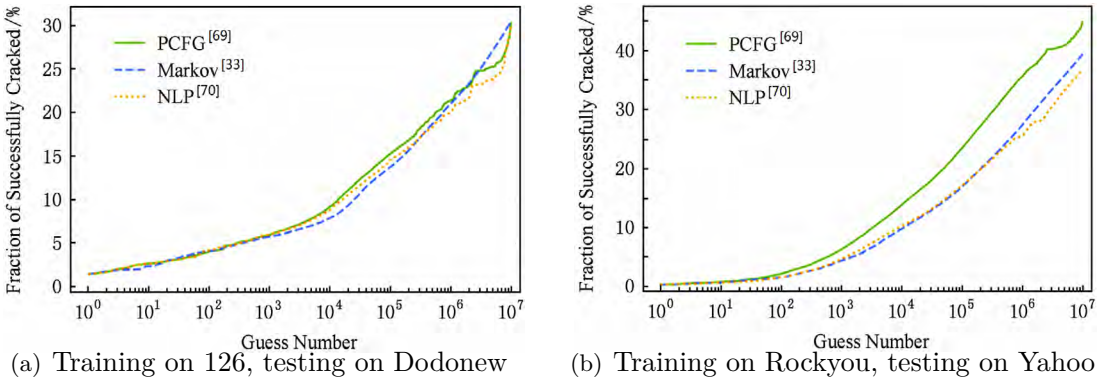
基于Markov的算法。2005年，Narayanan和Shmatikov[66]首次将Markov链技术引入到口令猜测中来。该算法的核心假设是：用户构造口令从前向后依次进行。它不像PCFG那样对口令进行分割，而是对整个口令进行训练，通过从左到右的字符之间的联系来计算口令的概率。类似PCFG 模型，Markov模型也分为训练和猜测集生成两个阶段。

在训练阶段，统计口令中每个子串后面跟的那个字符的频数。Markov模型有阶的概念， n阶Markov模型就需要记录长度为n的字符串后面跟的那个字母的频数。例如，在4 阶Markov中，口令abc123需要记录的有：开头是“a”的频数，“a”后面是“b”的频数，“ab”后面是“c”的频数，“abc”后面是“1”的频数，“abc1”后面是2的频数，“bc12”后面是“3”的频数。这样，每个字符串在训练之后都能得到一个概率，即从左到右，将长度为n的子串在训练结果中进行查询，将所有的概率相乘得到该字符串的概率。在4阶Markov模型下，口令abc123的概率计算如下：

$$\mathrm{Pr(abc123)} = \mathrm{Pr(a|\vdash)} * \mathrm{Pr(b|a)} * \mathrm{Pr(c|ab)} * \mathrm{Pr(1|abc)} * \mathrm{Pr(2|abc1)} * \mathrm{Pr(3|bc12)}。$$

其中，$\vdash$表示起始符号，$\mathrm{Pr(3|bc12)} = \frac{\mathrm{Count(3|bc12)}}{\sum_{\alpha \in \Sigma} \mathrm{Count}(\alpha|\mathrm{bc23})}$，其中字符集$\Sigma$一般是在95个可打印ASCII字符0x20~0x7E中增加一个起始符号$\vdash$，共96个字符。其它概率部分也以相同的方式进行计算。这样就能获得每个字符串的概率，按照概率递减排序即可获得一个猜测集。

在2014年，Ma等[34]首次将平滑技术和正规化技术运用到Markov模型中。平滑技术是为了消除数据集中过拟合(Overfitting)问题，主要有Laplace平滑和Simple Good-Turing平滑两种；正规化技术是为了使得攻击算法所生成的猜



(a) Training on 126, testing on Dodonew (b) Training on Rockyou, testing on Yahoo

图 1.7 三个漫步口令猜测概率攻击算法对比

测的概率总和始终为1，形成一个概率模型，主要有End-Symbol正规化和长度分布正规化两种。下面以Laplace平滑技术和End-Symbol正规化技术为例说明。Laplace平滑是在正常训练之后，对每个字符串的频数都加δ(建议取0.01)，然后再计算字符串的概率。例如，对于字符串“bc123”，其概率计算如下：

$$\Pr(3|\text{bc12}) = \frac{\text{Count}(3|\text{bc12}) + \delta}{\sum_{\alpha\in\Sigma}\text{Count}(\alpha|\text{bc12}) + \delta\cdot|\Sigma|}.$$

End-Symbol正规化在每个口令的最后加了一个结尾符号(假设记为$\perp$)，把$\perp$当作一个字符，训练时将$\perp$一起加入频数统计，除了口令开头不能出现$\perp$以外，在口令其它地方都当作正常字符。生成猜测集时，只有以$\perp$结尾的符串才能够作为猜测，其它字符串都不能作为猜测输出。

基于NLP的算法。2014年，Veras等[69]指出口令中包含大量深层次语义信息，比如一个口令以“ilove···”起始，它后面接男女姓名的可能性远大于“123”或“abc”。但是，当前的 PCFG 算法和 Markov 算法都没有利用这些深层次语义信息。于是，Veras等在PCFG算法将口令进行L、D、S分段的框架内，进一步对L段进行语义挖掘，提出了融合语义的NLP算法。该算法两个核心点：一是分词(Segmentation)，因为口令的词段间没有明确的分割符；二是词性标注(Part-of-Speech tagging，POS tagging)，即对词语标注一个合适的词性，也就是确定这个词是名词、动词、形容词或其它词性。为了有效分词和词性标注，Veras等收集整理了一个英语语料库集合，主体为当代美国英语语料库(COCA)，另加英文人名、城市地名等语料库。基于该语料库，他们使用NLP方法对口令进行切词和词性标注，并在此基础上对英文用户口令所具有的语义含义进行分析、抽象和实例化。NLP算法的猜测集生成阶段与PCFG 算法相同。

如图1.7显示，PCFG算法在小猜测次数下(即在线猜测攻击)最优，Markov算法在大猜测次数下(即离线猜测攻击)开始显示优势，NLP攻击效果介

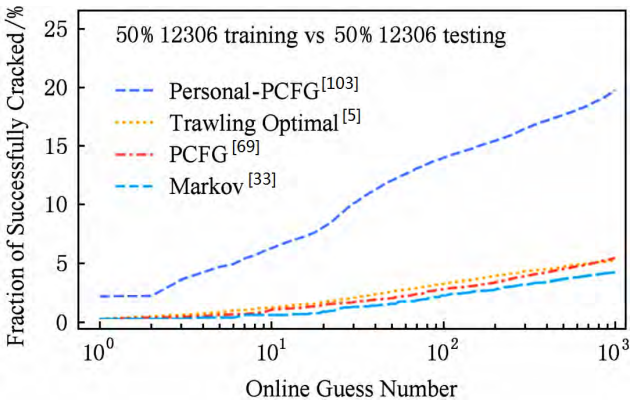


图 1.8 定向口令猜测算法与漫步猜测算法对比

于PCFG和Markov之间。因此，本文主要关注PCFG算法和Markov算法。

(2)定向口令猜测攻击算法

定向口令猜测攻击的目标是尽可能以最少尝试次数猜测出所给定用户在给定服务(如网站、个人电脑)的口令。因此，攻击者会利用与攻击对象相关的个人信息(Personal information, PI)，以增强猜测的针对性。用户的个人信息有很多种，比如用户个人可标识信息PII(姓名、生日、年龄、职业、学历、性别等)，身份认证凭证(如用户在该网站的旧口令，在其它网站泄露的口令)。当前定向口令猜测研究尚在起步阶段，主要集中在如何利用PII方面。

Personal-PCFG。2016年，Li等[103]首次提出了基于PCFG的定向攻击猜测算法，称为Personal-PCFG。该算法基于漫步PCFG攻击模型[68]，基本思想与PCFG模型完全相同：将口令按字符类型和长度进行切分。为实现这一思想，文献[103]首先将个人信息分为6大类(即用户名A、邮箱前缀E、姓名N、生日B、手机号P和身份证G)，并将这6种PII字符类型视为与漫步PCFG模型里的L、D、S同等地位，这样Personal-PCFG中有9种类型字符；接着，在训练过程中，将训练集中每个口令如同漫步PCFG攻击模型[68]那样，按相应字符类型及其长度进行分段[①]。比如口令wang123!被切分为$N_4D_3S_1$分段，因为“wang”属于姓名N 这一大类，长度为4；li123!被切分为$N_2D_3S_1$分段，因为“li”属于姓名N 这一大类，长度为2。剩余训练过程与漫步PCFG模型[68]类似。

猜测集生成阶段分两步。第一步，运行漫步PCFG模型[68]的猜测集生成过程，产生中间猜测集，该猜测集既包含123456这样的可以直接使用的猜测，也会包含带PII类型字符的中间猜测(如N_3，N_2123)；第二步将中间猜测里的PII类型字符替换为攻击对象的相应长度的PII信息，如将N_4123替换为wang123(假设攻击对象姓名为“Wang Lei”)。

[①]文献[103] 只考虑长度大于等于3的PII信息。

图1.8将Personal-PCFG[103]这个定向口令猜测攻击算法与漫步攻击算法(Markov[34]、PCFG[68]和Trawling-optimal[5])进行了对比。结果显示，在猜测数为10~1000时，Personal-PCFG[103]的攻击效率比PCFG[68]高出362%~571%, 比理想漫步算法(Trawling-optimal[5])高出313%~485%。

需要指出的是，Personal-PCFG[103]按长度来切分PII，在训练时既会造成对有些类别PII使用率的高估，也会造成对另外一些PII使用率的低估(详见节3.3.4)。比如，对于用户“Liu Wei”的两个口令`liu123!`和`wei123!`，Personal-PCFG将它们都会切分为$N_3D_3S_1$；对于用户“Wang Lei”的两个口令`wang123!`和`lei123!`，Personal-PCFG将它们分别切分为$N_4D_3S_1$和$N_3D_3S_1$。这是不合理的：高估了“姓氏+`123!`”的概率，同时低估了“名字+`123!`”的概率。这里仅以姓名为例说明，对于其它类型PII 存在类似问题。

1.2.4 口令组成规则

人们早就认识到系统自动分配的口令可用性差[6, 109]。然而，当允许用户选择自己的口令后，他们倾向于选择容易记忆的短字符串，而不是任意长的随机字符序列作为口令，这使用户生成的口令存在很大被攻击者猜测的风险[5, 65, 75]。近期口令研究的一个好消息是[89, 110, 111]，设计得当的口令生成策略(Password creation policy，PCP)可以帮助用户生成安全且容易记忆的口令，这缓解了口令可用性与安全性的矛盾。如图1.9，一个口令生成策略一般由口令组成规则(Password composition rules, PCR)和口令强度评测器(Password strength meter, PSM)组成。前者要求用户生成的口令要符合一定的复杂性(例如，“必须包含字母和数字”)来促进用户选择口令强度较高的口令[95, 112]，后者在用户注册时为用户提供口令强度的视觉(或文字)反馈[110, 111]。本小节主要关注口令组成规则PCR，下一小节关注口令强度评测器PSM。

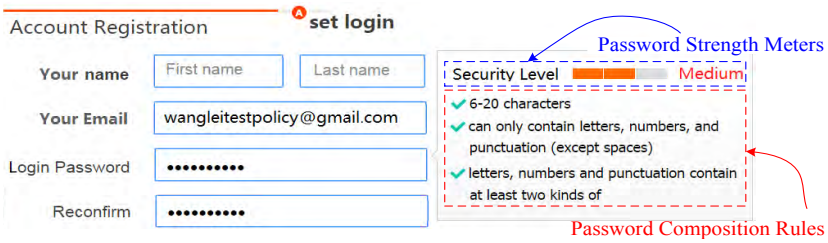


图 1.9 口令生成策略示例

互联网上几乎每个Web服务，无论新旧，都执行了某种口令组成规则PCR。然而，这些PCR到底有多少效果是一个值得研究的问题。2007年，Furnell[77]对10 个欧美流行网站进行调研，发现各个网站口令组成规则和强度评价差异很大，并且没有网站可以达到理想的标准。2010年，Bonneau 和

Preibush[81]对口令组成规则PCR进行了第一次大规模的实证研究。通过调研150个网站，他们发现各个网站的PCR差异很大，缺乏一个广泛认可的行业标准。同年，Florencio和Herley[113] 调研了75个知名网站的PCR强弱取向，发现更高的安全需求(例如，网站规模、资产价值和安全威胁严重程度)一般不是网站选择更严格的PCR 的主导因素。相反，这些大规模、高价值的网络服务(如Paypal和网银Citibank)接受弱口令，而那些不用为可用性差后果买单的网站(例如，政府和教育网站)通常实施更严格的PCR。

为了解各种网络服务是否随着时间推移改进他们的PCR，2011年Furnell[114]调研了10个知名网站，并与它们在2007年时的情况进行了对比。令人失望的是，在四年期间这些网站的PCR几乎没有任何增强，有的甚至没有发生变化，而安全威胁却大大增加。大多数关于业界PCR实施情况的调研工作[77, 81, 113, 114]都是在五年前进行的，但是互联网在这期间已经迅速发展。在2010年初，Twitter拥有2600万个月活跃用户，现在这个数字已经增长了十倍[①]; 2010年11月，Gmail 拥有1.39亿活跃用户，现在这个数字超过5亿[②]。此外，现有工作的另一个局限在于很少关注非英文网络服务。

因此，这里我们调研了来自50个知名网站的PCR现状，包括20个英文网站和30个中文网站(特别关注中文Web服务)。首先，我们选择了10个最感兴趣的网络服务主题，这些主题与我们的日常网络生活高度相关：网络门户，IT公司，电子邮件，安全公司，电子商务，游戏，技术论坛，社会论坛，学术服务和非营利组织。然后，对于每个主题，我们根据Alexa全球500强网站列表(`http://www.alexa.com/topsites`)的流量排名选择前5个网站。一些公司(例如，Microsoft和Google)提供了多种服务(例如，电子邮件，搜索，新闻，产品支持)并且带有一些附属网站，但它们通常依赖于相同的身份认证系统(例如，Windows Live和Google Account)以管理所有用户身份凭证，故我们将所有关联网站视为一体。同样，对于每个主题，我们也选择其在Alexa中国500强网站排名列表中的前10个网站。这样，每个主题选择了15个网站：5个来自英文，10个来自中文。此外，我们从每个主题的这15个网站中随机选择5个网站，从而产生了本文工作中使用的50个网站(见表1.4)：20个来自英文，30个来自中文。

表1.4的结果显示，50个网站中没有两个网站使用相同的口令组成规则PCR，这意味着工业界似乎没有公认可行的做法。一般来说英文网站执行比中文同行更严格的口令策略，但总体来说，尽管长期使用口令并且明白PCR的重要性，许多领先的web服务在执行PCR方面却似乎做得并不好。2010年，Bonneau和Preibusch[81]发现口令策略的许多方面都没有标准化，而我

[①] `http://www.statista.com/statistics/282087/`
[②] `http://thefusejoplin.com/2015/01/choose-google-gmail-yahoo-mail/`

们的结果表明，经过五年的发展，针对口令组成策略这个最基本的问题，业界仍然没有公认的答案。更糟糕的是，来自权威部门的PCR建议也彼此不同。比如，2016年，US-CERT 建议口令“包括大写字母，小写字母，数字和特殊字符”[95]； 2017 年，NIST SP800-63B不建议对口令设置任何复杂性限制，建议口令至少长位8位，通过设置口令黑名单列表来避免弱口令[38]；2017年，斯坦福大学信息安全办公室[112]建议采用适应性PCR：口令至少8位，如果是长为8~11则要求包括大写字母、小写字母、数字和特殊字符，如果是长为12~15则仅要求包括大写字母、小写字母和数字，等等。并且，这些PCR建议往往可执行性差，例如，US-CERT除了前面提到的口令字符复杂性限制，进一步要求“在不同的系统使用不同的口令”[95]。这种混乱大大损害了这些建议的权威性。并不奇怪，大部分知名网站(如Yahoo, Apple, Microsoft和Kaspersky)都执行自己的一套不必要、不合逻辑的PCR规则。总之，信息安全背景、雄厚的资本和强大的工程能力与PCR优劣的相关性并不明显。

2006年，Grimes[115]根据PCR对口令空间增长的贡献程度，指出增加口令的长度比增加复杂性更安全更可用。2010年，基于模拟的方法，在漫步在线猜测攻击模型下，Weir等[93]证实了Grimes的观点。2015年，基于模拟的方法，在漫步离线猜测攻击模型下，Dell'Amico和Filippone[105]证实了Grimes的观点。2016年，基于漫步猜测攻击模型，Shay 等[116]进一步指出，口令规则“口令长度12位以上，包含两类字符”要比策略“必须8位以上，包括字母、数字和特殊字符”更可用、更安全。但是，这些研究都仅仅建立在漫步猜测攻击模型之上。如果考虑更现实的定向猜测攻击，结论是否依然有效值得研究。这就首先依赖于提出高效的定向猜测攻击算法。

除了推荐长口令，现在另一广泛推荐的口令设置规则(见[38, 98, 117])是执行一个中等大小的黑名单(Blacklist)，在黑名单里的口令禁止用户选择。但是，这会缩减本已狭小的用户口令空间[118]。Schechter等不建议使用黑名单，而是使用基于流行度的规则(详见节2.2.1)：只有当一个口令的频率已经超过网站预先设定的安全阈值(如$\frac{1}{10^6}$)，则拒绝该口令。无论是黑名单还是基于流行度的规则，目前都没有关于它们会在多大程度上降低系统可用性的研究：有多少比例用户会被烦扰(被阻止使用当前选择的口令，被迫选择另外一个新口令)？

1.2.5 口令强度评价方法

造成当前弱口令频繁出现的一个直接原因在于普通用户的口令安全意识不足，甚至是不正确的[88, 119]，不知道如何正确构造(设置)与给定网络服务的重要程度相匹配的口令。造成用户安全意识缺陷的原因有两个:(1)口令是私密数据，普通用户往往不知道其他用户的口令是什么样，当选择大众化的口令或口

表 1.4　50个知名网站的口令组成规则PCR调研†

Web Services		Len. limits Min	Len. limits Max	Charset Requirement	Rules Explicit	Accept Symbol	Using blacklist	Checking user PII
Web Portal	Sina	6	16	∅	Yes	Yes	No	No
	China.com	6	-	1^+lower,1^+upper,1^+digit	No[a]	Yes	No	No
	Tecent	6	16	Not a number with len<9	Yes	Yes	No	No
	Ifeng	6	20	∅	Yes	Yes	No	Account
	Yahoo	7	30	∅	No	Yes	No	Both[b]
IT Corp.	Microsoft	8	16	Any 2 charsets	Yes	Yes	No	Both
	Intel	8	15	1^+letter,1^+digit,1^+symbol	Yes	Yes	No	Account
	Apple	8	32	1^+lower,1^+upper,1^+digit	Yes	Yes	No	Account
	Lenovo	8	20	Any 2 of letter,digit,symbol	No	Yes	No	No
	Huawei	8	60	1^+letter,1^+digit,1^+symbol	Yes	Yes	No	Account
Email	139	6	16	Not a number with len<8	No[a]	Yes[c]	Yes	No
	163	6	16	∅	Yes	Yes	Yes	Account
	AOL	8	16	∅	Yes	Yes	Yes	Both
	Sohu	6	16	∅	Yes	Yes	Yes	No
	Gmail	8	-	∅	Yes	Yes	Yes	Both
Security Corp.	Rsing	6	-	∅	Yes	Yes	No	NO
	Symantec	8	25	1^+letter,1^+digit	Yes	Yes	No	Account
	Kaspersky	6	16	∅	Yes	Yes	No	NO
	McAfee	8	32	1^+letter,1^+digit,no symbol	Yes	No	No	No
	360	6	20	∅	Yes	Yes	Yes	No
Ecommerce	Taobao	6	20	Any 2 of letter,digit,symbol	Yes	Yes	No	Account
	Jd.com	6	20	∅	Yes	Yes	Yes	Account
	Dangdang	6	20	∅	Yes	Yes	No	No
	Amazon	6	-	∅	Yes	Yes	No	No
	Meituan	6	32	∅	Yes	Yes	No	No
Gaming	17173	6	20	Not digits only	No[a]	Yes	Yes	No
	Duowan	8	20	Not a number with len<9	Yes	Yes	No	No
	4399.com	6	20	∅	Yes	Yes	No	No
	Sdo.com	6	30	Only letter and digit	Yes	No	Yes	No
	Wanmei	6	16	Only letter and digit	Yes	No	No	No
Technical Forum	CSDN	6	20	∅	Yes	Yes	No	No
	51CTO	8	20	∅	Yes	Yes	No	No
	ChinaUnix	6	24	Any 2 of letter,digit,symbol[d]	Yes	Yes	No	Account
	Hack80	9	-	∅	Yes	Yes	No	No
	Pediy.com	5	-	∅	No[e]	Yes	Yes	No
Social Forum	Tianya	6	-	1^+letter,1^+digit	Yes	Yes	Yes	No
	BBS.xiaomi	8	16	Any 2 of letter,digit,symbol	Yes	Yes	No	No
	Renren	6	20	∅	Yes	Yes	No	No
	Twitter	6	-	∅	Yes	Yes	Yes	Account
	Facebook	6	-	∅	Yes	Yes	Yes	No
Academic Service	WoS	8	-	1^+letter,1^+digit,1^+special	Yes	Yes	No	No
	CNKI	6	20	No symbol(except '_')	Yes	No	Yes	No
	Cjc.ac.cn	1	-	∅	Yes	Yes	No	No
	Easychair	6	40	Not digits only	No	Yes	No	No
	Edas	7	-	1^+letter,1^+digit	No	Yes	Yes	No
Non-profit Org.	IEEE	8	64	1^+digit	Yes	Yes	Yes[f]	No
	ACM	6	26	∅	Yes	Yes	No	No
	W3C	8	-	∅	Yes	Yes	No	No
	CCF	6	32	∅	No	Yes	No	No
	CACR	6	-	∅	Yes	Yes	No	No

† '-' means a length limit of larger than 64; '∅' means no charset requirement; 'Blacklist' means a list of banned popular passwords or structures (e.g., repetition); 'User PII' considers two types of a user's personal information, i.e. name and account name.

[a] China.com, 139 and 17173 only explicitly require that password must be no shorter than 6, yet when one submits a password that do not fulfill the charset requirement, they prompt that more type(s) of character(s) is(are) needed.

[b] Yahoo checks whether user's personal name are incorporated in the password yet it is case sensitive, e.g., "wanglei123" will not be blocked if we input the surname 'Wang' instead of 'wang'.

[c] 139 only accepts six kinds of symbols (i.e., `_~@#$^`).

[d] ChinaUnix explicitly states that a password must contain two types of characters, yet it accepts passwords (e.g., "123456789" and "qwertasdfg") that are measured as "medium" or "strong".

[e] There is no explicit rule in Pediy.com, yet when one submits a password shorter than 5, it prompts that an accepted password must be no shorter than 5.

[f] IEEE's blacklist only includes one item (i.e., "password"), which is explicitly stated.

令构造模式时却并不知情，甚至自认为是“独特的”，“巧妙的”[119]；(2)绝大多数主流网站的口令强度评测器PSM设计是启发式的，过于简单，没有投入足够的努力(“bear no indication of any serious efforts”[120])，向用户反馈的口令强度结果不准确。我们对节1.2.4中提到50个知名网站的PSM进行了调研，结论与[120]一致(见表1.5)。并且，在表1.6 中我们基于16个典型的弱口令，揭示这些网站的PSM反馈结果常常与其它网站相互冲突，这会不可避免造成用户的困惑、挫败感和误解。这两条原因都突出了在用户生成口令时，网站向用户提供及时、准确、一致的口令强度反馈结果的必要性。并且，近期研究表明，表现形式醒目、反馈结果准确的PSM确实能够显著提高口令的安全性[110, 111]。

鉴于此，近年来学术界、工业界和标准组织在PSM设计方面投入了大量的努力，提出了一系列PSM。根据[25, 93], 标准组织界影响力最大的PSM当属NIST entropy[37]; 文献[121]调研了当前工业界22个主流PSM，表现最突出的为Zxcvbn[122]和KeePSM[123]; 学术界最先进的两个算法为PCFG-based PSM[70]和Markov-based PSM[71]。根据底层设计思想的不同，可以将上述PSM分为三类：基于规则(如[37])、基于模式检测(如[122, 123])和基于攻击(如[70, 71])。

(1)基于规则的口令强度评价方法

影响最大的基于规则的方法当属NIST PSM[37]。当前在主流网站上应用的口令强度评价方法绝大多数为基于规则的方法，沿用了NIST PSM的思想：口令强度依据长度和所包含的字符类型而定。典型的代表就是腾讯网站的PSM[124]：(1)当口令长度$len < 6$或$len \leq 8$且仅由数字组成时，只进行警告，不输出强度值；(2)当口令长度$len \geq 6$，且仅由一种字符组成时，评价为“弱”；(3)当口令长度$len > 8$，包含两类字符为“中”，包含三类或四类字符为“强”。这类方法的最大优点是简单，比如腾讯涉及PSM的代码只有12行；缺陷也是非常明显的：评价结果不准确，容易误判—低估强口令和高估弱口令都会发生。

(2) 基于模式检测的口令强度评价方法

此类方法(如Zxcvbn[122]和KeePSM[123])主要目标是检测口令的各个子段所属的构造模式(如键盘模式、顺序字符模式、首字母大写等等)。接着，对发现的各个模式赋予相应的分数(score)。然后，将口令的所有模式的分数加和，得到该口令的总得分，即为口令强度值。比如，Zxcvbn[122]主要考虑了以下三类模式:(1)键盘模式，如qwert, asd, zxcvbn; (2)常见语义模式，如日期、姓名；(3)顺序字符模式，如123456、gfedcba。这些模式被视为弱口令的标志，会给一个较低的分数。此外, Zxcvbn[122]还考虑了字典模式：将口令的子段与构造的一系列字典(如top-10000口令, 1003个男性姓名，3814个女性姓名)进行匹配，根据查找比对的次数进行相应赋值。如果口令的一个子段不属于上述四类模式，则被视

表 1.5　50个知名网站的口令强度评测器PSM调研†

Web Services		Strength score scale	Verbal or visual	Monotonicity	Least score enforcement
Web Portal	Sina	Very weak, Weak, Medium, High	Both	Yes	Weak
	China.com	Weak, Medium, Strong	Both	Yes	Medium
	Tecent	Weak,Medium,Strong	Both	Yes	∅
	Ifeng	Low, Medium, High	Both	Yes	∅
	Yahoo	1(Easy to guess),2(weak),3(Medium), 4(Strong), 5(Very Strong)[a]	Visual[b]	Yes	3(Medium)
IT Corp.	Microsoft	∅	None	–	∅
	Intel	∅	None	–	∅
	Apple	Weak, Medium, Strong	Verbal	No	Medium
	Lenovo	∅	None	–	∅
	Huawei	∅	None	–	∅
Email	139	∅	None	No	∅
	163	Weak,Medium,Strong	Both	Yes	∅
	AOL	Weak, Strong, Brilliant	Both	Yes	∅
	Sohu	Weak, Medium, Strong	Both	Yes	∅
	Gmail	Too short, Weak, Fair, Slightly strong, Strong	Both	No	Fair
Security Corp.	Rsing	∅	None	–	∅
	Symantec	∅	None	–	∅
	Kaspersky	∅	None	–	∅
	McAfee	∅	None	–	∅
	360	∅	None	–	∅
Ecommerce	Taobao	Low, Medium, High	Both	Yes	Medium
	Jd.com	Weak, Medium, Strong	Both	Yes	∅
	Dangdang	Weak, Medium, Strong	Both	Yes	∅
	Amazon	∅	None	–	∅
	Meituan	Weak, Medium, Strong	Both	–	∅
Gaming	17173	Weak, Medium, Strong	Verbal	–	∅
	Duowan	∅	None	–	∅
	4399.com	∅	None	–	∅
	Sdo.com	Weak, Medium, Strong	Both	–	∅
	Wanmei	Low, Medium, High	Both	–	∅
Technical Forum	CSDN	Low, Medium, High	Both	Yes	∅
	51CTO	Weak, Medium, Strong	Both	Yes	∅
	ChinaUnix	Weak,Medium,Strong	Both	Yes	Medium
	Hack80	Weak, Medium, Strong	Both	Yes	∅
	Pediy.com	∅	None	–	∅
Social Forum	Tianya	∅	None	–	∅
	BBS.xiaomi	∅	None	–	∅
	Renren	Weak, Fair, Very brilliant	Verbal	Yes	∅
	Twitter	Too obvious/short, NSE, Can be more secure,Ok, Medium, Strong,Perfect	Visual[b]	No	Not secure enough (NSE)
	Facebook	∅	None	–	∅
Academic Service	WoS	∅	None	–	∅
	CNKI	∅	None	–	∅
	Cjc.ac.cn	∅	None	–	∅
	Easychair	∅	None	–	∅
	Edas	Weak, Medium, Strong	Verbal	No	∅
Non-profit Org.	IEEE	Should be stronger, Good, Great	Both	No	∅
	ACM	∅	None	–	∅
	W3C	Very weak, Weak, Sufficient, Strong, Very strong	Verbal	Yes	Sufficient
	CCF	∅	None	–	∅
	CACR	∅	None	–	∅

'∅' stands for no strength scale; '-' stands for non-existence; 'Monotonicity' stands for whether an additional character contributes to a better score).

[a] According to the results obtained in 2013 [120], the password meter of Yahoo was "Weak, Strong, Very strong". Yet, at the time of this writing Yahoo has changed its policy and divides its strength bar into five scales. Although Yahoo's meter only verbally displays the strength score when the password is "1 (Easy to guess)" or "2 (Weak)", and for the other cases, only a visual progress bar is in place, fortunately one can identify the total number of such cases (i.e., three). In line with its scores in 2013, we suppose the three scores corresponding to these three scales that are not verbally displayed are "3(Medium)", "4(Strong)" and "5(Very Strong)", respectively.

[b] The password meters of Yahoo and Twitter only verbally displays the strength score when a password can not be accepted. When a password can be accepted, *only* a visual progress bar is in place. Consequently, their meters is deemed to be visually displayed.

表 1.6 50个知名网站对16个典型弱口令的接受情况

	123456	123456789	5201314	woaini	iloveyou	password	woaini1314	password123	wanglei	wanglei123	wanglei1	wanglei12	Wanglei123	wang.lei	wanglei@123	Wanglei@123
Sina	,VW	✓,W	,VW	✓,W	,VW	,VW	✓,S	✓,S	✓,W	✓,S	✓,M	✓,M	✓,S	✓,M	✓,S	✓,S
China.com	⊗,W	⊗,W	⊗,W	⊗,W	⊗,W	⊗,W	⊗,M	⊗,M	⊗,W	⊗,M	⊗,M	⊗,M	✓,M	⊗,M	✓,S	✓,S
Tecent	,W	✓,W	,W	✓,W	✓,W	✓,W	✓,M	✓,M	✓,W	✓,M	✓,W	✓,M	✓,S	✓,M	✓,S	✓,S
Ifeng	✓,L	✓,L	✓,L	✓,L	✓,L	✓,L	✓,M	✓,M	✓,L	✓,M	✓,M	✓,M	✓,H	✓,M	✓,H	✓,H
Yahoo	⊗,W	✓,M	,W	,W	,W	,W	✓,M	✓,S	,W	✓,M	✓,M	✓,M	⊗,W	✓,M	✓,M	,W
Microsoft							✓									
Intel															✓	✓
Apple	,W	,W	⊗,W	,W	,W	,W	,M	,M	,W	,M	⊗,W	,W	✓,M	,W	,M	✓,S
Lenovo							✓	✓		✓	✓	✓	✓	✓	✓	✓
Huawei															✓	✓
139				✓,W	✓,W	✓,W	✓,M	✓,M	✓,W	✓,M	✓,M	✓,M	✓,S		✓,S	✓,S
163	,W	,W	,W	,W	,W	✓,W	✓,M	✓,M	✓,W	✓,M	✓,M	✓,M	✓,S	✓,M	✓,S	✓,S
AOL		,W			,W	,W	✓,S	✓,S		,S	,S	,S	,S	,W	,B	,B
Sohu	,W	✓,W	✓,W	✓,W	,W	,W	✓,M	✓,M	✓,W	✓,M	✓,M	✓,M	✓,M	✓,W	✓,M	✓,M
Gmail	,TS	,W	,TS	,TS	⊗,W	,W	,W	,W	,W	✓,F	✓,F	✓,SS	✓,SS	✓,F	✓,S	✓,S
Rsing	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Symantec			,				✓	✓		✓	✓	✓	✓		✓	✓
kaspersky	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
McAfee							✓	✓		✓	✓	✓	✓			
360			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Taobao	,L	,M	,M	,L	,M	,M	✓,M	✓,M	,L	✓,M	✓,M	✓,M	✓,M	✓,M	✓,M	✓,H
Jd.com	,W	,W	,W	,W	,W	,W	,M	✓,S	✓,W	✓,M	✓,M	✓,M	✓,M	✓,M	✓,S	✓,S
Dangdang	✓,W	✓,W	✓,W	✓,W	✓,W	✓,W	✓,M	✓,M	✓,W	✓,M	✓,M	✓,M	✓,S	✓,M	✓,S	✓,S
Amazon	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Meituan	✓,W	✓,W	✓,W	✓,W	✓,W	✓,W	✓,M	✓,M	✓,W	✓,M	✓,W	✓,M	✓,S	✓,W	✓,S	✓,S
17173	,W	,W	,W	,W	,W	✓,W	✓,M	✓,M	✓,W	✓,M	✓,M	✓,M	✓,S	✓,M	✓,S	✓,S
Duowan		✓			✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
4399.com	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Sdo.com	,W	,M	✓,M	✓,W	✓,M	✓,M	✓,M	✓,S	✓,M	✓,M	✓,M	✓,M	✓,M			
Wanmei	✓,L	✓,L	✓,L	✓,L	✓,L	✓,L	✓,M	✓,M	✓,L	✓,M	✓,L	✓,M	✓,H			
CSDN	✓,L	✓,L	✓,L	✓,L	✓,L	✓,L	✓,H	✓,H	✓,L	✓,H	✓,M	✓,M	✓,H	✓,M	✓,H	✓,H
51CTO	✓,W	✓,W	✓,W	✓,W	✓,W	✓,W	✓,M	✓,M	✓,W	✓,M	✓,M	✓,M	✓,S	✓,M	✓,S	✓,S
Chinaunix	,W	✓,M	,W	,W	,W	,W	✓,M	✓,M	,W	✓,M	✓,M	✓,M	✓,M	✓,M	✓,S	✓,S
Hack80		✓,W					✓,M	✓,M		✓,M		✓,M	✓,M		✓,S	✓,S
Pediy.com		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Xiaomi							✓	✓		✓	✓	✓	✓	✓	✓	✓
Tianya							✓	✓		✓	✓	✓	✓	✓	✓	✓
Renren	✓,W	✓,W	✓,W	✓,W	✓,W	✓,W	✓,M	✓,M	✓,W	✓,M	✓,M	✓,M	✓,S	✓,M	✓,S	✓,S
Twitter	,TO	,TO	✓,CMS	✓,NSE		,TO	✓,OK	,TO	✓,NSE	✓,OK	✓,NSE	✓,CMS	✓,M	✓,CMS	✓,S	✓,P
Facebook			✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
WoS															✓	✓
CNKI			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CJC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Easychair				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Edas	,W	,W	,W	,W	,W	,W	✓,W	,W	,W	✓,W	✓,W	✓,W	✓,M	✓,W	✓,S	✓,S
IEEE							✓,G			✓,G	✓,SBS	✓,G	✓,G		✓,G	✓,G
ACM	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
W3C	⊗,VW	,VW	⊗,VW	⊗,VW	,VW	,VW	✓,Suf	✓,Suf	⊗,VW	✓,Suf	,W	,W	✓,S	,W	✓,Suf	✓,VS
CCF	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CACR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

1°Notations: ✓: Accepted; : Rejected by password composition rules; : Rejected by password strength meter; ⊗: Rejected by both the password rule and meter.
2°Abbreviations: B = Brilliant; G = Good; H = High; L = Low; M = Medium; P = Perfect; S= Strong; W = Weak; SS = Slightly strong; ; TO = Too obvious; TS = Too short; TW = Too weak; VS = Very strong; VW = Very weak; CMS = Can be more secure; NSE = Not secure enough; SBS= Should be stronger, Suf = Sufficient.
3° The evaluation result for password A vs. site B is in the "X,Y" format, where "X" indicates whether A is accepted or rejected by B, and "Y" indicates A's strength score given by B's password meter. If B is with no meter, "Y" is naturally absent. We note that for three sites with meters (i.e., AOL, Twitter and Hack80), the strength score "Y" is also absent in cases where the password A does not comply with B's password rules.

为随机字符段，该子段被赋予一个较高的分数。

这类方法比基于规则的方法更科学，但仍属于启发式方法，没有自适应性；一些弱模式(比如字符跳变Leet: `password`→`p@ssword`)如果没有被考虑到，就会漏检，造成弱口令被高估为强口令。

(3) 基于攻击算法的口令强度评价方法

安全永远是相对的。相对于某种攻击者模型而言，一个口令可能是安全的；相对于另外一种攻击者模型而言，可能是非常脆弱的。比如，"Wangping.123"可以很好抵抗漫步在线猜测攻击(如Markov[34], PCFG[68])，但相对定向在线猜测攻击(如Personal-PCFG[103])而言，却明显是弱口令。因此，评价口令强弱的

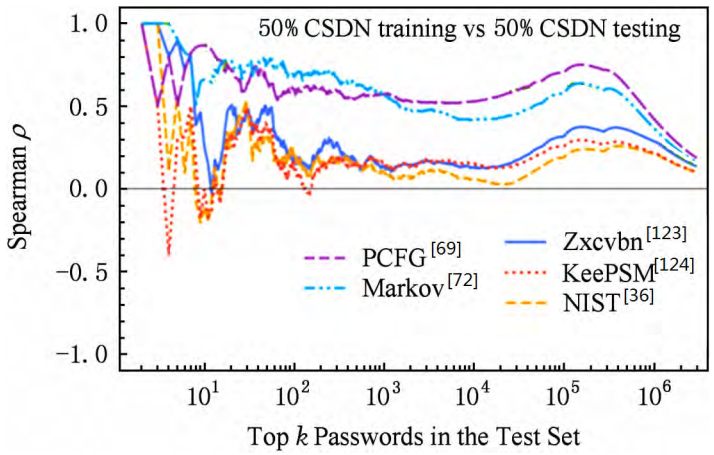


图 1.10　当前5种主流口令强度评价算法对比

一个自然途径就是，使用攻击算法对给定口令进行攻击，依据攻击的难易程度进行强弱判定。基于这一思想，近期提出了两个基于漫步攻击算法的PSM，即PCFG-based PSM[70]和Markov-based PSM[71]。

图1.10采用文献[71]中推荐的Spearman系数对前述5个主流PSM进行了对比，其中训练集为50% CSDN，测试集为剩余50% CSDN。对于一个给定的测试口令集，PSM会对其中每个口令输出一个强度值(概率值)；如果将这些口令按其强度值排序，会产生一个口令强度序列。Spearman系数用以衡量一个PSM输出的强度序列与理想PSM输出的强度序列间的相关关系：两个序列完全相同则Spearman系数为1，倒序则Spearman系数为-1，Spearman系数越大表明两个序列越接近。因此，Spearman系数越大表明其越接近理想PSM。图1.10 显示，在识别弱口令方面，PCFG-based PSM > Markov-based PSM > Zxcvbn > KeePSM > NIST PSM；在识别强口令方面，PCFG-based PSM > Markov-based PSM > Zxcvbn > KeePSM > NIST PSM。由于PSM最核心功能是防止用户选择弱口令：准确向用户反馈口令的强度，当发现用户选择了弱口令时，进行重点提示或阻止。因此，总体来说，PCFG-based PSM 表现最优，学术界的PSM要优于工业界PSM，标准组织NIST的PSM的评价结果准确性最低；在猜测数为$10^2 \sim 10^4$时，所有PSM的Spearman系数都低于0.8，都不够准确。

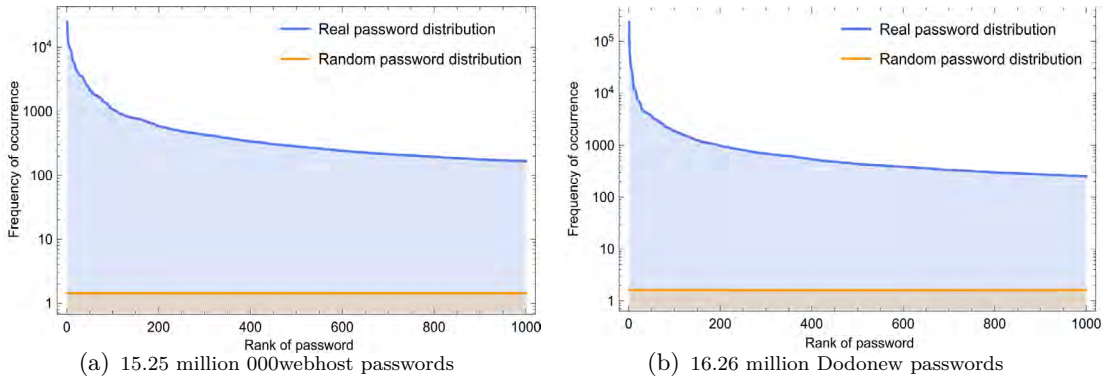
需要指出的是，本节所讨论的5个主流PSM都基于一个基本假设：用户在注册时，从无到有开始构造口令。但现实是，用户在向一个新网站注册口令时，往往是重用(51%)一个现有的口令，或者修改(26%)一个现有口令，而仅有21%的用户从无到有构造一个全新的口令[7]。这会给口令强度评价带来偏差。随着越来越多的网站被攻破、用户口令被泄露，设计更符合实际的、基于口令重用假设的PSM是一个具有重要现实意义的研究方向。

1.2.6 口令分布及其评价指标

“口令服从什么分布”是口令研究领域最为基础性的问题之一。如节1.2.2所示，用户选择口令时有一系列明显的偏好和倾向，因此口令不太可能是均匀随机分布。这一观点已被广为接受(见[5, 125])。尽管如此，口令研究领域长期以来却一直假设用户随机均匀选择口令。这一方面是因为，在均匀分布假设下容易推导出明确的数学结论；另一方面是因为如果不假设口令服从均匀分布，我们不知道口令到底服从什么确切的分布。

(1) 口令分布

为了对口令分布有一个具体的认识，我们以两个大规模口令集为例说明(见图 1.11)。我们其它的口令集也显示相似的分布，这里由于空间限制不再详述。显然，它们都不是均匀分布。那么一个关键的问题出现了：*如果用户选择的口令不服从均匀分布，那么它们到底服从何种分布？*



(a) 15.25 million 000webhost passwords　(b) 16.26 million Dodonew passwords

图 1.11 两个大规模真实口令集的频率分布

这个问题近几十年来一直是公开问题。这有两方面原因：一是现实生活中口令是敏感数据，不容易收集，因而真实口令集匮乏；二是解决这个问题涉及到跨学科知识及其最新进展，比如计算统计学和自然语言处理(NLP)。自2009年第一个千万级口令集 Rockyou 泄露以来(见表1.1)，数以百计的知名网站被攻陷[26, 126–128]，这为研究口令分布提供了充足原始数据。因为人类自然语言满足Zipf分布[129]，很自然的一个想法是，人类生成的口令也可能满足Zipf分布。2012年，Malone和Maher[125] 分析了3200万Rockyou数据和另外3个小于10万的数据集，他们将整个口令集输入Zipf模型，发现拟合出来的参数通不过 Kolmogorov-Smirnov (KS)检验。因此，他们得到结论：口令不服从 Zipf 分布。同年，Bonneau[5] 使用类似方法分析了7000万Yahoo口令，也否定了口令服从Zipf分布的可能性。这样，口令到底服从什么分布没有得到明确的答案，相关研究又陷入瓶颈。

因此，在众多必须依赖于口令分布明确假设的工作中，绝大多数(如基于口令的安全协议中的认证[22]、加密[130]、签名[131]和秘密共享[132])不情愿地做了不合实际的假设，认为用户选择的口令服从均匀随机分布，这样分析问题最为简单方便；剩余的工作中，极少数为了实现安全性分析，对口令分布做了各种各样的假设(如[71]中的二项分布和[133, 134]中的最小熵模型)。至于那些不需要依赖明确口令分布假设的工作，它们(如基于流行度的口令生成规则[135]和口令分布的强度评价指标[5])通常只是简单地避免对口令分布作假设。

正如本文将在第二章中证实的，上述关于口令分布的不合实际的假设常常会引起严重的安全性和可用性问题。文献 [21, 22, 130, 131, 136]中假设口令服从均匀分布，其攻击者优势的安全归约结果，相比于真实分布，被低估了2到4个数量级。至于那些规避口令分布假设的工作(如[5, 135])，许多重要的特性和安全目标都无法进行深入讨论和研究，因为它们往往都隐性依赖于明确的口令分布假设。比如，如果一个web服务强制实行文献[135]中提出的基于流行度的口令组成规则PCR，那么有多大比例用户可能会被阻止？这种预测对评估PCR的合理性十分重要，但若尚未明确口令服从何种分布，几乎不可能作出预测。

(2) 口令分布安全性评价指标

2012年，Bonneau[5]指出，采用攻击算法来评估口令分布的安全性可能带来不确定性，因为攻击算法的效率严重依赖于攻击模型以及模型参数(如训练集，平滑方法，归一化方法)的选择。相比之下，基于统计学的评价指标则避免了这一难题。当前，广泛采用的统计学指标主要有以下6类，下面分别介绍，最后利用它们来测量一些典型的真实口令集。不失一般性，设分布为$\mathcal{X}$，样本空间大小为N，事件$x_1, x_2, \cdots, x_N$的概率分别为$p_1, p_2, \cdots, p_N$。不失一般性，设$p_1 \geq p_2 \geq \cdots \geq p_N$。主要有六种统计学指标：

1) *Shannon*信息熵(Shannon Entropy)[137]被广泛用来测量一个分布的不确定性: $H_1(\mathcal{X}) = \sum_{i=1}^{N} -p_i \cdot \log p_i$。
2) 最小熵(Min Entropy)[138]被用来测量一个分布中概率最大事件出现的不确定性: $H_\infty(\mathcal{X}) = -\log_2(p_1)$。
3) 猜测熵(Guessing entropy)[139]被用来测量当一个漫步攻击者按最优方式攻击时，猜测出$\mathcal{X}$中任一元素需要的平均猜测次数： $G(\mathcal{X}) = \sum_{i=1}^{N} p_i \cdot i$。其等价的比特表现形式为$\widetilde{G}(\mathcal{X}) = \log_2(2 \cdot G(\mathcal{X}) - 1)$。
4) β-*success-rate*[140]被用来测量当一个漫步攻击者被限制至多猜测β次时，其平均成功率：$\lambda_\beta(\mathcal{X}) = \sum_{i=1}^{\beta} p_i$。其等价的比特表现形式为$\widetilde{\lambda}_\beta(\mathcal{X}) = \log_2(\beta / \lambda_{\beta(\mathcal{X})})$。
5) α-*work-factor*[141]被用来测量当一个漫步攻击者想要达到至少α的成功率

时，它需要对每个帐户至少发起的猜测次数为：$\mu_\alpha(\mathcal{X}) = \min\{j|\sum_{i=1}^{j} p_i \geq \alpha\}$。其等价的比特表现形式为$\tilde{\mu}_\alpha(\mathcal{X}) = \log_2(\mu_\alpha(\mathcal{X})/\lambda_{\mu_\alpha(\mathcal{X})})$。

6) α-*guesswork*[5]用来测量当一个漫步攻击者想要达到的成功率时，它需要对每个帐户平均发起的猜测次数：$G_\alpha(\mathcal{X}) = (1-\lambda_{\mu_\alpha})\cdot\mu_\alpha + \sum_{i=1}^{\mu_\alpha} p_i \cdot i$。其等价的比特表现形式为 $\widetilde{G}_\alpha(\mathcal{X}) = \log_2(\frac{2\cdot G_\alpha(\mathcal{X})}{\lambda_{\mu_\alpha}} - 1) + \log_2(\frac{1}{2-\lambda_{\mu_\alpha}})$。

2012年，Bonneau[5]指出，虽然对每个帐户发起$\mu_\alpha(\mathcal{X})$ 次猜测确实能够保证α的成功率，但对于有些帐户，攻击者并不需要$\mu_\alpha(\mathcal{X})$次猜测就能攻破。α-work-factor指标不能刻画这一情形，这是它的一个固有缺陷。α-guesswork这一指标正好克服了这一缺陷。

上述6个指标中，信息熵和最小熵天然以bit为单位，而其它指标有的是百分比，有的是猜测次数。Bonneau[5]巧妙地使用“有效密钥长度”的思想，将它们都等价转换为以bit单位，以便进行对比。如表 1.7 所示，最小熵、 β-success-rate[140] 和小成功率的α-guesswork 适于用来衡量一个口令分布抵抗在线猜测攻击的能力，而信息熵和大成功率的α-guesswork适于用来衡量一个口令分布抵抗离线猜测攻击的能力。

表 1.7 真实口令分布的安全性统计学指标评价结果(单位:比特)

Password distribution	Resistance against online guessing					Resistance against offline guessing				
	H_∞	$\widetilde{\lambda}_{10}$	$\widetilde{\lambda}_{10^2}$	$\widetilde{\lambda}_{10^3}$	$\widetilde{G}_{0.1}$	$\widetilde{G}_{0.2}$	$\widetilde{G}_{0.3}$	$\widetilde{G}_{0.5}$	H_1	G
Dodonew	6.11	8.25	10.80	13.51	14.42	17.97	19.97	21.65	21.76	22.64
CSDN	4.77	6.58	9.56	12.56	6.09	13.94	17.41	20.30	19.47	21.30
126	5.76	8.10	10.68	13.15	12.71	15.68	17.51	19.82	20.16	21.13
12306	8.37	9.61	11.52	13.83	14.75	16.07	16.40	16.61	16.63	16.71
Rockyou	6.81	8.93	11.10	13.11	12.77	14.88	16.77	19.79	21.07	22.65
Yahoo	8.05	9.95	11.76	13.55	13.93	15.72	16.63	17.68	17.88	18.03

表 1.7 显示，在 6 个口令分布中，CSDN抵抗在线猜测攻击能力最差，12306最优；12306抵抗离线猜测攻击能力最差，Dodonew最优。如节1.2.2所言，在这6个网站中，12306提供的是电子售票服务，直接与财产相关，因此用户会倾向于选择强口令，不难解释为什么12306抵抗在线猜测攻击能力最优。出乎意料的是，CSDN 执行了最长的口令长度要求(即“口令长度不小于8”)，但其抵抗在线猜测攻击能力最差，这可能因其仅是一个技术交流论坛。这证实：网站服务类型是影响口令强度的重要因素，口令生成规则可能次之。

需要指出的是，本节的6个指标都是通用的，都建立在对口令分布不作任何假设的基础之上。不难看出，它们不仅仅可以用来测量口令，也可以用来测量任何概率空间。这意味着，如果我们获得关于口令分布的更多认识，则可以进一步研究它们的性质。尤其是α-guesswork当前已在口令研究中广泛使用(见[84, 142–146])，但其一个明显缺陷，如原作者所指出的，是无法判定其单调性，导致在比较两个口令分布的安全性时十分繁琐。消除这一缺陷有赖于确切的口令分布函数，而后者当前是公开问题。

1.2.7　基于口令的身份认证协议

如节1.2.1所提到的，基于口令的身份认证协议(简称口令认证协议)由于不涉及人的因素，研究的问题容易刻画，研究的目标明确单一，可借鉴现有丰富的认证密钥协商协议设计和分析技术(如[147–149])。因此，这方面成果非常丰富，成百上千的口令认证协议先后被提出，知名的如EKE[54]、SRP[20]、KOY[21]、J-PAKE[22]。目前对口令认证协议的分类还没有一个统一的标准，但大体上可分为以下五类：

- 依据涉及认证因子的多少，可分为基于口令的单因子认证协议PAKE(如[21, 54, 56])、基于口令的双因子认证协议(如[150, 151])和基于口令的多因子认证协议(如[15, 60])；
- 依据协议参与方的多少，可分为两方口令认证协议(如[21, 54])、三方口令认证协议(如[57, 152])和多方口令认证协议(如[59, 153])；
- 依据协议是否使用公钥基础设施PKI，可分为基于PKI的口令认证协议(如[56, 154]和非基于PKI的口令认证协议(如[21, 54])；
- 依据协议的安全性证明是否使用了形式化方法，可分为可证明安全的口令认证协议(如[21, 57])和启发式安全的口令认证协议(如[54, 153])；
- 依据协议是否实现了匿名性，可分为匿名口令认证协议(如[155–157])和非匿名口令认证协议(如[21, 150, 158])；

对PAKE协议来说，在2000年以前，主要关注的是如何提高两方PAKE协议的效率，并且实现可证明安全性；2000年以后，随着互联网技术的发展，关注点开始转入研究各种网络应用环境下的口令传输安全问题(如三方PAKE[57]、Two-server PAKE[58]、Multi-server PAKE[59]和Gateway-based PAKE[159])。

对基于口令的双因子认证协议[①]来说，如图1.12，2004年以前都基于抗窜扰(tamper-resistant)智能卡假设：智能卡内秘密参数不能被攻击者分析出来[160–162]。自1999年Kocher等[163]发现使用差分能耗分析可以分析出智能卡内参数以来，边信道攻击技术不断进展，出现了一系列如计时攻击[164]、逆向工程[165]等手段可分析出普通智能卡内秘密信息。一旦攻击者得到智能卡内秘密参数信息，那些基于抗窜扰智能卡假设的协议就不再安全。鉴于这一严峻现实，2004年Ku等[166]首次提出一个基于非抗窜扰智能卡的双因子协议。不幸的是，在该协议提出不久，被Yoon等[167]发现它易遭平行会话攻击。然而，Yoon等提出的改进协议依然存在多种安全缺陷[168, 169]。如图1.12，整个双因子认证协议的研究历史，就是一个“攻击→改进→攻击→改进”的历史。

虽历经大量的努力，在非抗窜扰智能卡假设下，目前还没有一个双因子认

[①]由于基于“口令+智能卡”双因子认证协议是最流行的双因子认证协议，本文主要关注此类双因子协议。

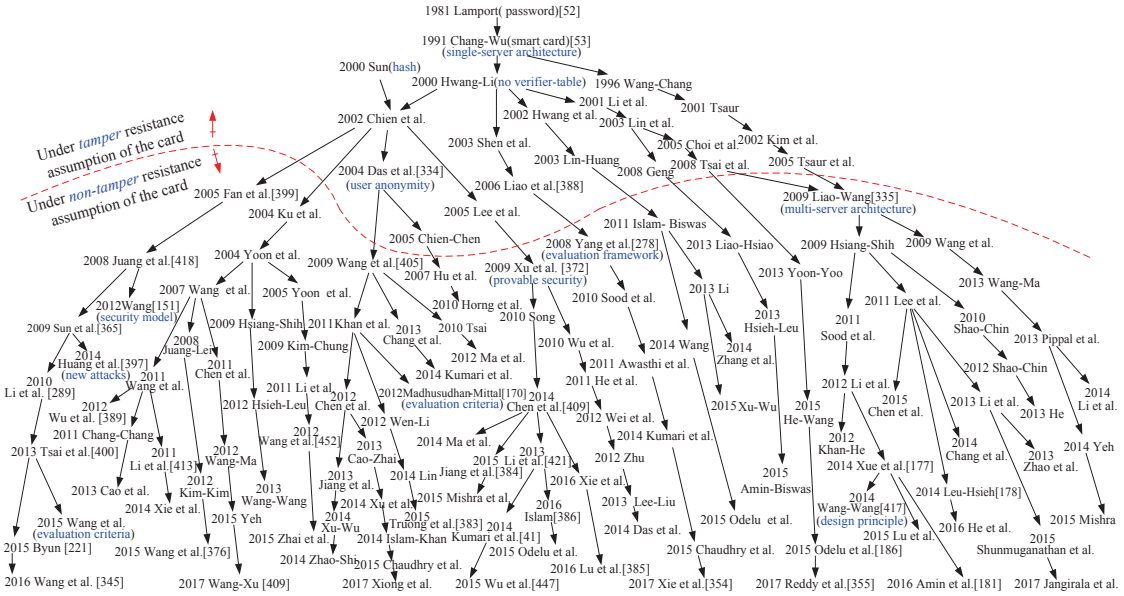


图 1.12 基于“口令+智能卡”双因子认证协议的一个简要历史

证协议能实现文献[170]中所有评价指标 (见表1.8和1.9)。这到底是因为协议设计不理想，还是因为协议评价指标体系本身存在问题？尚不得知。此外，为防止用户隐私(如个人偏好、登录历史、位置、身体状况)泄露和滥用[171, 172]，设计匿名双因子认证协议的需求不断增长。由于智能卡是资源受限设备，学者们尝试仅采用较为轻量的对称密码技术，来设计匿名双因子认证协议(如[173–178])。遗憾的是，其中绝大多数都陆续被指出无法实现宣称的匿名性。目前，仍有大批学者未放弃尝试(见[179–182])，经我们分析，这些尝试依然没有成功。这自然就产生一个猜测：在非抗窜扰智能卡假设下，公钥技术是双因子协议实现匿名性的必要条件。证实或否定这个猜测具有重要理论价值。

对基于口令的多因子认证协议来说，其应用环境主要是电子政务、电子商务、电子医疗等少数高安全需求环境。近年来随着敏感应用的不断联网和攻击者能力的不断增强，此类协议开始受到学界关注，研究点主要集中于如何利用丰富的PAKE协议和双因子协议来构造“口令+智能卡+生物特征”的多因子协议(如[183–186])。文献[15, 60]进一步探讨了如何基于双因子协议，来模块化设计基于口令的多因子认证协议，这方面研究趋于成熟。因此，关键点又回到如何设计出安全高效的基于口令的双因子认证协议。

需要指出的是，当前99%以上的口令认证协议在分析协议安全性时，都假设口令是均匀分布的；剩余极少数几个协议(如[133, 134])意识到“口令均匀分布的假设”不够合理，为避免低估攻击者的优势(advantage)，采用了最小熵模型(min-entropy model)。但是，如第二章所显示的，最小熵模型又走向另外一个极端：严重高估了攻击者的优势。

表 1.8　安全目标

SR1	抗DoS攻击
SR2	抗仿冒攻击
SR3	抗并行会话攻击
SR4	抗口令猜测攻击
SR5	抗重放攻击
SR6	抗智能卡丢失攻击
SR7	抗窃取验证项攻击
SR8	抗反射攻击
SR9	抗内部攻击

表 1.9　理想属性

DA1	服务器端无验证表项
DA2	用户能自由地选择口令
DA3	不泄露口令
DA4	口令相关的
DA5	双向认证
DA6	会话密钥协商
DA7	前向安全性
DA8	用户匿名
DA9	智能卡撤销
DA10	口令错误输入及时检测

1.3　有待解决的关键问题

口令由于使用简单、成本低廉、容易修改，克服了基于物理硬件的认证技术和生物特征认证技术的缺陷，人们逐渐认识到：在可预见的未来，基于口令的身份认证技术仍将是鉴别用户身份最主要的方式。因此，口令研究显得十分必要，近年来也涌现出了一批相关成果。但如节1.2中所指出的，口令研究整体尚处于起步阶段，学术界虽然针对一些现实口令安全问题提出了解决方案，但往往缺乏可行性和前瞻性，指导性不强，口令学术界没有能够引领口令工业界，人们面临的口令安全问题越来越严重。

本文认为，产生这一不理想研究现状的原因有三：

- **客观原因I**：涉及多学科交叉知识，研究的难度大。口令安全研究不仅需要大量真实口令数据，而且涉及多学科交叉知识，如密码学、系统安全、自然语言处理、统计学和心理学。
- **客观原因II**：涉及人的因素，研究的问题复杂。口令生命周期9个环节中有8个涉及到人的因素，导致研究的问题往往复杂多样，充满不确定性，难以准确刻画、定义，不容易采用现有密码学工具建模求解。因此，以往研究主要集中于不涉及人的因素的口令传输环节，即如何设计更安全高效的口令密码协议，缺少对人的因素的关注，而现实中用户是最薄弱环节。
- **主观原因**：把口令研究视为实验科学，研究的视角片面。现有口令研究常忽略问题背后的深层次理论问题，多局限于实验实证或对现象的简单统计，极少尝试采用数学理论模型或形式化方法，缺乏从理论层面进行规律性、原则性和系统性的探索。

鉴于此，本文以用户脆弱行为为切入点，采用数据驱动的方法，基于可证明安全理论、统计学习理论和自然语言处理技术，侧重从理论的视角研究下列6 个亟待解决的关键问题：

(1) **口令服从什么分布**？这一问题是身份认证领域最为基础性的问题之一，也

一直是公开问题。个别研究曾探索口令是否服从Zipf分布，但得到了否定的结论。现有研究在必须对口令服从什么分布作出假设时，普遍假设口令满足均匀分布，或者某种便于分析的分布；在不必对口令分布作出假设时，往往规避对涉及口令分布相关领域的探索。这不仅严重阻碍了对口令安全性的研究，还产生了不符实际的研究结果、缺乏可行性的安全方案。

(2) **如何科学评估定向在线口令猜测威胁**？现有研究主要集中于评估漫步口令猜测威胁，鲜有涉及定向在线口令猜测。由于后者利用各类个人信息，如显式PII(姓名、生日等)、隐式PII(性别、宗教信仰等)以及用户在其它网站的口令，攻击者可显著提高猜测成功率。近年来，大规模个人信息泄露事件不断发生，定向在线口令猜测成为一个越来越严峻的威胁。如何科学评估其威胁，进而设计针对性的防御措施，尤为迫切。

(3) **如何最优地攻击honeywords**？近年来，上百家世界知名网站(如Yahoo，Linkedin和Dropbox)泄露了口令文件，但都是在口令泄露多年后才被发现，然后才通知用户修改口令，往往为时已晚。如何在服务器端主动、及时检测到口令文件的泄露，是一项亟待解决的课题。2013年，图灵奖得主Rivest 及其合作者提出了honeywords的思想，并给出了几个启发式honeywords生成方法，但未给出严格的安全性分析(理论证明或实证评估)。他们将“如何对这些honeywords生成方法进行安全性分析”遗留为公开问题，而这一问题的解决有赖于对另一更基本问题的深刻认识：攻击者如何进行最优攻击。这涉及对相关问题的准确刻画和复杂的数学证明。

(4) **如何科学地评估用户口令的强度**？当前主流的口令评测器 PSM 都建立在一个不符合实际的假设之上：用户在注册时，从无到有开始全新构造一个口令。针对中文和英文用户的调查研究都显示，用户在向一个新网站注册口令时，往往是重用或者修改一个现有口令，而很少从无到有构造一个全新的口令。这意味着，现有PSM 的评价结果会存在严重偏差。随着越来越多的网站被攻破，用户口令被泄露的可能性越来越大，设计更符合实际的、基于口令重用假设的 PSM 是一个亟待解决的课题。

(5) **能否仅采用对称密码技术实现口令认证协议的匿名性**？为防止用户隐私泄露和滥用，学者们提出了大量的基于口令的匿名双因子认证协议。由于智能卡是资源受限设备，大多数协议仅采用了较为轻量的对称密码技术。遗憾的是，在非抗窜扰智能卡假设下，这些协议都陆续被指出无法实现所宣称的匿名性。目前，大批学者仍在不断尝试。我们对匿名性的本质计算复杂度产生一个猜测：在非抗窜扰智能卡假设下，公钥技术是双因子协议实现匿名性的必要条件。证实或否定这个猜测具有重要理论和现实价值。

(6) **如何全面准确地评价一个基于口令的双因子协议**？在非抗窜扰智能卡假设

下，设计一个实用且满足一系列严苛的安全目标(如抗智能卡丢失攻击)和理想属性(比如本地口令更新)的双因子认证协议仍是一个尚未解决的难题。近年来，数以百计的方案被提出，然而其中大部分在提出不久就被指出存在这样或那样的问题：要么不能满足一些关键的安全目标，要么缺失一些重要的理想属性。为了改变这种局面，学者们绞尽脑汁想要设计出满足所有目标的理想协议(至今没有人成功)，却少有人关注两个更本质的问题：(1)在协议设计中，是否存在一些内在的局限阻碍我们设计出一个理想的协议？(2)当前广泛采用的协议评价指标体系是否自身不够合理，比如，指标元素有无缺漏和歧义，指标元素间是否存在冗余和内在冲突？

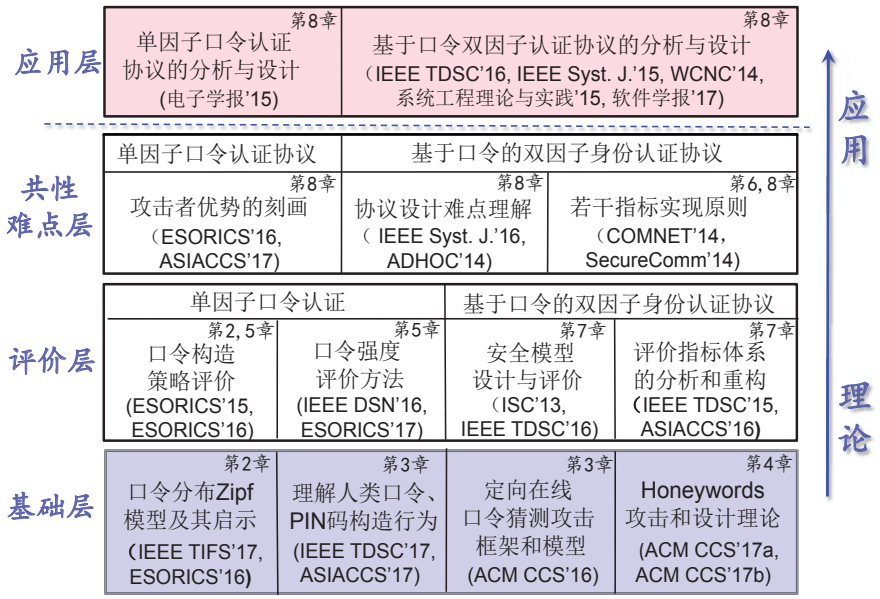


图 1.13　本文研究工作框架

1.4　本文工作

围绕上述6个关键问题，我们开展了一系列如图1.13所示的工作，本文正是这些工作的综合、集成和提炼。它们共同构成一个多层次、系统性的研究框架。其中，基础层致力于发现物理现象和理解人类行为规律，这是口令安全领域最基础、本原的工作，直接应对问题 1、2 和 3；评价层致力于研究口令认证评价体系，包括标准、方法和安全模型等方面，将解决问题 4 和 6；难点层承上启下，致力于研究如何实现一些挑战性的评价指标，如探究实现匿名性的本质计算复杂度(问题 5)，寻求协议可用性和安全性的合理平衡，这为设计安全高效的协议提供直接指导；应用层是难点层所揭示的协议评价指标实现原则和必要条件的综合集成，重点研究如何设计适于各类应用环境的口令认证协议。我们相信，这些工作将为口令安全研究从艺术走向科学奠定重要理论和方法基础。

具体来说，本文有以下 7 项主要贡献：

(1) **发现口令服从Zipf分布，并给出精确分布函数**。通过引入自然语言处理技术和统计计算理论，利用14个大规模真实口令集，提出了两个口令分布理论模型：PDF-Zipf 和 CDF-Zipf。其中，PDF-Zipf 模型需要去除低频口令后拟合，而CDF-Zipf模型可对整个口令集进行拟合。实验结果显示，PDF-Zipf模型的决定系数(R^2)达97%以上，CDF-Zipf模型得到的理论分布与实际分布的最大累积分布误差在0.48%~4.59%(平均1.84%)。两个模型的结果都表明，人类口令服从Zipf分布。进一步提出一个口令分布强度评价标准，并揭示三项重要应用：(i)刻画口令猜测攻击者的优势；(ii)预测基于流行度的PCR对可用性的影响程度；(iii)研究α-guesswork的单调性。

(2) **提出一套定向在线口令猜测攻击理论模型**。针对“在未获取网站服务器口令存储文件的情况下，攻击者如何利用个人信息来加速在线口令猜测”这一问题，提出一个定向在线口令猜测攻击框架。该框架包含 7 个理论模型，以刻画 7 种不同现实能力的定向在线猜测攻击者。大规模真实数据测试结果显示，在允许猜测100次的情况下，如果仅知道目标用户的一些常见PII，成功率可达20%；如果知道用户在其它网站曾泄露的一个口令，成功率可达77%。这一结果远超出此前人们的预期，引起美国国家身份认证指南NIST SP800-63-3关于防御在线猜测部分的修订。

(3) **提出一个Honeywords攻击和设计理论体系**。 首先指出Juels-Rivest 在 ACM CCS'13 上首次提出的4个主要 honeywords 生成方法都存在严重安全缺陷，攻击者通过1次猜测分辨出真口令的概率为36.8%，远高于宣称的“5%左右”。并且指出，此类启发式方法无法简单修补。进一步提出一个 Honeywords 攻击理论体系，回答了“给定不同攻击能力，攻击者如何进行最优攻击”这一重要理论问题，为成功解决Juels-Rivest遗留的三个公开问题奠定了理论基础。反过来，攻击者的最优攻击方法可被用来设计最优 honeywords 生成方法，成功摆脱了启发式设计方法。本文研究使 honeywords 生成方法的设计和评估从艺术走向科学。

(4) **设计一个基于重用行为的口令强度评测算法**。首先开展了对442名用户中文口令使用习惯的调查，发现在向一个新网站注册口令时，77.4%的用户会重用或修改一个现有口令，而仅有14.5%从无到有构造一个全新的口令。基于这一发现，设计了一个基于模糊概率上下文无关文法(fuzzy PCFG)的口令强度评测算法fuzzyPSM。fuzzyPSM使用一个弱口令集A作为基础口令集，使用另外一个较强的口令集B作为修改口令集，接着学习用户从A取口令然后修改为B中相似口令的方法和相应频率，得到一个口令概率模型fuzzy PCFG。在口令评测阶段，基于fuzzy PCFG模型，对给定口令计算出一个概率值作为该口令的强度。大规模实验表明，在防御在

线口令猜测攻击方面，fuzzyPSM 优于学术界和工业界现有的 5 个主要算法，而防御在线攻击正应是 PSM 的主要目标。

(5) **严格证明基于口令的双因子认证协议实现匿名性的本质计算复杂度。**由于智能卡是资源受限设备，学者们尝试各种巧妙的办法，仅基于对称密码技术来设计匿名双因子认证协议。遗憾的是，其中绝大多数陆续被指出无法实现用户行为不可跟踪性。基于我们分析和研究300多个双因子身份认证协议的经验，发现那些存在匿名性问题的协议往往仅采用了对称密码技术，而实现了匿名性的协议往往采用了公钥密码技术。本文采用反证法，基于Halevi–Krawczyk (1999)和Impagliazzo–Rudich (1989) 这两个经典结果，在抗碰撞 Hash 函数存在的假设下，成功证明：公钥密码技术是实现匿名性的基本组件。这意味着，数以百计的仅采用对称密码技术的协议设计尝试都是徒劳，无论协议设计如何“巧妙”，证明多么“严格”。

(6) **构建一个基于口令的双因子认证协议评价指标体系。**系统研究了当前广泛采用的 Madhusudhan-Mittal(2012)双因子认证协议评价指标体系各元素间关系，指出并证明三个关键指标间存在本质冲突，无法同时实现。这意味着二十年来，数以百计的双因子认证协议无法实现所宣称的目标。此外，发现一些指标元素存在冗余和歧义。为此，本文深入分析了现有 5 个协议评价指标体系的优势和不足，在考虑安全威胁最新进展的基础上，构建了一个新的协议评价指标体系，并基于 67 个典型协议的评估结果显示新指标体系的可行性和先进性。评估结果不仅有利于学界更好地理解现有协议，而且揭示了设计实用双因子协议的挑战性。

(7) **设计一个可证明安全的基于口令匿名双因子认证协议。**基于上述理论成果，设计了一个“口令+智能卡”双因子协议，实现了随机预言机模型下可证明安全性。提出将“模糊验证因子”与“Honeywords”技术相结合的思想，使双因子认证协议可及时识别攻击者的在线猜测行为，在满足可用性指标的同时，实现超越传统上限的安全性。这一思想为本领域一个公开问题提供了肯定性的答案：能否在实现口令本地安全更新的同时，实现双因子安全性？基于14个大规模真实口令集，验证了所提思想的有效性。

1.5　论文结构

本文紧紧围绕 6 个拟解决的关键问题展开工作。由于这6个问题层层递进，论文后续章节内容依次安排如下：

第 2 章研究口令的分布问题，提出口令Zipf分布理论。首先指出现有两个相关研究的失误之处，提出两个新的口令分布模型，并通过决定系数和参数检验两个指标来评估所提模型的有效性，进一步设计一个口令分布强度评价标准，

最后探讨口令的Zipf分布新发现所带来的三项重要启示。

第 3 章关注如何科学评估定向在线口令猜测威胁，提出定向口令攻击理论。首先给出个人信息的明确定义和分类，然后揭示用户的三种脆弱口令行为，接着提出一个在线口令猜测框架TarGuess—包含7个理论猜测模型以应对 7 种不同能力的定向猜测攻击者，最后基于大规模真实数据显示TarGuess的有效性。

第 4 章研究如何及时检测口令文件的泄露，提出honeywords最优攻击理论。首先对honeywords系统中攻击者的能力和安全目标进行精确刻画；接着，针对Juels-Rivest提出的4个主要 honeywords 生成方法给出现实攻击方法，显示它们都存在严重安全缺陷；然后，提出一套honeywords最优攻击理论模型，并显示我们前述攻击方法是最优的；最后，基于最优攻击理论，利用PCFG、Markov、List三个口令概率模型及其定向版本，设计一套honeywords生成方法以适于4类不同安全威胁的应用环境。

第 5 章研究用户口令的强度评测问题，提出基于用户口令重用行为评价方法。通过对442个中文用户的口令使用习惯进行调研，指出当前主流PSM的基本假设不符合实际情况，进一步提出一个符合实际的基于用户口令重用行为的方法fuzzyPSM，最后在大规模真实数据集上测试了fuzzyPSM的准确性。

第 6 章关注如何在口令认证协议中实现匿名性，提出匿名性的本质计算复杂度理论。首先，分析了Fan 等和Xue等提出的两个匿名双因子认证协议，以它们为典型揭示此类协议设计中存在的挑战和难题；采用反证法，进一步提出一个协议设计基本原则：在非抗窜扰智能卡假设下，公钥技术是实现匿名性的基本组件；最后，显示所提原则的普适性。

第 7 章研究如何公平合理地评价口令认证协议，提出一套包含1个现实的攻击者模型和12项具体评价指标的系统性评价方法。首先，分析了Tsai等和Li等提出的两个双因子认证协议，揭示一些安全目标和理想属性间存在的冲突；接着，梳理了当前主流的Madhusudhan-Mittal协议评价标准各指标元素间的相互关系，显示该标准不足之处；然后，提出一个新的协议评价指标体系；最后，采用新的指标体系对67个具有代表性的协议进行评测，显示新指标体系的有效性。

第 8 章研究如何应用前述4项理论和2项方法，设计新的安全高效的口令认证协议。首次将系统安全中的Honeywords技术引入安全协议设计，提出将“Honeywords”与“模糊验证因子”相结合的思想，使双因子认证协议可及时识别攻击者的在线猜测行为，在满足可用性指标的同时，实现超越传统上限的安全性。在随机预言机模型下，证明了协议的安全性。新协议实现了第7章所提评价标准中所有12个评价指标，解决了 Huang 等2014年初提出的公开问题。

最后是本文的结束语部分，对论文研究工作进行总结和展望。

第二章　口令分布函数

“口令服从什么分布”是口令安全领域最为基础的公开问题之一。本章通过引入自然语言处理技术和统计计算理论，指出现有研究失败的根本原因，首次提出了两个口令分布理论模型：PDF-Zipf 和 CDF-Zipf。基于14个大规模真实口令集的实验表明，CDF-Zipf模型可以精确地刻画真实口令的分布，比如1600万Dodonew口令的真实分布和理论分布间的最大CDF误差仅为0.004926。本章将为后文6章奠定理论基础。

2.1　研究背景

口令自计算机诞生以来，一直是最流行的身份认证方法。虽然无处不在，但这种身份认证方法常处于两难境地：用户如何生成一个口令，使得强大攻击者难以破解但是普通用户却易于记忆？由于人类认知能力有限，用户难以记忆完全随机的口令。尤其随着需要管理的帐户的不断增多，记忆的负担不断增大[82]，用户选择的口令往往是高度可预测的[25, 65]，比如倾向于选择与日常生活相关的口令(如姓名，生日，爱好，甚至爱人和朋友的相关信息[103])。这意味着用户口令很可能是从一个狭小空间中选取的，易遭字典猜测攻击。

为解决这一著名的“安全性 vs. 可用性”两难问题，学者们提出了众多不同的口令生成策略，例如随机生成[143]，基于规则[37]和基于流行度[135]。它们的目标是使用户新创建的口令满足给定的规则，以达到一个可接受的口令强度。如表1.4和1.5所示，现实中不同网站执行的口令组成规则(PCR)和口令强度评测方法(PSM)各异。这种多样性的口令生成策略，导致同一口令往往在不同网站间得到迥异的境遇(见表1.6)：在有的网站被接受，在有的网站被拒绝。例如口令`password123`，被Gamil评价为“弱”而拒绝；在Apple评价为“中”，但因为不满足“1^+ lower,1^+ upper,1^+ digit”的口令组成规则PCR而被拒绝；在京东评价为“强”，并被接受。

产生这种严重不一致的口令评价结果和接受情况，直接原因是当前尚不存在统一的、公认的口令生成策略标准，更深层次原因是网站的差异化利益需求。口令研究领域一个难得的好消息是，如果设计合理，口令生成策略可以明显地提高口令安全性且不降低可用性[187]。因此，一大批学者致力于口令生成策略的设计和分析[73, 86, 93]。更严格的口令策略[Φ]可能使口令更难被攻破，然而用户构造和记忆口令的难度也随之增加，因此降低了可用性[110]。文献[188] 中的结果显

[Φ]本章中，“口令策略”如果没有特别说明，即为“口令生成策略”，二者表达相同的含义。

示，不恰当的口令策略会增加用户的精神和认知负担，对用户工作效率带来负面影响，最终的结果是，用户会竭力规避这种不友好的策略。

因此，不同服务类型的系统作出了不同的选择。对于电子商务网站如eBay，门户网站如Yahoo，订单接收网站如Kaspersky，可用性是重要的属性，因为任何隐性降低用户体验的做法都会导致用户流失，减少营利。所以他们倾向于使用更宽松的口令生成策略[113]。另一方面，对于安全攸关的网站，必须防范攻击者非法访问其上有重要价值的资源，例如保存敏感文件的云存储网站和处理课程分数的大学教务网站。因此它们更倾向于对用户口令进行严格的约束(如要求包含数字和特殊字符，拒绝诸如`pa$$word123`这样的流行口令)。

既然不同系统常常会执行不同的口令生成策略，一系列现实问题产生了：口令策略设计者如何评估策略的有效性？管理员怎样为系统选择合适的策略？另外，系统中的用户总体会随着时间的推移而变化，这导致即使口令策略保持不变，口令分布也会在一段时间(如一年)后产生很大的差异，特别是对大型的网络服务提供商来说。在这种情况下，安全管理员应该测量口令分布的强度，并可能需要据此调整口令生成策略。忽视口令分布的变化，或采取不恰当的举措，都可能带来后果巨大但并不易觉察的安全性和可用性问题。

因此，对口令分布的强度进行精确评测十分必要。如果缺少这一评测，安全管理员就难以回答“应如何调整口令策略”这一关键问题。换而言之，口令策略应该增强以增加安全性，或者保持不变，或者放松一点来换取可用性？总的来说，设计、选择和调整一个恰当的口令生成策略的关键，在于*如何精确地评测基于该策略产生的口令分布的强度*。需要注意的是，这里我们假设每一个身份认证系统都已经执行了某种口令策略(如[70, 189])，该策略的调整主要涉及更新一些规则和调整口令强度阈值。

2.1.1 研究动机

不可避免地，实现口令分布强度的精确评测依赖于另一个更为基础问题的成功解决：如何精确地刻画一个给定的口令分布？换而言之，用户生成的口令服从什么分布函数？尽管口令已经被研究了四十多年，这一基础性问题至今仍未解决。这有两方面原因：一是现实生活中口令是敏感数据，不容易收集，因而真实口令集匮乏；二是解决这个问题涉及到跨学科知识及其最新进展，比如计算统计学和自然语言处理(NLP)。因此，在众多必须依赖于口令的分布明确假设的工作中，绝大多数(如基于口令的密码协议中的认证[22]、加密[130]、签名[131]和秘密共享[132]，更多示例见节2.6.1) 不情愿(Unwittingly)地做了不合实际的假设：用户选择的口令服从均匀分布；剩余的工作中，极少数为了便于安全性分析，对口令分布作了各种各样的假设(如[71]中的二项分布假设

和[133, 134]中的最小熵模型)。至于那些不需要依赖明确口令分布假设的工作，它们通常只是简单地 *规避*口令分布假设，比如基于流行度的口令生成策略[135]和口令分布的强度评价指标[5]。

正如我们将在本章中证实的，上述关于口令分布的不合实际的假设常常会引起严重的安全性问题和可用性问题(如节2.6.1所示，文献 [21, 22, 130, 131, 136]中假设口令均匀分布，其安全规约的精度，相比于实际分布，降低了2到4个数量级)。至于那些规避口令分布假设的工作(如[5, 135])，实际上许多重要的特性和目标都依赖于未被讨论的假设。例如，如果一个web服务强制执行文献[135]中提出的口令构造策略，那么会有多大比例用户被烦扰？这种预测十分重要，但如果不对口令分布做任何假设，几乎不可能作出预测。

尽我们所知，Malone-Maher的工作[125]可能是与本章最为相关的工作。他们利用了四个真实口令集(其中三个的规模小于10^5)，首次尝试研究了口令的分布问题。因为人类语言服从Zipf分布[129]，Malone和Maher 把整个口令集输入Zipf模型，得到的实验结果显示：用户口令不大可能是Zipf分布。这一结果与我们将要展示的结果恰恰相反。几乎与Malone-Maher同时，Bonneau在文献[5]中也研究了口令的分布问题。Bonneau采用了与文献[125]本质上完全相同的数据拟合方法，自然地，也得到了与[125]相同的结论。因此，本章我们主要以Malone-Maher的工作为例进行讨论。

2.1.2　本章贡献

本章通过引入自然语言处理技术和统计计算理论，指出现有研究的根本失误之处，利用14 个大规模真实口令集，首次提出了两个口令分布理论模型：PDF-Zipf 和 CDF-Zipf。具体来说，本章主要取得了以下四方面贡献：

- **两个Zipf模型**。我们提出两个Zipf模型来刻画口令分布，其实验结果一致地显示用户口令满足Zipf定律。特别地，我们的PDF-Zipf模型适用于刻画流行口令，结果表明：(1)用户口令中易受攻击的部分(如频数$f \geq 4$ 的流行口令)*天然地*服从Zipf分布；(2)剩下的部分*很可能*服从Zipf 分布。受PDF-Zipf分布模型的启发，我们进一步提出了适于刻画整个口令集的CDF-Zipf分布模型，且得到了更好的拟合效果：实际数据和理论模型之间的累积分布函数(CDF)的最大误差为0.48%~4.59% (均值1.84%)，远优于PDF-Zipf模型下相应的结果6.26%~27.88%(均值 16.56%)。
- **一个评价指标**。我们提出了一个新的指标来评测给定口令分布的强度。这个指标利用了我们对口令分布的 *精确*刻画，所以它克服了现有的安全指标中的许多潜在问题(如基于攻击行为指标的不确定性[70] 和α-guesswork的非单调性[5])。我们的新指标能以严密的数学方式来评测给定的口令分

布(包括明文和hash形式的口令)的强度。这使得安全管理员可以更好地评价创建口令集时的口令策略的安全合理性。

- **三项启示**。基于CDF-Zipf这一精确口令分布模型，可以推导出一系列函数式用以：(1)精确刻画口令猜测攻击者的优势；(2)预测基于流行度的PCR带来的可用性降低程度；(3)研究α-guesswork的单调性。这些都是前人想做，因尚不知道具体的口令分布函数，而没有做到的。
- **大规模的评估**。我们的实验评估基于14个大规模真实口令集，具有广泛的代表性：来自4种不同语言，共包含1.13亿口令，涵盖12种主流的网络服务和5种口令构造策略。大规模实验结果表明，每个用户口令都可被看作从Zipf 概率分布空间中抽取的一个实例。这否定了[5, 125]中宣称的“用户口令不太可能服从Zipf分布”。

2.2 相关工作

本节我们简要地回顾口令生成策略和口令攻击方面的一些相关工作。

2.2.1 口令生成策略

1990年，Klein[6]率先提出了“主动式口令检查”的概念，通过事先检查新提交的口令是否安全，来促使用户产生更安全的口令分布。口令强度的检查标准可分为两类：一类是构造一个口令组成规则PCR，如最小长度限制和字符种类要求；另一类是使用一个口令强度评价算法，根据安全阈值决定接收还是拒绝用户口令，最典型的就是维护一个“口令黑名单”。尽管作者称这项技术为“主动式口令检查”，事实上这就是我们今天所熟知的“口令生成策略”(见图1.9)。

自Klein的开创性工作之后，学者们提出了一系列主动式口令检查方法，研究重点在于如何减少匹配用户口令与黑名单中的“弱”口令时所消耗的时间和空间(如Opus[106])。也有学者尝试针对不同网站，设计可调的口令生成策略，其中最有影响力的当属美国国家身份认证指南NIST SP800-63[37]。然而，通过使用口令攻击算法攻击在不同口令组成规则(PCR)下模拟生成的口令集，Weir等[93]发现，口令组成规则PCR无法确保口令分布达到预期的安全性。

2012年，Houshmand和Aggarwal[70]提出一种既能增强安全性又不降低可用性的全新口令策略：它首先根据基于PCFG的攻击结果，分析用户选择的口令是弱口令还是强口令，如果是弱口令则进行轻微修改以达到可接受的强度。Houshmand-Aggarwal 的这个策略就是我们今天所熟知的“口令强度评价器(PSM)”(见图1.9)。与此同时，Claude等[71]也指出了一个类似的基于Markov的PSM。Houshmand-Aggarwal 的PSM 能更准确地评测一个单独口令

的强度，并且相比PCR，可进行更灵活的调整，因为其调整只涉及更新一段连续区间内的某个阈值点。2014年，Das等指出，用户在注册新帐户时，常重用(或简单修改)一个现有的口令，而不是构造一个全新的口令[7]。这意味着现有的PCR和PSM都没有考虑这一现实，对口令的评测是不准确的。

Schechter等[135]提出的基于流行度的口令生成策略，可能与本章的口令分布强度指标(见节2.5)最为相关。他们创新性地提出使用基于流行度的预言机来替代传统的口令生成策略(如PCR和PSM)，如果一个口令超过给定的流行度阈值$\mathcal{T}$(如$\mathcal{T}=\frac{1}{10^4}$)则被拒绝。这一策略能有效阻止漫步猜测攻击(见表1.3)：漫步攻击者猜测一次的成功率不会超过$\mathcal{T}$。这对那些拥有千万级以上用户的大型服务系统来说，尤其有效。但是，Schechter等[135]策略最关键之处是如何选取阈值$\mathcal{T}$。如果$\mathcal{T}$ 过大，会危及安全性；如果$\mathcal{T}$过小，会有大批用户被要求重新选择口令，增加用户烦扰，降低系统可用性。针对互联网级系统，Schechter等建议$\mathcal{T}=\frac{1}{10^6}$。但给定$\mathcal{T}$，到底会在多大程度上影响可用性，鲜有公开研究。

基于提出的CDF-Zipf模型，我们建立了理论预测模型并发现，如果直接使用 Schechter 等建议的阈值$\mathcal{T}=\frac{1}{10^6}$[135]，会拒绝相当大比例的用户，给用户带来负担和困扰，严重降低系统可用性。比如，天涯(www.tianya.cn) 4000万用户中，34.89%使用了频率高于$\mathcal{T}=\frac{1}{10^6}$的口令，这意味着超过三分之一的用户都有潜在可能被拒绝接受现有口令，被要求生成和记忆一个新口令。如果能够合理选择阈值$\mathcal{T}$，Schechter等的策略仍有广泛应用前景，而我们节2.6的预测模型正致力于解决这一关键问题。

2.2.2　口令攻击

基于口令的身份认证技术存在多种攻击方式，如在线口令猜测攻击、离线口令猜测攻击、键盘记录、肩窥、社会工程学等等。本章只考虑前两种攻击，因为其它攻击方式与口令强度和口令集的强度均无关，故不在本章讨论的范围之内。在线口令猜测的防御措施主要有基于机器学习的检测算法[19]、登录速率限制或帐户锁定策略[37]等，这些手段均不涉及密码学。相反的，离线猜测在攻击者$\mathcal{A}$的本地机器上进行，完全由攻击者$\mathcal{A}$控制，因此如果有充足的时间和强大计算能力，$\mathcal{A}$可以进行足够多的猜测。

Fiorencio等[190]探讨了离线猜测会真正构成威胁的一些场景，发现一个口令在抵抗这两种不同猜测时，存在巨大的强度“鸿沟”：“鸿沟”的左端是抗在线猜测所需要的强度，“鸿沟”的右端是抗离线猜测所需要的强度，在“鸿沟”内渐近地提高口令强度不会带来任何安全性提升。因此，他们对促使用户生成强口令以抵抗离线猜测的作法提出了质疑。不难看出，如果口令以salted-hash 形式被泄露，但网站在短时间内(如几天)及时发现，这样的“鸿沟”(包括相应的

质疑)显然会消失。在这段时间内，离线猜测构成现实威胁，在“鸿沟”内渐近地提高口令强度可带来切实的安全性提升。这从侧面显示了服务器正确存储用户口令和及时检测口令文件泄露(如honeywords技术[191])的重要性。

因此，恰当地评测基于口令的认证系统抗离线猜测的能力是至关重要的。在本章中，这主要通过对比搜索空间的大小(如猜测的次数)和在该搜索空间下，可猜测出的口令比例来实现。这个方法只依赖于攻击技术和用户选择口令的方式，既与系统的独特配置环境无关(如使用了什么Hash函数，SHA-1、PBKDF2还是CASH[192]?)，也与攻击者的能力无关。系统环境和攻击者的能力决定的是攻击者每一次猜测付出的代价[104]。针对离线猜测的一些安全对策，如加盐来对抗预计算技术(如彩虹表)或者增加Hash循环次数，可提高攻击的单次成本，它们只是衡量口令系统对离线攻击的抵抗能力时的一个参数。结合考虑这个单次成本和搜索空间的大小，我们就可以明确地计算攻击者$\mathcal{A}$的成本-收益。本章采用这一评测方法。

口令搜索空间的大小本质上取决于用户选择口令的方式。众所周知，用户倾向于选择易于记忆的口令，如字典中的单词或者个人相关信息(姓名、生日等)[65, 193]。然而由于口令生成策略的限制，用户极少原封不动地使用这些信息，而是先经一定的便于记忆的方法修改后使用。比如，流行口令 `pa$$word`就是由单词`password`跳变两个字符生成的。

为模拟用户的口令生成行为，学者们设计了各种启发式变换规则(如插入、删除、字符跳变)来对字典中单词生成变种。一些广泛使用的基础词典可参见[194]。这种模拟技术最早出现在1979 年Morris-Thompson[1]的工作中，他们分析了3000 个真实口令抗离线猜测的强度。这一开创性工作吸引了大量后续的跟进研究(如[6, 66, 109])，甚至一些专门的自动化软件破解工具也出现了。比如，著名的John the Ripper(JtR)[195]，至今仍偶尔有口令研究(如[68, 187, 196])利用这些破解软件来实施字典攻击，在这些研究中安全性往往是它们的次要目标。

直到近几年，口令攻击研究才逐渐由艺术发展为科学。Narayanan 和 Shmatikov[66]首次提出了一种基于Markov链的先进口令攻击算法，该算法不再构造启发式变换规则来模拟用户口令生成行为，而是生成音律上与单词十分相近的猜测。Narayanan-Shmatikov算法的测试集由142个Hash口令组成，成功破解了其中的96个(67.6%)。然而，他们的算法并不是标准的基于字典的猜测攻击，因为它只能产生音律上与单词相似的口令，其它类型的口令无法自动生成。此外，测试集过小，无法准确评估算法的有效性。

2009年，Weir等[68]提出一种全新的基于概率上下文无关文法(PCFG)的口令攻击算法。PCFG-based算法可通过训练集来自动学习用户的口令生成行为，并在大规模真实口令集上测试了有效性。在该算法的训练阶段，训练集中每

个口令被视为字母段(记作L)、数字段(D)和特殊字符段(S)的组合。进一步，口令pa$$word123被抽象为结构 $L_2S_2L_4D_3$。通过对训练集中每个口令进行字段抽取和结构抽取，就自动获得了用户的口令变换规则，形成一个概率上下文无关文法(PCFG)。在该算法的猜测生成阶段，由于任一PCFG文法都对应一个语言，该语言中的每个单词即为一个口令猜测，其中每个猜测都可计算出一个概率。最后，将这些猜测按概率递减排序，即得到猜测序列。同样的猜测次数下，PCFG算法[68]可比JTR[195]多破解28%到129%的口令。

2014年，Ma等[34]引入平滑化和正规化等自然语言处理技术，改进了基于Markov链的口令攻击算法。他们发现，如果选择了正确的阶数和恰当的方法来解决数据稀疏和正规化问题后，Markov-based攻击算法将优于PCFG-based攻击算法。2015年，Ur等[197]调研了：(1)学者们提出的上述攻击算法与现实中黑客们使用的算法相比，有何优劣；(2)攻击算法的选择将如何影响研究结论。他们发现每种攻击算法都对配置非常敏感，基于单一的攻击方法来评测单个口令的强度可能会低估口令在经验丰富的攻击者面前的脆弱性，但口令分布的强度的评测可以依据单一攻击方法。

2016年，Li等[103]首次研究了攻击者利用个人显式PII可在何种程度上提升猜测效率，提出了定向在线猜测算法Personal-PCFG。在12306.cn网站泄露的13 万带6种显式PII数据上的攻击结果显示，Personal-PCFG效率可比传统的漫步猜测算法 PCFG 高出309%∼634%。但是，如节1.2.3所指出的，Personal-PCFG[103]的PII匹配方法在严重高估一些PII的使用频率的同时，会低估严重另外一些PII的使用频率，大大降低了算法的有效性。

2.3　预备知识

本节我们首先描述将使用到的数据集，然后展示用户口令的一些基本信息，最后介绍一些将使用到的统计学习技术。

2.3.1　口令数据集的介绍

我们收集了过去10年间泄露的14个大规模真实口令集(见表2.1)。它们来自不同服务类型和规模的网站，口令的泄露方式，用户的地理位置、语言、信仰和文化背景也各异，这将充分显示本章所提出的模型是普适的，可被用于精确刻画用户口令的分布特性。这14个口令集都或被攻击者窃取，或被内部人员泄露，随后被公开在互联网上。一些早期泄露的口令集已广泛在口令研究中使用(如[34, 92, 93, 197])。我们意识到，尽管这些口令集是公开的，它们包含了邮箱、姓名和口令等个人隐私信息。因此，我们将所有的用户姓名、邮箱保密，仅展示有关口令的聚合信息，这样将避免对受害者造成二次伤害。攻击者很可

表 2.1 本章使用的14个真实口令集的基本信息

口令集	服务类型	地址	语言	泄露时间	泄露途径	口令总数	独立口令数
Tianya	社交网站	中国	中文	2011年12月	黑客攻击	30,233,633	12,614,676
Dodonew	游戏、电子商务	中国	中文	2011年12 月3日	黑客攻击	16,231,271	11,236,220
CSDN	程序员论坛	中国	中文	2011年12月	黑客攻击	6,428,287	4,037,610
Rockyou	游戏	美国	英文	2009年12月	SQL注入	32,603,388	14,341,564
000webhost	免费建站空间	美国	英文	**2015年10月**	PHP编程错误	15,251,073	10,583,709
Battlefield	游戏网站	美国	英文	2011年6月	黑客攻击	417,453	542,386
Myspace	社交网络	美国	英文	2006年10月	钓鱼攻击	41,545	37,144
Single.org	约会	美国	英文	2010年10月	查询字符串注入	16,250	12,234
Faithwriters	写作论坛	美国	英文	2009年3月	SQL注入	9,709	8,347
Hak5	黑客论坛	美国	英文	2009年7月	黑客攻击	2,987	2,351
Gmail	邮箱	俄罗斯	主要为俄文	2014年9月	钓鱼攻击	4,929,090	3,132,028
Mail.ru	邮箱	俄罗斯	俄文	2014年9月	钓鱼软件	4,932,688	2,954,907
Yandex.ru	搜索引擎	俄罗斯	俄文	2014年9月	钓鱼软件	1,261,810	717,203
Flirtlife.de	婚恋网站	德国	德文	**2006年5月**	黑客攻击	115,589	343,064

能已利用这些口令集作为训练集或者攻击字典，来危害用户的帐户安全；我们对它们的利用，将有助于系统管理员和普通用户更好保护帐户的安全。

前3个数据集(Tianya、Dodonew和CSDN)来自中文网站。我们以相应网站的域名来对口令集命名(如“Tianya”数据集就来自www.tianya.cn)。这些网站在2011 年 12 月被攻陷[28]，口令文件在互联网上可公开下载，我们在当时收集到了这些公开数据。Tianya是中国影响最大的论坛之一；Dodonew也是一个流行游戏论坛，且能进行货币充值；CSDN是中国最大的程序员社区网站。

接下来的7个口令数据集(Rockyou、000webhost、Battlefield、Myspace、Singles.org、Faithwriters、Hak5) 来自英文网站。3260万Rockyou数据来自一个游戏论坛，泄露于2009年12月[198]。1500万000webhost数据泄露于2015年10月，000webhost官方确认了这一泄露，且公告称这是攻击者利用一个旧版本PHP程序造成的后果[199]。54.2万Battlefield口令是名为LulzSec的黑客组织在2011年6月公开的。4.1万的Myspace口令泄露于2006年10月，Myspace是美国著名社交网站。攻击者创建了虚假的Myspace登录界面，然后实施了一个典型的社会工程学(即网络钓鱼)攻击。由于现存若干个Myspace数据集的版本，可能因此不同学者会使用不同时期的版本，我们使用[194]中提及的包含41,545个明文口令的版本。“Singles.org”和“Faithwriters”的用户基本都是基督教徒: Singles.org是一个标识为基督徒的约会网站，Faithwriters.com是一个基督徒写作论坛。前者通过查询字符串注入入侵，有16,250个口令被泄露；而后一个被SQL注入，泄露了9,709个口令。 Hak5数据集被一个名为ZF0(Zero for 0wned)的组织泄露[200]，它只是`www.hak5.org`网站整个口令集的一小部分。令人奇怪的是，Hak5是一个黑客站点，其口令分布是本章最弱的一个(见节2.5)。本章中，我们将这个数据集作为正常口令分布的一个反例。

接下来3个口令集(492.9万Gmail，493.2万Mail.ru，126.2万Yandex.ru)来自俄文用户，2014 年9月被俄罗斯黑客泄露，且其中90%以上仍是活跃用户[201]。据称这三个网站并未遭受入侵，这些口令是攻击者通过网络钓鱼和其它形式攻

击(如键盘记录)捕获。

最后一个口令集来自德国婚恋网站Flirtlife.de，攻击者窃取了用户口令文件，内含343,064个口令。这一泄露得到Flirtlife.de官方确认[202]。在2006年5 月泄露被发现时，这些用户里仍有一半是活跃的。

2.3.2　用户口令的基本信息

表2.2显示了各个口令集的字符组成信息。中文俄文用户更倾向于仅使用数字构造口令，而英文用户更喜欢使用字母。这与文献[84]一致。一个合理的解释可能是，母语非英语的用户不太熟悉英文单词和字母。000webhost中极少有口令单独由字母或数字构成，但有54.42%的口令由小写字母后接数字组成，另外仅有7.92%口令含特殊字符，我们猜测000webhost在泄露前可能执行了一个“包含字母+数字”的口令策略[①]。有趣的是，18.24%的Myspace用户通过在小写字母序列后添加数字“1”来构造自己的口令。这可能是由于Myspace强制用户口令包含至少一个数字；超过10%的Mail.ru口令由数字后面接字母组成，这远远高于其它网站，原因尚不得知。

表2.3显示了用户口令的长度分布。我们可以看到，最流行的口令长度介于6 到10之间，平均占整个数据集的85.01%。很少有用户选择长度超过12的口令，而Dodonew和000webhost除外。一个原因可能是，Dodonew是一个允许货币充值的网站，用户认为他们的帐户非常重要，因此选择更长的口令；000webhost是一个免费建站空间，用户在其上建立的站点往往属于重要资产，并且用户都是网站管理员，因此倾向于选择更安全更长的口令。有趣的是，相比于其它口令集，CSDN口令集长度为6和7的口令要少得多。这可能是由于CSDN(以及许多其它的网络服务)在刚建站时，执行了一个宽松的口令策略，随后执行了一个严格的策略(如要求口令长度不小于8)。000webhost、Battlefield和Yandex.ru中长度小于6的口令都低于1%，一个可能的原因是这些网站后来执行了“口令长度不低于6”的策略。我们也注意到，Singles.org 的口令都不超过 8 个字符，这可能是由于Singles.org 阻止用户选择长于 8 的口令。许多金融类公司现今仍执行这样的口令生成策略[203]，一个可能的原因是增加口令长度会导致系统兼容问题。

2.3.3　线性回归

在统计学中，广泛使用线性回归来拟合可能存在线性关系的观测数据，从而来求解两个变量之间的关系。通常，线性回归是指在给定x值的条件下，y的条件均值是x的仿射函数：$y = a + b \cdot x$，其中x是自变量，y是因变量。b是拟合

[①]这一猜测得到了证实，见http://www.liangxin.name/?post=403

表 2.2 口令集的字符组成信息

Dataset	[a-z]+	[A-Z]+	[A-Z a-z]+	[0-9]+	[a-zA -Z0-9]+	[a-z]+ [0-9]+	**[a-z]+1**	[a-zA-Z]+ +[0-9]+	[0-9]+ [a-zA-Z]+	[0-9]+ [a-z]+
Tianya	9.96%	0.18%	10.29%	**63.77%**	98.05%	14.63%	0.12%	15.64%	4.37%	4.11%
Dodonew	8.79%	0.27%	9.37%	**20.49%**	82.88%	**40.81%**	1.39%	42.94%	7.31%	6.95%
CSDN	11.64%	0.47%	12.35%	**45.01%**	96.31%	26.14%	0.24%	28.45%	6.46%	5.88%
Rockyou	41.71%	1.50%	44.07%	15.94%	96.25%	27.70%	4.55%	30.18%	2.75%	2.54%
000webhost	**0.04%**	**0.00%**	0.26%	**0.02%**	**93.08%**	54.42%	4.66%	60.95%	8.43%	7.28%
Battlefield	32.11%	0.29%	34.01%	9.23%	98.06%	34.93%	5.08%	39.58%	3.39%	3.05%
Myspace	7.18%	0.31%	7.66%	0.71%	89.95%	**65.66%**	**18.24%**	**69.77%**	6.02%	5.66%
Singles.org	60.20%	1.92%	**65.82%**	9.58%	99.78%	17.77%	2.73%	19.68%	1.92%	1.77%
Faithwriters	54.40%	1.16%	**59.04%**	6.35%	99.57%	22.82%	4.13%	25.45%	2.73%	2.37%
Hak5	18.61%	0.27%	20.39%	5.56%	92.13%	16.57%	2.01%	31.80%	1.44%	1.21%
Flirtlife.de	57.46%	8.34%	73.29%	13.37%	97.95%	6.78%	0.85%	7.88%	0.65%	0.59%
Gmail	39.87%	0.00%	39.87%	**15.70%**	98.04%	31.33%	3.76%	31.33%	2.93%	2.93%
Mail.ru	24.18%	0.32%	26.37%	**24.66%**	97.43%	20.28%	0.75%	22.97%	**12.50%**	**11.65%**
Yandex.ru	17.30%	0.25%	18.51%	**48.19%**	98.42%	15.24%	0.66%	17.21%	6.48%	6.02%

*注：前两行为正则表达式，如`[a-z]+`表示仅由小写字母构成的口令，`[A-Za-z]+`表示仅由字母构成的口令，`[a-zA-z]+[0-9]+`表示小写字母串后面接数字串的口令。

表 2.3 口令集的长度分布

Length	1-5	6	7	8	9	10	11	12	13-16	17-30	30+
Tianya	1.81%	33.77%	13.92%	18.10%	9.59%	10.28%	5.53%	2.88%	4.05%	0.07%	0.00%
Dodonew	1.84%	9.71%	13.45%	18.49%	20.29%	14.69%	3.10%	1.34%	10.24%	6.79%	0.04%
CSDN	**0.63%**	**1.29%**	**0.26%**	36.38%	24.15%	14.48%	9.78%	5.75%	6.96%	0.32%	0.00%
Rockyou	4.31%	26.05%	19.29%	19.98%	12.12%	9.06%	3.57%	2.10%	3.12%	0.39%	0.00%
000webhost	**0.02%**	**5.70%**	**7.92%**	21.81%	15.41%	14.51%	10.49%	7.67%	14.35%	2.13%	0.00%
Battlefield	**0.00%**	20.29%	14.67%	28.75%	14.91%	10.25%	5.02%	3.12%	2.83%	0.15%	0.00%
Myspace	1.54%	15.67%	23.40%	22.78%	17.20%	13.65%	2.83%	1.13%	1.15%	0.48%	0.17%
Singles.org	13.10%	32.05%	23.20%	31.65%	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**
Faithwriters	1.17%	31.97%	20.95%	22.71%	10.35%	5.98%	3.24%	1.87%	1.53%	0.20%	0.01%
Hak5	1.71%	12.96%	8.50%	20.89%	8.94%	30.83%	3.58%	3.08%	6.90%	2.44%	0.17%
Flirtlife.de	27.01%	25.82%	12.29%	20.84%	6.24%	3.19%	1.74%	1.19%	1.37%	0.31%	0.00%
Gmail	4.13%	18.73%	13.49%	28.94%	13.86%	13.86%	3.10%	1.90%	1.81%	0.18%	0.00%
Mail.ru	2.04%	24.09%	14.76%	23.63%	11.50%	8.36%	5.24%	4.12%	5.48%	0.75%	0.00%
Yandex.ru	0.64%	30.17%	13.85%	22.40%	10.36%	8.24%	5.70%	3.60%	4.17%	0.87%	0.00%
Average	4.40%	20.95%	14.53%	24.27%	12.62%	11.23%	4.61%	2.85%	3.94%	0.58%	0.03%

回归线的斜率，a是截距。线性回归最常用的方法是最小二乘法。此方法通过计算从每个数据点到回归线的垂直距离的总和，使该总和最小的回归线即为最优回归线。比如，如果一个点正好位于回归线上，那么它的垂直距离是0。在拟合过程中，决定系数$R^2 \in [0,1]$是衡量回归线与实际数据点拟合优劣的一个评价指标：*R^2越接近1越好*。决定系数$R^2 = 1$则表明所有数据点完全在回归线上。

2.3.4 Kolmogorov–Smirnov检验

除了决定系数R^2，我们进一步使用假设检验来测量样本和理论分布模型之间的“距离”。由于口令不太可能服从正态分布，所以应采用非参数检验。Kolmogorov–Smirnov(KS)检验是针对离散数据的最流行的非参数检验方法[204, 205]。它检验的是实际数据的累积分布函数(Cumulative distribution function，CDF)$F_n(x)$与理论分布的CDF $F(x)$的之间的距离：

$$D = \sup_x |F_n(x) - F(x)|,$$

其中n是样本大小，$\sup_x$是距离集合的上确界。那么，$D \in [0,1]$表示两条CDF曲线$F_n(x)$和$F(x)$之间的最大距离，故D越小越好。

可以采用此统计量D进行严格的假设检验。需要注意的是，本章中的口令

分布理论模型是通过每个口令集的数据计算得到的，因此口令分布会随口令集(样本)选取的不同而不同。换句话说，口令分布不是固定不变的。因此，我们不能通过标准公式$\Pr(\sqrt{n}D > x) = 2\sum_{i=1}^{\infty}(-1)^{i-1}e^{-2i^2x^2}$来计算 p- 值。我们采用文献[205]的节4.1中推荐的Monte Carlo方法：(1)使用理论模型(如PDF-Zipf)来拟合一个给定的口令集，得到口令分布函数，并计算相应的 KS 统计量D；(2)使用从实际口令集拟合得到的分布参数，生成一些(如建议的2500 个)合成数据集；(3)使用理论模型来拟合每个合成数据集，并计算出相应的合成KS距离D'；(4)p-值定义为所有合成 KS 距离D'中值大于D的比例。

更具体地说，基于Monte Carlo的 KS 检验方法执行如下步骤：(1)先对原始样本$sample_0$做参数估计，求出理论分布fit_0；(2)根据理论分布fit_0的参数抽样，抽取n个合成样本($10^3 \leq n \leq 10^4$)，记为$sample_i(1 \leq i \leq n)$，每个样本大小与原始样本$sample_0$相同；(3)对$n$个样本$sample_i(1 \leq i \leq n)$使用与原始样本$sample_0$相同的方法做参数估计，求出$n$个合成样本对应的分布$fit_i$；(4)求$sample_0$和$fit_0$的 KS 距离$D_0$，$sample_i$与$fit_i$ 的 KS 距离$D_i(1 \leq i \leq 2500)$，则p-值为$\frac{|\{i|D_i>D_0,1\leq i\leq n\}|}{n}$。本章取文献[205]中推荐的$n = 2500$。

我们的原假设是口令分布理论模型(PDF-Zipf或CDF-Zipf)是可以接受的，反之为备择假设。大的p-值使我们更有信心相信，实际口令数据的分布与理论模型没有显著差异。

2.4　用户口令中的Zipf定律

本节我们提出两种Zipf理论模型来刻画用户口令的分布。

2.4.1　PDF-Zipf模型

最初，概率上下文无关文法(PCFG)这种机器学习技术广泛在自然语言处理(NLP)领域应用，但Weir等[68]成功利用它来自动学习用户的口令构造行为。最近，NLP技术在口令研究中也显示了更多的用武之地，比如用来评估口令中的语义对安全性的影响[69]，用来解决口令概率模型的稀疏性问题[34]等等。

受这些工作启发，本章试图探讨存在于自然语言中的Zipf定律[Φ]是否也存在于口令中。Zipf定律最早是用来刻画自然语言中单词的“排名 vs. 出现频率”关系：在语料库里，一个单词出现的频率与它在频率表里的排序成反比。更确切地说，对于按频率递减排序的自然语言语料库，单词的排名r和它的频率f_r满足关系：$f_r = \frac{C}{r}$，其中C是一个取决于特定语料库的常数。这意味着，最流行的单词出现频率大约是第二流行的单词的两倍，第三流行的单词的三倍，以此类

[Φ]Zipf分布也被称作Pareto分布或幂率分布，当统计量连续时它们可以彼此互相推导得出[206]。

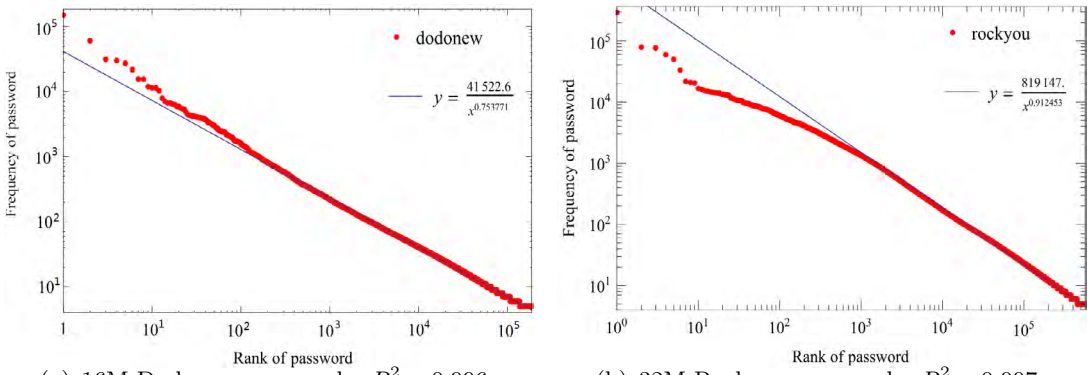
推。Zipf定律已被成功地(即$R^2 \approx 1$)用来刻画战争频繁程度分布[205]，软件包大小分布[207]和互联网的拓扑结构[208]。

有趣的是，通过去除每个口令集中最不流行那些口令(即口令出现少于3或5次)，然后进行线性回归拟合，我们发现现实中用户的口令分布服从相似的规律：对于口令集$\mathcal{DS}$，一个口令的排名 r 及其频率 f_r 满足方程

$$f_r = \frac{C}{r^s}\,, \tag{2.1}$$

其中C和s是依赖于口令集$\mathcal{DS}$的常数，可能是由许多复杂的深层次因素决定的，比如网站的服务类型、采取的口令策略，以及用户的人口统计学因素(如年龄、性别、教育程度、职业和语言)等等。通过绘制双对数坐标图(在本章中以10为底取对数)，坐标轴分别取log(排名顺序)和log(频率)，Zipf 定律表示为一条直线，可以被更容易地观察到。换句话说，$\log(f_r)$与$\log(r)$是线性相关的：

$$\log f_r = \log C - s \cdot \log r\,. \tag{2.2}$$



(a) 16M Dodonew passwords: $R^2 = 0.996$　(b) 32M Rockyou passwords: $R^2 = 0.997$

图 2.1 使用PDF-Zipf模型拟合真实口令集。Dodonew包含1600万中文用户的口令，RockYou包含3200万英文用户的口令。虽然少数流行的口令不在拟合线上，但是相比于那些在拟合线上的点，它们的数目可以忽略不计。

图2.1(a)显示，Dodonew的1623万口令服从Zipf定律，决定系数 $R^2 = 0.995531$，非常接近于1。这表明，回归线 $\log f_r = 4.618284 - 0.753771 \cdot \log r$ 很好地拟合了Dodonew的流行口令(即$f_r \geq 5$的口令)。安全问题正是源自口令集中那部分流行的口令，因为它们都是在猜测攻击下脆弱的口令：聪明的攻击者会首先尝试最流行的口令，然后第二流行的口令，以此类推，这样收益最大[5]。如图2.1(b)和图2.2 所示，其余12个数据集的流行口令也满足Zipf定律，相应的回归线能很好地吻合来自各个口令分布的数据点。由于版面的限制和前文提到的Hak5口令集的缺陷，在这里我们不展示其Zipf 曲线，实际上其线性拟合也有很高的决定系数$R^2 = 0.923$。由于Zipf 理论模型是尝试模拟口令的概率分布函数(Probability distribution function, PDF)，故我们称之为PDF-Zipf模型。

表 2.4 PDF-Zipf模型：采用线性回归拟合14个口令集

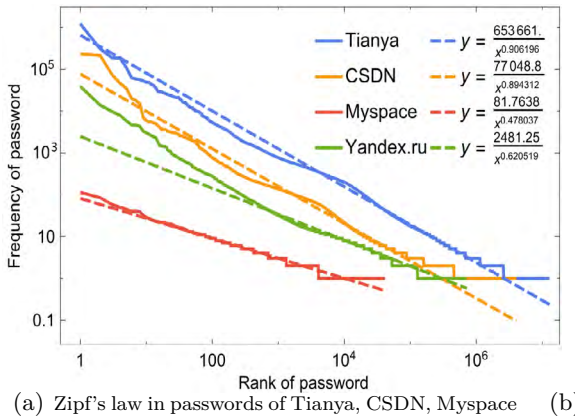
Dataset	Least frequency	Fraction of PWs in LR	Unique PWs in LR(N)	Zipf regression line ($\log y$)	R^2	KS test Statistic D	KS test p-value
Tianya	5	0.50443286	486,118	$5.806523 - 0.905773*\log x$	0.994204954	0.161718	$< 10^{-4}$
Dodonew	5	0.21640911	187,901	$4.618284 - 0.753771*\log x$	0.995530686	0.164640	$< 10^{-4}$
CSDN	5	0.29841262	57,715	$4.886747 - 0.894307*\log x$	0.985106832	0.268982	$< 10^{-4}$
Rockyou	5	0.49600581	563,074	$5.913362 - 0.912453*\log x$	0.997298647	0.193567	$< 10^{-4}$
000webhost	5	0.19687867	229,725	$7.354124 - 0.624446*\log x$	0.989437653	0.111546	$< 10^{-4}$
Battlefield	3	0.20773582	15,178	$3.214071 - 0.673161*\log x$	0.98384909	0.209718	$< 10^{-4}$
Myspace	3	0.08094836	706	$1.722674 - 0.459808*\log x$	0.965861431	0.126651	$< 10^{-4}$
Singles.org	3	0.22135384	658	$1.875405 - 0.518096*\log x$	0.970277755	0.100741	$< 10^{-4}$
Faithwriters	3	0.12472963	242	$1.583425 - 0.486348*\log x$	0.974175889	0.140562	$< 10^{-4}$
Hak5	3	0.15400067	76	$1.579116 - 0.643896*\log x$	0.922662999	0.278761	$< 10^{-4}$
Flirtlife.de	3	0.52644283	6,898	$3.305541 - 0.752347*\log x$	0.986011446	0.066174	0
Gmail	5	0.29617143	77,397	$4.903847 - 0.799443*\log x$	0.995817202	0.217463	$< 10^{-4}$
Mail.ru	5	0.33034872	83,914	$4.332851 - 0.732599*\log x$	0.970047769	0.168754	$< 10^{-4}$
Yandex.ru	5	0.34210777	26,003	$3.394671 - 0.620519*\log x$	0.972507203	0.097279	$< 10^{-4}$

PW=Passwords; LR=Linear regression; R^2=Coefficient of determination.

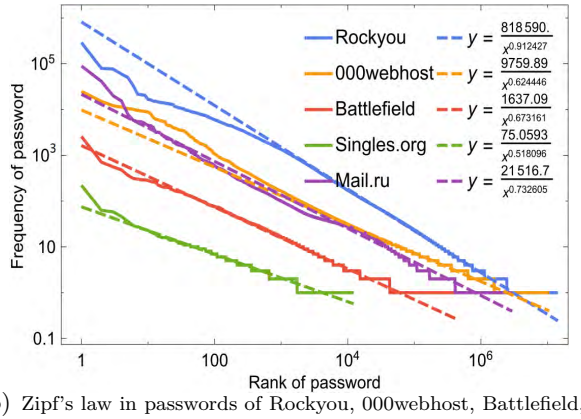
如表2.4中R^2这一列所显示的，每个线性回归(除了Hak5)的决定系数都大于0.965，接近于1，表明非常成功的拟合。至于“Hak5”，其R^2值为0.923，尽管可以接受，但仍不如其它口令集。一个可能的原因是它仅含有2000多个口令，不能很好代表整个www.hak5.org的口令分布。更重要的是，数据泄漏的方式可能直接影响R^2。如表2.4所证实的，相比那些因网站被攻陷而泄露口令集，因网络钓鱼攻击而泄露的口令集一般有较低的R^2值。这是因为网络钓鱼攻击一般只能获得网站整个口令数据的有限部分，而如果成功攻陷网站，该网站所有的用户口令会都会被攻击者得到。

但是，我们目前尚不能得出结论：口令的整个分布服从PDF-Zipf模型。实际上，KS检验都得到了足够低的p-值(见表2.4参数)，我们有足够的信心拒绝假设“口令服从所拟合的分布”。需要特别注意的是，这并不与我们的结论“用户口令服从Zipf定律”相矛盾：如果我们可以更准确地计算出Zipf模型参数，我们很可能可以获得足够大的p-值，顺利通过KS检验。节2.4.3中，我们通过在PDF-Zipf模型基础上，提出改进的CDF-Zipf模型，来证实这一点。

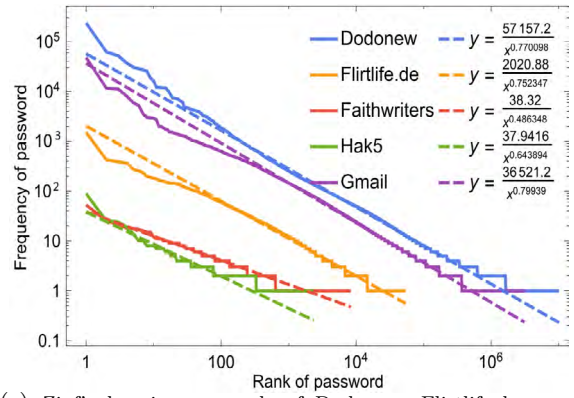
我们将在节2.4.2详细说明为什么需要去除那些低频口令后再拟合。至于为什么选择一个较小的值(如3或5)作为最小频次的阈值(LF)，其基本思想来自于统计学中发现的一个规律(见文献[205]的图3)：当样本大小小于样本空间时，线性回归先随着LF逐渐增加而变优，直到达到最优点$\hat{p}$，之后随着LF 逐渐增加，回归缓慢地恶化(由于减少了拟合中使用的样本数量)。我们进行了一系列的增量实验，以找到确切的LF使得回归达到$\hat{p}$，发现对于百万以上的大规模数据集来说，一个可行的设置是令$LF = 5$，否则令$LF = 3$。需要注意的是，要衡量一个模型是否适于刻画一个数据集，分布函数$f(x)$ 应至少在2~3个数量级的区间$[x_{min}, x_{max}]$内有效[207]，即$x_{max}/x_{min} \geq 10^{2\sim3}$。除了Hak5，前述所有13 个口令集得到的拟合分布函数均满足这个条件。



(a) Zipf's law in passwords of Tianya, CSDN, Myspace and Yandex.ru



(b) Zipf's law in passwords of Rockyou, 000webhost, Battlefield, Singles.org and Mail.ru



(c) Zipf's law in passwords of Dodonew, Flirtlife.de, Faithwriters, Hak5 and Gmail

图 2.2 在“对数-对数”坐标系下采用PDF-Zipf模型拟合真实口令集。详细拟合参数见表2.4。为便于显示，14个口令集以尽量减少线条交叉为原则分为3组。

另外两个关键参数是N和s，其中N是回归中使用的独立口令数量，s是回归线的斜率的绝对值。虽然N和s之间没有明显的关系，但是我们发现：(1) N与总口令数量有着密切的联系——N越大，后者将更大；(2)参数$s \in [0, 1]$，这不同于其他具有Zipf参数$s > 1$的社会现象(如战争频繁程度和姓氏的频率[205])。

2.4.2 PDF-Zipf模型的合理性

在上节中，我们的结果显示，Zipf定律可以很好地刻画流行口令的分布。在接下来，我们将解释PDF-Zipf模型的合理性，并使用模拟实验证明：用户口令的非流行部分，很有可能也服从Zipf分布。

2012年，Malone和Maher[125]首次尝试采用真实口令集来研究口令的分布，得到的结论是他们的数据集(包括32M RockYou)“实际上不可能是Zipf分布”。他们指出，“尽管Zipf定律不能完全描述我们的数据，但它提供了一个合理的模型，特别是在用户口令分布的长尾部分(即非流行部分)”。这一结论与我们本章中的结论正好相反，接下来我们将证实Malone-Maher尝试失败的主要原因是，他们将整个口令集输入到PDF-Zipf模型。

如表2.4所示，非流行部分的口令(即$f_r < LF$的那部分口令)占整个口令

集的比重在50%∼92%，构成口令分布的长尾部分(见[125]中的图1)或称为“嘈杂的尾部”[206]。但根据大数定律，它们的经验概率(即在口令集中出现的频率)不能反应他们在口令分布中的真实概率。更确切地说，对于给定口令pw_i，每次观测可以看作一个服从Bernoulli分布的随机变量，其均值$\mu = p_{pw_i}$，方差$\sigma = p_{pw_i}(1-p_{pw_i})$[5]，其中$p_{pw_i}$是$pw_i$的真实概率。在抽$|\mathcal{DS}|$个样本之后，$pw_i$的经验概率$\frac{f_{pw_i}}{|\mathcal{DS}|}$是一个服从二项分布的随机变量，且$\mu$=$p_{pw_i}$，$\sigma$=$\sqrt{\frac{p_{pw_i}(1-p_{pw_i})}{|\mathcal{DS}|}}$，其中$f_{pw_i}$是$pw_i$在口令集$\mathcal{DS}$中的频率。由于一般有1-$p_{pw_i}$≈1，这得到了一个相对标准误差(RSE)：

$$\frac{\sigma}{\mu} = \sqrt{\frac{p_{pw_i}(1-p_{pw_i})}{|\mathcal{DS}|}} \cdot \frac{1}{p_{pw_i}} \approx \sqrt{\frac{f_{pw_i}}{|\mathcal{DS}|^2}} \cdot \frac{|\mathcal{DS}|}{f_{pw_i}} = \sqrt{\frac{1}{f_{pw_i}}}$$

这意味着只有f_{pw_i}比较大时，真实概率p_{pw_i}可以由经验概率$\frac{f_{pw_i}}{|\mathcal{DS}|}$来近似。比如，$f_{pw_i}$>4时，我们可以确保RSE <$\frac{1}{2}$；$f_{pw_i}$<3时，我们可以确保RSE>$\frac{1}{\sqrt{3}}$。因此，当整个口令集被输入PDF-Zipf模型时，这些不流行的口令将在很大程度上干扰拟合。这解释了为什么[125] 和本章得到了截然相反的结论，也构成为什么需要去除非流行口令后再进行拟合的直接原因。

我们发现存在一个更本质(但微妙)的原因：即使用户生成的口令分布完美地服从Zipf分布，千万规模的样本(如3000万Tianya和3200万RockYou)仍然太小，以至于无法显现出服从Zipf 分布这种本质特征。比如，CSDN的口令策略为“由字母和数字组成，长度为8到16”。这意味着在这一策略下，用户的口令(通过一个随机变量$\mathcal{X}$表示)将有大约$|\mathcal{X}|$=$62^{16} - 62^8$ ≈$4.8{\cdot}10^{29}$种不同的可能。但整个CSDN网站的口令集大小为$6.42{\cdot}10^6$，相比$|\mathcal{X}|$是一个非常小的样本。由于Zipf分布具有口令概率的多项式递减的特性(见公式2.1)，在小样本中低概率事件(如$f < 3$)将压倒高概率事件，因此在这样的小样本中，如果不排除非流行的事件，不大可能反映出高概率事件的真实分布。

这表明，当把规模相对较小的口令集整个输入PDF-Zipf模型时，线性回归结果将受这些非流行的口令的干扰，而且即使口令集的流行部分本身表现出良好的Zipf特性，线性回归的实验结果也无法显示这一点。

应当指出的是，即使这些非流行口令在原始状况下不服从Zipf分布，但这并不与我们关于“一个网站的完整口令集极有可能服从Zipf分布”的猜想相矛盾。表2.4显示，一般来说，口令集越大(即样本数量越大)，流行口令部分(即用于拟合的部分)越大。基于这种趋势可以预见，如果口令集足够大，非流行的口令将仅占很小比例，是否去除它们将基本不会影响拟合的效果。这意味着，整个口令集将呈现Zipf特性。幸运的是，Wang 等[209]近期的一项关于用户PIN码[Φ]分布

[Φ]Personal identity number (PIN)是口令的一种特殊形式，通常由固定长度的纯数字构成，一般在自动取款机ATM，移动设备锁屏控制，电子门禁等系统中使用。

的工作，正好印证了这一推测。他们的结果显示，大多数PIN码数据集可以整个输入PDF-Zipf 模型，即使将$f_r < 10$的PIN码去除，处理后的PIN码数据集仍然达原数据集94%以上，它们很好地服从Zipf定律($R^2 > 0.97$)。

表 2.5 PDF-Zipf模型：研究样本大小和最小频率(LF)阈值对线性拟合效果的影响。粗体表示的是一组实验中最好的拟合结果。

Zipf N	Zipf s	**Sample size**	**LF**	# of Unique passwords	Passwords used in regression(%)	Fitted N	Fitted s	R^2
1000	0.9	100	**1**	71.197	100.00%	71.197	0.429486	0.754566
1000	0.9	100	2	71.262	41.10%	12.361	0.641264	0.884263
1000	0.9	100	3	70.963	**27.20%**	5.307	**0.719897**	0.894042
1000	0.9	100	4	71.068	20.59%	3.173	0.683547	0.916477
1000	0.9	100	5	70.765	17.01%	2.215	0.622484	0.953243
1000	0.9	200	1	123.933	100.00%	123.933	0.516278	0.822066
1000	0.9	200	2	124.103	51.49%	27.074	0.688394	0.923847
1000	0.9	200	3	123.795	36.71%	12.145	0.761613	0.935451
1000	0.9	200	4	124.121	**29.57%**	7.392	**0.785336**	**0.930795**
1000	0.9	200	5	123.954	25.08%	5.242	0.784747	0.921241
1000	0.9	500	1	245.459	100.00%	245.459	0.633549	0.895852
1000	0.9	500	2	246.040	65.37%	72.899	0.724630	0.951529
1000	0.9	500	3	245.482	50.10%	34.245	0.796940	0.969880
1000	0.9	500	4	245.697	42.34%	21.499	0.819386	0.970288
1000	0.9	500	5	245.586	**37.51%**	15.372	**0.834885**	**0.966581**
1000	0.9	1000	1	389.360	100.00%	389.36	0.730031	0.937941
1000	0.9	1000	2	388.014	76.00%	148.053	0.756649	0.965318
1000	0.9	1000	3	388.733	61.18%	74.478	0.807381	0.979783
1000	0.9	1000	4	388.774	53.08%	47.184	0.833071	0.983395
1000	0.9	1000	5	388.839	**47.69%**	33.829	**0.847137**	**0.983550**
1000	0.9	2000	1	573.821	100.00%	573.821	0.835995	0.964407
1000	0.9	2000	2	573.607	85.62%	286.058	0.790817	0.977339
1000	0.9	2000	3	574.446	72.75%	158.041	0.818059	0.985691
1000	0.9	2000	4	574.011	64.39%	102.03	0.840089	0.989460
1000	0.9	2000	5	574.229	**58.66%**	73.534	**0.854452**	**0.990812**
1000	0.9	5000	1	828.243	100.00%	828.243	0.963949	0.963691
1000	0.9	5000	2	828.466	**95.20%**	588.56	**0.861714**	**0.989008**
1000	0.9	5000	3	827.675	87.58%	397.276	0.842637	0.991843
1000	0.9	5000	4	828.601	80.29%	276.308	0.849865	0.993588
1000	0.9	5000	5	828.281	74.49%	203.349	0.859765	0.994832
1000	0.9	10000	1	953.483	100.00%	953.483	1.013698	0.943442
1000	0.9	10000	2	953.545	98.85%	838.141	0.929787	0.985080
1000	0.9	10000	3	953.125	**95.82%**	686.791	**0.884120**	**0.994655**
1000	0.9	10000	4	953.483	91.47%	541.471	0.867965	0.996179
1000	0.9	10000	5	953.365	86.84%	425.614	0.866388	0.996641
1000	0.9	20000	1	995.527	100.00%	995.527	0.994645	0.943878
1000	0.9	20000	2	995.514	99.90%	975.432	0.968837	0.969613
1000	0.9	20000	3	995.521	99.43%	928.450	0.938901	0.986291
1000	0.9	20000	4	995.550	98.33%	855.099	0.912336	0.994446
1000	0.9	20000	5	995.544	**96.49%**	763.027	**0.894282**	**0.997415**

*更多细节和120个详尽的实验详见`http://t.cn/R4ccgiF`.

为进一步证明我们关于用户生成的口令样本服从Zipf定律的猜想，我们从完美的*Zipf*理论分布中随机抽取一些样本，然后观测这些来自理论分布的样本的拟合特性，看这两类样本具有相同的拟合特性。我们考察了3组可能会影响拟合的变量，即Zipf分布(3种)，样本大小(8种)和最小频率阈值LF(5种)，在固定两组变量的情况下，观察另一组变量对拟合效果的影响。为此，我们进行了一系列共120(=3·5·8)个PDF-Zipf模型下的拟合实验。

具体来说，假设随机变量$\mathcal{X}$服从Zipf定律，且$\mathcal{X}$有N=10^3 个可能的取值$\{x_1, x_2, \cdots, x_{10^3}\}$。不失一般性，定义分布律为$\{p(x_1)=\frac{C/1^s}{\sum_{i=1}^{N}\frac{C}{i^s}}=\frac{1/1^s}{\sum_{i=1}^{N}\frac{1}{i^s}}, p(x_2)=\frac{1/2^s}{\sum_{i=1}^{N}\frac{1}{i^s}}, \cdots, p(x_N) =\frac{1/N^s}{\sum_{i=1}^{N}\frac{1}{i^s}}\}$，其中样本空间$N$和斜率$s$唯一确定了Zipf分

布函数。为了保证稳定性，每个实验进行10^3次；为了便于比较，每个实验只改变一组变量中的一个参数。不失一般性，表2.5中只包含40个实验，其中Zipf N固定为10^3，Zipf s固定为0.9，样本大小从10^2变化到10^4，LF从1逐步增加到5。读者可参考`http://t.cn/R4ccgiF`中共计120个实验结果。需要注意的是，表2.5中的一些整数统计量(如Zipf N)赋了小数值，因为我们对1000次重复实验取了均值。

我们的120个实验结果表明，给定一个Zipf分布(即确定Zipf参数N和s)，无论样本空间是小于、等于或大于N，刚开始时更大的LF值将导致更好的回归(即拟合得到的Zipf s接近给定的Zipf s，决定系数R^2更接近1)，但随着LF进一步增加，拟合会恶化。具体来说，当样本量小于N时，随着LF的增加拟合得到的s先增加后减小；当样本大于N时，随着LF的增加，拟合得到的s先减小后增加。因此，我们可以找出最优的拟合(表2.5加粗部分)，从它们可以看到，样本量越大，更大比例的流行事件将用于拟合。这一特性与我们前面观察到的真实口令集的拟合特性相吻合。

特别地，当样本容量足够大时(如，$10^4 \gg N{=}10^3$)，流行事件(如，$f \geq 4$)的比例总是超过90%，它们很好地服从Zipf定律($R^2 \geq 0.99$)。这一现象和拟合PIN码时的现象(见[209])完全一致，也和我们对口令集拟合特性的推断一致。此外，当样本容量远小于样本空间N时，非流行的事件占多数，那么即使流行事件实际上服从Zipf定律，非流行的部分可能会妨碍整体拟合，因此需要先去除。现实生活中，口令集的大小通常远远小于口令样本空间，这正显示了节2.4.1中我们的数据处理方法的合理性。总的来说，14个真实口令集的Zipf分布拟合特性，与完美Zipf分布下120个模拟实验的结果相一致，这从侧面为我们“整个口令集服从Zipf分布”的猜想提供了证据。

2.4.3　CDF-Zipf模型

如上节所指出的，PDF-Zipf模型的一个不足之处在于它不能很好地刻画非流行的那部分口令。因为根据大数定律，像PDF-Zipf这样的基于口令概率的理论模型，本质上难以直接刻画非流行的口令分布(见节2.4.2)。我们也曾尝试各种方法调节从流行口令拟合得到PDF-Zipf的参数，来刻画非流行的那部分口令的分布，但我们总是陷入两难境地：如果非流行的口令被很好的刻画，整体拟合效果将不佳；如果不考虑非流行的口令，整体拟合会有很好的效果(即得到大的决定系数)。幸运的是，PDF-Zipf模型让我们看到了口令可能服从Zipf分布的一道曙光，现在关键的问题是如何优化我们的模型。

既然一个分布的概率密度函数PDF和它的累积分布函数CDF可相互转化，如果直接对口令分布的CDF建模会怎样呢？出乎意料的是，我们发现每个口

表 2.6 CDF-Zipf模型: 对14个真实口令集的拟合结果

	C'	s'	数据覆盖率	统计量 D	p-值
Tianya	0.062239	0.155478	100.00%	0.022798	0.0052
Dodonew	0.019429	0.211921	100.00%	0.004926	0.0124
CSDN	0.058799	0.148573	100.00%	0.022319	$<10^{-4}$
Rockyou	0.037433	0.187227	100.00%	0.045874	$<10^{-4}$
000webhost	0.005858	0.281557	100.00%	0.006170	$<10^{-4}$
Battlefield	0.010298	0.294932	100.00%	0.010557	0.0032
Flirtlife.de	0.034577	0.291596	100.00%	0.036448	$<10^{-6}$
Myspace	0.005646	0.403400	100.00%	0.008262	0.4440
Singles.org	0.020122	0.337620	100.00%	0.013786	0.1048
Faithwriters	0.013588	0.364763	100.00%	0.014524	0.2076
Hak5	0.029029	0.351792	100.00%	0.018078	0.5104
Mail.ru	0.025211	0.218212	100.00%	0.020773	0.0016
Gmail	0.020963	0.225653	100.00%	0.020543	0.0048
Yandex.ru	0.044248	0.197234	100.00%	0.013345	$<10^{-4}$

令集的完整CDF图像可以很好地由Zipf定律拟合(见图2.3(a)~2.3(e)的绿色点划线)。由于这种模型直接拟合口令集的CDF图像，我们称之CDF-Zipf模型：

$$F_r = C' \cdot r^{s'}, \tag{2.3}$$

其中F_r为排名至r的口令的累积频率，C'和s'是取决于口令集的常数，可以通过线性回归计算得到。因为$r = 1, 2, 3, \cdots$，$F_r(\cdot)$是一个阶梯函数。因此，我们有：

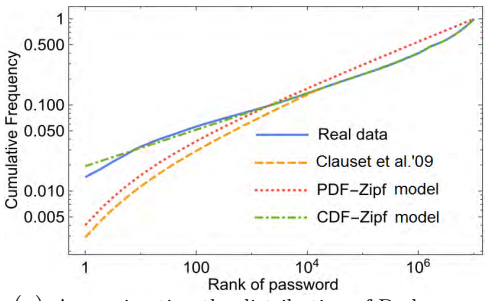
$$f_r = F_r - F_{r-1} = C' \cdot r^{s'} - C' \cdot (r-1)^{s'}. \tag{2.4}$$

需要注意的是，如果将F_r看作连续函数，f_r可用F_r的导数来近似表示：$f_r \approx d(F_r)/dr = C' \cdot s' \cdot r^{s'-1}$，这隐含着Zipf定律。

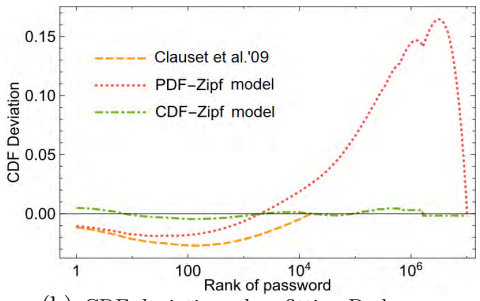
如图2.4，基于我们的CDF-Zipf模型，对所有的14个数据集，拟合得到的KS统计量D(即一个理论模型的CDF曲线和实际数据之间的最大距离)，总是优于PDF-Zipf模型下的结果。本章线性回归过程中，采用著名的黄金分割法[210]来计算CDF-Zipf参数。

表2.6显示，在我们的CDF-Zipf模型拟合得到的KS检验统计量D为0.006170 ~ 0.045874(均值0.018457)。它总是比PDF-Zipf模型得到的D 值小(见表2.4)。这意味着，CDF-Zipf模型的最大CDF 距离总是小于PDF-Zipf模型。KS检验的p-值结果表明，对于9个口令集，我们有足够的信心去拒绝“它们都来自所拟合的Zipf分布(见表2.6参数)”。正如前面所提到的，这并不与我们的结论“用户口令服从Zipf分布”相矛盾。事实上，低的p-值是由于“给定一个足够大的样本，非常小且非显著的差异会被认为统计学上显著的，此时统计显著性失去现实意义”[211]。这就是统计学里著名的现象：样本大小会对假设检验的实际意义产生显著影响[212]。

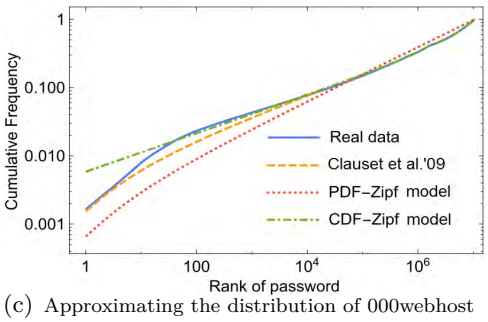
我们的测试性实验发现，当将口令样本的容量限制在一个与文献[205, 212]中样本容量可比拟的范围时(即10^5量级)，大部分基于CDF-Zipf模型的



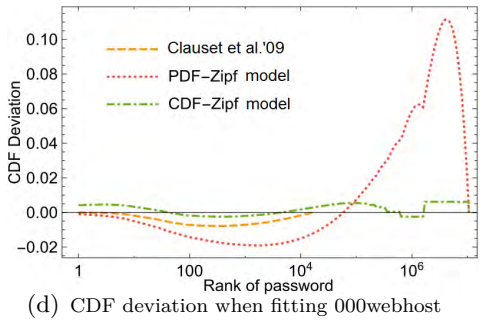
(a) Approximating the distribution of Dodonew



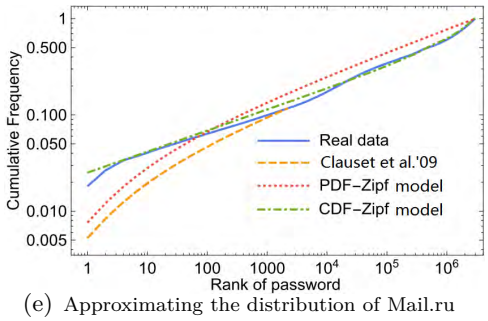
(b) CDF deviation when fitting Dodonew



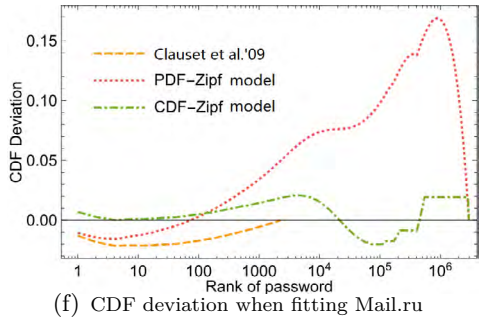
(c) Approximating the distribution of 000webhost



(d) CDF deviation when fitting 000webhost



(e) Approximating the distribution of Mail.ru



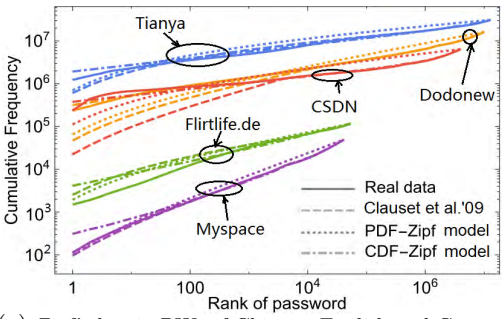
(f) CDF deviation when fitting Mail.ru

图 2.3 三种不同Zipf模型的比较。基于三个典型口令集为例来说明；“CDF误差”的绝对值等于KS 统计量D。结果表明，CDF-Zipf模型优于其它两个模型：更低的CDF误差和100%的数据覆盖率。更多细节见表2.8。

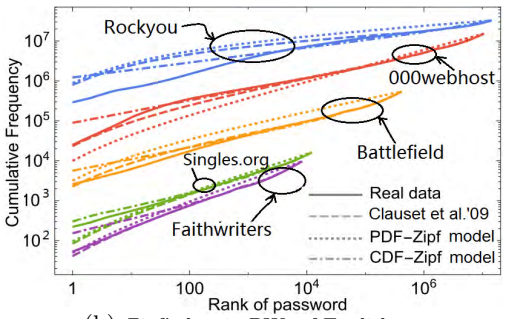
拟合可通过KS检验(即p-值> 0.01)。表2.6中基于四个小口令集(即Myspace，singles.org、Faithwriters和Hak5)的拟合证实了这一点：CDF-Zipf模型下，KS检验的p-值> 0.01。同时需要指出的是，虽然未来有可能出现可更精确的拟合口令Zipf分布的新模型，但CDF-Zipf模型确保在无论何种新模型下，KS 统计量D的最大改进量将被限制在区间$[0,\ 0.018457]$ 内。也就是说，可改进的空间非常有限。特别是针对1623万Dodonew口令和1525 万000webhost口令来说，改进的最大空间甚至不到0.0062。所有这些，都表明我们CDF-Zipf 模型的准确性。下文将进一步显示该模型的优越性。

2.4.4　模型的对比

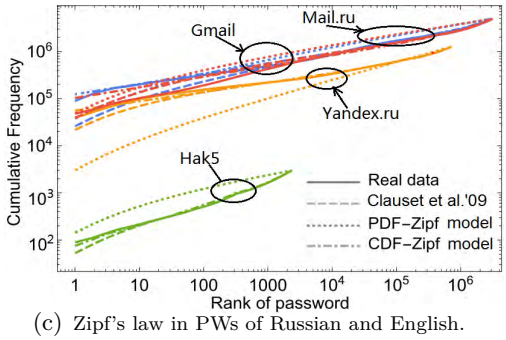
为获得更准确的Zipf分布的参数，我们进一步尝试使用Clauset等[205]提出的



(a) Zipf's law in PWs of Chinese, English and German.



(b) Zipf's law in PWs of English.



(c) Zipf's law in PWs of Russian and English.

图 2.4 Zipf定律存在于14个真实口令集中：口令集来自4种不同语言，采用3种不同Zipf拟合模型。各个模型下拟合出的详细Zipf参数参见表2.4，2.6和2.7。一般来说，这3个模型的优势从高到低可排序为: CDF-Zipf $\geq$ Clauset et al. 2009[205] $\geq$ PDF-Zipf。

一种较复杂的Zipf模型。他们的模型最初是为了刻画一般性的幂律分布，而不是专门针对口令分布而设计。Clauset'09模型需要先确定幂律分布的参数，然后将幂律分布的参数转化为Zipf分布的参数。针对离散变量(如本章中的口令排名r)，我们依文献[205]中建议，基于该文中的式3.7来拟合14个口令集[Φ]。如表2.7所示，Clauset等模型[205]拟合得到的KS统计量D取值范围为0.001079~0.108263(均值0.031883)。其中，四个拟合通过了KS检验(即p-值>0.01)。

表 2.7 Clauset et al. 2009 模型: 14个口令集的拟合结果

	x_{min}	Power-law α=1+$\frac{1}{1-s'}$	Dataset coverage	Statistic D	p-value
Tianya	5	2.099455	50.37%	0.046035	$<10^{-4}$
Dodonew	36	2.399279	15.15%	0.026957	$<10^{-4}$
CSDN	62	2.613033	20.52%	0.089842	$<10^{-4}$
Rockyou	5	2.061061	49.62%	0.108263	$<10^{-4}$
000webhost	23	2.423542	8.75%	0.007795	0.4160
Battlefield	5	2.378776	15.18%	0.008973	$<10^{-4}$
Flirtlife.de	4	2.233860	46.15%	0.064640	$<10^{-4}$
Myspace	7	2.964454	4.91%	0.001079	0.5080
Singles.org	4	2.885328	16.60%	0.009791	$<10^{-4}$
Faithwriters	4	2.999978	9.07%	0.001250	0.8368
Hak5	5	2.370435	9.41%	0.012166	0
Mail.ru	55	2.306956	11.72%	0.021279	0.8156
Gmail	9	2.215576	24.33%	0.025722	$<10^{-4}$
Yandex.ru	35	2.098622	17.18%	0.022568	$<10^{-4}$

表2.8详细比较了三种Zipf模型。从KS统计量D方面来看，我们可以发现：(1)CDF-Zipf模型和Clauset等的模型[205]总是优于PDF-Zipf模型；(2)对于百万规

[Φ]本章使用Clauset等[205]提供的源码`http://tuvalu.santafe.edu/~aaronc/powerlaws/`。

模以上的口令集，CDF-Zipf模型通常表现最佳，而当口令集很小(如小于100万)时，Clauset等的模型[205]表现最佳。为更好地理解这一点，可参见图2.4。从口令集的覆盖范围方面来看，我们可以发现：在所有情况下，CDF-Zipf模型表现最佳(即实现100%的覆盖率)，PDF-Zipf模型表现次之，而Clauset等模型[205]最差。综合来看，本章提出的CDF-Zipf模型在三种Zipf模型中表现最优。

我们进一步研究了在拟合给定口令集时，三种Zipf模型的时间复杂度。具体地说，PDF-Zipf模型，CDF-Zipf模型和Clauset等的模型[205] 的时间复杂度分别是$\mathcal{O}(|\mathcal{DS}|)$、 $\mathcal{O}(|\mathcal{DS}|\cdot log|\mathcal{DS}|)$ 和 $\mathcal{O}(|\mathcal{DS}|)$。举个具体的例子。在一台普通的多核处理器上(i7-4790K 4.00 GHz CPU和16G内存)上，拟合3200万RockYou口令，PDF-Zipf 模型需要32.40秒，CDF-Zipf模型需要14.67个小时，Clauset等的模型[205]需要69.39秒。总的来说，给定中等的计算资源，即使是拟合现实中的大口令集，三个Zipf模型均可以在可接受的时间内完成。

表 2.8 口令分布的三个Zipf模型对比

Dataset	KS statistic D			Dataset (distribution) coverage		
	PDF-Zipf	CDF-Zipf	Clauset et al.[205]	PDF-Zipf	CDF-Zipf	Clauset et al.[205]
Tianya	0.161718	**0.022798**	0.046035	52.86%	**100.00%**	50.37%
Dodonew	0.164640	**0.004926**	0.026957	29.92%	**100.00%**	15.15%
CSDN	0.268982	**0.022319**	0.089842	31.67%	**100.00%**	20.52%
Rockyou	0.193567	**0.045874**	0.108263	51.86%	**100.00%**	49.62%
000webhost	0.111546	**0.006170**	0.007795	23.09%	**100.00%**	8.75%
Battlefield	0.225527	0.010557	**0.008973**	17.14%	**100.00%**	15.18%
Mail.ru	0.168754	**0.020773**	0.021279	35.15%	**100.00%**	11.72%
Gmail	0.217463	**0.020543**	0.025722	32.05%	**100.00%**	24.33%
Flirtlife.de	0.062585	**0.036448**	0.064640	46.15%	**100.00%**	46.15%
Myspace	0.126651	0.008262	**0.001079**	9.66%	**100.00%**	4.91%
Singles.org	0.100741	0.013786	**0.009791**	16.60%	**100.00%**	16.60%
Faithwriters	0.140562	0.014524	**0.001250**	9.07%	**100.00%**	9.07%
Hak5	0.278761	0.018078	**0.012166**	11.28%	**100.00%**	9.41%
Yandex.ru	0.097279	**0.013345**	0.022568	37.18%	**100.00%**	17.18%
Average	0.165627	**0.018457**	0.031883	28.84%	**100.00%**	21.35%

*粗体表示相应的模型在给定评价指标下表现最优。

总结。我们的两个Zipf模型，特别是CDF-Zipf，表明现实中用户生成的口令服从Zipf分布，这与现有研究(如[5, 125])的结论正好相反。对比结果表明，我们的CDF-Zipf模型在KS统计量D和数据覆盖范围两方面均优于另外两个模型，而我们的PDF-Zipf模型劣于Clauset等模型[205]。特别地，我们的CDF-Zipf模型非常准确：其最大CDF误差(即KS统计量D)为0.006170∼0.045874(均值0.018457)。尽我们所知，本章使用的口令数据是现有口令研究中迄今为止规模最大、最多元化的数据集，具有充分的代表性，这显示了本章结论的普适性。可以预期，本章提出的两个口令Zipf分布模型将大为改善人们对用户口令分布的理解，将为整个口令安全研究带来基础性影响。下面两节中，我们将初步展示“口令服从Zipf分布”这一科学发现所产生的重要价值。

2.5 口令分布的强度指标

本节将研究如何准确地评测一个给定口令集的安全强度。基于CDF-Zipf模型，我们提出一个简洁但准确的口令分布强度统计学评价指标。

2.5.1 我们的指标

正常情况下，一个聪明的(漫步)猜测攻击者$\mathcal{A}$，总是会先尝试猜测最有可能成功(即概率最大)的口令，然后再尝试第二可能的口令等等，按成功率递减的顺序以此类推，直到目标口令被正确匹配。在极端情况下，如果攻击者$\mathcal{A}$得到了整个明文口令集，那么$\mathcal{A}$可以直接获得最优的猜测顺序，此时的攻击称为最优攻击[5, 104]Φ。相应地，对于一个给定的口令分布$\mathcal{X}$，Boztas[140]将$\mathcal{A}$以最优顺序对每个帐户猜测n次时的优势$\lambda_n^*(\mathcal{X})$，作为口令分布$\mathcal{X}$的强度指标：

$$\lambda_n^*(\mathcal{X}) = \sum_{r=1}^{n} p_r(\mathcal{X}) = \frac{1}{|\mathcal{DS}|}\sum_{r=1}^{n} f_r(\mathcal{X}), \tag{2.5}$$

其中$|\mathcal{DS}|$是口令集的大小，n是猜测次数。不足的是，Boztas的这一指标只是进行原始的口令频率加和，没有利用任何关于口令确切分布的信息。

根据节2.4.3的结果，口令的分布可以很好地由Zipf定律来刻画：$p_r(\mathcal{X}) = f_r(\mathcal{X})/|\mathcal{DS}| = F_r(\mathcal{X}) - F_{r-1}(\mathcal{X}) \approx C' \cdot r^{s'} - C' \cdot (r-1)^{s'}$。因此，$\lambda_n^*(\mathcal{X})$可由下式来近似:

$$\lambda_n^*(\mathcal{X}) = \sum_{r=1}^{n} p_r(\mathcal{X}) = F_n(\mathcal{X}) \approx C' \cdot n^{s'} = \lambda_n(\mathcal{X}). \tag{2.6}$$

如下文将证实的，由于$\lambda_n(\mathcal{X})$和最优攻击者的优势$\lambda_n^*(\mathcal{X})$具有充分的一致性，$\lambda_n(\mathcal{X})$可以作为口令分布强度的上限。因此，我们提出$\lambda_n(\mathcal{X}) = C' \cdot n^{s'}$作为口令分布$\mathcal{X}$的一个强度评价指标。

2.5.2 指标有效性评估

需要注意的是，即使我们的CDF-Zipf模型与实际口令数据拟合得非常好，式2.6中左部的$\lambda_n^*(\mathcal{X})$并不确切地等于右部的$\lambda_n(\mathcal{X})$。我们根据式2.5绘制$\lambda_n^*(\mathcal{X})$关于n的函数，根据式2.6绘制将$\lambda_n(\mathcal{X})$关于n的函数，并测量两条绘制的曲线的吻合程度。这里我们以Dodonew和000webhost这两个典型口令集(分布)为例作图说明。在图2.5(a) 中，我们为Dodonew的1623万口令绘制$\lambda_n^*(\mathcal{X})$(见图2.5(a) 中蓝色线)和$\lambda_n(\mathcal{X})$(见图2.5(a)中绿色线)，两条曲线最大偏差0.49%(均值0.19%)；图2.5(b)中，我们为 000webhost 的1525万口令绘制 $\lambda_n^*(\mathcal{X})$ 和 $\lambda_n(\mathcal{X})$，两条曲

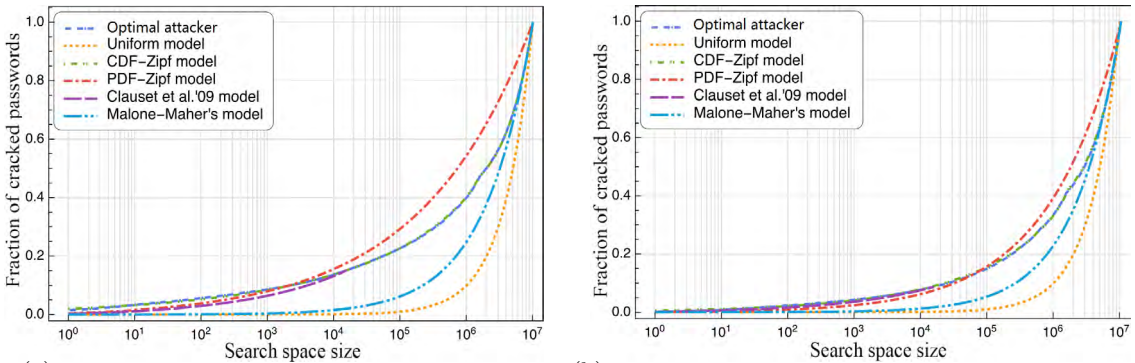
Φ需要注意的是，最优攻击仅具有理论价值，用以表征攻击者可以采取的最佳攻击策略，给出$\mathcal{A}$的优势的上限。现实中，如果$\mathcal{A}$已经获得了所有的明文口令，没有必要为这些口令排序，然后再尝试猜测它们。

表 2.9 $\lambda_n^*(\mathcal{X})$ 和 $\lambda_n(\mathcal{X})$ $(1 \le n \le |\mathcal{DS}|)$之间的偏差

Dataset	Maximum deviation	Average deviation
Tianya	2.28%	1.81%
Dodonew	0.49%	0.19%
CSDN	2.23%	1.15%
Rockyou	4.59%	3.57%
000webhost	0.62%	0.54%
Battlefield	1.06%	0.65%
Mail.ru	2.08%	1.75%
Gmail	2.05%	1.70%
Flirtlife.de	3.64%	2.83%
Myspace	0.83%	0.93%
Singles.org	1.38%	0.90%
Faithwriters	1.45%	0.99%
Hak5	1.81%	0.79%
Yandex.ru	1.33%	1.05%
Average	**1.85%**	**1.35%**

线最大偏差0.62%(均值0.54%)。表2.9中总结了每个口令集的$\lambda_n^*(\mathcal{X})$ 和 $\lambda_n(\mathcal{X})$ $(1 \le n \le |\mathcal{DS}|)$这两条曲线之间的最大偏差和平均偏差。

如表2.9所示，除了RockYou和flirtlife.de之外，其余12个口令分布的$\lambda_n^*(\mathcal{X})$和$\lambda_n(\mathcal{X})(1 \le n \le |\mathcal{DS}|)$这两条曲线之间的平均偏差均小于2%(即从0.19%到1.81%)。这表明$\lambda_n(\mathcal{X})$曲线可用来很好地模拟$\lambda_n^*(\mathcal{X})$ 曲线。换言之，强度指标$\lambda_n(\mathcal{X})$和最优攻击者的优势$\lambda_n^*(\mathcal{X})$具有充分的一致性，$\lambda_n(\mathcal{X})$可以作为口令分布强度的上限。这是符合预期的，因为：(1)如式2.6，$\lambda_n(\mathcal{X})$曲线本质上是CDF-Zipf 模型下的CDF 曲线；(2) 如节2.4.3所示，一个真实口令分布的CDF曲线可由CDF-Zipf模型的CDF曲线很好地近似。如图2.5所示，图中两个口令集的这两条CDF曲线几乎完全重合。图2.5中我们还将CDF-Zipf模型与另外4种口令分布模型进行了对比，以评估它们在模拟最优攻击者优势$\lambda_n^*(\mathcal{X})$时的优劣。结果表明，CDF-Zipf 模型表现最佳，Clauset等模型[205]次之。



(a) Approximating the optimal attacker against Dodonew (b) Approximating the optimal attacker against 000webhost

图 2.5 在两个样本数据集(16.2M Dodonew和15.3M 000webhost)上基于CDF-Zipf模型的指标(即$\lambda_n(\mathcal{X})$)对最佳攻击的一致性。

小结。由于最优(漫步)口令猜测攻击者的优势$\lambda_n^*(\mathcal{X})$，可以视为口令分布$\mathcal{X}$强度的一个上限，而$\lambda_n(\mathcal{X})$可以很好地模拟这一优势。自然地，我们提出$\lambda_n(\mathcal{X}) =$

$C' \cdot n^{s'}$作为衡量$\mathcal{X}$强度的一个指标，其中n是猜测次数，C'和s'可通过CDF-Zipf模型对分布$\mathcal{X}$拟合得到。一般来说，很难直接测量一个口令生成策略的强度。幸运的是，通过测量在此策略下生成的口令集的强度可实现这一点(见[86, 93])。本节的$\lambda_n(\mathcal{X})$指标为实现这种测量提供了一个简洁、精确的方法。

2.6 三项启示

基于“口令服从Zipf分布”这一发现，本节探讨其对口令研究领域的三个启示。首先，我们指出绝大多数基于口令密码协议(如认证, 加密, 签名等)，其安全归约结果的精度，相比传统基于均匀分布的假设，可以提高2至4个数量级。由于基于口令的密码协议是最常见、最主要的密码协议之一，本章这一贡献将产生具有重要现实意义的影响。其次，我们发现，如果采用当前广泛推荐的安全参数，基于流行度的口令生成策略(见[135])将严重降低系统的可用性。最后，广泛使用的口令分布强度评价指标α-guesswork过去被认为总是依赖于攻击成功率α，而我们发现在两种情形下(共4种情况)，α-guesswork不依赖于α，即α-guesswork在50%情况下是非参数化的。特别地，基于14个大规模真实口令集，我们证实了本节所提结论的有效性。

2.6.1 对基于口令的安全协议的启示

本小节以最常见的基于口令的身份认证协议为例，来探讨“口令服从Zipf分布”这一发现对基于口令的密码协议的启示。

(1) 对基于口令的单因子认证协议的启示

可以预期，“口令服从Zipf分布”这一发现对基于口令的单因子身份认证协议(即PAKE 协议)影响最为深远：当前数以百计的PAKE协议的基本假设不再成立。根据涉及认证因子的多少，基于口令的认证协议可以分为基于口令的单因子协议(如两方PAKE[21]和多方PAKE[213])、基于口令的双因子协议(如[151])和基于口令的多因子协议(如[214])。这里我们首先讨论对PAKE协议的启示。

基于均匀分布的安全性表达函数。在可证明安全的PAKE协议文献中(如随机预言机模型下的协议[22, 213, 215, 216] 和标准模型下的协议[21, 132, 217, 218])，一般都遵循相似的流程：先假设“每个用户U的口令pw_U从大小为$|\mathcal{D}|$的字典 $\mathcal{D}$ 中均匀随机选取，其中 $|\mathcal{D}|$ 是一个与安全参数 k 无关的常数”[21]；然后描述协议所基于的安全模型；最后给出如文献[21]的关于安全性的“标准”定义：

> “… …协议$\mathcal{P}$是一个安全的基于口令的单因子认证密钥交换协议，仅当对所有大小为$|\mathcal{D}|$口令字典和对所有概率多项式时间的至多进

表 2.10 各个真实口令集中Top-x口令所占的比重†

Datasets	Top 1	Top 3	Top 10	Top 10^2	Top 10^3	Top 10^4	Top $\frac{1}{10}$	Top $\frac{1}{10^2}$	Top $\frac{1}{10^3}$	Top $\frac{1}{10^4}$
Tianya	3.98%	5.59%	7.43%	11.50%	16.04%	25.78%	58.21%	41.19%	27.45%	16.70%
Dodonew	1.45%	2.15%	3.28%	5.60%	8.59%	13.62%	39.97%	22.72%	13.66%	8.61%
CSDN	3.66%	8.15%	10.44%	13.26%	16.54%	23.91%	42.66%	28.46%	20.62%	14.97%
Rockyou	0.89%	1.37%	2.05%	4.55%	11.30%	22.31%	57.28%	39.30%	24.24%	12.84%
000webhost	0.16%	0.34%	0.79%	2.32%	4.30%	7.71%	34.09%	15.17%	7.83%	4.36%
Battlefield	0.48%	0.71%	1.14%	3.21%	8.13%	17.91%	30.57%	13.49%	5.78%	2.16%
Myspace	0.23%	0.52%	1.07%	3.55%	11.48%	36.49%	24.71%	7.37%	2.25%	0.62%
Singles.org	1.36%	2.10%	3.40%	9.19%	26.35%	86.26%	29.09%	10.06%	3.65%	1.36%
Faithwriters	0.55%	1.03%	2.17%	7.76%	24.33%	100.00%	22.62%	7.06%	1.90%	0.00%
Hak5	2.98%	4.63%	7.21%	17.04%	54.86%	100.00%	26.02%	9.69%	3.82%	0.00%
Flirtlife.de	1.30%	2.00%	3.47%	10.83%	28.51%	58.01%	48.73%	22.52%	7.92%	2.55%
Mail.ru	1.82%	3.06%	4.05%	6.37%	9.94%	17.40%	43.88%	24.92%	12.46%	7.81%
Gmail.ru	0.97%	1.43%	2.08%	3.88%	8.66%	17.77%	41.65%	23.79%	12.63%	5.76%
Yandex.ru	3.10%	4.98%	7.66%	12.26%	17.43%	26.53%	44.29%	24.53%	16.61%	11.55%
Avg. above	**1.78%**	**2.98%**	**4.24%**	**7.38%**	**12.94%**	23.09%	44.13%	25.61%	14.92%	8.73%
Uni. dist.	0.0001%	0.0003%	0.0010%	0.01%	0.10%	1.00%	10.00%	1.00%	0.10%	0.01%

† In this Section, we only consider datasets (in Table 2.1) that are with a size larger than 10^5; "Uni. dist." stands for uniform distribution.

行$Q(k)$次在线猜测的攻击者$\mathcal{A}$，存在一个可忽略的函数$\epsilon(\cdot)$，满足：

$$\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k) \leq Q(k)/|\mathcal{D}| + \epsilon(k), \tag{2.7}$$

其中，$\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k)$是$\mathcal{A}$攻击$\mathcal{P}$时的优势。" ①

通常，用户生成的口令可提供约$20 \sim 21$比特[5]的抗离线猜测攻击的安全性，故有效口令空间的大小约为$2^{20} \sim 2^{21}$。这意味着，一个实现式2.7所示安全目标的PAKE协议，能够保证一次在线猜测的成功率$\frac{1}{|\mathcal{D}|}$不高于$1/2^{20}$~$1/2^{21}$。

事实并非如此，如式2.7的PAKE安全性表达函数，会向普通用户和安全管理员传达一种过于乐观的安全感。如表2.10所示，当对现实世界中网站进行10^3次在线猜测时，攻击者的实际优势比均匀模型下安全性表达函数所刻画的攻击者的优势高出2到4个数量级。比如，当$Q(k)$=3时，攻击者对游戏兼电子商务网站Dodonew的实际优势达到1.45%；$Q(k)$=10时，达到 3.28%；$Q(k)$=100时，达到5.60%。它们远远超出了式2.7预测的理论结果(如表2.10最后一行所示)。

为谨慎起见，少数PAKE文献(如[21, 217])对式2.7进行了补充说明，指出"假设口令服从均匀分布、字典大小固定，只是为了描述方便，实际上文中的安全性证明可以通过扩展其来处理更复杂的情况，比如口令不服从均匀分布、不同用户有不同的口令分布、或是口令字典的大小$|\mathcal{D}|$与安全参数k有关等等"。然而，对读者来说，这样的补充说明不仅模糊了协议可实现的安全性目标，同时也降低了用户对协议$\mathcal{P}$到底可在多大程度上保护系统的信心：当用户选择的口令不服从均匀分布时，$\mathcal{P}$的安全性表达函数未知，亦即可提供的安全性保障是个未知数。这违背了设计可证明安全协议的初衷："通过具体明确的安全性归约，清晰地刻画协议安全性所固有的量化本质，以提供量化的安全保障" [219]。

① 此定义的显示了一个安全PAKE协议$\mathcal{P}$ 所提供的"本质安全性"：$\mathcal{P}$可抗离线猜测攻击，只有不可避免的在线猜测攻击对$\mathcal{A}$破坏$\mathcal{P}$的安全性有帮助。

我们的基于Zipf分布的安全性表达函数。幸运的是，有了节2.4中的Zipf理论，我们可以不再对口令作不合实际的均匀分布假设。现实中，由于系统分配的口令可用性低[65]，大多数网络服务允许用户自己生成口令。如前文所述，这通常会导致口令服从Zipf定律。因此，对口令分布做Zipf 假设是必要且合理的。在节2.4.2中的PDF-Zipf模型下，自然地得到：

$$\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k) = \frac{C/1^s}{\sum_{i=1}^{|\mathcal{D}|}\frac{C}{i^s}} + \frac{C/2^s}{\sum_{i=1}^{|\mathcal{D}|}\frac{C}{i^s}} + \cdots + \frac{C/Q(k)^s}{\sum_{i=1}^{|\mathcal{D}|}\frac{C}{i^s}} = \frac{\sum_{j=1}^{Q(k)}\frac{1}{j^s}}{\sum_{i=1}^{|\mathcal{D}|}\frac{1}{i^s}} + \epsilon(k), \tag{2.8}$$

在我们的CDF-Zipf模型下(见式2.4)，可以自然地得到：

$$\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k) = C' \cdot Q(k)^{s'} + \epsilon(k), \tag{2.9}$$

其中参数 C, C', s ands'与节2.4中式2.1和式2.4的相应含义一致。

图2.6表明，相比式2.8，式2.9可以更好地刻画$\mathcal{A}$的优势。如节2.4.4所证实的，这一结果是合理的。

我们的基于流行度策略的安全性表达函数。如图2.6和表2.10所示，即使攻击者只尝试攻击极小一部分的流行口令(如$Q(k) = 100$)，可以攻破的用户帐户数量也不可小觑。换言之，即使部署的认证协议实现了可证明安全，如果系统的口令服从Zipf定律，仍然不能保证系统的安全性。我们可以采取基于流行度的口令策略(见[135])来解决这一问题，而此时用户口令将是扭曲的Zipf 分布，几乎不可能采用明确的数学模型来刻画，在这种情况下如何给出$\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k)$确切的表达函数就成为最大难题。从一个安全的PAKE协议所能提供的“本质安全性”的概念中(具体见式2.7)，我们得到灵感—既然只有在线猜测攻击对攻击者破坏PAKE协议的安全性有帮助[21, 134]，那么可设法避开这个难题，放弃先刻画口令确切分布再给出安全性表达函数的想法。相反，针对每一个类似[135]的基于流行度的口令策略，我们可为攻击者的优势计算出一个严格上限。具体来说，将式 2.7修正如下：

$$\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k) \leq F_1 \cdot Q(k)/|\mathcal{DS}| + \epsilon(k), \tag{2.10}$$

其中F_1是数据集$\mathcal{DS}$中top-1口令的频率，$|\mathcal{DS}|$ 是系统的用户数量，其它符号与式2.7中的含义一致。注意，字典$\mathcal{D}$是口令样本空间，是一个集合，而数据集$\mathcal{DS}$则是一系列(具体的)口令样本，是一个多重集。因此，$F_1/|\mathcal{DS}|$的值恰好就是口令策略中维护的概率阈值$\mathcal{T}$(如$\mathcal{T}$ =$1/10^6$[135])。对于要达到一级安全性资质的系统来说，一次在线猜测，攻击者的成功率不应超过$1/1024$[37]，即$F_1/|\mathcal{DS}| \leq 1/1024$；类似地，为达到二级安全性资质，系统应保证$\mathcal{T} = F_1/|\mathcal{DS}| \leq 1/16384$。比如，对于要达到二级安全性的游戏和电子商务网

站www.dodonew.com，F_1应不大于991($\approx$ 16231271/16384)。不难发现，当$F_1=1$且$|\mathcal{DS}|=|\mathcal{D}|$时，式2.7实际上是式2.10的一个特殊情况。

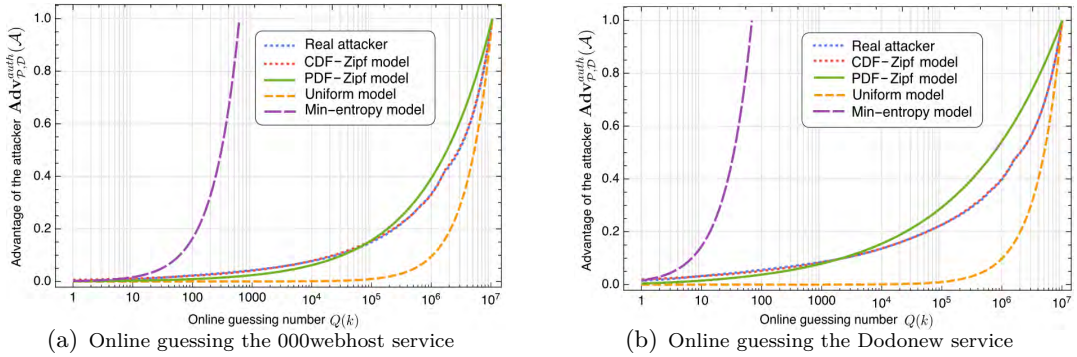
基于最小熵的安全性表达函数。2015年，Abdalla等[134]提出了一个可证明安全的PAKE协议，未使用传统如式2.7的安全性表达函数，而使用如下表达函数：

$$\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k) \leq Q(k)/2^m + \epsilon(k), \tag{2.11}$$

其中m是口令集的最小熵[5]。[Φ]实际上，不难看出Abdalla等的安全性表达函数本质上与式2.10 相同，因为我们可以推出$m=-\log_2(F_1/|\mathcal{DS}|)$。然而，文献[134]中并未给出为什么要选择式 2.11 而不是式 2.7 的依据或理由。相比之下，从口令生成策略的角度，我们的式 2.10比式2.11 更为具体、更易理解。

此外，与我们的式2.10一样， Abdalla等的式2.11(即最小熵模型)也仅在执行了基于流行度的口令策略(如[135])的情况下有效，因为该策略会导致口令分布不再严格服从Zipf定律。然而，如果不执行这种策略，用户口令就会服从Zipf定律，那么式2.10和式2.11都将失效。文献[134]中并未指出这一点。最近的研究[121]和我们对50家知名网站(见表1.4)的调研表明，基于流行度的策略尚未被应用到主流web服务或是口令管理器中。

还需注意，如果m被定义为口令的熵，我们则可以推出$m=\sum_{r=1}^{|\mathcal{D}|}-p_i\log_2 p_i$，其中$p_i$是 $\mathcal{D}$ 中降序排名第i位的口令的频率(比如，$p_1=F_1/|\mathcal{DS}|$)，那么式 2.11实际上与式2.7是等价的，它们刻画的是在线猜测难度的平均值。这很好地解释了Benhamouda等(见[220]的6.1 节)声称“任何敌手的优势都可以被式2.7或式2.11等价刻画”的原因。然而，如前所述，如果m被定义为给定口令分布的最小熵，式2.7与式2.11(或等价的式2.10)将有显著的差异。



(a) Online guessing the 000webhost service　　(b) Online guessing the Dodonew service

图 2.6 在允许$Q(k)$次在线猜测的情况下，五种攻击者的优势对比：真实攻击者、基于均匀分布的攻击者[22, 215, 217]、基于最小熵模型的攻击者[133, 134], 基于PDF-Zipf模型的攻击者和基于CDF-Zipf模型的攻击者。基于CDF-Zipf 模型的攻击者优势曲线与真实攻击者几乎重合，表现最优。

[Φ]我们注意到，在文献[134]的节5.2∼5.4中，m被重新定义为口令的熵(而不是最小熵)，这种不一致将导致安全性表达函数的值存在巨大差异。我们推测原文此处可能存在笔误。

对比与小结。在图2.6中，我们以猜测000webhost 和Dodonew这两个网站的口令为例，对比了现有的两个PAKE安全性表达函数(即式2.7 和式2.11)和基于Zipf口令分布假设的两个安全性表达函数(即式2.8和式2.9)在模拟真实攻击者的优势时的优劣。现实中，一般通过帐号锁定、登录速率限制或异常登录检测[19]等手段来抵御在线猜测攻击，因此攻击者$\mathcal{A}$无法进行大量的登录尝试，可进行的猜测次数十分有限(通常$Q(k) \leq 10^4$)。可以看出，本节所提出的基于CDF-Zipf模型的安全性表达函数能很好地模拟真现攻击者—它的优势曲线几乎与真实攻击者$\mathcal{A}$的曲线完全重合，远远优于其它三个模型。比如，当$Q(k)$=10^2时，$\mathcal{A}$攻击Dodonew的实际优势达到5.60%；当$Q(k)$=10^3时，优势则达到8.59%。它们远远超出了由式2.7(见[22, 215, 217])给出的0.00098% 和0.0098%，也远远小于由式2.11(见[133, 134])给出的100%和100%。幸运的是，当$Q(k)$=10^2时，我们的CDF-Zipf 模型(即式2.9)可精确地预测$\mathcal{A}$的优势为5.15%；当 $Q(k)$=10^3时，可精确地预测为8.40%。综上，基于CDF-Zipf模型的表达函数表现最佳，可精准刻画真实攻击者的在线猜测优势。

(2) 对基于口令的多因子认证协议的启示

不失一般性，这里我们以应用最广泛的基于智能卡的口令认证协议为例。设计双因子协议的主要目标是实现“真正的双因子安全性”[151]，即攻击者$\mathcal{A}$只有同时拥有口令和智能卡这两个认证因子才能登录服务器。这意味着，仅拥有口令因子或智能卡因子，$\mathcal{A}$ 将无法成功登录。

设计具有“真正双因子安全”协议的关键在于在智能卡被攻击者窃取的情况下，如何抵御离线口令猜测攻击[151]。这意味着，此时双因子协议的安全性仅仅依赖于口令的安全性。攻击者$\mathcal{A}$可以使用窃取的智能卡，分析出智能卡内安全参数，然后使用猜测$pw_1, pw_2, \cdots$尝试登录，直至服务器锁定帐户。由于这种在线猜测攻击是无法阻止、不可避免的，那么一个双因子协议$\mathcal{P}$ 能实现的最优安全性就是，确保这种在线猜测攻击是攻击者$\mathcal{A}$所能进行的最有效的攻击。因此，称协议$\mathcal{P}$是安全的，当且仅当：

$$\mathrm{Adv}^{\mathrm{2fa}}_{\mathcal{A},\mathcal{P}}(k) \leq Q(k)/|\mathcal{D}| + \epsilon(k), \tag{2.12}$$

其中$\mathrm{Adv}^{\mathrm{2fa}}_{\mathcal{A},\mathcal{P}}(k)$是在对 $\mathcal{P}$进行$Q(k)$次在线猜测攻击时，$\mathcal{A}$的优势。

在绝大多数可证明安全的双因子协议文献中(如[221]中的2.2.1节，[215]中的定义1)，都有类似式2.12的安全性表达函数。如上文所述，我们的Zipf理论推翻了这种不切实际、极具误导性(传递了一种安全保证的错觉)的安全性表达函数。对比结果表明，本章提出的式2.9那样的安全性表达函数更加准确、更为可取(见图2.6)：

$$\mathrm{Adv}^{\mathrm{2fa}}_{\mathcal{A},\mathcal{P}}(k) = C' \cdot Q(k)^{s'} + \epsilon(k), \tag{2.13}$$

其中$\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k)$是$\mathcal{A}$对$\mathcal{P}$进行$Q(k)$次在线猜测攻击时的优势，$C'$和$s'$是使用CDF-Zipf模型计算得到的参数。

(3) 对其它基于口令的密码协议的启示

不失一般性，这里我们主要采用典型示例，来阐明我们的安全性表达函数2.9在不同应用场景下的普适性。示例包括基于口令的秘密共享[136]，基于口令的签名[131]和基于口令的加密[130]。

2016年，Jarecki等[136]提出了一种高效的、可组合的基于口令的秘密共享(PPSS)协议。文献[136]的定义2假设“$pw \leftarrow_R \mathcal{D}$”，表示口令$pw$是从口令空间$\mathcal{D}$中均匀随机抽取的。进一步，他们将协议的安全性定义为$\mathrm{Adv}^{\mathrm{ppss}}_{\mathcal{A}}(k) = (q_S + q_U)/|\mathcal{D}| + \epsilon$。根据CDF-Zipf理论，我们提出的式2.9那样的安全性表达函数更加准确、更为可取：

$$\mathrm{Adv}^{\mathrm{ppss}}_{\mathcal{A}}(k) = C' \cdot Q(k)^{s'} + \epsilon(k), \tag{2.14}$$

类似地，对基于口令的签名协议(PBS)[131]，当定义“盲性”时PBS通常涉及口令分布的明确假设。绝大多数相关工作均假设“$pw \leftarrow_R \mathtt{PW}$”[131]。对基于口令的加密协议(PBE)，绝大多数相关工作假设“$pw \leftarrow_\$ A_1(\lambda)$”（见[130]中的图7），即口令是均匀随机抽取的。所有这些基于口令的密码协议，都可以基于“口令Zipf服从分布”这一假设重新设计，并给出类似式 2.9的更切实际的安全性表达函数。

小结。尽我们所知，本章首次关注了口令安全研究与基于口令的认证协议研究间的结合点。基于对口令的确切分布的认识，我们成功建立了更为准确、更切合实际的安全性表达函数(即$\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k) = C' \cdot Q(k)^{s'} + \epsilon(k)$)，来刻画基于口令的认证协议的形式化安全结果。给定目标系统，其口令分布函数$f_r = C' \cdot r^{s'} - C' \cdot (r-1)^{s'}$中参数$C'$和$s'$的值可以通过从具有相同(似)服务类型、相同语言和相同口令策略网站泄露的口令集的CDF-Zipf参数来近似，实现对C'和s'的可靠预测。比如，如果拟将口令认证协议部署在一个北美英文游戏网站上，则可以设置C'=0.010298、s'=0.294932，它们来自泄露的Battlefiled网站。一般来说，对于高价值服务系统，C'和s'的值可选自Dodonew/000webhost；对于中等价值服务系统，C'和s'的值可选自Gmail/Battlefield；而对于较低价值服务系统，C'和s'的值则可选自 Tianya/Rockyou。这样，我们就能够在没有口令数据的情况下，对口令系统的安全规定进行准确、定量和切实有效的评价。

本小节中我们主要以基于口令的认证协议为例。事实上，只要一个密码协议的安全性表达函数本质上依赖于用户口令分布的明确假设，如PBE[130]、PBS[131]和PPSS[136, 216]，本节揭示的结果就可以很容易地应用到此协议中。

2.6.2 对口令生成策略的启示

如节2.6.1所示，由于用户倾向选择流行的口令，这种口令使$\mathcal{A}$在极少的猜测次数$Q(k)$的情况下，也能获得极高的优势$\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k) = C' \cdot Q(k)^{s'} + \epsilon(k)$。为解决这一问题，传统的应对办法是执行一个流行口令“黑名单”(见表1.4)，但这会缩小本已十分受限的口令空间，并且会迫使一大批用户放弃旧口令而选择、记忆一个新口令，严重降低用户体验。近来，基于流行度的口令生成策略受到越来越多关注和推荐(见[135, 190])，这种策略禁止用户选择流行度超过给定阈值$\mathcal{T}$(如$\mathcal{T} = 1/10^{6}$[135])的口令，一方面不会降低用户口令空间，同时可确保系统安全性达到一定保障：$\mathcal{A}$在线猜测一次的成功率不超过阈值$\mathcal{T}$。

但是，这种策略带来了一个新的问题：如何合理选择阈值$\mathcal{T}$，使系统在保证一定安全性的同时，尽量减少对用户的影响？这需要我们在给定阈值$\mathcal{T}$ 的情况下，可以预测有多大比例用户会受到影响。基于CDF-Zipf模型，本节提出一套精确的预测模型。

(1) 预测模型

根据我们的CDF-Zipf模型，口令的排名r和排名至r的累积频率F_r满足等式$F_r = C' \cdot r^{s'}$，其中参数C'和s'是常数，它们可在拟合具体口令集的过程中计算得到。换言之，

$$P(rank \leq r) = C' \cdot r^{s'}, \tag{2.15}$$

通常，口令集中独立口令的数量是有限的(记为N)。故

$$P(rank \leq N) = C' \cdot N^{s'} = 1 \;\Rightarrow\; N = (\frac{1}{C'})^{1/s'}. \tag{2.16}$$

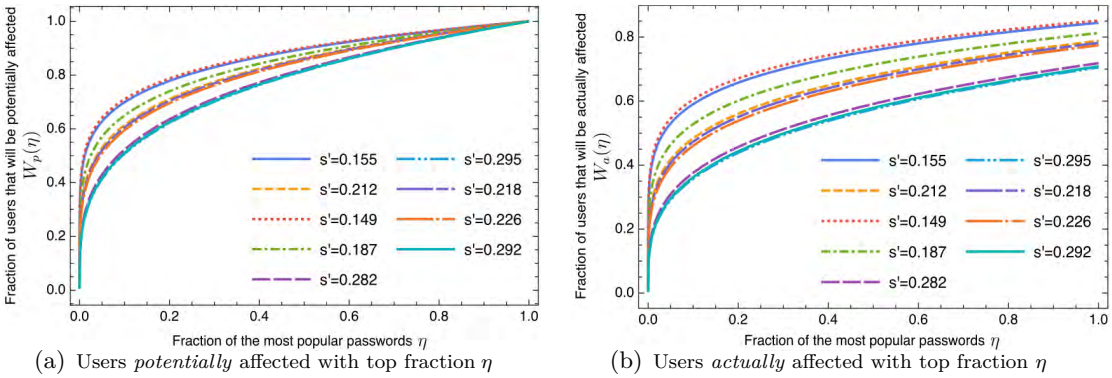
由于排名前η(如$\eta = 1\%$)的独立口令数则是$\eta \cdot N$，那么前 $\eta \cdot N$个独立口令的累积频率即为：

$$P(rank \leq \eta \cdot N) = C' \cdot (\eta \cdot N)^{s'} = C' \cdot (\eta \cdot (1/C')^{1/s'})^{s'} = C' \cdot \eta^{s'} \cdot (1/C') = \eta^{s'}. \tag{2.17}$$

这意味着，将潜在有比例为$W_p(\eta) = P(rank \leq \eta \cdot N) = \eta^{s'}$的用户被阻止使用现有口令。为便于理论分析，我们假设口令的频率f_r是一个连续型变量，则：

$$f_r = \frac{d(F_r)}{dr} = \frac{d(C' \cdot r^{s'})}{dr} = C' \cdot s' \cdot r^{s'-1}. \tag{2.18}$$

现在，我们可以计算出η与流行度阈值$\mathcal{T}$之间的关系。对于频率为$\mathcal{T}$的



(a) Users *potentially* affected with top fraction η　(b) Users *actually* affected with top fraction η

图 2.7 假设限制使用排名前η的口令，根据CDF-Zipf模型预测潜在和实际受影响的用户的比例。其中，Zipf参数s'的值见表2.6。

口令，记其排名为$r_{\mathcal{T}}$。一方面，根据式 2.18，有$\mathcal{T} = C' \cdot s' \cdot r_{\mathcal{T}}^{s'-1}$；另一方面，有$\eta = r_{\mathcal{T}}/N$。进一步结合式2.16，我们可以得到：

$$\eta = \frac{r_{\mathcal{T}}}{N} = (\frac{\mathcal{T}}{C' \cdot s'})^{\frac{1}{s'-1}} \cdot \frac{1}{N} = (\frac{\mathcal{T}}{C' \cdot s'})^{\frac{1}{s'-1}} \cdot (C')^{\frac{1}{s'}}. \tag{2.19}$$

根据2.18，排名为$\eta \cdot N$的口令的频率应为$C' \cdot s' \cdot (\eta \cdot N)^{s'-1}$。因此，在那些潜在受到影响的$W_p(\eta)$口令中，实际上不会受到影响的用户比例为$C' \cdot s' \cdot (\eta \cdot N)^{s'-1} \cdot \eta \cdot N = s' \cdot \eta^{s'}$。因此，根据式2.19，实际受到影响的用户比例为：

$$W_a(\eta) = W_p(\eta) - s' \cdot \eta^{s'} = (1-s') \cdot \eta^{s'} = (1-s') \cdot (\frac{\mathcal{T}}{C' \cdot s'})^{\frac{s'}{s'-1}} \cdot C' . \tag{2.20}$$

有一个微妙之处需特别注意。$W_p(\eta)$和$W_a(\eta)$相互关联但具有不同含义，都可以用作衡量可用性受影响程度的指标。比如，假设一个大规模的英文社交网站 www.example.com打算执行一个基于流行度的口令策略，其中$\mathcal{T} = 1/10^6$，那么我们可以做出预测(基于Rockyou的CDF-Zipf 参数)：将会有$W_p(\eta)$=36.08%的帐户，其口令的流行度超过了阈值$\mathcal{T} = 1/10^6$。意味着，这36.08%的用户具有同等可能性被要求更换口令。然而，最终实际上只有$W_a(\eta)$=29.32%的帐户被要求更换口令，因为当这$W_a(\eta)$=29.32%的帐户被迫更换口令后，其余$W_p(\eta) - W_a(\eta)$=6.76%的帐户，其口令的流行度将不会超过阈值$\mathcal{T} = 1/10^6$，是符合策略的，将不会被烦扰。相比之下，如果系统执行的是传统的口令“黑名单”策略且想达到同等安全性，那么整个$W_p(\eta)$=36.08%的帐户将被迫更换口令。这首次量化显示了基于流行度口令策略的优越性。

(2) 实验结果

在图 2.7中，我们以 η 为 x-轴，针对不同的 s' 值(表2.6)分别绘制了 $W_p(\eta)$ 和 $W_a(\eta)$ 的曲线。可以看到，W_p 和W_a的曲线在$\eta \leq$10%时快速上升，这清楚地揭示了一个事实：即使只有很少部分的流行口令被限制使用，也会有相当大一

批用户会受到影响。在7个百万以上用户规模的网站中(见表2.1)，当$\mathcal{T}$=$1/10^6$时，根据W_a=$(1-s')\cdot(\frac{\mathcal{T}}{C'\cdot s'})^{\frac{s'}{s'-1}}\cdot C'$可以预测，实际会对13.98%~36.42%(平均31.14%)的用户造成不便。

为了检验我们的理论是否与实际吻合，我们总结了14个真实口令集的统计结果(见表 2.11)。一般来说，当$\mathcal{T}<1/16384$ 时，理论的W_a是实际W_a 的1/2~1。这意味着我们的预测模型可作为可用性影响程度的一个保守指标。

表 2.11 阈值$\mathcal{T}$影响的流行口令比重η的和影响的用户比重W_a

		$\mathcal{T}$ =1/1024		$\mathcal{T}$ =1/10000		$\mathcal{T}$ =1/16384		$\mathcal{T}$ =1/100000		$\mathcal{T}$ =1/1000000	
		η	W_a	η	W_a	η	W_a	η	W_a	η	W_a
Large sites	Tianya	0.00%	6.61%	0.00%	10.76%	0.00%	11.64%	0.04%	16.84%	0.44%	30.76%
	Dodonew	0.00%	2.30%	0.00%	4.61%	0.00%	5.19%	0.02%	7.76%	0.40%	14.65%
	CSDN	0.00%	9.46%	0.00%	12.28%	0.00%	12.87%	0.09%	16.59%	0.84%	24.46%
	Rockyou	0.00%	1.22%	0.00%	4.13%	0.00%	5.69%	0.03%	13.76%	0.49%	27.17%
	000webhost	0.00%	0.07%	0.00%	1.36%	0.00%	1.72%	0.01%	3.30%	0.27%	7.40%
	Battlefield	0.00%	0.42%	0.04%	2.33%	0.08%	3.26%	1.07%	9.34%	100.00%	58.25%
	Flirtlife.de	0.06%	3.00%	1.99%	18.54%	3.31%	24.10%	27.11%	51.38%	100.00%	94.61%
	Mail.ru	0.00%	3.08%	0.00%	5.40%	0.01%	6.05%	0.09%	9.52%	2.84%	24.68%
	Gmail	0.00%	1.24%	0.00%	2.94%	0.01%	3.87%	0.13%	9.64%	2.47%	21.89%
	Yandex.ru	0.00%	7.22%	0.03%	11.73%	0.05%	12.81%	0.59%	17.75%	18.01%	40.48%
Small sites	Myspace	0.01%	0.23%	1.01%	3.22%	1.68%	5.40%	100.00%	58.49%	100.00%	95.85%
	Singles.org	0.16%	2.51%	14.17%	18.04%	100.00%	25.34%	100.00%	87.77%	100.00%	98.78%
	Faithwriters	0.19%	1.26%	100.00%	16.54%	100.00%	49.06%	100.00%	91.65%	100.00%	99.17%
	Hak5	3.24%	8.00%	100.00%	76.54%	100.00%	85.68%	100.00%	97.65%	100.00%	99.77%

小结。我们提出的一套可用性预测模型(即η，$W_p(\eta)$和$W_a(\eta)$)表明，即使针对拥有千万量级用户的大规模的网络服务，设置阈值$\mathcal{T}=1/10^6$ 极可能会严重降低系统可用性。相比之下，如果令$\mathcal{T}=1/16384$，同样符合2 级认证标准[37]，但仅有低于12.87%的用户会被烦扰。综合来看，针对大规模的网络服务(如用户规模在百万量级以上)，我们认为$\mathcal{T}=1/16384$是一个更可行、更可接受的阈值，尤其是对那些注重用户检验的服务来说，如电子商务和游戏。

2.6.3 对口令分布强度评价指标的启示

2012年，Bonneau提出了一个新颖的口令分布强度评价指标α-guesswork[5]，该指标当前已在口令研究领域广泛使用(见[84, 142–146])。但如Bonneau所承认的，α-guesswork存在一个明显缺陷—它是完全参数化的，在成功率$\alpha\in[0,1]$整个区间内无单调性，这导致在比较两个口令分布的强度时十分繁琐，必须整个[0, 1]区间内逐一进行对比。幸运的是，基于“口令服从Zipf 分布”这一发现，本节指出α-guesswork在两种(共四种)情形下实际上是非参数化的，在整个[0, 1]区间内具有单调性。α-guesswork这一本质特性的揭示将使其更易于应用。

(1) α-guesswork评价指标回顾

为便于理解，这里的符号与[5]保持一致。$\mathcal{X}$表示给定的口令分布，每个口令x_i从$\mathcal{X}$中以概率p_i 随机选取，其中$\sum p_i$=1。不失一般性，设$p_1\geq p_2\geq\cdots\geq p_{\mathcal{N}}$，

其中$\mathcal{N}$是$\mathcal{X}$中独立口令的数目。在我们回顾α-guesswork $G_\alpha(\mathcal{X})$之前，先回顾两个更早的基于统计学的口令分布强度评价指标，即β-success-rate $\lambda_\beta(\mathcal{X})$[140]和$\alpha$-work-factor $\mu_\alpha(\mathcal{X})$[141]：

$$\lambda_\beta(\mathcal{X}) = \sum_{i=1}^{\beta} p_i, \tag{2.21}$$

它测量的是$\mathcal{A}$对每个帐户至多可进行β次猜测的情况下，$\mathcal{A}$的平均优势。

$$\mu_\alpha(\mathcal{X}) = min\Big\{j | \sum_{i=1}^{j} p_i \geq \alpha\Big\}, \tag{2.22}$$

其中0<α≤1。$\mu_\alpha(\mathcal{X})$代表当 $\mathcal{A}$想要攻破不低于α比例的帐户时，需要对每个帐户进行的最少猜测次数。不难看出，当对每个帐户进行μ_α次猜测时，λ_{μ_α}代表$\mathcal{A}$的实际优势且有 $\lambda_{\mu_\alpha} \geq \alpha$ 。结合上述两个定义，α-guesswork[5]被定义如下：

$$G_\alpha(\mathcal{X}) = (1 - \lambda_{\mu_\alpha}) \cdot \mu_\alpha + \sum_{i=1}^{\mu_\alpha} p_i \cdot i. \tag{2.23}$$

式2.23考虑的是漫步猜测攻击者，其底层原理有二：(1)对于每个不在$\mathcal{A}$的猜测集里的目标口令，$\mathcal{A}$将尝试μ_α次猜测，这产生了式2.23右部的第一部分；(3)对于每个在$\mathcal{A}$ 的猜测集里的目标口令，$\mathcal{A}$将按最优顺序猜测，共产生$\sum_{i=1}^{\mu_\alpha} p_i \cdot i$次尝试，即为式2.23右部的第二部分。$G_\alpha(\mathcal{X})$测量的是，为达到成功率$\alpha$，$\mathcal{A}$必须对每个帐户进行的平均猜测次数。$G_\alpha(\mathcal{X})$较好的刻画了现实中攻击者的特点：$\mathcal{A}$注重成本和收益，将避免去攻击那些低频口令。

(2) α-guesswork指标的缺陷

α-guesswork评价已广泛用于最近的研究中(见[84, 142–146])，相应的论文也获得了“NSA 2013年度最佳网络安全学术论文奖”[222]。然而，它不具有单调性，总是参数化于α。比如，$G_{0.51}(\mathcal{X}_A) > G_{0.51}(\mathcal{X}_B)$永远不能保证$G_{0.52}(\mathcal{X}_A) \geq G_{0.52}(\mathcal{X}_B)$成立。Bonneau[5]坦诚地写道：“我们不能依赖于任何单一的 α 值，因为每个值提供的信息都代表完全不同的攻击场景。”因此，为了公平比较两个口令$\mathcal{X}_A$和$\mathcal{X}_B$的强弱，我们必须绘制整个[0, 1]区间内$G_\alpha(\mathcal{X}_A)$和$G_\alpha(\mathcal{X}_B)$的曲线，十分繁琐。这一缺陷本质上是因为α-guesswork没有利用所测量的数据集的分布函数信息，实际上它可以用来评测服从任意离散分布的任何数据集，不仅仅是用户生成的口令集。

(3) 我们的新发现

有趣的是，基于口令分布的确切知识(即口令服从Zipf分布)，我们发现在总共四种情形中，有两种情形下G_α具有单调性，不再是参数化的。需要注意，λ_β表示在最优猜测顺序下，$\mathcal{A}$进行β次猜测的成功率。这意味着，以口令排

名r为x-轴，λ_β的曲线与口令集$\mathcal{X}$的CDF曲线本质上是同一条曲线。如节2.5所证实的，我们的CDF-Zipf模型可以很好地刻画$\mathcal{X}$的CDF曲线。因此，我们有：

$$\lambda_\beta = \sum_{i=1}^{\beta} p_i \approx F_\beta = C' \cdot \beta^{s'}. \tag{2.24}$$

其中β是在最佳攻击序列下进行的在线猜测的次数。

定理 1 对于两个口令分布 $\mathcal{X}_A$ 和 $\mathcal{X}_B$，假设$C'_A \le C'_B$, $s'_A \le s'_B$。则

$$G_\alpha(\mathcal{X}_A) \ge G_\alpha(\mathcal{X}_B),$$

其中$0 \le \alpha \le 1$。如果上述两个不等式中任一个的条件是严格的，则$G_\alpha(\mathcal{X}_A) > G_\alpha(\mathcal{X}_B)$，其中 $0 < \alpha \le 1$。

证明。首先，由 $C'_A \le C'_B$ 和 $s'_A \le s'_B$，我们可以得到$\lambda_\beta(\mathcal{X}_A) \le \lambda_\beta(\mathcal{X}_B)$：

$$\lambda_\beta(\mathcal{X}_A) = \sum_{i=1}^{\beta} p_i^A \approx F_\beta(\mathcal{X}_A) = C'_A \cdot \beta^{s'_A} \le C'_B \cdot \beta^{s'_B} = F_\beta(\mathcal{X}_B) \approx \sum_{i=1}^{\beta} p_i^B = \lambda_\beta(\mathcal{X}_B), \tag{2.25}$$

其中 $1 \le \beta \le \max\{|\mathcal{X}_A|, |\mathcal{X}_B|\}$。接着，由式 2.22 和 $\lambda_\beta(\mathcal{X}_A) \le \lambda_\beta(\mathcal{X}_B)$ (即，式 2.25)，可以自然地得到 $\mu_\alpha(\mathcal{X}_A) \ge \mu_\alpha(\mathcal{X}_B)$。然后，我们可以推出

$$\begin{aligned} G_\alpha =& (1-\lambda_{\mu_\alpha}) \cdot \mu_\alpha + \sum_{i=1}^{\mu_\alpha} p_i \cdot i = \sum_{i=1}^{\mu_\alpha} \sum_{j=1}^{i} p_i + (1-\lambda_{\mu_\alpha}) \cdot \mu_\alpha \\ =& \sum_{j=1}^{\mu_\alpha} \sum_{i=j}^{\mu_\alpha} p_i + \sum_{j=1}^{\mu_\alpha} (1-\lambda_{\mu_\alpha}) = \sum_{j=1}^{\mu_\alpha} (1-\lambda_{\mu_\alpha} + \sum_{i=j}^{\mu_\alpha} p_i) \\ =& \sum_{j=1}^{\mu_\alpha} (1-\lambda_{j-1}). \end{aligned}$$

由于 $\mu_\alpha(\mathcal{X}_A) \ge \mu_\alpha(\mathcal{X}_B)$ 及 $\lambda_j(\mathcal{X}_A) \le \lambda_j(\mathcal{X}_B)$，我们有

$$G_\alpha(\mathcal{X}_A) \ge G_\alpha(\mathcal{X}_B).$$

如果定理1的两个条件中的任一个是严格的，那么有 $G_\alpha(\mathcal{X}_A) > G_\alpha(\mathcal{X}_B)$，其中 $0 < \alpha \le 1$。

推论 1 假设 $C'_A \ge C'_B, s'_A \ge s'_B$。则

$$G_\alpha(\mathcal{X}_A) \le G_\alpha(\mathcal{X}_B),$$

这一推论显然成立，因为它恰好是定理1的逆否命题。

给定两个口令集A和B，我们可以先用黄金分割法(见节2.4.3)来获得它们的CDF-Zipf模型参数(即C'_A、s'_A、C'_B和s'_B)，然后分别将C'_A与C'_B、s'_A与s'_B进行对比。这样会出现四种情况，其中两种情形下(即$\{C'_A \ge C'_B, s'_A \ge s'_B\}$和 $\{C'_A \le$

$C'_B, s'_A \leq s'_B\}$)，进一步根据上述定理和推论，可以发现α-guesswork[5]具有单调性。因此，在这两种情形下，α-guesswork的应用可大为简化。

可类似地证明，在上述两种情形下，节2.5中提出的口令分布强度评价指标$\lambda_n(\mathcal{X})$是随猜测次数n单调的。需要指出的是，虽然$G_\alpha(\mathcal{X})$ 和$\lambda_n(\mathcal{X})$都是关于口令分布强度评价指标，都考虑的是最优漫步猜测攻击，它们各自有不同的侧重点，是互补的：$G_\alpha(\mathcal{X})$ 关注的是，为达到成功率α，$\mathcal{A}$针对每个帐户进行的平均猜测次数，即给定α时$\mathcal{A}$的工作量；$\lambda_n(\mathcal{X})$关注的是，针对每个帐户至多进行n次猜测，$\mathcal{A}$可达到的成功率上限，即限定n时$\mathcal{A}$的优势上限。

小结。本节揭示了“口令服从Zipf分布”这一发现产生的三个重要启示。现有基于口令的安全协议、口令生成策略和评价指标或是基于“口令服从均匀分布”的假设，或是简单地规避对口令分布做明确假设，但用户生成的口令实际上服从Zipf定律，一个与均匀分布迥异的分布，因此在新的口令分布假设下修正(或改善)现有研究是十分必要的。我们设计了更准确的安全性表达函数，以刻画可证明安全的口令密码协议所能提供的安全性保障；提出了合理的流行度阈值$\mathcal{T}$，以更好地权衡安全性和可用性；证明了α-guesswork评价指标的一个新的本质特性，以促进其更方便地应用。特别地，我们在14个大规模口令集上进行了充分实验，证实了所提结论的*有效性*。由于我们的口令集包含1.28亿真实口令，涵盖了各种互联网服务和多样化的用户群体，因此我们的结果也具有*普适性*。可以预期，“口令服从Zipf分布”这一发现将会被用于其它一些场合，一个典例就是用来评估那些需要重建用户口令分布的算法的优劣(如honeywords生成算法、口令hash算法和用户调查程序等)。

2.7　本章小结

本节成功地解决了三个基本问题：(1)*用户生成的口令服从什么分布？*(2)*如何准确地评测一个口令分布的强度？*(3)*口令分布的新发现会带来什么启示？*通过引入自然语言处理技术、统计计算技术和可证明安全理论，基于14 个大规模口令集的1.13亿个真实口令，提出了PDF-Zipf和CDF-Zipf 两个理论模型刻画口令分布，尤其是CDF-Zipf模型可以精确地刻画整个口令集的分布。基于确切的口令分布函数，提出了一种新的口令分布强度评价指标$\lambda_n(\mathcal{X})$。

我们从口令密码协议、口令生成策略和口令分布强度评价指标三方面，揭示了口令分布的新发现所产生的重要影响。可以预见，本节所提出的“用户口令服从Zipf分布”这一基础性理论必将对更多口令相关研究领域产生深远影响，为它们未来的理论发展和实际应用奠定基础。截止目前，本节所提出的口令Zipf分布理论和一些数值结果已经被初步采用到多个口令研究领域，如基因

组口令保护系统[223]、口令哈希函数[192]、口令管理器[224]等等。

针对这一有趣且富有挑战性的课题，还有更多的工作可做。比如，如何更准确(且高效地)确定口令分布的Zipf参数？导致口令服从Zipf分布的底层动力学机制是什么？随着时间的推移，系统的口令分布将如何演变？极高价值服务(如电子银行)的口令是否也服从Zipf定律？解决这些问题往往涉及比本节更广范围内的交叉学科知识，需要更多来自不同领域的研究力量的合力攻关；我们也喜忧参半地祝福，随着更多类型、更高价值的系统被攻破，更丰富、更大规模的用户帐户信息被公开，解决这些问题的客观条件越来越好。

第三章　定向在线口令猜测模型

本章针对“如何科学评估定向在线口令猜测威胁”这一公开难题，首次进行了系统性的研究。提出了一个定向在线口令猜测攻击框架，包含7个理论攻击模型以刻画7种不同能力的现实攻击者。大规模真实数据测试结果显示，在允许猜测100次的情况下，如果仅知道目标用户的一些常见个人信息，成功率可达20%；如果知道用户在其它网站曾经泄露的一个口令，成功率可达73%。这一结果大大超出了此前人们的预期，已经引起美国国家身份认证标准NIST SP800-63-3的修订。本章将为后文第四、五、八章奠定理论基础。

3.1　研究背景

口令在当前各类计算机系统中广泛应用，在可预见的未来仍无可替代。为研究口令的安全性，学者们提出了众多口令猜测概率模型，如Markov n-grams[34]和概率上下文无关文法PCFG[68]。这些模型的一个共同点是，都致力于刻画一个漫步离线猜测攻击者，这种攻击者拥有泄露的口令文件，其攻击目标在于尽可能多地恢复出文件中的口令。如文献[190] 所指出的，不论是漫步离线猜测攻击，还是定向离线猜测攻击，只在非常受限的一种情形下才构成现实威胁：服务器端采用了正确的口令加盐hash方法，口令文件被泄露且未被察觉。近来，学术界(见[25, 190])认识到，防御离线猜测攻击的责任应当在服务器端，即服务器端应确保口令文件安全存储，而对普通用户来说，他们只需要确保自己的口令可以抵抗在线猜测攻击即可。

只要服务器公开接受用户请求，在线口令猜测攻击可以由任何人在任何时候仅通过一个浏览器即可发起，其面临的最主要限制是服务器允许的猜测次数。*漫步在线猜测*主要利用了用户倾向于选择流行口令的行为(见节2.6)，这一般通过在服务器端部署一些安全措施来防范，如可疑登录检测[19]、登录频率限制和帐户锁定[38]。然而，*定向在线猜测*(见图3.1)不仅利用普通用户使用流行口令的脆弱行为，而且会利用用户重用口令和使用个人信息构造口令的脆弱行为。

随着大规模个人信息泄露事件的不断发生(见[225–227])，各种类型的个人可标识信息(Personally Identifiable Information，PII)和用户在其它网站使用的口令都越来越容易被攻击者获取，定向在线猜测逐渐成为一个十分严峻的现实威胁。比如，2016年4 月发生在土耳其的一起公民信息泄露事件，涉及土耳其总人口的64%，共计5000万土耳其公民[227]；据CNNIC的2015年度报告，6.68亿中国网民中超过78.2%都曾遭遇过PII数据泄露[228]; 最近在美国发生的一系列信

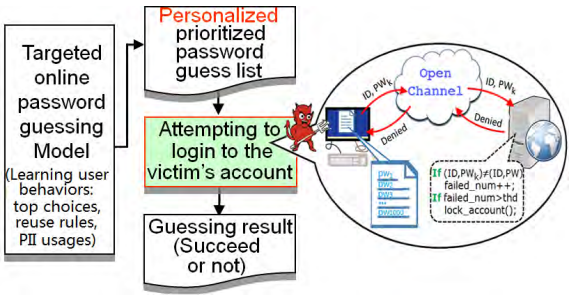


图 3.1 定向在线口令猜测

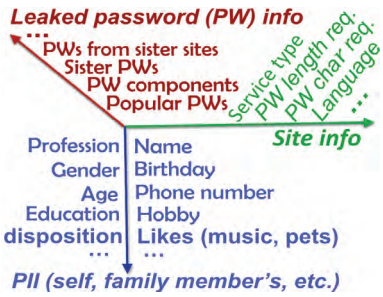


图 3.2 $\mathcal{A}$ 可利用的相关信息

息泄露事件，让超过2.53亿的美国网民成为PII和口令泄露的受害者[229]。

这意味着，现有建立在那些漫步猜测概率模型(如[34, 66, 68, 197])之上的口令生成规则(如[111, 116])和口令强度评价算法(如[71, 105])，只考虑了十分受限的离线猜测威胁，而无法防御越来越现实、危害越来越大的定向在线猜测攻击。这种研究重点的错位，导致学术界提出的口令安全解决方案并不比工业界现行方案更令人信服，最终是学术界没有能够引领工业界(见[25])，工业界各行其是，总体呈现一片混乱的局面(见[73, 120])。

定向在线猜测研究的核心问题是如何刻画攻击者$\mathcal{A}$的猜测模型，以有效捕捉$\mathcal{A}$可利用的多维度信息(见图3.2)，而$\mathcal{A}$被允许的猜测次数通常却十分有限—美国国家身份认证指南NIST SP800-63-2[37]规定，对于等级1或2的系统，30天内每个帐户允许的错误口令登录次数不应超过100。下面详细解释，为什么设计定向在线口令攻击的猜测模型是一个挑战。

第一，不同用户选择口令的习惯差异巨大。有的直接使用一个现有的口令，有的可能会在现有的口令基础上做一些修改；有的把PII加入自己的口令中，而有的就不这样做；有的喜欢用数字构造口令，有的可能更喜欢使用字母，如此等等。这意味着，给定一个目标系统，用户彼此间的口令会有很大的差异。因此，只能为所有用户生成一个同样的猜测集的漫步口令猜测模型(如[34, 66, 68, 197])不适于刻画定向猜测。

第二，用户的PII严重异构。有些PII(如姓名和爱好)由字母组成，有些PII(如生日和手机号)由数字组成，还有些PII(如用户名)可能会同时包含字母、数字和特殊符号。如图 3.2 所示，有些PII(比如姓名和爱好)可直接作为口令的组成部分，而另外一些PII(如性别和受教育程度)不能直接作为口令的组成部分。后文我们将会证实，绝大多数的PII对用户的口令选择都有一定影响。因此，在允许的猜测次数极其有限的情况下，如何大规模、自动化地将这些严重异构的PII引入猜测模型中是困难的。

第三，用户在修改现有口令时，有一系列各异的变换规则可供使用。如文献[7, 230]所述，给定一个口令，可通过很多变换规则来产生一个新口令，如插

入、删除、大小写变换、字符跳变(如password→ passw0rd)等等，这些变换规则也可以综合使用(如password→ Passw0rd1)。针对每个不同用户，如何将这些规则进行优先级排序是困难的。

第四，用户会选择什么样的口令变换规则，往往与上下文相关。假设攻击者$\mathcal{A}$ 试图猜测Alice在eBay的帐户口令，$\mathcal{A}$还知道Alice已经30多岁了。那么，如果$\mathcal{A}$知道Alice的一个从Yahoo泄露的姊妹口令Alice1978Yahoo，考虑到人类行为的惯性，$\mathcal{A}$尝试口令Alice1978eBay的成功率将比尝试Alice1978高。而如果泄露的口令是123456，猜测Alice1978的成功率则会更高。如果进一步考虑网站的口令生成策略(假设eBay要求口令长度为10位以上)，情况又会更复杂。要处理这些上下文相关性，需要一个自适应的、可识别语义的跨站猜测模型。

3.1.1　相关研究

Zhang等[11]研究了如何通过用户在一个帐户上的旧口令预测同一个帐户的新口令，提出了一个启发式预测算法。Das等[7]首次研究了用户口令重用行为带来的威胁，提出了一个跨站口令猜测算法。但是，Das等算法远未达到最优：(1)没有考虑利用常见的流行口令(如口令12345678、iloveyou和 pa$$w0rd)，这类口令与重用和个人PII都无关；(2)假设所有用户都按照固定的优先级选择口令变换规则，但他们自己也承认，用户采用的口令变换规则会因人而异、动态变化且与上下文环境紧密相关；(3)没有考虑混合变换规则的情况；(4)算法是启发式的，缺乏严格的理论基础。

Li等[103]研究了用户的PII会在何种程度上影响口令安全性。他们发现60.1%的用户在口令中使用了至少一种自己本人的PII，并设计了一个可识别PII语义的定向在线口令猜测模型Personal-PCFG。Personal-PCFG考虑了六种PII：姓名、生日、电话号码、身份证号码、邮箱地址和用户名。然而，如本文节3.4所显示的，Personal-PCFG采用了基于长度的PII匹配和替换方法，不能精确地捕捉口令中的PII使用频率，以致口令猜测效率低下。

3.1.2　本章贡献

本文主要作出了以下贡献：

- **一个实用框架**。为科学评估定向在线口令猜测攻击带来的威胁程度，我们设计了一个名为*TarGuess*的攻击框架。*TarGuess*基于严格的数学概率模型，摆脱了启发式设计，刻画了7种最为典型的基于不同个人信息组合的定向在线猜测攻击场景。
- **四个概率模型**。基于PCFG、Markov和贝叶斯理论，提出了4个概率模型，以建模最具代表性的4种定向在线猜测攻击场景。这4个模型的攻击效果都

表 3.1 定向口令猜测环境下用户个人信息(PI)的释义†

Different kinds of personal info		Degree of secrecy	Roles in PWs	**Considered** in this work(✓)	Not Considered in this work(×)
PII	Type-1	Semipublic	Explicit	Name, Birthday, Phone #, NID	Place of birth, Likes, Hobbies, etc.
	Type-2	Semipublic	**Implicit**	Gender, Age, Language	Faith, Disposition, Education, etc.
User identification credentials		Private	Explicit	Passwords, PIN	Finger prints, Private keys, etc.
		Public	Explicit	User name, Email address	Debit card number, Health IDs, etc.
Other personal data		—	—	—	Employment, Financial records, etc.

† PI = Personal information; PII=Personally identifiable information; PW = password; PIN = Personal Identification Numbers; NID = National identification number, e.g. social security number.

远优于当前最先进的同类算法。我们进一步讨论了如何基于这4个模型，来建模另外3个定向攻击场景。

- **大规模实验。** 我们进行了一系列实证实验来显示所提框架和模型的有效性和普适性。实验结果表明，相当大比例用户的口令无法抵抗我们的定向在线猜测攻击，这一攻击的危害程度当前被学术界和工业界严重低估了。
- **新的见解。** 我们发现，对定向在线猜测攻击来说，基于类型的PII标签比基于长度的PII标签更为有效；简单地在猜测模型中增加PII的种类，并不一定会提高攻破率，这是反直觉的发现；随着猜测数的增大，单次猜测的成功率依Zipf定律递减。

3.2 预备知识

本节对个人信息进行分类，并详述安全模型。

3.2.1 个人信息分类

定向猜测攻击和漫步猜测攻击的最突出区别在于前者使用了目标用户的特定数据。这些数据也被称为“个人信息”(Personal information, PI)，有时还被称为“个人可标识信息”(Personally identifiable info, PII)[103, 231]，但它们的含义在不同场合差异很大[232, 233]。通常来说，一个用户的个人信息是指“和此用户有关的任何信息”[233]，这个范围要比PII更广。为更好地理解个人信息对用户口令安全性的影响，我们在口令猜测语境下对个人信息进行了首次分类(见表3.1)。这一分类使系统化地研究定向口令猜测成为可能。

我们将个人信息PI分为三类，每一类的私密程度、在口令构造中的作用、含有的元素都各不相同。第一类是用户的PII(如姓名和性别)。它们天然是半公开的：对朋友、同事、熟人是公开的，而对陌生人是秘密的。第二类是用户认证凭证，有些(如用户名、邮箱)是公开的，有些(如口令)是保密的。剩下的用户个人信息归于第三类，与本文的讨论无关。我们进一步把PII分为两类：Type-1和Type-2。Type-1 PII(如姓名和生日)可以直接作为口令的一部分，而Type-2 PII(如性别和受教育程度[86]) 则不能直接作为口令的一部分，但会影响用户的口令生成行为。这两类PII将对设计定向猜测模型产生重要影响。

表 3.2 四种最具代表性的定向在线口令猜测场景

Attacking scenario	Exploiting *public* info (e.g., datasets and policies)	Exploiting user *personal* info †			Existing literature	**Our model**
		One sister PW	Type-1 PII	Type-2 PII		
Trawling #1	✓				Ref.[34, 66, 68]	
Targeted #1	✓		✓		Ref.[103]	TarGuess-I
Targeted #2		✓			Ref.[7]	
	✓	✓			None	TarGuess-II*
Targeted #3	✓	✓	✓		None	TarGuess-III
Targeted #4	✓	✓	✓	✓	None	TarGuess-IV

*As public password datasets are readily available, TarGuess-II and [7] is comparable because they exploit the same type of user PII.

†A total of 7($=C_3^1+C_3^2+C_3^3$) scenarios result from combining the three types of personal info. With TarGuess-I∼IV, all 7 cases will be tackled in Section 3.4.

这里我们着重强调一种特殊的用户个人信息—用户在其它系统中使用的口令。如文献[7, 234]所述，在向一个新系统注册口令时，用户倾向于重用或修改一个现有口令(称之为*姊妹口令*，Sister password)作为新帐户的口令。随着大规模口令泄露事件不断发生(见[28, 225, 229])，用户的姊妹口令越来越容易被攻击者获取。比如，近期针对4000美英用户的一个调查显示，27% 被访者承认自己口令曾被泄露[235]；Shape公司2017年初发布的数据泄露报告显示，2016年里至少有51个公司的33亿口令遭遇泄露，黑客会使用这些泄露口令来尝试登录用户在其它网站的帐户，其中电商、金融、旅游、政府网站是主要攻击目标[33]。

3.2.2　安全模型

不失一般性，本文我们主要关注最为基本、常见的客户端-服务器用户认证架构(见图3.1的右半部分)。在定向在线口令猜测攻击中，涉及三个实体：用户U，认证服务器S和攻击者$\mathcal{A}$。

用户U在服务器S上注册了一个帐户。口令只储存在S上，但U在其它网站上的口令可能已经泄露[228, 236]。服务器S可以在远程(如电子商务网站)，也可以是本地的(如受口令保护的设备)。为安全模型更符合实际，我们假设S 执行了一些常见的在线猜测预防机制，如可疑登录检测、登录频率限制和帐户锁定等[19, 38]，因此$\mathcal{A}$的猜测次数将十分受限(如10^{2}[37, 38])。$\mathcal{A}$知道U的一些个人信息，比如$\mathcal{A}$可能是一个好奇的朋友，忌妒的妻子，竞争对手，甚至是地下市场收买个人信息的黑客团体。

攻击者$\mathcal{A}$可获取的关于U的信息类型多种多样(见图3.2)，并且如何综合利用这些严重异构的信息没有明显的规律可循，这给刻画$\mathcal{A}$ 的猜测行为带来了巨大困难。为解决这一难题，我们假设$\mathcal{A}$可以获取所有的公开信息(如遭遇泄露的公开口令集和目标网站的口令策略)，然后根据$\mathcal{A}$ 获知的U的个人信息类型的不同，定义一系列攻击场景(见表3.2)。我们的解决办法是合理的，因为：(1)$\mathcal{A}$是聪明的，会尽可能利用各种公开信息来提高猜测成功率；(2)$\mathcal{A}$会根据拥有U的个人信息的不同，来调整、优化其攻击策略。

表 3.3 10个口令集的基本信息

Dataset	Web service	Language	When leaked	Total PWs	With PII
Dodonew	E-commerce	Chinese	Dec., 2011	16,258,891	
CSDN	Programmer	Chinese	Dec., 2011	6,428,277	
126	Email	Chinese	Dec., 2011	6,392,568	
12306	Train ticketing	Chinese	Dec., 2014	129,303	✓
Rockyou	Social forum	English	Dec., 2009	32,581,870	
000webhost	Web hosting	English	Oct., 2015	15,251,073	
Yahoo	Web portal	English	July, 2012	442,834	
Rootkit	Hacker forum	English	Feb., 2011	69,418	✓
Xiaomi*	Mobile, cloud	Chinese	May, 2014	8,281,385	
Xato	Synthesised	English	Feb., 2015	9,997,772	

*Xiaomi passwords are in salted-hash and will be used as real targets.

表 3.4 个人信息数据集的基本信息

Dataset	Language	Number of Items	Types of PII useful for this work
Hotel	Chinese	20,051,426	Name, Gender, Birthday, Phone, NID
51job	Chinese	2,327,571	Email, Name, Gender, Birthday, Phone
12306	Chinese	129,303	Email, User name, Name, Gender, Birthday, Phone, NID
Rootkit	English	69,324	Email, User name, Name, Age, Birthday

注意，本文只考虑了$\mathcal{A}$至多拥有U的一个姊妹口令的情形。这是因为，在我们历时6年收集的5.48亿公开泄露的口令数据中，不超过1.02%的邮箱地址能匹配1个以上的口令，不超过1.73%的用户名能匹配1个以上的口令。类似的，在Das等[7]收集的796万口令数据中，只有152个(0.00191%)邮箱地址可以匹配1个以上的口令。因此，*假设绝大多数用户都泄露了一个姊妹口令，并且$\mathcal{A}$可以利用U的这个姊妹口令来提高猜测效率，是合理的。*

3.3 用户构造口令的行为

本节展开一个关于人类构造口令行为的大规模实证研究，包括用户如何选择流行口令，如何重用口令和如何使用自己的PII构造口令三个视角。

3.3.1 数据集

本章的实验评估使用了10个大规模真实口令集(见表3.3)，来自5 个英文网站和5个中文网站，涵盖众多主流的网络服务。它们由黑客或者内部人员泄露，在互联网上可公开下载，其中一些已经被广泛用于漫步口令概率模型[34, 105, 197]。Rootkit 口令集原本含有71,228个经MD5哈希过的口令，我们通过TarGuess-IV和一些其它的漫步猜测模型(如[34, 105, 197])，在一周内成功恢复出了其中的97.46%。这些口令集总共含有9583万个明文口令。在节3.6中我们将说明这些口令集的角色，即如何使用它们。

为充分地理解用户PII在生成口令过程中的作用，我们需要大规模带PII的口令集。但是如表3.4所示，只有两个口令集天然带有用户的PII。为扩充带PII的用户口令数量，我们进一步使用了两个辅助的中文PII数据集，可将它们与前述不带PII 的口令集通过邮箱地址匹配，以补充不带PII的口令集中缺失的PII。最终，共产生7个带PII 的口令集(见表3.6的第1行)。其中，中文带PII的

表 3.5 用户选择的top-10口令

Rank	Dodonew	CSDN	126	12306	Rockyou	000webhost	Xato	Yahoo	Rootkit
1	123456	123456789	123456	123456	123456	abc123	123456	123456	123456
2	a123456	12345678	123456789	a123456	12345	123456a	password	password	password
3	123456789	11111111	111111	**5201314**	123456789	12qw23we	12345678	welcome	**rootkit**
4	111111	**dearbook**	password	123456a	password	123abc	qwerty	ninja	111111
5	**5201314**	00000000	000000	111111	**iloveyou**	a123456	123456789	abc123	12345678
6	123123	123123123	123123	**woaini1314**	**princess**	123qwe	12345	123456789	qwerty
7	a321654	1234567890	12345678	123123	1234567	secret666	1234	12345678	123456789
8	12345	**88888888**	**5201314**	000000	**rockyou**	********†	111111	sunshine	123123
9	000000	111111111	**18881888**	qq123456	12345678	asd123	1234567	**princess**	qwertyui
10	123456a	147258369	1234567	1qaz2wsx	abc123	qwerty123	dragon	qwerty	12345
%	3.28%	10.44%	3.52%	1.28%	2.05%	0.79%	1.46%	1.01%	3.94%

†The letter-part (i.e., `YfDbUfNjH`) of the 7th most popular password `YfDbUfNjH10305070` in 000webhost can be mapped to a Russian word which means “navigator”. Why it is so popular is beyond our comprehension.

口令集里的个人信息比较完整，但由于英文Rootkit口令集有17.90%的姓名项和54.04%的生日项空白，导致匹配之后产生的3个带PII的英文口令集个人信息不完整。这些缺失PII项可能会导致针对英文用户的定向猜测攻破率降低。尽我们所知，迄今所有评估定向在线口令猜测安全威胁的研究中，本工作所使用的口令集规模最大、最为多样。

3.3.2　使用流行口令

表3.5展示了在不同服务中用户选择流行口令的情况。令人担心的是，只需要尝试最流行的前10个口令，就可以攻破0.79%~10.44%的用户帐户。总的来说，中文流行口令比英文流行口令的分布更加集中(见表3.5最后一行)，因此中文用户更容易遭受在线猜测的威胁。大部分中文流行口令都仅由数字组成，而英文流行口令一般是有涵义的字母串或是由键盘键位模式构成的口令。爱情在流行口令中也显示了重要地位——`iloveyou`和`princess`在两个英文网站中排前10，而`5201314`，在中文里谐音为“我爱你一生一世”，在3个中文网站的流行口令中排前10。文化因素也是产生流行口令的重要原因(如`18881888`)，此外网站名称(如`rockyou`和`rootkit`)也对口令构造有重要影响。

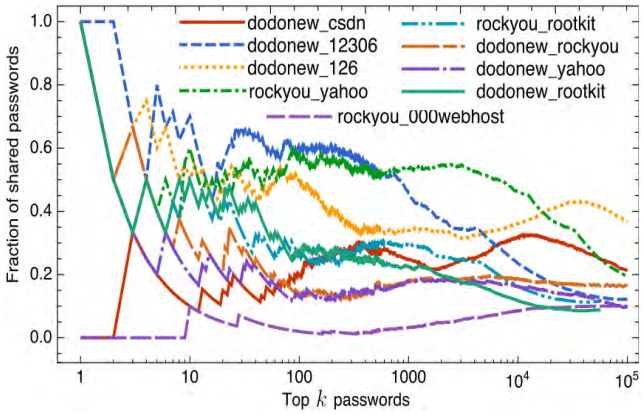


图 3.3 两个网站之间口令的共用比例。

图3.3显示了在不同的排序阈值k下，两个不同服务中top-k口令相同的比例。一般来说，具有相同语言的两个服务，他们的口令集间top-k口令相同的比例会

明显更高一些。另外，如果k大于10，任何两个服务的口令集之间这一比例都会小于60%。这表明，语言和服务类型对用户的口令构造行为有重要的影响。

需要注意的是，Rockyou和000webhost两个英文口令集之间的公共口令比例非常少。我们仔细观测了这两个口令集，发现000webhost口令集里99.29%的口令都同时含有字母和数字(见表2.2)，说明该网站执行了较强的口令生成策略。表3.5也印证了这一点：000webhost口令集中top-10的口令都是由字母和数字组成。类似地可以发现，CSDN执行了“口令长度必须不低于8”的策略。

3.3.3 重用口令

尽管用户现在维护(记忆)的口令帐户数量很可能是10年前的好几倍，但是人类大脑的记忆能力却几乎保持稳定。因此，用户倾向于在不同的服务中重用口令[190, 234]。当前，已有一些针对欧美用户口令重用行为的实证研究(见[7, 85])。然而尽我们所知，尚没有针对中文用户的类似研究，尽管中文网民数量已经在2016年6月达到了7.10亿[237]，占整个全球互联网网民的20%以上。

为弥补这一空白，我们通过email来匹配12306和Dododnew这两个口令集，并进一步去除口令相同的帐户。这产生了一个新的口令集12306&Dodonew，其中每个用户都有两个互异的姊妹口令。类似地，我们总共生成了2个匹配的中文口令集和3个匹配的英文口令集(见图3.4)。在匹配的过程中，我们发现34.02%~71.11%中文用户的姊妹口令对是完全一样的(这些口令对被剔除了)，而只有6.25%~21.96%的英美用户有相同的姊妹口令(见节3.5.1)。这意味着，英美用户口令重用的程度更低一些。

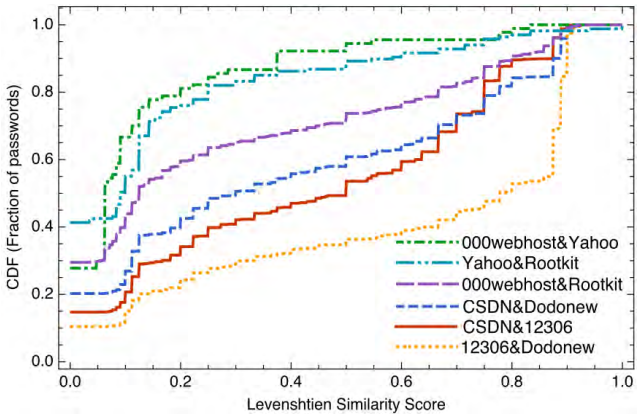


图 3.4 采用文氏距离来评价同一个用户在两个不同网络服务上的口令的相似度。大多数用户使用比较复杂的方法来修改口令。

我们采用广泛接受的文氏距离(Levenshtein-distance)这一指标，来评估同一用户的两个不同口令间的相似度。图3.4表明，中文用户的姊妹口令通常比英美用户的相似度更高，这意味着中文用户修改口令的方式更简单。约30%中

文用户的互异口令对间的相似度在0.7∼1.0，而英美用户相似度在这个区间的比例不到20%。我们进一步使用了“最长公共子序列”(the longest-common-subsequence)这一相似度评估指标，得到了类似的结论。结果表明，大多数用户都会用比较复杂的方式来修改口令，这使得建立一个刻画用户修改口令行为的模型具有相当挑战性。

需要指出的是，在我们的数据集中，英美用户重用口令现象少并且修改口令的方式更复杂，这极可能是因为Rootkit是一个黑客论坛，而且000webhost的用户一般是网站管理员。意味着，这些用户会比普通用户具有更高的安全意识。因此，在口令集Rootkit&000webhost, Rootkit&Yahoo和000webhost&Yahoo中的用户，会具有比普通英文用户更安全的口令重用行为。

2014年，Das等[7]发现在普通的英文用户中，有43%使用完全相同的姊妹口令对，这和我们在中文用户中揭示的结果大致相同，但比我们的英文用户中的结果高出了2∼6倍以上。他们也发现，大约有30%的互异口令对间的相似度在0.7∼1.0，这与我们中文用户中的研究结果基本一致。另外，根据对中文普通用户(见节5.3)和英文普通用户[7, 119]的调查显示，这两个用户群体在口令重用上有很多相似的行为。总之，无论是实证还是用户调查的结果都表明中文和英文普通用户中有很多相似的口令重用行为，而本章的英文用户(黑客和网站管理员)代表的应是具有较高安全意识的用户。

3.3.4　利用个人信息构造口令

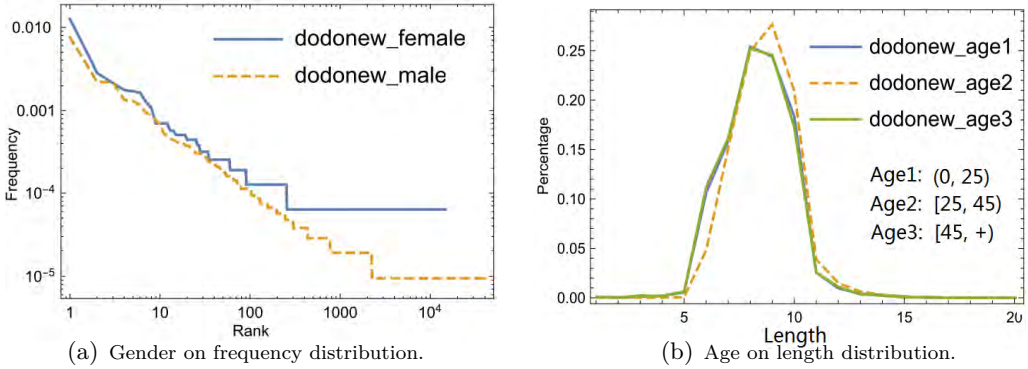
表3.6中展示了用户使用他们自己的显式PII来构造口令的比例。由于有些口令集没有PII(见表3.3)，我们将这些口令集与表3.4中相同语言的PII数据集通过匹配邮箱地址的方式关联起来，生成了7 个含有PII的口令集，每个含有PII信息的口令集样本大小见表 3.6 的第一行。这样，本文含有PII的口令数量远比文献[103] 丰富。结果正如我们预想的那样，这些高度异构的显式PII都是口令的组成部分，其中使用频率最高的PII是姓名和生日。特别地，有不可忽略比例的用户(0.75%∼1.87%)使用自己的全名(Full name)作口令，1.00%∼5.16%的中文用户使用自己的生日作口令。出乎意料的是，使用用户名或邮箱地址作为口令的频率也很高，分别占0.77%∼5.07%和0.54%∼2.34%。相比之下，本文英文用户在PII的使用上更加谨慎，一个合理的解释是他们是黑客用户，具有比普通用户更高的安全意识和安全行为能力。

图3.5展示了type-2 PII对用户口令行为的影响：(1)Dodonew中的女性用户的口令分布更加集中；(2)Dodonew中≤24岁用户的口令长度分布，与≥46 岁用户相似(配对χ^2检验，p值为0.009)，与25∼45岁的用户具有显著差异(配对χ^2检验，p-值$\leq 10^{-6}$)。其它口令集也得到了类似的结果。

表 3.6 用户使用(只使用)自己本人的可标识信息构造口令的比例†

Typical usages of PII (examples)	PII-Dodonew (161,510)		PII-126 (30,741)		PII-CSDN (77,439)		PII-12306 (129,303)		PII-Rootkit (69,330)		PII-Yahoo (214)		PII-000webhost(2,950)	
Full_name (`lei wang`, `john smith`)	4.68	0.82	3.00	1.32	4.85	1.81	5.02	1.13	1.38	0.75	2.34	1.87	2.44	1.32
Family_name (`wang`, `smith`)	11.15	0.01	6.16	0.00	9.75	0.00	11.23	0.00	2.28	0.78	4.67	1.87	3.73	1.46
Given_name (`lei`, `john`)	6.49	0.07	4.10	0.12	6.26	0.08	6.61	0.07	0.49	0.07	0.93	0.00	0.75	0.20
Abbr. full_name (`wl`, `lwang`, `js`, `jsmith`)	13.64	0.02	6.36	0.00	9.42	0.00	13.13	0.00	0.15	0.01	0.00	0.00	0.20	0.00
Birthday(19820607,06071982,07061982)	3.12	1.00	3.70	2.77	6.29	5.16	4.33	1.77	0.08	0.06	0.47	0.00	0.10	0.07
Year of birthday (1982)	8.92	0.00	8.84	0.01	11.37	0.00	10.78	0.00	0.75	0.01	1.40	0.00	1.12	0.00
Date of birthday (0607, 0706)	8.32	0.00	10.48	0.02	11.84	0.00	10.03	0.00	0.44	0.01	0.47	0.00	0.58	0.00
Abbr. birthday(198267, 820607)	2.37	0.59	2.60	1.71	2.89	1.45	3.31	1.12	0.10	0.05	0.00	0.00	0.20	0.14
Family_name+birthday (`wang19820607`)	0.08	0.08	0.05	0.05	0.03	0.03	0.14	0.14	0.00	0.00	0.00	0.00	0.00	0.00
Family_name+year of birth (`wang1982`)	0.55	0.22	0.20	0.07	0.22	0.07	0.64	0.25	0.01	0.00	0.00	0.00	0.00	0.00
Family_name+date of birth (`wang0607`)	0.12	0.09	0.05	0.03	0.08	0.04	0.16	0.12	0.01	0.00	0.00	0.00	0.00	0.00
User name (`icemoon12`, `bluebirdz`)	1.54	1.14	0.54	0.38	0.61	0.43	1.96	1.32	1.59	0.92	2.34	1.40	2.20	1.32
Email_prefix(`l0veu4ever`@example)	5.07	3.07	2.52	1.60	4.35	2.48	3.03	1.82	0.77	0.44	4.21	1.87	1.32	0.78
Chinese Phone number (`13511336677`)	0.10	0.10	0.48	0.45	0.50	0.45	0.07	0.01	—	—	—	—	—	—
'a'+birthday(`a19820607`)	0.16	0.13	0.04	0.02	0.03	0.02	0.16	0.12	0.00	0.00	0.00	0.00	0.00	0.00
Full_name+1 (`wanglei1`, `johnsmith1`)	1.49	0.22	0.51	0.03	0.84	0.03	1.65	0.17	0.06	0.01	0.00	0.00	0.03	0.00

†All the decimals in the table use '%' as the unit. For instance, 4.68 in the top left corner means that 4.68% of the 161,510 PII-associated Dodonew users employ their full name to build passwords; 0.82 means that 0.82% of these 161,510 Dodonew users' passwords are *just* their full names.



(a) Gender on frequency distribution. (b) Age on length distribution.

图 3.5 type-2 PII对用户口令生成的影响。性别和年龄都表现出了明显的影响。

基于类型的PII匹配。为了更加准确地识别口令中的PII信息，我们使用了基于类型的(type-based)PII字段匹配方法：在PCFG模型[68]的L、D、S标签基础上，我们引入了PII 主标签(如表示姓名的N，生日的B)和子标签的概念。其中，每一个PII主标签代表某一种PII；PII子标签带下标，其下标表示某一种PII的不同子类的编号，而不是代表长度。比如N_1表示姓氏(如li)，B_5表示出生年份(如1982)等等。更详细信息见节3.4.1。

这与Li等[103]基于长度的PII匹配方法有本质不同。Li等的PII标签，与原始PCFG模型[68]中L、D、S标签本质上类似，其下标代表某一种PII的长度。为避免误匹配，Li等只考虑了长度≥3的PII字段。比如，数字串19720123中所有长度≥3的子串(如197, 972, 720)都会被识别为一个生日的匹配。但是，这种匹配方法会在有些情况下低估PII使用率，而在另外一些情况下高估PII使用率。比如，如果目标用户的姓名是"Wei Li"，生日是19720123，口令是li.7201，那么这个口令会被标记为$L_2S_1Birth_4$，因为姓氏li的长度小于3。最流行的前50个中文姓氏拼音中，有20%长度都小于3(如li，wu和he)，这意味着很大一批姓氏的使用率会被基于长度的PII匹配方法低估。12306 口令集的13万用户中，有30,926(23.9%)个用户的姓氏拼音长度小于3，而其中4,346 个用户确实在他们的口令中包含了自己的姓氏，但文献[103]直接忽略了这种情况。

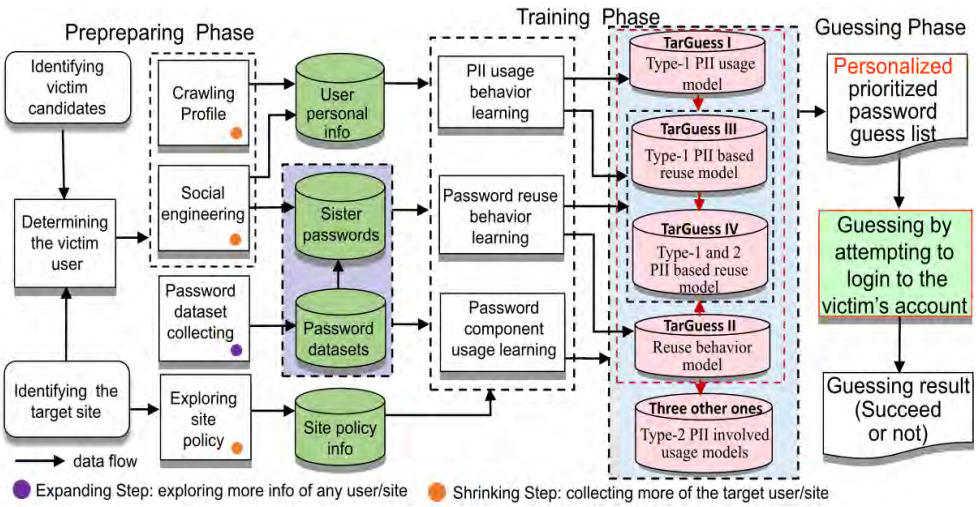


图 3.6　TarGuess 攻击框架概览。

另一方面，包含在流行数字口令(如123456，123456789，5201314)中的一些数字片段(如123，520，201)可能会和生日或者手机号正好重合，这样会导致对生日或者手机号使用频率的高估。节3.4.1我们进一步指出，这种基于长度的PII匹配方法还会导致在猜测集生成时的严重缺陷。遗憾的是，无论增加或减少匹配长度的阈值，都无法消除基于长度的PII匹配方法带来的这些问题。

小结。迄今所有评估定向在线口令猜测安全威胁的研究中，本工作所构造的含有PII的口令集规模最大、最为多样。特别是我们首次考虑了(具有较高安全意识)的英文用户。尽管用户的三种脆弱口令行为会增加口令被在线定向猜测的潜在风险，我们的结果显示，多样的上下文环境(如语言、服务类型和口令策略)、复杂的口令变换规则和异构的PII，使设计自动化的模型来评估这种威胁成为一个挑战，尤其是在猜测次数受限的情况下(如NIST[37, 38]只允许100次猜测/月)。

3.4　TarGuess：一个定向在线猜测框架

本节提出一个实用的框架*TarGuess*，以有效解决一个现实且兼具挑战性的问题：如何对各种定向在线猜测的攻击情景进行建模？如图3.6所示，TarGuess由三个阶段构成：准备、训练和猜测。其中，第一和第三阶段的设计比较直观，最核心的部分在第二阶段。TarGuess刻画了四种最具代表性的定向在线猜测场景，每一场景都基于$\mathcal{A}$可获取的不同种类的个人信息(见表3.2)：(i)Type-1 PII；(ii)姊妹口令；(iii)i和ii的结合；(iv)iii和type-2 PII的结合。

为对这4种最具代表性的攻击场景进行建模，基于一些统计计算理论如PCFG、Markov和贝叶斯理论，我们提出了4个相应的理论猜测模型：TarGuess-I∼IV。进一步，基于TarGuess-I∼IV，我们较好地解决了剩余三种攻击场景的建模问题。

3.4.1 TarGuess-I

TarGuess-I的目标是通过利用用户U的一些type-1 PII(如姓名和生日)，对U的口令进行在线猜测。它建立在Weir等[68]提出的基于PCFG的概率模型之上。Weir等PCFG模型在刻画漫步猜测攻击的场景时取得了很好的效果(见[34, 76])。在PCFG模型的训练阶段，每一个口令都被看作是字母段(L)，数字段(D)和特殊字符段(S)的组合。比如`loveyou@1314`会被解析为L段“loveyou”，S段“@”和D段“1314”，产生基本结构$\mathsf{L}_7\mathsf{S}_1\mathsf{D}_4$；类似地，`wanglei@1982`会被解析成$\mathsf{L}_7\mathsf{S}_1\mathsf{D}_4$。

我们的新模型。为感知口令中的PII语义，除了PCFG[68]模型中的L、D、S标签外，我们还定义了一系列基于个人信息类型的PII标签(如N_1∼N_7和 B_1∼B_{10})。对于每一个PII标签，它的下标数字代表这个类型的PII用法的一个细分(即子类型标签)，而不是像L，D，S标签那样，下标数字表示相应的长度。比如，N代表姓名信息的用法，而N_1表示姓名全称，N_2表示姓名全称的缩写(如“`lei wang`”缩写为`lw`)，⋯; B代表生日信息的用法，B_1表示以年月日的格式使用生日(如19820607)，B_2表示以月日年的格式使用生日，⋯。这样，我们就产生了一个可感知PII的上下文无关文法 $\mathcal{G}_I = (\mathcal{V}, \Sigma, \mathcal{S}, \mathcal{R})$，其中：

1) $\mathcal{S} \in \mathcal{V}$是起始符号。

2) $\mathcal{V} = \{\mathcal{S}; \mathsf{L}, \mathsf{D}, \mathsf{S}; \mathsf{N}_1, \cdots, \mathsf{N}_7; \mathsf{B}_1, \cdots, \mathsf{B}_{10}; \mathsf{A}_1, \mathsf{A}_2, \mathsf{A}_3; \mathsf{E}_1, \mathsf{E}_2, \mathsf{E}_3; \mathsf{P}_1, \mathsf{P}_2; \mathsf{I}_1, \mathsf{I}_2, \mathsf{I}_3\}$是一个有限的变量集合[Φ]，其中：(1) N_1 和 N_2 已经在前文定义了，N_3 表示姓氏(如 `wang`)，N_4 表示姓名，N_5 表示姓名的首字母+姓氏(如`lwang`)，N_6 表示姓氏+姓名的首字母(如 `wangl`)，N_7 表示姓氏并且首字母大写(如 `Wang`)；(2) B_1 和 B_2 已在前文定义，B_3 表示以日月年的格式使用生日(如 07061982)，B_4 表示出生日期，B_5 表示出生年份，B_6 表示出生年份+月份(如198206)，B_7 表示出生月份+年份(如 061982)，B_8 表示出生年份的后两位数字+以月日表示的日期(如 820607)，B_9表示以月日表示的日期+出生年份的后两位数字(如 060782)，B_{10} 表示以日月表示的日期+出生年份的后两位数字(如 070682)；(3)A 表示用户名信息的用法，A_1 表示用户名的全称(如 `icemoon12`)，A_2 表示用户名的第一个字母段(如 `icemoon`)，A_3 表示用户名的第一个数字段(如 12)；(4) E 表示邮箱前缀信息的用法，E_1 表示邮箱前缀的全称(如 `loveu1314@example.com`中的`loveu1314`)，E_2 表示邮箱前缀的第一个字母段(如 `loveu`)，E_3 表示邮

[Φ]PII标签变量的数量以及它们具体的定义，都依赖于用于训练的PII的特性(比如美国的手机号码为10位，而中文为11位)和攻击者$\mathcal{A}$倾向选择的粒度(如$\mathcal{A}$有可能更偏好于使用4种类型的姓名用法，而不是我们上面的7种)。这里我们针对中文用户的情况给出了一个典型定义，可以很容易地调整它来适应其他用户群体。此外，我们还可以通过预先确定一些运行模式：中国模式，美国模式，德国模式等等，使$\mathcal{G}_I$具有普适性。这样，$\mathcal{G}_I$就无需特别定制，TarGuess-I算法的输入只需要攻击对象的PII以及一种运行模式。

箱前缀的第一个数字段(如1314)； (5)P 表示手机号码的用法， P_1 表示整个手机号， P_2 表示手机号的前三位数字， P_3 表示手机号的后四位数字；(6)I 表示中文身份证号的用法， I_1 表示身份证号后4位数字，I_2 表示身份证号前三位数字，I_3表示身份证号前6位数字[①]

3) Σ={95个可打印的ASCII字符，$Null$}是一个与$\mathcal{V}$不相交的有限集合并且包含$\mathcal{G}_I$中的所有终结符。

4) $\mathcal{R}$ 是形如$A \to \alpha$规则的有限集合，其中$A \in \mathcal{V}$ 并且 $\alpha \in \mathcal{V} \cup \Sigma$ (见图 3.7)。

一个概率上下文无关文法(PCFG)是一种特殊的上下文无关文法：对文法规则中一个具体的左侧变量(如 L_4)，所有由该变量导出的规则(如 $L_4 \to$ love 和 $L_4 \to$ Suny)的概率加和等于1[68]。上文提出的文法$\mathcal{G}_I$确实能够满足这一条件。因此，TarGuess-I的训练阶段可以自动派生出PCFG算法。可参考[68]了解更多背景信息。通过文法$\mathcal{G}_I$，在生成猜测的过程(见图 3.7)我们可以得到：(1)所有通过基本结构实例化得到的只包含L， D 和 S标签的终结符(如love@1314)及其概率；(2)所有包含PII标签的预终结符(亦称为中间猜测，如N_3B_5和$N_3$1234)及其概率。给定目标用户，最终的猜测集来自这些终结符，以及使用目标用户的PII实例化后的预终结符。

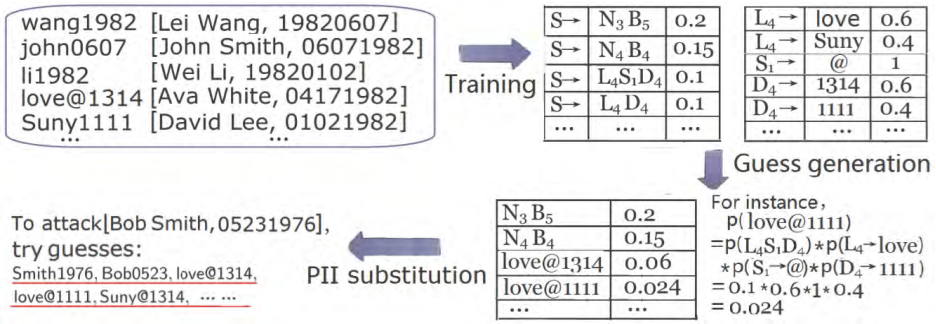


图 3.7 TarGuess-I示意图。

注意，为提高PII匹配的准确性，我们按照最长前缀(the longest prefix)的原则进行匹配，并且只考虑$len \geq 2$的PII字段。比如，如果john06071982与John Smith 的用户名“john0607”相匹配，那么根据最长前缀的规则它会被解析为A_1B_5，而不是N_3B_2。此外，我们在$B_1 \sim B_{10}$的定义中只考虑MMDD格式的日期，然而有不少用户更倾向于使用日期的缩写(如使用“198267”而不是“19820607”)。因此，在训练阶段匹配生日信息的时候，如果存在缩写的情况，那么这个字段的完整形式对应的标签计数+1；在口令生成的过程中，完整形式和缩写这两种情况都会被生成。比如，如果N_3B_2这个结构被用作生成猜测，那么“john06071982”和“john671982”都会被生成。

[①]我们对PII用法的定义，是通过在含有用户个人信息的126 口令集(即PII-126)上反复进行调整和训练而得到。为遵循基本的机器学习原则，该口令集在后文不再用作测试集。

我们基于类型的PII标签具有广泛的适用性。上文我们介绍了如何在PCFG中加入基于类型的PII标签来建立可识别语义的文法。实际上，这些标签还可以应用于许多其它的猜测模型(如Markov模型[34]和在节3.4.2提出的TarGuess-II)，以扩展为识别PII语义的猜测模型。比如，在建立可识别PII的基于Markov的模型时，我们只需要在n-阶Markov模型的字符集里Σ(如在[34]中Σ = {95 个可打印的ASCII字符})加入这些基于类型的PII标签{N_1, ⋯, N_7; B_1, ⋯, B_{10}; A_1, A_2, A_3; E_1, E_2, E_3; P_1, P_2; I_1, I_2, I_3}，然后所有针对这些基于类型的PII标签的操作，都与在Σ中最初的字符的操作相同。

$\mathcal{G}_I$具有很强的自适应性。一方面，当我们需要考虑新的语义用法(如网站名称)或者新的type-1 PII用法(如兴趣爱好)时，只需要简单地定义相应的基于类型的标签(如W表示网站名称的用法，H表示兴趣爱好的用法)，正如我们定义N和B标签一样。在训练和猜测集生成阶段，所有与H和W相关的操作都与N和B标签的操作类似，可以说是“即插即用”的。另一方面，即使训练集没有用户的生日信息，而TarGuess-I 定义了B标签，$\mathcal{G}_I$依然可以正常运作—它只是简单地把口令中的生日信息解析成D标签，而不会解析成B标签。换句话说，$\mathcal{G}_I$是“自卸载”的，所以上述情况中不需要特意去掉与B相关的标签。

一项独立的研究。在最近的一篇文献中(发表于2016年四月)，Li 等[103]提出了一种基于长度的PII字段匹配方法，并给出了一个基于type-1 PII的定向在线口令猜测模型Personal-PCFG。本章工作与他们的工作是独立的。

除了PCFG [68]中的L、D、S标签外，Li等还提出了六种PII标签：N表示姓名，B表示生日，E表示邮箱，A表示用户名，P表示电话号码，以及I表示身份证号。与我们提出的基于类型PII字段匹配的方法不同，[103]中的每一种PII标签都使用下标表示匹配的PII字段的长度len([103]中只考虑了$len \geq 3$)，本质上类似于Weir等PCFG模型[68]中的L、D 和S标签。因此，根据[103]，`wanglei@1982`会被解析为N段“wanglei”，S段“@”以及B段“1982”，而它的基本结构是$\mathsf{N}_7\mathsf{S}_1\mathsf{B}_4$；“`loveyou@1314`”会被解析为$\mathsf{L}_7\mathsf{S}_1\mathsf{D}_4$。在每个帐户允许100次猜测的情况下，他们提出的Personal-PCFG算法使用“完美的字典”，攻破了17%左右的测试集数据。

然而，我们发现了Personal-PCFG的一个严重缺陷。他们的模型与Weir等[68]的模型一样都使用了基于长度的标签：这样虽然可以区分不同长度的字段，但却对字段的子类型不敏感。比如，`john@1982`和`wang@1982`都会被解析为$\mathsf{N}_4\mathsf{S}_1\mathsf{B}_4$，因为“`wang`”和“`john`”长度都为4，尽管“`wang`”是一个姓氏，而“`john`”是一个姓名。如图 3.8所示，这不仅导致在训练阶段高估和低估一些PII 类型的使用频率，而且会导致在猜测集生成阶段产生不合理的情况。在训练阶段中，由于“`wang`”，“`smith`”和“`li`”都是姓氏，并且“`1982`”是用

户的出生年份，所以N_4B_4的概率应该是0.6而不是0.4，而L_2B_4的概率应该是0而不是0.2。在猜测集生成阶段，“lee”的长度是3，但是没有对应的N_3B_4 的结构，所以lee1977这个猜测的概率为0，因此不会被Personal-PCFG生成。

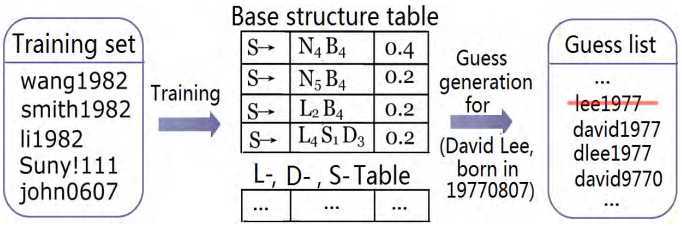


图 3.8 Personal-PCFG[103]存在的严重缺陷示例。

根据我们的文法$\mathcal{G}_I$，wang1982和li1982都会被解析成相同的基本结构N_3B_5；john1982会被解析成N_4B_5。因此，对于生于1977年的用户“David Lee”，TarGuess-I模型会根据基本结构N_3B_5生成lee1977 这个猜测。这样，就克服了文献[103]存在的问题。

评估。为实现公平对比，TarGuess-I和Personal-PCFG[103]一样，使用了 PII-12306 口令集，并且尽可能采用与他们的实验相同的设置(见图 3.9)。唯一的例外是，我们没有使用直接从测试集得到的“完美的(L-)字典”，因为这样不仅会产生过拟合的问题[34, 68]，而且在实践中也不现实。反而，我们所有的实验都是直接从12306训练集中学习得到L-字典，这样是口令攻击实践中受推荐的做法[34, 105]。如图3.9所示，在$10\sim10^3$ 次猜测下，我们的TarGuess-I的攻破率比Personal-PCFG[103]高出37.11%~73.33%；与其它三种漫步攻击模型[5, 34, 68]相比，TarGuess-I高出412%~740%。

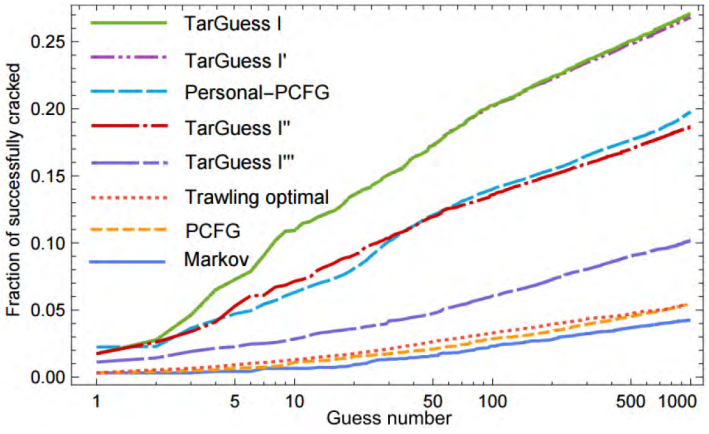


图 3.9 TarGuess-I模型(及其变体)与Personal-PCFG模型[103]的对比，训练集为50%的12306口令集，测试集为剩余的50%。TarGuess-I和Personal-PCFG两者都使用了12306口令集的六种type-1 PII，其中TarGuess-I′除去了电话号码和身份证号信息，TarGuess-IÆ进一步地除去邮箱和用户名信息，而TarGuess-IÆ′更进一步地除去生日信息。结果显示，TarGuess-I和I′的表现都十分突出。

我们还进一步测试了每种PII对TarGuess-I有何种程度的影响。如图 3.9所示，在给定用户的邮箱、用户名、姓名、生日、电话号码和身份证号等6类PII信息的条件下，我们的TarGuess-I能够在100次猜测以内，以平均20.26%的概率攻破用户的口令。在只知道姓名和生日的条件下，攻破率是13.61%；在只知道姓名的条件下，攻破率是6.04%。我们的结果表明，邮箱、用户名、姓名和生日对于在线攻击者而言是十分有价值的信息，而电话号码和身份证号只能使成功率略有提升。有趣的是，Personal-PCFG[103]比TarGuess-I′多使用了两种PII，而效果却更差，这表明简单在算法中加入更多的PII并不总是能得到更好的结果。

小结. 我们首次提出了基于类型的PII标签的概念，并在此基础上构建了一个可识别PII语义的PCFG模型，TarGuess-I。进一步，我们展示了利用这些基于类型的PII标签，将其它漫步猜测模型(如n阶Markov[34])扩展为定向攻击模型的可能性。在给定一个12306用户的邮箱、用户名、姓名和生日的条件下，TarGuess-I在10^2次猜测以内能以20.18%的成功率攻破用户的口令。在$10\sim10^3$次猜测以内，TarGuess-I可比Personal-PCFG[103]多攻破37.11%~73.33%的口令。特别地，TarGuess-I具有高度自适应性。

3.4.2 TarGuess-II

Targuess-II的目标是，在给定用户U在其它网站(如Dodonew)泄露的口令PW_s的条件下，在线猜测U在目标网站中(如CSDN)的帐户口令PW_x。这是一项十分具有挑战性的任务，原因有二。首先，在线猜测允许的猜测次数十分受限。其次，在用户通过修改PW_s来产生PW_x的时候，可以有许多变换的方法，如插入、删除、大写、字符跳变(如`password`→ `passw0rd`)和混合变换(如`password`→ `Passw0rd`)。这个过程受多种因素影响，如口令生成策略、网站的价值以及用户的创造力。而且，即使攻击者$\mathcal{A}$知道U 更倾向于插入三个数字，她也不知道U会具体插入哪些数字序列。如节3.3.2 所示，有相当比例的用户倾向使用流行口令而不是修改现有PW_s。Das等[7]2014年开展了关于跨站点定向猜测的研究，提出了一个基于U的一个姊妹口令来进行跨站点猜测的启发式算法。尽我们所知，他们的算法是当前唯一与本小节相关的工作，但由于我们在节3.1中指出的四个原因，这一算法并不是很有效。

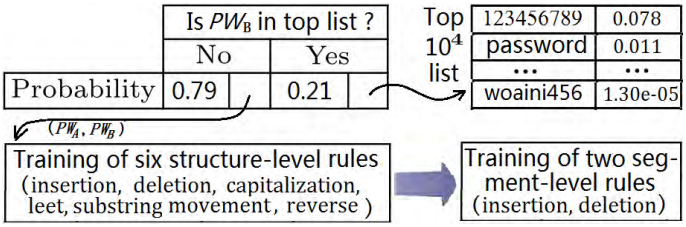


图 3.10 TarGuess-II的训练过程。

表 3.7 大写变换规则C的训练 (C_1: 整体大写； C_2: 首字母大写; C_3: 整体小写； C_4:首字母小写)

	No	C_1	C_2	C_3	C_4
Probability	0.95	0.01	0.03	0.003	0.007

表 3.8 字符跳变规则L的训练

	No	$L_1: a \leftrightarrow @$	$L_2: s \leftrightarrow \$$	$L_3: o \leftrightarrow 0$	$L_4: i \leftrightarrow 1$	$L_5: e \leftrightarrow 3$
Prob.	0.95	0.02	0.01	0.01	0.005	0.005

在本项工作中，我们采用数据驱动的方法。我们使用两个口令集A和B作为训练集，其中口令集A来自PW_s所在的网站或与其相似的网站，口令集B来自与目标网站(如CSDN)有着相同口令生成策略、服务类型和语言的网站。进一步，通过email将口令集A和B进行匹配，所有相同的口令对都忽略不计，这样就产生了一个由互异口令对$\{PW_A, PW_B\}$组成的列表。接着，我们判断PW_B是否只是一个流行口令，如果不是则计算PW_B 是如何由PW_A修改得到的。为判断PW_B是否是那些最为流行的口令，我们从大量泄漏的口令集中，综合考虑口令生成策略和语言，为目标网站(如CSDN)构建一个排名前10^4的流行口令列表$\mathcal{L}$。比如，令 $\mathcal{L}$=$\{pw \mid \text{len}(pw) \geq 8$ 并且$\text{Pr}_{csdn}(pw) * \text{Pr}_{126}(pw) * \text{Pr}_{dodonew}(pw)$的值排名前$10^4\}$。

如图3.10所示，在TarGuess-II的训练阶段中，第一步判断$PW_B \in \mathcal{L}$与否。如果是，那么$PW_B \in \mathcal{L}$的发生频次+1；否则，(PW_A, PW_B)首先通过结构级的训练，然后是字段级训练。在结构级训练的过程中，TarGuess-II首先使用L、D、S标签解析口令，正如TarGuess-I一样。比如，abc123解析为$\mathsf{L_3D_3}$。根据[7, 234]，我们主要考虑六种结构级变换规则：插入(如$\mathsf{L_3D_3} \rightarrow \mathsf{L_3D_3S_2}$)，删除(如$\mathsf{L_3D_3S_2} \rightarrow \mathsf{L_3D_3}$)，大写变换$C$，字符跳变$L$，子串移动$SM$ (如abc123→123abc)和翻转变换R(如abc123→321cba)。

字段级变换规则主要有两种：插入(如$\underline{\mathsf{L_3}:\text{abc}} \rightarrow \underline{\mathsf{L_4}:\text{abcd}}$)和删除(如$\underline{\mathsf{L_3}:\text{abc}} \rightarrow \underline{\mathsf{L_2}:\text{bc}}$)。对于这8种变换规则，每一种都有几个最基本的子类型。具体来说，对于前述两个层次的插入变换(见图3.11)，分别有尾部插入ti（ti'）以及首部插入hi（hi')；对于删除变换，分别有尾部删除td （td'）以及首部删除hd（hd')；对于大写变换，如表3.7所示有四种类型；对于字符跳变，如表5.4所示我们考虑五种子类型；对于翻转变换，如表3.10所示我们考虑两种子类型。注意，我们这里可以通过插入和删除变换的组合，可完成从abc123变换到abc!123，以此实现口令的中间插入变换。

结构级训练的第一步是使用文氏距离(LD)的算法(只针对插入和删除操作)去衡量PW_A 和PW_B的相似度d_1=LD(PW_A, PW_B)。然后，考虑到这些变换规则的流行度$C > L > SM > R$ [7, 234]，并且规则R会比SM带来更多的改变，我们对结构级的部分变换规则(除了插入和删除)C、L、R、SM，进行排序，并以此顺序尝试由PW_A变换得到PW'_A。对于每个规则，我们计算d_2=LD(PW'_A, PW_B)。如果$d_2 > d_1$，那么这条规则称为是“有效的”，然

表 3.9 子串移动规模SM的训练

	substring moved SM	
	Yes	No
Prob.	0.03	0.97

表 3.10 翻转规则R的训练(R_1: 翻转全部; R_2: 翻转每个段)

	No	R_1	R_2
Probability	0.97	0.02	0.01

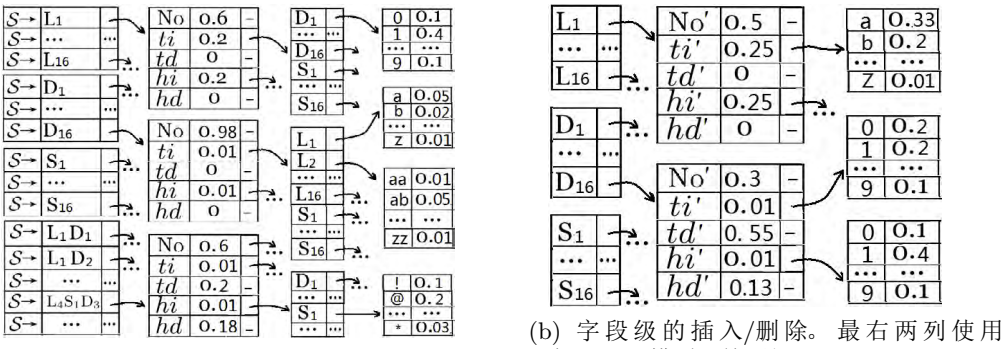
后PW_A 会更新为PW'_A，并且相应的规则(见表 3.7 到 3.10)出现频次+1。然后我们对PW'_A执行下一条规则来生成PW_AÆ，并计算d_3=LD(PW_AÆ, PW_B)。如果$d_3 > d_2$，那么这条规则是“有效的”并进行计数。如此反复，直到由PW_A变换得到PW_B。

通过这些有效的规则，假设PW_AÆ′由最初的PW_A变换而来。为避免无用的变换稀释有效变换的比例，我们规定，如果LD(PW_AÆ′, PW_B)小于预先定义的阈值(如在这项工作中建议为0.5)，那么此前PW_A经历的所有“有效的”规则都不作计数，然后开始训练下一个训练集中的口令对；否则，PW_AÆ′和PW_B都可用L、D 和S解析，如$L_4D_3S_2$和L_6S_1。由于我们在结构级训练中不考虑口令中字段的长度，所以$L_4D_3S_2$和L_6S_1会看作是LDS 和LS。我们使用文氏距离来计算d_3=LD(“LDS”, “LS”)的同时，文氏距离算法会返回一个编辑路径，记录从“LDS”转换为“LS”的过程：首先在S_2 段使用规则td，然后在D_3段使用规则td，最后在S_1段使用规则ti，生成基本结构为L_4S_1的PW_AÆÆ。相应地，所有相关规则的出现频次(见图3.11(a))都会进行更新。

现在进入了字段级的训练阶段，这一步的重点在于口令中L-、D- 或 S-字段的内部变换。比如，对于PW_AÆÆ(结构为L_4S_1) 和PW_B(L_6S_1)，我们使用文氏距离指标去测量它们L-段之间的相似度。与结构级训练类似，采用文氏距离作为判定标准来更新所有相关规则(见图3.11(b))的出现频次。在我们的实验中，我们发现图3.11(b)最右边两列的概率使用n-阶Markov模型[34]进行计算会比上述训练的效果要好，因为前者是在百万级规模的口令集训练得到的，而我们的训练集中互异口令对数量小得多，而这会产生数据稀疏性的问题。幸运的是，在百万级的口令集上训练得到的n-阶Markov模型可以很好地解决这个问题。

基于上述两个训练阶段，可产生一个基于口令重用的上下文无关文法$\mathcal{G}_{II} = (\mathcal{V}, \Sigma, \mathcal{S}, \mathcal{R})$，其中：

1) $\mathcal{V} = \{\mathcal{S}; \mathsf{L}, \mathsf{D}, \mathsf{S}; L, R, C, SM; ti, td, hi, hd; ti', td', hi', hd'\}$ 是一个变量的有限集合。
2) $\mathcal{S} \in \mathcal{V}$ 是起始符号。
3) Σ={95个可打印的ASCII字符； $C_1, \cdots, C_4$; R_1, R_2; $L_1, \cdots, L_5$; Yes, No; No′}是一个与$\mathcal{V}$不相交的有限集合。
4) $\mathcal{R}$是形如$A \to \alpha$规则的一个有限集合，其中 $A \in \mathcal{V}$并且$\alpha \in \mathcal{V} \cup \Sigma$(见图3.11和表3.7∼3.10)。



(a) 结构级的插入/删除

(b) 字段级的插入/删除。最右两列使用n阶Markov模型训练更好。

图 3.11　两种层次的插入和删除的训练过程。由于口令长度$len \leq 16$ [34]的口令的比例超过99%，所以我们只考虑$len \leq 16$的段。图3.11(a)中最右两列使用PCFG[68]在百万量级以上的口令集上训练效果更好。

注意，$\mathcal{G}_{II}$确实是一个概率上下文无关文法，因为对于$\mathcal{G}_{II}$中的任何一个左侧变量(如$R \rightarrow$)，所有由它出发的规则(如$R \rightarrow$No，$R \rightarrow R_1$和$R \rightarrow R_2$)的概率加和等于1。在猜测集生成阶段使用$\mathcal{G}_{II}$，我们可以生成一个带有概率的猜测集。比如，给定口令password，Pr(“Pa$$word123”)= Pr($\mathcal{S} \rightarrow \mathsf{L}_8$)* Pr($\mathsf{L}_8 \rightarrow ti$)* Pr($ti \rightarrow \mathsf{D}_3$)* Pr($\mathsf{D}_3 \rightarrow$123) $*P(C \rightarrow C_1)$ *Pr($L \rightarrow L_2$) *Pr($L \rightarrow L_2$) *Pr($R \rightarrow$No)*Pr($SM \rightarrow$No) =1 * 0.1 * 0.15 * 0.08 * 0.03 * 0.01 * 0.01 * 0.97 * 0.97 = 3.39 * 10^{-9}，其中，相关的概率可参考表3.7~3.10。然后，所有$\mathcal{G}_{II}$生成的猜测的概率都应乘上系数α。这个系数α表示不选择流行口令的用户的比例(如图 3.10中的0.79)。而在排名前10^4的流行列表中的口令的概率则乘上$1-\alpha$。最后，合并这两个带有概率的列表并对其进行降序排序，然后我们从中选择排名前k(如k=10^3)的猜测作为最终的猜测集。

图3.12显示了TarGuess-II与Das等[7]的模型的对比结果。由于这两个模型都使用了目标用户的相同的个人信息，所以它们是可比的。给定一个用户U在Dodonew使用的口令，在100次猜测以内，Das等[7]的算法有8.98% 的成功率能攻破U在CSDN的口令，而TarGuess-II有20.19%，效果提升了124.83%。在节3.5的一系列10个实验中，使用相同的个人信息，在100次猜测以内，TarGuess-II的攻破率比Das等算法[7]要高出8.12%~300% (平均 111.06%)。现在自然产生了一个新的问题：除了使用U的一个姊妹口令，如果$\mathcal{A}$再额外使用一些U的PII信息，攻击会不会更加有效？接下来，我们首次尝试回答这个问题。

3.4.3　TarGuess-III

TarGuess-III的目标是使用用户U的一个姊妹口令和一些PII在线猜测U的口令。这是十分现实的：如果攻击者$\mathcal{A}$想定向攻击U并且知道U的一个姊妹口令，那么$\mathcal{A}$也很有可能获得了U的一些PII(如邮箱、姓名)。据我们所知，目前尚没

有公开研究关注过这种攻击场景。这里我们主要考虑type-1 PII(如姓名和生日)，而type-2 PII(如性别和年龄)会在节3.4.4进行讨论。

由于允许的猜测次数受限，利用更多的信息通常也意味着更复杂的问题需要TarGuess-III考虑：到底哪些猜测的排序应提前，哪些猜测的排序应推后，依据是什么？这比设计TarGuess-II更具挑战。假设$\mathcal{A}$ 希望得到目标Alice Smith 在eBay 的帐户，eBay 要求口令的长度8^+，并且$\mathcal{A}$ 知道Alice是1978年出生的，Alice 的 Yahoo 帐户曾经泄露的一个口令为`Alice1978Yahoo`。那么对于`Alice1978eBay`、`Alice1978` 和`12345678`这三个猜测，$\mathcal{A}$应该先尝试哪一个来攻击eBay呢？如果Alice泄露的口令是`123456`，那$\mathcal{A}$的选择会改变吗？为了回答这些问题，我们需要一个自适应的、可识别PII的跨站猜测模型。

幸运的是，我们发现在TarGuess-II的$\mathcal{G}_{II}$文法中引入基于PII的标签(我们曾在TarGuess-I的$\mathcal{G}_{I}$文法中使用，见节3.4.1)，就可以构建一个可识别PII的基于口令重用的文法$\mathcal{G}_{III}$，就可以很好地解决上述挑战。具体来说，除了$\mathcal{G}_{II}$中的L、D、S标签，我们的$\mathcal{G}_{III}$文法进一步引入6大类型的PII标签。如同$\mathcal{G}_{I}$，我们在文法$\mathcal{G}_{II}$的变量集合$\mathcal{V}$中增加一系列基于类型的PII标签变量(如节3.4.1中所述的N_1 ~N_7和B_1~B_{10})。

在训练阶段，口令中的PII的字段都会被替换为相应的PII标签，而对于PII标签只适用$\mathcal{G}_{II}$中定义的6种结构级变换规则，而$\mathcal{G}_{III}$的所有其它部分与$\mathcal{G}_{II}$相同。在猜测集生成阶段，根据$\mathcal{G}_{III}$我们可以得到：(1)所有从只含有L、D和S标签的基本结构实例化得到的终结符(如`love@1314`)及其概率；(2)所有作为中间猜测的含有PII标签的预终结符(如$\mathsf{N}_4\mathsf{B}_5$和$\mathsf{N}_3$1314)及其概率。对于这些中间猜测，当给定目标用户后，我们可进一步使用目标用户的PII对其进行实例化。

如同$\mathcal{G}_{I}$，我们的$\mathcal{G}_{III}$也具有高度的适应性。原因与$\mathcal{G}_{I}$的情况相似(见节3.4.1)。这意味着对于一个新的语义标签，如W_1代表网站名的使用，可以很容易地加入到$\mathcal{G}_{III}$ 的PII标签集合中。这样，$\mathcal{G}_{III}$可以把`Alice1978Yahoo`解析为$\mathsf{N}_4\mathsf{B}_5\mathsf{W}_1$，并且`Alice1978eBay`会以最高的概率生成，因为从$\mathsf{N}_4\mathsf{B}_5\mathsf{W}_1$派生出`Alice1978eBay`的过程不涉及任何结构级或字段级变换规则。

如图3.12所示，即使用户U在CSDN的帐户没有完全重用他在Dodonew的口令，TarGuess-III仍然可以在100次猜测下，以23.48%的成功率攻破U的口令，这比TarGuess-II的结果高出16.3%。在那些使用TarGuess-III没能猜出的口令对中，超过80%的口令对都有较大的差异(LD相似度<0.5)。

3.4.4 TarGuess-IV

TarGuess-IV的目标是使用用户U的一个姊妹口令以及type-1和type-2 PII来在线猜测U的口令。相比TarGuess-III，TarGuess-IV模型中$\mathcal{A}$进一步知

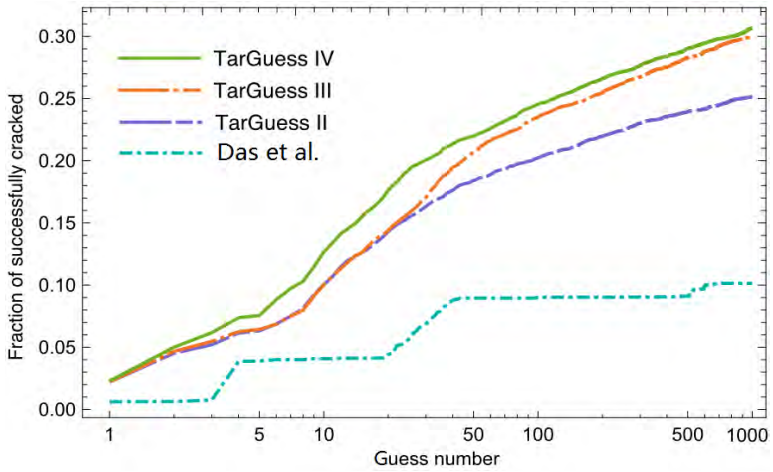


图 3.12 TarGuess II∼IV与Das等模型[7]的对比，在 66,573 对126 → CSDN的互异口令对集上训练，并在 30,8045 对Dodonew → CSDN的互异口令对集上测试。除了一个姊妹口令，TarGuess-III还利用了四种51job中的type-1 PII，而TarGuess-IV进一步利用了性别信息。

晓U的type-2 PII信息。设计TarGuess-IV的一个核心难点在于，type-2 PII信息(如性别)无法使用基于类型的PII标签，PCFG模型的L、D、S标签，或者Markov n-gram来直接进行匹配，进而也无法得到type-2 PII的使用频率。接下来，我们先证明一个定理，然后通过引入贝叶斯理论成功解决这一难题。

定理 2 *令事件pw表示用户U使用pw作为目标网站的口令，事件pw'表示U使用pw′作为她在另外一个网站的口令并且pw′被泄露，A_i $(i=1,2,\cdots,n)$ 表示用户的某一种PII属性，包括type-1和type-2的PII。假设$A_1,A_2,\cdots,A_n$ 是相互独立的，并且他们在事件pw'和(pw,pw')下也相互独立，*[①]*那么我们有*

$$\Pr(pw|pw',A_1,A_2,\cdots,A_n)=\frac{\prod_{i=1}^{n}\Pr(pw|pw',A_i)}{\Pr(pw|pw')^{n-1}},$$

证明　由于我们假设$A_1,\cdots,A_n$相互独立，因此可得$\Pr(A_1,\cdots,A_n)=\prod_{i=1}^{n}\Pr(A_i)$。又因为我们假设它们在事件$pw'$和$(pw,pw')$下也是相互独立的，因此$\Pr(A_1,\cdots,A_n|pw)=\prod_{i=1}^{n}\Pr(A_i|pw)$，$\Pr(A_1,\cdots,A_n|pw')=\prod_{i=1}^{n}\Pr(A_i|pw')$ 和 $\Pr(A_1,\cdots,A_n|pw,pw')=\prod_{i=1}^{n}\Pr(A_i|pw,pw')$。这样，我们可以得到：

[①]注意，定理2的假设并不与“$A_1,\cdots,A_n$ 与事件pw'和(pw,pw')是相关的”这一事实相矛盾。

$$\begin{aligned}\Pr(pw|pw', A_1, A_2, \cdots, A_n) =& \frac{\Pr(pw, A_1, A_2, \cdots, A_n|pw')}{\Pr(A_1, A_2, \cdots, A_n|pw')} \\ =& \frac{\Pr(A_1, A_2, \cdots, A_n|pw, pw') \cdot \Pr(pw|pw')}{\prod_{i=1}^{n} \Pr(A_i|pw')} \\ =& \frac{(\prod_{i=1}^{n} \Pr(A_i|pw, pw')) \cdot \Pr(pw|pw')}{\prod_{i=1}^{n} \Pr(A_i|pw')} \\ =& \frac{\prod_{i=1}^{n} (\Pr(A_i|pw, pw') \cdot \Pr(pw|pw'))}{(\prod_{i=1}^{n} \Pr(A_i|pw')) \cdot \Pr(pw|pw')^{n-1}} \\ =& \frac{\prod_{i=1}^{n} \frac{\Pr(pw, A_i|pw')}{\Pr(A_i|pw')}}{\Pr(pw|pw')^{n-1}} = \frac{\prod_{i=1}^{n} \Pr(pw|pw', A_i)}{\Pr(pw|pw')^{n-1}}. \end{aligned}$$ ■

上述定理表明，给定用户U的PII信息$A_1, A_2, \cdots, A_n$和在另一个网站曾用过的口令，预测U在目标网站的口令的问题可以通过“分而治之”的方法解决。具体来说，首先我们计算$\Pr(pw|pw')$，然后对每个A_i计算$\Pr(pw|pw', A_i)$，然后计算出最终的目标$\Pr(pw|pw', A_1, A_2, \cdots, A_n)$。幸运的是，$\Pr(pw|pw')$ 可以通过TarGuess-II计算得到，并且当A_i是一种type-1 PII时，$\Pr(pw|pw', A_i)$ 可以通过TarGuess-III获得。当考虑2^+种type-1 PII时(假设是$\{A_l, A_m, A_n\}$)，在定理2中，$\{A_l, A_m, A_n\}$可以先看作是一个整体的虚拟type-1 PII属性(如看作是A'_l)；在运行TarGuess-III 时，$\{A_l, A_m, A_n\}$仍各自分开来匹配、运算。剩下唯一的问题是，当A_i是一种type-2 PII时，如何计算$\Pr(pw|pw', A_i)$。

为解决这一问题，我们引入贝叶斯理论。不失一般性，我们假设A_k是一种type-2 PII。首先，

$$\begin{aligned}\Pr(pw, pw', A_k) =& \Pr(pw, A_k|pw') \cdot \Pr(pw') \\ =& \Pr[(A_k|pw)|pw'] \cdot \Pr(pw|pw') \cdot \Pr(pw').\end{aligned}$$

进一步，我们使用$\Pr(A_k|pw)$来近似$\Pr[(A_k|pw)|pw']$。这样，可得：

$$\begin{aligned}\Pr(pw|pw', A_k) =& \frac{\Pr(pw, A_k, pw')}{\Pr(pw', A_k)} \\ =& \frac{\Pr[(A_k|pw)|pw'] \cdot \Pr(pw|pw') \cdot \Pr(pw')}{\Pr(pw', A_k)} \\ \approx& \Pr(pw|pw') \cdot \frac{\Pr(A_k|pw) \cdot \Pr(pw')}{\Pr(pw', A_k)},\end{aligned} \tag{3.1}$$

其中，$\Pr(pw|pw')$在贝叶斯理论中称为先验概率，$\Pr(A_k|pw) \cdot \frac{\Pr(pw')}{\Pr(pw', A_k)}$这个系数表示$(pw', A_k)$对于事件$(pw', pw)$ 发生概率的影响。

现在，$\Pr(pw|pw', A_k)$可以彻底地计算：(1)$\Pr(pw|pw')$可以通过TarGuess-II计算得到；(2)$\Pr(pw', A_k) = c_1$是一个常数，因为当我们攻击U时，事件(pw', A_k)是一个已知和确定的事实。而生成猜测时只需要相对顺序，故也不需要知道c_1确切的值；(3)$\Pr(pw')$=c_2也是一个常数，因为当我们攻击U时，事件pw'是一个已知和确定的事实；(4)$\Pr(A_k|pw)$可以通过统计训练集中，有多少比例带有属性A_k的用户选择口令pw。当训练的数据量足够大(如>10^6)时，$\Pr(A_k|pw)$可以通过直接计数得到。否则，我们需要使用平滑技术(如Laplace和Good-Turing)来克服数据稀疏性的问题，以提高精度。

我们注意到，一些PII属性本质上是相互依赖的(如生日和年龄，姓名和性别)。幸运的是，由于大多数PII属性(见表3.1)都是相互独立的，所以定理2的实用性不会受到较大的影响，尤其是当TarGuess需要同时考虑较多的PII属性时。我们实验测试发现，即使TarGuess-III只使用了生日，而TarGuess-IV比它多使用一种PII(如年龄)，我们基于等式3.1仅调整那些不包含生日的猜测的概率，TarGuess-IV的效果仍然会比TarGuess-III好。

如图3.12所示，通过使用一个额外的type-2 PII(如性别)，在10~10^3次猜测以内，TarGuess-IV比TarGuess-III的攻破率提升 4.38%~18.19%。当分别允许10^2次猜测和10^3次猜测时，成功率达24.51% 和30.66%。这表明，type-2 PII对攻击者$\mathcal{A}$来说确实是一种很有价值的信息。据我们所知，这是迄今首次在口令猜测模型中引入type-2隐式PII.

3.4.5　解决其它攻击场景

如表3.2所示，从我们所关注的三大类型的个人信息，可产生7种(= C_3^1 + C_3^2 + C_3^3)组合，得到相应的7种攻击场景。这意味着，除了上文讨论过的4种最具有代表性的场景#1~#4，还剩下三种场景： #5 (type-2 PII)、#6 (type-1 PII + type-2 PII) 和#7 (1个姊妹口令 + type-2 PII)。除此之外，还有包含2^+个姊妹口令的场景：#8 (2^+姊妹口令)和#9 (2^+姊妹口令+ PII)。

当A_k是一种type-2 PII时，从等式3.1我们很自然地得到：

$$\Pr(pw|A_k) \approx \Pr(pw) \cdot \frac{\Pr(A_k|pw)}{\Pr(A_k)}, \tag{3.2}$$

其中，$\Pr(A_k) = c_3$是一个常量，而$\Pr(pw)$和$\Pr(A_k|pw)$都可以通过训练集统计得到，这就成功地解决了场景#5

为解决场景#6，我们需要推导一个新的公式。根据定理2，我们得到

$$\Pr(pw|A_1, A_2, \cdots, A_n) = \frac{\prod_{i=1}^{n} \Pr(pw|A_i)}{\Pr(pw)^{n-1}}, \tag{3.3}$$

其中，当A_i是一种type-1 PII时，$\Pr(pw|A_i)$可以通过TarGuess-I 得到；当A_i是一

表 3.11 9种算法在不同攻击场景下的训练集和测试集设置†

Experimental scenario*	Training set(s), with policy and language consistent with the test set								Test set (size; service)
	PCFG	Markov	Tra. opt.	Personal-PCFG	TarGuess-I	Das et al.	TarGuess-II‡	TarGuess-III, IV‡	
#1: 12306→Dodo	126	126	Dodo	PII-12306	PII-12306	126	12306, 126	12306, PII-126	49,775; Dodo
#2: 12306→CSDN	8^+Dodo	8^+Dodo	CSDN	8^+PII-Dodo	8^+PII-Dodo	8^+Dodo	12306, 8^+Dodo	12306, 8^+PII-Dodo	12,635; CSDN
#3: Dodo→CSDN	8^+Dodo	8^+Dodo	CSDN	8^+PII-Dodo	8^+PII-Dodo	8^+Dodo	Dodo, 8^+126	Dodo, 8^+PII-12306	5,997; CSDN
#4: Dodo→12306	Dodo	Dodo	CSDN	PII-Dodo	PII-Dodo	Dodo	Dodo, 126	PII-Dodo, 126	49,775; 12306
#5: CSDN→12306	Dodo	Dodo	12306	PII-Dodo	PII-Dodo	Dodo	CSDN, Dodo	CSDN, PII-Dodo	12,635; 12306
#6: CSDN→Dodo	126	126	Dodo	PII-12306	PII-12306	126	CSDN, 126	CSDN, PII-126	5,997; Dodo
#7: Rootkit→Yahoo	Rockyou	Rockyou	Yahoo	PII-Rootkit	PII-Rootkit	Rockyou	Rootkit, Xato	Rootkit, PII-Xato	214; Yahoo
#8: Rootkit→000web	L+D Rockyou	L+D Rockyou	000web	L+D PII-Rootkit	L+D PII-Rootkit	L+D Rockyou	Rootkit, L+D Xato	Rootkit, L+D PII-Xato	2,949; 000web
#9: 000web→Rootkit	Rockyou	Rockyou	Rootkit	PII-Xato	PII-Xato	Rockyou	000web, Xato	000web, PII-Xato	2,949; Rootkit
#10: Yahoo→Rootkit	Rockyou	Rockyou	Rootkit	PII-Xato	PII-Xato	Rockyou	Yahoo, Xato	Yahoo, PII-Xato	214; Rootkit

† Tra. opt. = Trawling optimal; Dodo=Dodonew; 000web=000webhost; 8^+=$len \geq 8$; PII-X=the PII-associated list X; L+D=Passwords with both letters and digits.

*$A \to B$ means that: (1) for the four password-reuse-based algorithms (i.e., Das et al.'s algorithm [7], TarGuess-II∼IV), a user U's password at service A can be used by $\mathcal{A}$ to help attack U's account at service B; and (2) for the other five algorithms, U's password at A is not involved, and only U's password at B is used as the target. Note that, every user's passwords in both A and B now have been associated with PII (see Tables 3.12 and 3.13) to facilitate the four PII-based algorithms.

‡When training TarGuess-II∼IV, U's one sister password comes from the 1st dataset, and $\mathcal{A}$ uses it to guess U's password from the 2nd dataset.

种type-2 PII时，可以使用式3.2得到。这样我们就解决了场景#6。

由于我们的定理2是对type-1和type-2 PII都适用，针对场景#7，我们可以首先使用TarGuess-II生成一个猜测集，然后根据式3.1调整猜测的概率。这样，场景#7也得到了妥善的解决。

对于场景#8和#9，它们不能很好地通过前述理解模型来解决。一种比较简单有效的现实(非理论上)解决办法是，重复使用我们的TarGuess算法，不过这并不是最优的解法。如何使用理论模型，来刻画这两个攻击场景是我们的下一步工作。不过，如节3.3.3所述，只有极少部分的用户曾经泄露过两个或以上的口令，因此，解决#8 和#9这两种场景的需求，远不如我们已经解决的前述7种定向猜测场景迫切。

小结。本节为定向在线猜测攻击设计了一系列逻辑严密的概率理论模型，以刻画7种不同现实能力的定向在线猜测攻击者。从理论上，我们的TarGuess-I 和II明显优于相关的模型(见[7, 103])，而TarGuess-III 和IV首次解决了“已知用户的PII和她在其它系统曾泄露的一个口令，如何加速在线猜测”这一难题。基于TarGuess-I∼IV，我们进一步展示如何解决剩余三种场景。下面我们通过大规模实验，进一步证实TarGuess-I∼IV的有效性。

3.5　实验

本节首先描述我们的实验设置，然后评估TarGuess-I∼IV，并将它们与其它五个主流的口令猜测模型/算法进行对比。

3.5.1　实验方法

在训练过程中，九个口令猜测算法中的其中三个(即Markov[34]、PCFG[68]和最优漫步算法Trawling optimal[5])只需要提供一个用作训练的口令集，另外两个(即Das等算法[7] 和 TarGuess-II∼IV)需要提供用户的姊妹口令对，还有四个(即Personal-PCFG[103]，TarGuess-I，III和IV)需要提供用户的PII。然而，原始的口令集只有两个(即 12306 和 Rootkit)带有PII(见表3.3)。因此，如节3.3.4所述，我们将1600万 Dodonew 口令集与 51job 和 12306 通过 email 匹配，得到161,510条含有PII的口令集，记为PII-Dodonew；类似地，将 000webhost 与Rootkit 通过email匹配，得到2,950条含有PII的口令集，记为PII-000webhost。通过将 000webhost 与 Rootkit 匹配，保证了*我们实验中拥有PII信息的英文用户口令集都是来自拥有较高安全意识的 Rootkit 黑客用户*。由于 Rockyou 口令集没有邮箱地址和用户名信息，故我们将 Xato 与 Rootkit 通过用户名进行匹配，得到15,304 条含有PII的 Xato 口令集(记为PII-Xato)，以实现对 Rockyou 口令集的补充。

表 3.12 匹配的中文口令对集基本信息

Original	PII-12306(129,303)		PII-Dodonew(161,510)	
dataset	Total	Non-identical(%)	Total	Non-identical(%)
Dodonew	49,775	14,380 (28.89%)	—	—
CSDN	12,635	5,538 (43.83%)	5,997	3,957(65.98%)

表 3.13 匹配的英文口令对集基本信息

Original	PII-Rootkit(69,330)		PII-000webhost(2,950)	
dataset	Total	Non-identical(%)	Total	Non-identical(%)
000webhost	2,949	2,510 (85.11%)	—	—
Yahoo	214	167 (79.04%)	96	90 (93.75%)

如表3.12所示，我们还将CSDN和Dodonew与PII-Dodonew和PII-12306 通过email匹配，生成三个带PII的中文用户口令对集。如口令集Dodonew↔PII-12306中，一共有49,775个口令对，其中14,380个口令对是互异的。类似地，我们也生成了三个外国用户的口令对集(见表 3.13)，但其中的Yahoo↔PII-000webhost口令对集只有96条数据，不适于实验，所以我们将它们去除。这总共产生了5个口令对集。

如表3.11所示，我们设计了10个实验场景以评估TarGuess I∼IV的有效性。为了让我们的攻击实验更接近实际情景，我们按以下这3条规则来为给定攻击目标(测试集)选择相应的训练集：(1)训练集和测试集来自不同网站；(2)训练集和测试集的语言和口令生成策略相同；(3)训练集要尽可能大。规则1防止了过拟合的情况，规则2和3保证了实验的合理性和算法的有效性。为了让实验中的对比更公平，全部9个待测算法都使用相同的测试集，而同类型的算法(如TarGuess-I和Li等算法[103]) 使用相同的个人信息数据。

3.5.2 实验结果

对于在线猜测攻击者来说，允许的猜测的次数是最稀缺资源，而网络带宽和计算能力一般不是瓶颈。比如，本节我们进行的全部10个实验中，训练过程在普通的个人电脑上均可以在65.3秒内完成，而对每个用户生成1000个猜测的时间只需要不到2.1秒。因此，我们采用“猜测数量-成功率”图来评估TarGuess-I∼IV，并将它们与现有的五个主流口令猜测算法(即PCFG [68]，Markov [34]，最优漫步算法[5]，Personal-PCFG[103]和Das的跨站攻击算法[7])进行了对比。

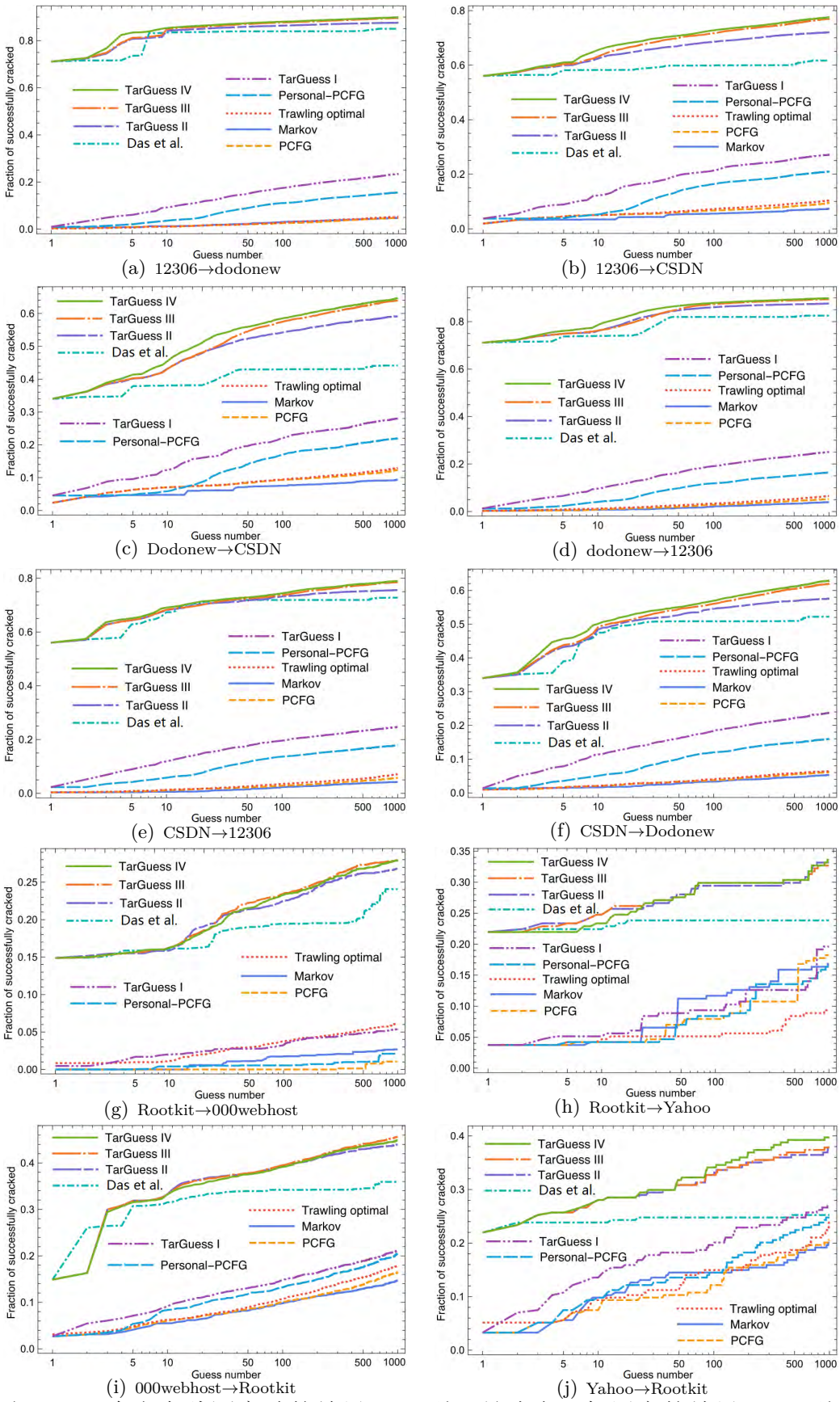
图3.13(a)∼3.13(j)表明，在$\mathcal{A}$拥有相同的个人信息和每个帐户猜测100次的情况下：(1)TarGuess-I的攻破率明显优于同类算法Personal-PCFG[103]，超出11.17%∼509%(平均 84%)；(2)TarGuess-II在针对互异口令对的攻击上表现优异，超出Das等算法[7]8.12%∼300%(平均 111.06%)；(3)TarGuess-III和IV在攻击普通中文用户时分别可达72.34%和73.09%的成功率，攻击具有较高安全意识的

英文用户也可达31.61%+。在绝大多数情况下，如果增加猜测次数，TarGuess-I~IV的领先优势会进一步扩大。如节3.3.1所述，英文口令集(即Rootkit及其匹配口令集)有大量PII信息缺失，否则我们的攻击效果会更好。

这里特别指出，TarGuess-IV首次刻画了一类十分强大而又现实的攻击者，他们能利用目标用户的一个姊妹口令以及type-1和2的PII实现跨站猜测。图3.13(a)~3.13(f)表明，在允许10次猜测的情况下，TarGuess-IV在针对不同网络服务的普通用户时取得了45.49%~85.33%的成功率(平均65.70%)；在允许100次猜测的情况下，成功率为56.96%~88.02%(平均73.09%)；在允许1000次猜测的情况下，成功率为62.95%~89.87%(平均77.32%)。若想使用漫步离线猜测算法[34, 197]达到这么高的成功率，需要对每个用户进行10^{13}次以上猜测，即使使用高性能的计算机本地运行也需要数天。

我们发现，定向猜测和漫步猜测算法生成的口令猜测(Password guesses)的攻击效率服从*Zipf*分布。排名靠前的猜测会有很高的成功率，比如在前10次猜测中，TarGuess-I在所有中文网站的实验都有超过9.25%的成功率。然而随着猜测次数的增加，每次猜测的成功率迅速下降。有趣的是，我们大多数实验都显示，对于8个可实际应用的算法(除了最优漫步算法[5])中任意一个算法，其生成的口令猜测可成功攻破的帐户数目f_n与该帐户已进行的猜测次数n之间关系大致满足Zipf 定律(见第2.4.3章)：$f_n = C * n^s$，其中$s \in [0.15, 0.30]$和$C \in [0.001, 0.01]$都是由测试集决定的常数。如果将f_n和n在对数- 对数坐标系中绘图，得到的将是一条直线(见图3.14，基于实验12306→Dodonew绘制)。图3.14中的3簇平行线与3类猜测算法相对应：漫步算法，基于PII的定向猜测算法，基于姊妹口令的定向猜测算法。这种成功率多项式时间递减的现象意味着：攻击者$\mathcal{A}$会在某个时间点停止攻击，超过该点时的每次攻击成功率过小而不足以弥补发动攻击的成本。根据3类不同的猜测策略，会有3 个停止攻击的时间点，然而现有的口令安全指南[38, 190] 均只考虑了漫步攻击的情况。

当$\mathcal{A}$拥有用户的一个姊妹口令时，TarGuess-IV在100次猜测以内针对普通用户的帐户，可以达到73%的成功率；即使没有姊妹口令，TarGuess-I也可以在100次猜测内达到20%的成功率，10^3次猜测时达到25%，而10^6次猜测时达到50%。这表明大部分普通用户的口令只需要少量的定向在线猜测尝试就可以猜中(如NIST[37, 38]建议为100)，这否定了NIST在2016年声称的在线猜测攻击“可以轻易地通过限制用户登录的频率来防范(can be readily addressed by throttling the rate of login attempts permitted)”。由于普通用户的口令强度尚不足以抵御在线定向猜测，而且距$[10^6, 10^{14}]$的“在线-离线鸿沟”[190]还有很长一段距离，所以那些关注于抵御离线攻击的努力(如[116, 238]关注10^{10}或更高的猜测次数)本可以发挥更好的作用。网站应该主要侧重于抵御在线攻击并保护好



(a) 12306→dodonew (b) 12306→CSDN

(c) Dodonew→CSDN (d) dodonew→12306

(e) CSDN→12306 (f) CSDN→Dodonew

(g) Rootkit→000webhost (h) Rootkit→Yahoo

(i) 000webhost→Rootkit (j) Yahoo→Rootkit

图 3.13 10个定向猜测实验的结果。(a)到(f)是攻击一般用户的结果，(g) 到(j)是攻击具有较高安全意识用户的结果。结果显示了TarGuess-I∼IV的有效性。

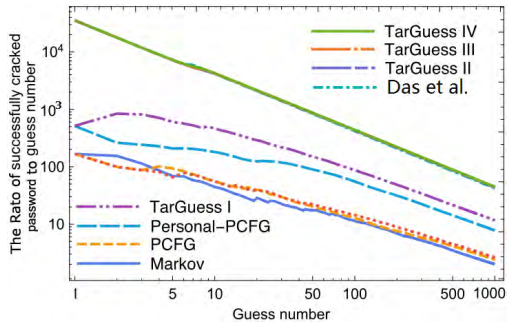


图 3.14　8个猜测算法产生的口令猜测的攻击效率服从Zipf定律。

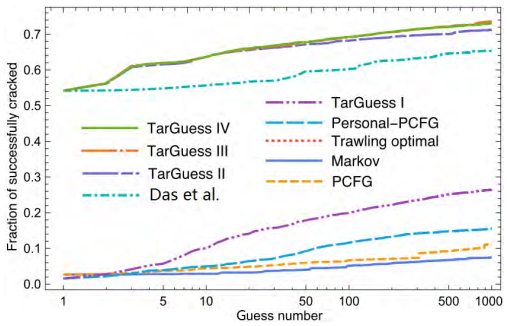


图 3.15　进一步实验：12306→Xiaomi。结果与图3.13(a)~3.13(f)的数据一致。

口令的hash文件，而用户的主要关注点应在于尽力避免在节3.3提及的三种不良口令习惯，以抵御在线猜测攻击。总的来说，定向在线口令猜测是一种危害程度远超学术界(见文献 [116, 121, 238])预期的现实威胁，比漫步猜测的危害要大得多。本章的模型可以用于更好地评估现有和未来的口令安全策略(如[73, 116, 121])，也可以离线猜测的方式运行来协助电子取证。

3.5.3　进一步实验

上面的10个实验显示了我们提出的4个模型的有效性，但实验中测试集都是明文口令。一个很自然的问题出现了：我们提出的4个模型在攻击已知信息很少的*真实账户*时仍然有效吗？为证实这一点，我们来进一步猜测Xiaomi口令集的口令。Xiaomi口令集来自于世界知名的手机制造商Xiaomi公司，其中的口令都是加盐后MD5哈希过的*非明文口令*。

我们攻击Xiaomi口令集的实验设置，与表3.11所示的场景#2中猜测Dodonew口令集的实验相同。测试集包含5,284条哈希过的口令，来自828万的Xiaomi口令集通过email与12306口令集匹配的结果。图3.15的实验结果表明，在10~10^3次猜测时，TarGuess-I比Personal-PCFG[103]的攻击效率高出70.58%~119%；TarGuess-II比Das的算法[7]高出73.66%~ 405%；TarGuess-III和IV的成功率为63.61%~73.56%。这一结果与节3.5.2中的10个实验的结果，尤其是攻击中文用户的结果，十分接近。这显示了本文模型的通用性。

3.6　本章小结

本章中，我们首次系统性地评估了定向在线猜测攻击者能够从用户的个人信息和泄露的口令中获得多大程度的攻击优势。我们提出了一个攻击框架，该框架包含7个严格的数学模型以刻画7种不同的定向在线猜测攻击场景。特别指出的是，TarGuess-I~IV 分别刻画了四种最为典型的攻击场景，较为妥善的解

决了“如何设计可识别上下文环境和PII语义的跨站口令攻击模型”这一难题。

大规模实证实验表明，TarGuess-I和II的效率显著优于其它同类算法，TarGuess-III 和IV在100次猜测以内针对普通用户可达到72%+的成功率，而针对具有较高安全意识的用户可达到32%的成功率。我们的结果说明现有的安全机制很难防范定向在线猜测攻击，这一攻击的危胁程度要远比预期中大。我们相信，本章这些新模型以及对定向猜测的新认识，会对现有的口令安全策略和未来的口令安全研究具有重要启发意义。

第四章　Honeywords攻击和设计理论体系

本章研究“如何评估和设计honeywords，以实现对口令文件泄露的及时检测”这一现实问题。为此，我们首先解决“如何最优地攻击honeywords”这一核心难点，首次提出一系列坚实的理论攻击模型，每种模型都基于攻击者不同的现实能力。然后，基于第二章中的口令Zipf分布理论和一系列口令猜测概率模型(例如Markov[34]、PCFG[68]和第三章中提出的TarGuess)，我们提出了4种新颖而有效的honeywords生成方法。本章将为第八章奠定理论基础。

4.1　研究背景

文本口令几十年来一直是应用最为广泛的身份认证方法，并且在可预见的未来仍无可替代[11, 16, 239]。但是，基于口令的身份认证系统长期面临一个固有的缺陷：服务器需要维护一个保存所有用户口令的文件，而这个文件对于外部攻击者和内部恶意人员来说都是有重要价值的攻击目标。当前，著名网站泄漏百万级的用户口令已经不再是新闻了，仅2016年以来遭遇泄露的就有Yahoo[127]、MySpace[240]、LinkedIn、Dropbox[29]、 Tumblr、VK.com和Last.fr[241]。

最令人不安的是，这些口令文件泄露均未被及时发现，当发现时往往距离初的泄漏时间已经过了数月，甚至数年之久。并且绝大多数情况是，攻击者已经充分利用了数据，然后将数据在网络上叫卖或公开，而网站此时才发觉口令文件泄露。比如，2016年10月，全球最大的免费建站空间网站Weebly要求用户更改口令，而事实上4300 万用户口令数据在八个月前已经泄漏；2016年9月，Yahoo被发现泄漏5亿用户口令，而泄漏被发现的原因是，攻击者在暗网上以1800美元的价格出售部分数据(约2亿条)，而泄漏发生的时间早在两年前，即2014年[242]；2016年6 月，Dropbox发现黑客公开了其6800万用户数据，然后要求用户更改口令，而事实上泄露发生在四年前[29]；2016年5月，Myspace被发现其3.6亿用户口令数据在黑市“Real Deal”上出售，而事后调查指出该数据泄漏其实发生在2008年[240]，历经八年时间网站才知晓，用户才被通知更新口令。

需要指出的是，虽然一些网站会采用推荐的方法将口令加盐哈希存储，但是攻击者可以利用基于机器学习的口令猜测算法(如[34, 69])和常见硬件如GPU(见[35, 243])来加速猜测，在可接受时间内恢复出相当大比例的口令(见[36]和节3.3.1我们对Rootkit的攻击结果)。如果网站采用的是通用的Hash函数(如MD5和SHA-1)，这对攻击者而言基本不构成障碍[35]。即使一些网站采用了专用的口令hash函数(如bcrypt和PBKDF2)，这仅仅只是在一定程度

上减慢攻击者的猜测速度，因为这类哈希函数会对攻击者和认证服务器增加同样倍数的时间成本，这个成本不能过高以致于影响用户体验。但是，攻击者很有可能会配备专用于口令攻击的硬件[244]，来大大降低这个成本。因此，一旦攻击者获得了服务器存储的口令哈希文件，就应当假设攻击者可以通过离线猜测攻击恢复出所有明文口令，而不应心存侥幸。另外，由于攻击者只需要进行本地离线猜测，那些专用口令哈希函数无助于检测口令文件泄漏。

对口令文件泄露的后知后觉不仅会给涉事网站带来直接的灾难性的损失(如资产和信誉)，而且由于用户常常在多个网站中重用口令，间接导致很多未发生泄露的网站也面临严重安全威胁。因此，设计口令文件泄漏检测方法，以实现对泄漏事件的及时感知和响应，是一项具有重要现实意义的研究课题。

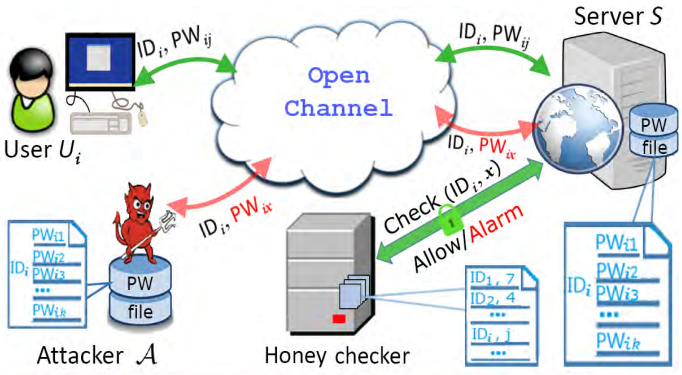
近来，学者们提出了一些方案试图彻底消除离线口令猜测：(1)使用与硬件设备相关的函数对口令进行运算后存储，如ErsatzPasswords[245]; (2)采用分布式密码技术，如门限口令认证秘密共享方案[246]。然而，这两种方案都需要对服务器端的认证系统进行大幅度的改变。另外，前者不支持以分布式的方式备份口令文件，且可扩展性差，所以不适于大规模的服务系统；后者需要改变客户端系统，这往往是难以接受的。

另一个更有前景的方案由Juels和Rivest在2013年提出，如图4.1所示，该方案对每个用户的帐户都关联一些诱饵口令(称为“honeywords”)[191]。其核心思想是，即使攻击者$\mathcal{A}$窃取了口令文件，并成功恢复出口令的明文，$\mathcal{A}$仍必须从一组故意生成的honeywords中找到用户的真实口令。当honeywords生成算法足够好做到以假乱真时，$\mathcal{A}$为了分辨出真实口令，唯一可做的是把服务器当作认证预言机，进行在线登录尝试。这样，进行在线登录尝试不仅会显著的延缓攻击者的猜测速率[19]，而且当$\mathcal{A}$使用honeywords登录时，服务器可以觉察到口令文件的泄漏。Juels-Rivest方案只需要对现有的服务器端系统做少量的修改，完全不涉及客户端系统，因此极具可行性。

4.1.1 动机和挑战

如文献[191, 247, 248]所指出的，honeywords方案的一个核心问题是*如何生成有效的*honeywords，*使其难以和真实口令相区分*。Juels-Rivest[191]提出了5种honeywords生成方法：4种适用于传统用户界面，1种适用于修改的用户界面。由于传统用户界面方案不需要改变用户使用习惯，因此更具可用性，故本文只关注传统用户界面的4种honeyword方法。从本质上来看，Juels-Rivest的4种方法都是基于字符随机替换的思想，因此天然无法抵抗可感知语义的攻击者，如文献[248–250]中给出的一些典型反例：`bound007` 和 `john1981`。

关于Juels-Rivest方法的有效性，原文[191]仅给出了一些启发式分析，文



Info of U_i	ID 4ever@foo.com;PW ******;Name john smith;Birthday 01171981
CCS'13[191] (k=14)	gracie52 tiger01 qwerty65 prince54 tiger81 gracie98 prince15 prince71 qwerty93 tiger66 qwerty10 gracie39 qwerty16 prince78
[this work] (k=14)	jsmith117 prince00 loveu123 qwertyu js128821 bond007 a123456 trustno1 rcv_11n1nj tiger81 letmein newy0rk 1989y2002r jan1981

图 4.1 采用honeywords技术的口令认证系统。为便于理解，这里假设口令都是明文存储的，但实际上是加盐哈希存储的。图的底部显示的是，受害者U_i的一些个人信息，以及由该用户真实口令“tiger81”生成的两组各19(=k-1)个honeywords。这两组honeywords分别由文献[191]两个启发式方法生成：tweaking-tail 和 modelling-syntax。

献[248–250]也只给出了一些零散的反例，目前既无严格的理论分析，也未曾在真实数据集上进行实证检验。本章首先通过实证实验显示，Juels-Rivest提出的4种启发式honeywords生成方法都无法提供预期的安全性，真实口令可以非常高的成功率被1次猜测出来。文献[248]提出了一个名为“honeyindex”的改进方法，但是honeyindex的设计和评估方法仍然是启发式的，honeyindex仍存在严重的安全性和可用性问题(见节4.3.4)。总之，*现有的honeywords生成方法(见[191, 248])均缺乏系统性的评估，无法达到所声称的安全性。*

存在上述问题的根本原因在于，*当给定honeywords生成算法时，攻击者如何最优地分辨真假口令仍有待解决。*在解决该难题之前，很难有效地评估(无论实验实证还是理论分析)一个honeywords生成算法的优劣。因此，在我们锻造和评估“盾”之前，必须先充分理解“剑”；只有经历最优的“剑”的测试，我们才知道“盾”可提供多大强度的抵抗力。对于一个honeywords攻击者，它可拥有哪些现实能力，又受哪些主要限制，通过这些能力，可以造成什么样的后果，有什么指标可以测量这些后果？本章首次对这些问题进行系统性的研究。

此外，*在定向猜测攻击下，一个honeywords生成方法可提供多大程度的安全性目前尚没有相关研究。*如本文第三章所指出的，定向猜测攻击者不仅可以利用用户选择流行口令的行为，而且可以利用受害者的个人身份信息(PII)。据我们所知，现有的研究工作[191, 248–250]主要关注的是无法识别用户PII的漫步猜测攻击者(见[5, 34, 68, 197])。然而，一方面如节4.2.1所示，现实中有相当大比例用户会使用自己的PII来构造口令—普通用户为38.33%~51.43%，黑客用

户为8.88%~15.97%；另一方面，用户的PII通常可以很容易的从社交网络[231]，或是从不断发生的大规模数据泄漏中获得[229, 251–253]。比如，发生在2016年9 月的Yahoo大规模口令泄漏事件，有5亿用户数据泄漏，信息包括姓名、电子邮箱、电话号码、生日等等[254]；2016年4月，土耳其的个人信息泄漏事件中有64%的总人口[227]受到影响；根据CNNIC 2015年的报告，中国6.68亿网民中有近80%的个人信息遭遇过泄漏[255]。鉴于定向口令猜测威胁的严重性和现实性，在该威胁下评估honeywords生成方法的有效性十分必要。

4.1.2 本章贡献

本章的主要贡献如下：

- **现有honeywords方法实验分析**。我们首次在大规模真实口令集上进行了一系列实验，发现Juels-Rivest的4种honeywords生成算法都无法实现预期的安全性。结果显示，真实口令可以以35.7%的概率被1次猜测区分出来，而不是预期的5%(如文献[191]中建议的，当每个帐户关联19个honeywords时)。定向攻击者1次猜测更是可以达到49.8%的成功率，这回答了[191] 中遗留的公开问题："对于给定的honeywords生成算法，定向攻击者分辨用户口令的能力如何?"遗憾的是，Juels-Rivest方法的这些缺陷是启发式方法固有的缺陷，难以简单修补，需要全新的设计方法。
- **新的honeywords攻击理论**。为刻画攻击者的最优策略，我们首次提出一系列honeywords攻击理论模型，每种模型都基于不同的攻击者能力。特别地，我们首次考虑了已知用户个人信息和已知用户注册顺序的攻击者，这些攻击者都是非常现实的攻击者。这些理论模型使我们可以利用真实数据集，来有效地评估honeywords生成算法的安全性(平坦性)，这回答了[191]中遗留的公开问题："是否有好的实验方法来测量honeywords生成算法的平坦性?"
- **新的honeywords生成方法**。基于一系列口令猜测概率模型(例如Markov[34]、PCFG[68]和第三章中提出的TarGuess)，我们提出了4种新颖而有效的honeywords生成方法。Juels和Rivest[191]简要提到了使用口令概率模型来构建honeywords生成方法的可能性，但他们明确的将其遗留为一个公开问题："潜在的口令猜测模型(如[68])是否可容易的加以利用?"本章将通过一系列探索性实验来显示，如何使用这些口令概率模型并不是显然的，而是需要一系列重大、新颖、创造性的努力。我们的honeywords设计方法不仅解决了Juels-Rivest提出的公开问题，而且为如何改造口令猜测模型来生成平坦的honeywords提供了指南，使得未来关于口令猜测模型的任何改进都可以很容易的应用到honeywords系统中。

- **大规模实验评估**。我们构建了基于所提方法的honeywords原型系统，并且在最强攻击者模型下，进行了大规模实验，证实了所提方法的有效性。本章的实验基于11个大型真实口令集，包含1.1277亿条口令，涵盖了各种流行的互联网服务，并且该实验是首次对honeywords进行实证评估。另外，为评估所提方法是否能抵御具有语义感知能力的人类攻击者，本章对11位受过充分训练的honeywords专家进行了用户调查，这是为此目的而进行的首次用户调查。结果表明，我们的honeywords方法可以抵御自动化的机器攻击者和手工的专家攻击者。

4.2　数据集、安全模型和评价指标

4.2.1　数据集

本章使用了11个大规模真实口令集(见表4.1)，它们来自6个英文和5 个中文网站。除了已在文献[86, 120, 256]中广泛使用的一些早期泄漏的7个口令集外，本章还使用了4个近两年内泄漏的口令集，这些新近泄漏的口令集可能会揭示最新的用户口令行为。这些口令集都是由黑客或内部人员泄漏，并且现在或曾经在互联网上可公开下载。由于经典的Rockyou口令集只含口令，无Email和用户名信息，所以在评估定向猜测威胁时，本章使用Xato来替代Rockyou。

其中，3个口令集以MD5哈希方式泄漏。为此，我们使用了多种漫步猜测模型[34, 197]和定向猜测模型TarGuess(见第三章)，在普通的带GPU的PC上破解了一个星期，成功恢复出了绝大多数明文口令。具体来说，Mango 包含1,113,638个口令，我们恢复了其中96.51%的口令明文；Rootkit包含71,228个口令，恢复了97.46%；QNB 包含97,415个口令[257]，恢复了79.86%。特别地，QNB这一英文口令集泄漏于2016年4月，来自中东地区的卡塔尔国家银行[257]。据我们所知，这是在学术研究中首次使用真实的电子银行数据集。总体来说，本章使用的口令集包含1.1277亿个明文口令，覆盖了各种流行的互联网服务。每个数据集的用途将在必要时指定。

特别地，3个口令集(12306、Rootkit 和 QNB)和与其关联的PII如表4.2 所示。为便于更全面的实证分析，本章使用了2个辅助PII数据集(Hotel 和 51job)，通过匹配电子邮箱和NID来扩充每个口令集，获得更多含有PII的账户。我们注意到，很多含有PII的账户缺失了一些重要的PII，但这些缺失的信息可从辅助PII数据集中得到一定的补充。

表4.3展示了各个口令集最流行的前10个口令。除Mango和QNB两个较新的口令集外，大多数口令集的top-10口令信息已经在节3.3.2进行了分析，这里不再赘述。Mango来自中文电商网站，其top-10口令的情况与其它中文网站没有显

表 4.1 10个口令集的基本信息†

Dataset	Web service	Language	When leaked	Total PWs	With PII
Tianya	Social forum	Chinese	Dec., 2011	30,901,241	
Dodonew	E-commerce	Chinese	Dec., 2011	16,258,891	
CSDN	Programmer	Chinese	Dec., 2011	6,428,277	
Mango	E-commerce	Chinese	July, 2015	1,074,742	
12306	Train ticketing	Chinese	Dec., 2014	129,303	✓
Rockyou	Social forum	English	Dec., 2009	32,581,870	
000webhost	Web hosting	English	Oct., 2015	15,251,073	
Yahoo	Web portal	English	July, 2012	442,834	
Rootkit	Hacker forum	English	Feb., 2011	69,418	✓
QNB*	E-bank	English	April, 2016	77,799	✓
Xato	Synthesised	English	Feb., 2015	9,997,772	

†PW stands for password, PII for personally identified information.
*QNB passwords were leaked in hash and will be used for real targets.

表 4.2 关于PII数据集的基本信息

Dataset	Language	Items num	Types of PII useful for this work
Hotel	Chinese	20,051,426	Name, Gender, Birthday, Phone, NID*
51job	Chinese	2,327,571	Email, Name, Gender, Birthday, Phone
12306	Chinese	129,303	Email, User name, Name, Gender, Birthday, Phone, NID
QNB	English	77,799	Email, User name, Name, Gender, Birthday, Phone, NID
Rootkit	English	69,418	Email, User name, Name, Birthday

*NID stands for National identification number, e.g., social security number.

著差异。然而，QNB所有的top-10口令都由字母和数字混合组成，并且$len \geq 8$，这表明QNB 极可能执行了一个“1^+ letter, 1^+ number, $len \geq 8$”的策略。这从表4.4中得到了印证。据我们所知，这是迄今所有泄露的口令集中，口令设置要求最为严格的。纵使应用了如此严格的要求，并且QNB 本身是电子银行，用户依然倾向使用最简单的方法、最简单的字母数字串(如`abcd1234`，`qatar1234`)来满足要求。高达0.7%的用户选择最流行的10个口令，这意味着$\mathcal{A}$仅通过简单尝试这些口令，其成功率就会达到0.7%。

表4.4展示了口令的字符组成信息。最显著的是，大部分中文(用户的)口令是只由数字组成的，而同样比例的非中文口令是只由字母组成的。总而言之，大部分口令包含数字(见列“[0-9]”)，同时极少含有特殊字符。000webhost 中，97.98%的口令包含字母和数字，但是只有0.28%只包含数字或只包含字母，这是由于000webhost要求口令必须同时包含数字和字母(见`http://www.liangxin.name/?post=403`)。QNB 也有同样的字符组成要求。有趣的是，英文用户有相当一部分倾向于在口令最后加上“1”，见表4.4的列“[a-z]+1$”。

图4.2展示了不同网站之间共有口令的比例。一般的，当k>10时，任意两个网站共有口令的比例少于60%。与预期相同，两个有不同口令生成策略(即字符和长度要求)的网站，共有比例远低于有相同口令生成策略的网站(如，Yahoo和000webhost)；同时，不同语言(见Yahoo和Dodonew)的共有比例远低于相同语言。中东英语的口令集 QNB 与英美的口令集，共有比例很低，即使

表 4.3 各口令集最流行的前10个口令

rank	Dodonew	CSDN	12306	Mango	Yahoo	000webhost	Rootkit	Xato	QNB
1	123456	123456789	123456	123456	123456	abc123	123456	123456	abcd1234
2	a123456	12345678	a123456	111111	password	123456a	password	password	qwer1234
3	123456789	11111111	5201314	000000	welcome	12qw23we	rootkit	12345678	qatar123
4	111111	dearbook	123456a	123456789	ninja	123abc	111111	qwerty	1234qwer
5	5201314	00000000	111111	123123	abc123	a123456	12345678	123456789	asdf1234
6	123123	123123123	woaini1314	123	123456789	123qwe	qwerty	12345	a1234567
7	a321654	1234567890	123123	123321	12345678	secret666	123456789	1234	1234abcd
8	12345	88888888	000000	1234567	sunshine	YfDbUfNjH10305070	123123	111111	m1234567
9	000000	111111111	qq123456	5201314	princess	asd123	qwertyui	1234567	qatar2022
10	123456a	147258369	1qaz2wsx	12345678	qwerty	qwerty123	12345	dragon	qatar1234
Sum of top-10 (%)	3.28%	10.44%	1.28%	13.06%	1.01%	0.79%	3.94%	1.46%	0.70%

表 4.4 各个口令集的字符组成信息†

Dataset	^[a-z]+$	[a-z]	^[A-Z]+$	[A-Z]	^[A-Za-z]+$	[a-zA-Z]	^[0-9]+$	[0-9]	Symbol only	**With symbol**	^[a-zA-Z0-9]+$	^[a-z]+[0-9]+$	^[a-zA-Z]+[0-9]+$	**^[a-z]+1$**
Tianya	9.91%	34.63%	0.18%	2.96%	10.24%	35.66%	**63.77%**	89.49%	0.03%	1.92%	98.08%	14.71%	15.73%	0.12%
Dodonew	10.30%	66.32%	0.30%	3.67%	10.92%	69.05%	**30.76%**	88.52%	0.02%	1.67%	98.33%	43.50%	45.74%	1.40%
CSDN	11.64%	51.39%	0.47%	4.57%	12.35%	54.33%	**45.01%**	87.10%	0.03%	3.69%	96.31%	26.14%	28.45%	0.24%
Mango	11.07%	35.42%	0.29%	1.78%	11.53%	36.50%	**63.25%**	88.23%	0.03%	0.95%	99.05%	17.57%	18.47%	0.24%
12306	5.26%	72.52%	0.05%	1.00%	5.42%	72.94%	**27.03%**	94.56%	0.00%	0.13%	99.87%	50.85%	51.50%	0.93%
Rockyou	41.71%	**80.58%**	1.09%	10.67%	44.07%	83.89%	15.94%	54.04%	0.01%	1.10%	96.25%	27.70%	30.18%	4.55%
000webhost	0.04%	**98.04%**	0.00%	18.94%	**0.26%**	**99.57%**	**0.02%**	**98.41%**	0.01%	6.92%	93.08%	54.42%	60.95%	4.66%
Yahoo	33.09%	**92.83%**	0.40%	8.51%	34.64%	94.06%	5.89%	64.74%	0.00%	2.85%	97.15%	38.27%	41.85%	4.80%
Rootkit	41.60%	**84.64%**	0.48%	9.27%	43.84%	85.84%	13.88%	53.97%	0.08%	6.10%	93.90%	19.19%	21.55%	1.81%
QNB	0.25%	**97.27%**	0.00%	13.13%	**0.28%**	**99.79%**	**0.19%**	**99.69%**	0.00%	8.14%	91.86%	66.85%	**76.09%**	4.10%

†Note that the first row is written in regular expressions. For instance, ^[a-z]+$ means passwords that are composed of *only* lower-case letters; [a-z] means the passwords that include lower-case letters as a substring; ^[0-9]+[a-z]+$ means the passwords composed of digits, followed by lower-case letters.

表 4.5 用户口令中PII的使用频率†

Typical usages of personal information	PII-Dodonew (161,517)	PII-12306 (129,303)	PII-CSDN (77,216)	PII-Mango (4,404)	PII-Rootkit (69,330)	PII-Xato (15,304)	PII-000web-host(2,950)	PII-Yahoo (214)	PII-QNB (77,799)
Name (7 subtypes, e.g., `johnsmith`, `john`, `jsmith`)	23.82%	23.83%	16.71%	16.33%	4.32%	2.38%	4.68%	2.34%	9.89%
Birthday (10 subtypes, e.g., `01171981`, `1981`, `011981`)	16.37%	18.75%	19.16%	16.33%	1.57%	0.22%	2.78%	1.87%	7.07%
Email_prefix (3 subtypes, e.g., `moon123`, `moon`, `123`)	8.60%	6.61%	8.65%	5.79%	3.75%	1.72%	6.17%	3.27%	4.74%
User name (3 types, e.g., `loveu1314`, `loveu`, `1314`)	10.53%	10.12%	7.46%	5.11%	3.45%	4.87%	3.12%	1.40%	7.66%
Phone # (3 subtypes, e.g., `4153022671`, `415`, `2671`)	1.00%	0.89%	1.43%	0.82%	–	–	–	–	0.38%
NID (3 subtypes, e.g., `620915337`, `620`, `5337`)	0.39%	0.71%	0.12%	0.27%	–	–	–	–	0.32%
Total personal information usages (all above)	51.43%	50.71%	46.87%	38.33%	12.76%	9.10%	15.97%	8.88%	26.71%

†The specific sub-types of each PII we consider are the same with that of TarGuess-I (see Sec. 3.4.1). For instance, 23.82% in the top left corner means that 23.82% of the 161,517 PII-associated Dodonew users employ at least one of their 7 sub-types of name information to build passwords.

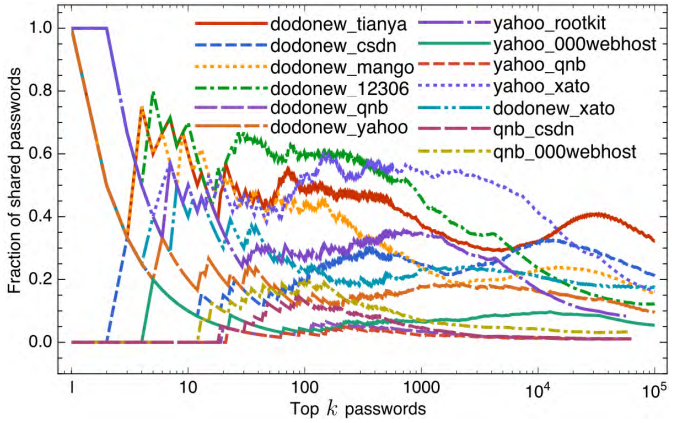


图 4.2 两个网站共有口令的比例。

两者执行相同的口令生成策略。比如QNB和000webhost都执行“1^+ letter, 1^+ number”的规则，并且86.4%的000webhost 口令和99.7%的QNB口令长度都不少于8，但是它们共有口令的比例总是低于20%。这表明文化对口令的分布有重要的影响。总体来说，本节显示网站服务类型、口令生成策略、语言和文化都对口令的分布有重要影响。

表4.5展示了9个关联了PII的口令集：前4个是通过将不含PII的4个中文口令集与12306、51job 和 Hotel 进行匹配得到；中间4个是将不含PII的4个欧美英文口令集与Rootkit匹配；最后1个是自身带PII的QNB。结果显示，用户倾向于使用他们自己的PII来构造口令。本章使用了第三章中提出的基于类型的PII 匹配方法来测量PII的使用情况，因为这种方法已被证实比其它方法(如[103])更准确。高达38.33%~51.43%的中文用户使用至少 1 种PII来构造口令，同时在欧美英文用户中这一比例是8.88%~15.97%，中东英文用户是26.71%。相比之下，含有PII的欧美英文用户在PII的使用上，有更安全的行为，这是因为，他们都是通过电子邮箱与Rootkit这个黑客论坛匹配得到的。换句话说，本章的含有*PII*的欧美英文用户都是黑客，他们很好的代表了安全意识强的用户。QNB用户比中文用户更少使用PII，一个合理的原因是 QNB 的账户是高价值的，所以相应的用户有更安全的行为[80]。虽然如此，仍然有超过四分之一的 QNB 用户使用PII构造口令。上述的结果展示了有大量的用户使用*PII*构造他们的口令，一个实用的*honeywords*生成方法应考虑这些用户行为。

我们还发现，虽然QNB的用户使用英语，但QNB口令与其它4个英文网站的口令差异较大。这符合预期，因为QNB的用户主要来自中东地区。因此，本章将用户分成三组：中文、欧美英文、中东英文。总之，本章数据集规模大，时间跨度长，服务类型多，用户群体较为全面，能很好的代表真实口令的分布。尽我们所知，这是迄今口令研究中使用的最大、最多元化的数据集之一。

表 4.6 本章考虑的四种攻击者的能力

Attacker types		PW file	Public info*	Personally identifiable info (e.g., name and birthday)	User registration order	Existing literature
Distinguishing attacker	$\mathcal{A}_1$	✓	✓			[191, 246, 248]
	$\mathcal{A}_2$	✓	✓	✓		None
	$\mathcal{A}_3$	✓	✓		✓	None
	$\mathcal{A}_4$	✓	✓	✓	✓	None

*Typical public information includes the various leaked password lists, password policy and all the cryptographic algorithms (e.g., password hash methods and the honeyword generation methods).

4.2.2　安全模型

本章主要关注的是honeywords系统自身的安全性，及其对背后的身份认证系统安全性的影响。不失一般性，我们考虑最一般的情形，即客户端-服务器架构。我们下面回顾honeywords系统，并讨论与之相关的主要攻击方式。

Honeywords系统。如图4.1所示，honeyword系统中总共有4个实体参与：一个用户U_i，一个认证服务器S，一个honeychecker，一个攻击者$\mathcal{A}$。用户U_i在服务器S上注册了账户及口令$(\mathrm{ID}_i, \mathrm{PW}_i)$，可能还提供一些PII(例如，Gmail注册时还需要姓名、生日、手机号和性别)。在服务器端，与传统口令认证不同的是，会执行一个命令$\mathsf{Gen}(k;\mathrm{PW}_i)$，来生成$k$-1个不同的、似真的诱饵口令(称为*honeywords*)，关联到用户U_i，文献[191]中建议k取20。口令PW_i和$k-1$个honeywords统称为k个*sweetwords*。一般地，有两种生成honeyword的方法：与真实口令相关的(例如，chaffing-by-tweaking[191])和与真实口令不相关的(例如，honeyindex[248]和本章的方法)。

现在，用户U_i在S中的记录可以被表示成$(\mathrm{ID}_i, \mathrm{SW}_i)$，其中$\mathrm{SW}_i = (\mathrm{sw}_{i,1}, \mathrm{sw}_{i,2}, \cdots, \mathrm{sw}_{i,k})$。$k$个sweetwords中只有一个与用户的真实口令相同，记为$\mathrm{sw}_{i,j}$。令C_i为用户U_i的真实口令在sweetword列表SW_i中的位置，因此$C_i = j$。S上的k个sweetwords应该被加盐哈希存储，加盐可以每个用户一个盐，也可以每个sweetword一个盐。记录(ID_i, C_i)存储在honeychecker 上，honeychecker是一个单独的、固化的、极简设计的计算机系统。这个系统可以被放置到不同的管理域中，运行不同的操作系统、软件堆栈、安全机制等，并确保分布式安全性[191, 247]。Honeychecker不可以公开访问，他只能与服务器S通过“专用、加密和经认证的”[191]信道交互。

当用户U_i使用$(\mathrm{ID}_i, \mathrm{PW}_i^*)$登录时，$S$首先检查$\mathrm{PW}_i^*$是否在$\mathrm{SW}_i$中。如果不在，那么拒绝登录。如果在，设其索引为$C_i^*$，$S$会提交一条命令$\mathsf{Check}(\mathrm{ID}_i, C_i^*)$到honeychecker。如果$C_i^*$=$C_i$，honeychecker指示$S$允许$U_i$登录。否则，honeychecker就检测到有人使用honeyword尝试登录，向S发出一个警报。根据预警策略，S会采取相应的措施，例如：1)在蜜罐系统上接受登录，并且更严格地监视登录者行为；2)如果使用honeyword的登录次数超过账户U_i预定义的阈

值$\mathcal{T}_1$(如$\mathcal{T}_1 = 3$)，锁定用户U_i，直到U_i更新现有口令；3) 如果所有用户的honeyword登录总次数超过预先定义的阈值$\mathcal{T}_2$，关闭计算机系统，并要求所有用户重置其口令。$\mathcal{T}_2$的值取决于系统的风险分析，这超出了本章考虑的范围。由于系统需要平衡honeyword区分(distinguishing)攻击者和DoS(Denial of Service)攻击者，因此$\mathcal{T}_2$不能太大也不能太小，不失一般性，本章令$\mathcal{T}_2$=10^4。

Honeyword 区分攻击者。如前所述，honeywords生成方法最核心的安全目标是，对给定用户U_i生成一组honeywords，使得它们难以和用户U_i的真实口令PW_i相区分。这个安全目标对应的是图4.1中的honeyword 区分攻击者(Distinguishing attacker)，该攻击者$\mathcal{A}$的目标是把服务器S看成查询预言机，试图将用户U_i 的真实口令和$k-1$个honeywords区分开来。$\mathcal{A}$ 使用honeyword登录时，会被honeychecker检测到。当U_i的honeyword 登录次数超过该用户的阈值$\mathcal{T}_1$(如3)时，$\mathcal{A}$ 导致U_i 的账户发出报警。如果$\mathcal{A}$ 的honeyword登录次数超过阈值$\mathcal{T}_2$(如10^4)，也会导致系统级的警报。因此，$\mathcal{A}$对单个用户和整个系统的honeyword登录次数都是极其受限的。

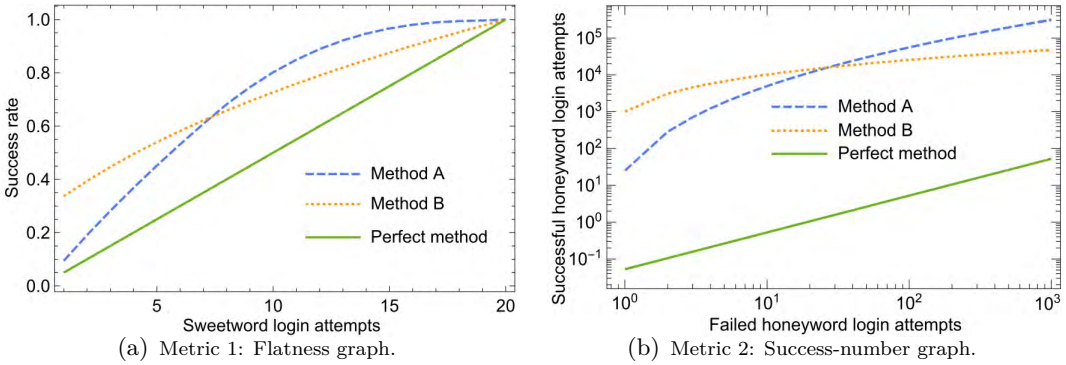
如表4.6所示，本章假设攻击者$\mathcal{A}$已经得到了服务器S存储口令的文件，还知道honeyword的生成方法和哈希方法，并且知道所有公开可得的信息(比如各种公开泄漏的口令集和目标网站的口令生成策略)，甚至于知道受害用户U_i的PII。这些关于$\mathcal{A}$的能力假设，都是非常现实的，但之前的研究(见[191, 247, 248, 250])通常只是作隐性假设，没有明确的指出。

本章还注意到，有时用户的注册顺序对$\mathcal{A}$是非常有用的。注册顺序这种信息通常是明确地存储在口令文件中(如Forbes、QNB和Tianya)，或者可以间接的由单调递增的用户注册号码推断出来。即使，这种信息不能从泄漏口令集中得到，它也可以从某些应用(如社交/程序员论坛、BBS讨论版)的用户主页上抓取到，或者可很大程度上根据用户首次参与评论、提问、回答的时间点来推测。这一能力对与真实口令无关的honeywords生成方法而言，是非常有用的。如节4.3.4中将指出的，已知该信息的$\mathcal{A}$可以很容易的攻破“honeyindex”方法。

其它攻击者。如[191]中指出的，honeywords所面临的现实威胁还包括攻击honeychecker系统、针对honeyword系统的DoS攻击、交叉攻击(当用户在不同系统上重用相同的口令时)。这些攻击方法与特定的honeywords生成方法无关，所以不是本章研究的范围。

4.2.3 评价指标

Juels-Rivest[191]提出了一个指标ϵ-flat来测量honeywords生成方法的安全性。ϵ-flat表示，在给定U_i的k个sweetwords，并且只允许对S提交 1 次在线猜测时，区分攻击者$\mathcal{A}$所能获得的最大成功率ϵ。然而，当$\mathcal{A}$可以对每个用户进行超过一



(a) Metric 1: Flatness graph.　(b) Metric 2: Success-number graph.

图 4.3 这两个指标测量的是，honeywords生成方法抵抗区分的能力。这里方法A和B都是概念性的例子，不是真实的攻击方法。

次的在线攻击时(例如$\mathcal{T}_1 > 1$时)，这个指标不足以评测一个方法的安全性。另外，ϵ-flat也不能反映一个方法生成的“低垂的果实”(即那些容易被攻击者分辨的honeywords)的脆弱性。

本章提出两种全新的指标：flatness图和success-number 图。这两个指标(见图4.3)分别测量了一个方法在平均和最差情况下，抵御区分攻击者的能力。

Flatness图测量的是分辨出真口令的概率，随每个用户sweetword登录次数的变化趋势。如图4.3(a)所示，曲线的点(x, y)表示真实口令被前x次猜测猜中的概率为y，其中$x \leq k$。实际上，还要求$x \leq \mathcal{T}_1$ (例如$\mathcal{T}_1 = 3$)。显然，flatness曲线上的点(x=1, y)，就对应了[191]提出的ϵ-flat指标，其中$\epsilon = y|_{x=1}$。Flatness图表现的是，在允许对每个用户尝试多次时，抵御区分攻击者的平均水平。

Success-number 图测量的是一个honeywords生成方法，会生成多大比例的“低垂的果实”。这个图描绘的是，成功登录(即，使用password登录)次数，随着失败登录(即，使用honeyword登录)次数的变化。图4.3(b)中曲线上点(x, y) 的意思是，在第x次honeyword登录尝试时，已经有y个真实口令被成功区分出来，其中$x \leq \mathcal{T}_2$(例如，$\mathcal{T}_2$=10^4)。一个 $\frac{1}{k}$-flat 的方法是理想的，因为它没有生成“低垂的果实”，k个 sweetwords是真实口令的概率都相等。

4.3　攻击现有方法

Juels-Rivest宣称他们的4种主要honeywords生成方法能达到这样的安全目标(见[191]中的表1)：当给定U_i的k个honeyowrds时，区分攻击者$\mathcal{A}$通过在线查询服务器S一次，成功率不会显著高于$\frac{1}{k}$。然而，基于前述的大规模真实口令集，本节揭示这些方法均未实现预期的安全性。

4.3.1 Juels-Rivest方法回顾

Juels-Rivest的4种主要方法(见[191]中的表1)，都是与真实口令直接相关的，即$k-1$个honeywords直接依赖于真实口令PW_i的结构/模式。

Tweaking by tail。他们的第一个方法是“修改”(“tweak”)真实口令PW_i中指定位置的字符，来生成$k-1$个honeywords。记t(例如，$t=2$或3) 为想要修改的位置数。Juels 和 Rivest[191] 主要讨论的是“tweaking by tail”，本章也主要关注该方法。PW_i的最后t个字符被随机地替换成同类字符：数字替换成数字，字母替换成字母，特殊字符替换成特殊字符。比如，如果PW_i是`trustno1`并且$t=3$，那么sweetword列表可以是`trustyp0`、`trustmi7`、`trustme5`等等。

Modeling syntax。他们的第二个方法受文献[258]启发，先将PW_i解析成由相同类型字符组成的字段(tokens)，加以标记，然后提取出结构(syntax)。比如，可从`trustno1`提取的结构为$\mathsf{L}_7\mathsf{D}_1$。当生成honeywords时，将结构中对应的标记随机替换成符合相应结构的字段。比如，结构$\mathsf{L}_7\mathsf{D}_1$可以生成以下honeywords：`dfdphus3`, `letmein4` 和 `kebrton9`。

Hybrid。他们的第三种方法是，结合tweaking-tail和modeling syntax两种方法，预期综合这两种方法的优点。具体来说，当给定U_i的口令PW_i，hybrid方法先使用modeling syntax方法生成a个种子sweetwords(包括PW_i)，然后对每个种子sweetword使用tweaking方法生成$b-1$个honeywords，这里$a\cdot b\geq k$。最终从这$a\cdot b-1$个honeywords中随机抽取$k-1$。推荐参数是$a\approx b\approx\sqrt{k}$。举个例子，假设$k=20$，$a=4$，$b=5$，那么从`trustno1`得到结构$\mathsf{L}_7\mathsf{D}_1$，先得到另外3个种子honeywords，`dfdphus3`、 `letmein4` 和 `kebrton9`，然后对4个种子sweetwords(包括口令`trustno1`)每个再生成4个新的honeywords，总计20个sweetwords。

Simple model。他们的第四种方法是使用真实口令集来辅助生成honeywords(见[191]的附录)。首先，混合大规模的真实口令集和8%的各种长度的随机字符串生成一个表L。然后，从L中随机抽取一个字符串，记为$w=w_1w_2\cdots w_d$，其长度为d。那么一个honeyword c，逐个字符记为$w_1c_2\cdots c_d$，用下述方法，j从2到d启发式的逐个生成对应字符：

- 以 10%的概率，从L中随机抽取一个字符串w，令 $c_j=w_j$；
- 以 40%的概率，从L随机抽取一个满足条件$c_{j-1}=w_{j-1}$的字符串，并且令$c_j=w_j$；
- 以 50%的概率，令 $c_j=w_j$。

需要指出的是，除了上述4种主要方法，Juels-Rivest还提出了一种基于传统用户界面的方法，称为“Chaffing with tough nuts”。它旨在生成一些超高强度的难以被恢复出明文的honeywords，用以增加$\mathcal{A}$离线攻击的计算量。它不

能单独使用，需要联合其它方法使用。然而，如[248]所指出的，$\mathcal{A}$是聪明的，会关注成本-收益，因此会简单的跳过这些“tough nuts”，避免攻击它们，这是因为一般用户很少使用“tough nuts”作为真实口令。从这方面来看，引入“tough nuts”反而实质上减少了有效honeywords的数量(因为每个用户总共只有k个honeywords)，实际上会降低安全性。因此，本章不考虑这个方法。

4.3.2　对Juels-Rivest方法的现实攻击

本节给出攻击Juels-Rivest方法的2个现实的honeywords区分策略，而每一策略下又有不同的参数设置，最终共产生9种具体的攻击方式。这两个攻击框架基于真实口令集，这与现有基于一些典型反例口令的启发式评估方法(见[191, 247, 248])有本质不同。为实现公平评估，本节考虑$\mathcal{A}_1$型攻击者，见表4.6。简言之，$\mathcal{A}_1$获得了口令的哈希文件，并且知道所有公开信息，但是不知道用户的PII和注册顺序。这样的攻击者能力假设与Juels-Rivest的工作[191] 和之前的一些研究工作[247, 248]相一致。本章的结果显示，即使对于这种最弱的攻击者，Juels-Rivest的方法也不能达到预期的安全性。

Top-PW attack。这是一个直观的攻击策略。区分攻击者$\mathcal{A}$拥有一个口令文件F，其中包含n个列表$\{\mathrm{SW}_1, \mathrm{SW}_2, \cdots, \mathrm{SW}_n\}$，并且每个列表包含$k$个不同的sweetwords。给定这样$n\cdot k$个sweetwords，在honeyword登录次数到达$\mathcal{T}_2$(例如，10^4) 之前，$\mathcal{A}$的目标是分辨出尽可能多的真实口令。注意，$\mathcal{A}$对每个账户最多只能进行$\mathcal{T}_1$(例如，1、3或10)次honeyword登录。$\mathcal{A}$可以简单的使用概率递减的顺序依次尝试$n\cdot k$ 个sweetwords，这里sweetword的概率可以直接来自于一个已知的概率分布$P_{\mathcal{D}}$。比如，来自一个泄漏口令集$\mathcal{D}$的经验分布：

1) $\forall \mathrm{sw}_{i,j} \in F$, 如果 $\mathrm{sw}_{i,j} \in \mathcal{D}$, 令$\Pr(\mathrm{sw}_{i,j})$=$P_{\mathcal{D}}(\mathrm{sw}_{i,j})$;

2) $\forall \mathrm{sw}_{i,j} \in F$, 如果$\mathrm{sw}_{i,j} \notin \mathcal{D}$, 令$\Pr(\mathrm{sw}_{i,j}) = 0$。

本章称$P_{\mathcal{D}}(\cdot)$为List口令模型，这里$\forall x \in \mathcal{D}$, $P_{\mathcal{D}}(x) = \frac{\mathtt{Count}(x)}{|\mathcal{D}|}$。本章实证的评估了这一攻击方法对[191]中4种honeyword 生成方法的攻击效果。更具体的，本章将Dodonew口令集随机的分成相等的两部分，Dodonew-tr和Dodonew-ts。使用[191]中每个方法，对Dodonew-ts中的每个口令，生成$k-1$个honeywords，同时Top-PW攻击中的$P_{\mathcal{D}}$来自于Dodonew-tr。如图4.4(a)∼4.4(d)，对[191]中4种方法的任意一种，Top-PW攻击可以从8,129,445个Dodonew-ts账户中至少分辨出615,664 个账户的真实口令。然而，根据Juels-Rivest的预期安全目标，攻击者应该只能分辨出Dodonew-ts中的526(=$\mathcal{T}_2/(k-1)$)个真实口令。也就是说，就success-number指标而言，他们的方法比宣称的至少弱1170倍。

本章还使用了Top-PW攻击策略去评估[191]中4种方法的平坦性(flatness)。简言之，当给定用户U_i的k个sweetwords时，$\mathcal{A}$简单的将这k个sweetwords按概率

Input: n个 sweetword 列表 $\{\mathrm{SW}_1, \mathrm{SW}_2, \cdots, \mathrm{SW}_n\}$，每个用户一个列表；每个用户允许使用 honeyword 登录的最大次数为$\mathcal{T}_1$；整个系统允许用户使用honeyword登录的最大次数为$\mathcal{T}_2$。
Output: 向量$\mathcal{V}$存储的是描绘success-number图相关的数据。

初始化空向量 $\mathcal{V}$;
初始化空优先级队列$crackQueue$；/* $crackQueue$ 的优先权值是sweetwords的条件概率。 */
$numFailure = 0$, $numSucess = 0$;
for $i = 1$ **to** n **do**
    $(p, sw) = getSweetword(\mathrm{SW}_i)$; /* 根据具体的攻击策略，从$\mathrm{SW}_i$中所有尚未尝试的sweetwords 中，抽取有最大优先权值p的sweetword sw。 */
    $crackQueue.Insert((p, \mathrm{SW}_i, sw))$;
end for
while $numFailure < \mathcal{T}_2$ **and** !$crackQueue.empty()$ **do**
    $(p, \mathrm{SW}_j, sw) = crackQueue.getFirst()$;
    $crackQueue.removeFirst()$;
    if $\mathrm{SW}_j.numFailure < \mathcal{T}_1$ **then**
        $success$=$Login(\mathrm{SW}_j.id, sw)$;/*以 $\mathrm{SW}_j.id$ 为用户名，sw 为口令，尝试登录用户 U_j。 */
        if $success$ **then**
            $numSuccess$++;
        else
            $\mathrm{SW}_j.numFailure$++, $numFailure$++;
            $\mathcal{V}.Insert((numFailure, numSuccess))$;
            $(p, sw) = getSweetword(\mathrm{SW}_j)$;
            $crackQueue.Insert((p, \mathrm{SW}_j, sw))$;
        end if
    end if
end while
Return $\mathcal{V}$;

算法 4.1 从给定的口令文件F中分辨出真口令，F中包含n个sweetwords列表

递减排序，这里概率直接来自于上述的已知概率分布$P_{\mathcal{D}}$。图4.4(e)显示，所有的方法都是0.35^{+}-flat的。换句话说，对k=20个sweetwords进行一次猜测，可以成功分辨出超过35%的真实口令。显然，这4种方法与理想方法之间有很大差距。

尽管这个攻击是有效的，但是它有明显的缺陷：当寻找下一个要猜测的sweetword $\mathrm{sw}_{i,j}$时，这个过程不关心$\mathrm{sw}_{i,j}$属于哪个列表。举个例子，假设$\Pr(\mathrm{sw}_{i,j})$=0.030，并且SW_i中的另外19个sweetwords概率都是0.029；$\Pr(\mathrm{sw}_{i+1,j})$=0.028，并且SW_{i+1}中的19个sweetwords概率都是0.001。然后问$\mathrm{sw}_{i,j}$和$\mathrm{sw}_{i+1,j}$哪个应该优先被攻击？$\mathrm{sw}_{i+1,j}$相对于SW_{i+1}，比$\mathrm{sw}_{i,j}$相对于SW_i更有可能是真口令。这种直觉产生了以下更有效的攻击策略。

Normalized top-PW 攻击。此攻击策略与上述策略的本质区别在于，现在$\mathcal{A}$尝试以归一化后的概率递减的顺序，攻击列表$\{\mathrm{SW}_1, \mathrm{SW}_2, \cdots, \mathrm{SW}_n\}$中的$n \cdot k$个sweetwords。更具体，每个sweetword $\mathrm{sw}_{i,j}$ $(1 \le i \le n$ 且 $1 \le j \le k)$的概率是直接来自于一个已知口令分布$\mathcal{D}$ (即List口令模型$P_{\mathcal{D}}(\cdot)$)，但是它已经对每个用户的k个sweetword 进行了归一化：

1) $\forall \mathrm{sw}_{i,j} \in \mathrm{SW}_i$, 如果$\mathrm{sw}_{i,j} \in \mathcal{D}$, 令 $\Pr(\mathrm{sw}_{i,j}) = P_{\mathcal{D}}(\mathrm{sw}_{i,j})$；
2) $\forall \mathrm{sw}_{i,j} \in \mathrm{SW}_i$，如果$\mathrm{sw}_{i,j} \notin \mathcal{D}$，令 $\Pr(\mathrm{sw}_{i,j}) = 0$；
3) $\forall \mathrm{sw}_{i,j} \in \mathrm{SW}_i$, 令 $\Pr(\mathrm{sw}_{i,j}) = \Pr(\mathrm{sw}_{i,j}) / \sum_{t=1}^{k} \Pr(\mathrm{sw}_{i,t})$。

Input: n个 sweetword 列表 $\{\mathrm{SW}_1, \mathrm{SW}_2, ..., \mathrm{SW}_n\}$, 每个用户一个列表。
Output: 向量 $\mathcal{V}$存储的是描绘flatness图相关的数据。

```
初始化空向量 V;
for i = 1 to n do
    r = 0;/* r 是用来记录成功分辨SW_i前的尝试次数。 */
    while !SW_i.isEmpty() do
        (p, sw) = getSweetword(SW_i);/*根据具体的攻击策略，从SW_i中所有尚未尝试的sweetwords中，抽取有最大优先权值p的sweetword sw。 */
        success=Login(SW_i.id, sw);/*以 SW_j.id 为用户名，sw 为口令，尝试登录用户 U_j。 */
        r++; /* SW_i的尝试次数加一。 */
        if success then
            break  /* 如果攻击成功，继续攻击下一个列表 SW_{i+1}。 */
        end if
    end while
    V.Insert(r);
end for
Return V
```

算法 4.2 对于给定的口令文件F，从每个*sweetword*列表中分辨真实口令

如果系统允许每个用户使用多次honeyword登录(即$\mathcal{T}_1 > 1$)，那么SW_i中任意一个被尝试后，剩余未被尝试的sweetwords应该重新进行归一化：

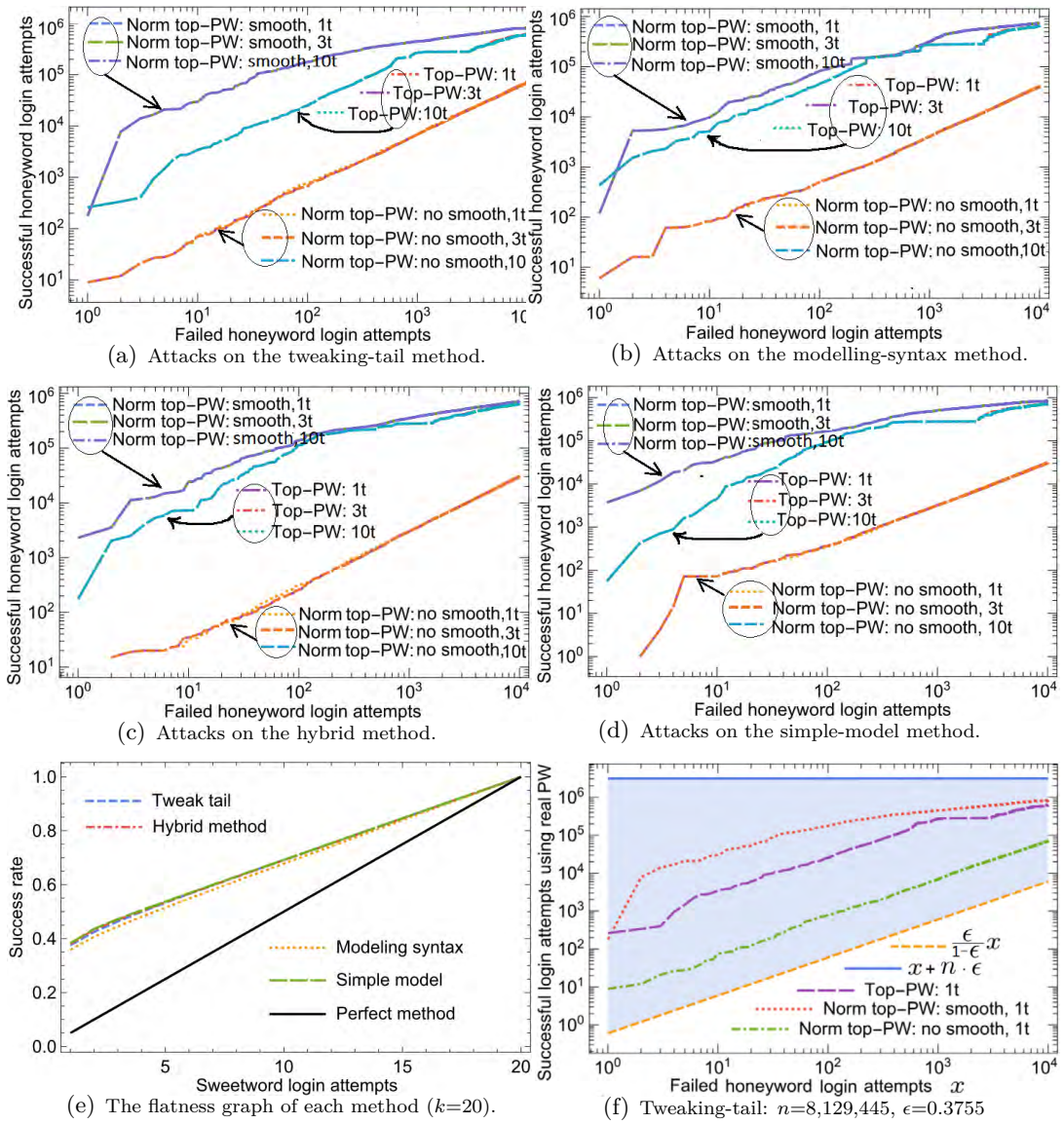
3′) 记$\mathcal{I}$为SW_i中已经被尝试登录的所有sweetwords的下标。$\forall \mathrm{sw}_{i,j} \in \mathrm{SW}_i \wedge j \notin \mathcal{I}$, 令 $\Pr(\mathrm{sw}_{i,j}) = \Pr(\mathrm{sw}_{i,j}) / \sum_{t \notin \mathcal{I}} \Pr(\mathrm{sw}_{i,t})$。

将SW_i中尚未尝试的sweetwords归一化，其中最大概率的sweetword将以其概率作为优先权放入优先级队列中，这对应于算法4.1中的`getSweetword`(SW_i)函数。一旦`getSweetword`(SW_i)函数实例化，算法4.1将会生成success-number图。类似的，经过步骤1∼3，SW_i中的每个sweetword都会赋予一个(归一化)的概率，算法4.2中的`getSweetword`(SW_i)就由此而实例化了。算法4.2运行后将会生成flatness图。

可以预见到口令文件F中的很多sweetwords都没有出现在分布$\mathcal{D}$中，这些sweetwords的概率会被赋值成0，这会严重影响sweetword排序的准确性。现实中，不同口令集的分布差别很大，见图4.2，这一点是合乎预期的。例如，当使用Dodonew-tr作为训练集(即$\mathcal{D}$)，Dodonew-ts作为测试集时，F中有超过74%的sweetwords概率都被赋值为0。直观的说，没有观测到某个事件，不代表这个事件的概率是0。在机器学习中，这个问题被称为数据稀疏问题，可以使用平滑技术加以解决。在语言模型中，有一系列平滑技术，如Laplace和Good-Turing，这些技术被广泛的使用在口令研究中，如[34]和第三章。然而，这些方法不能直接在本情景下使用，因为大部分的sweetwords都是没有出现的。因此，本章设计了一种“+1”平滑方法：

2′) $\forall \mathrm{sw}_{i,j} \in \mathrm{SW}_i$, 如果 $\mathrm{sw}_{i,j} \notin \mathcal{D}$, 令 $\Pr(\mathrm{sw}_{i,j}) = \frac{1}{|\mathcal{D}|+1}$。

本章尝试了Laplace、Good-Turing和“+1”平滑方法，发现本章的“+1”



(a) Attacks on the tweaking-tail method. (b) Attacks on the modelling-syntax method.

(c) Attacks on the hybrid method. (d) Attacks on the simple-model method.

(e) The flatness graph of each method (k=20). (f) Tweaking-tail: n=8,129,445, ϵ=0.3755

图 4.4 针对Juels-Rivest的4种honeywords生成方法的攻击结果，以success-number图和flatness图为评价指标。每种方法都应用了9 种攻击方式，训练集为50%的Dodonew(记为Dodonew-tr)，测试集为剩余的50%(记为Dodonew-ts)。子图(a)~(d)显示，当$\mathcal{T}_2$=10^4 时，带平滑的“norm top-PW”策略攻击每种方法时，均可成功区分出至少71万真口令。这里1t表示$\mathcal{T}_1=1$，3t表示$\mathcal{T}_1=3$ 等等。子图(e)揭示了4种方法都是0.35^+-flat的，比[191]中期望的结果高 7 倍。子图(f)示例了flatness和success-number两个指标间的关系。

平滑效果最好。因此，本章使用“+1”方法。注意，对Juels-Rivest方法，主要攻击的是流行口令，“+1”平滑针对这种口令是很有效的。在第4.5.3节中，本章将会提出一个新的平滑方法来攻击不流行的口令。

如图4.4(a)~4.4(d)所示，对于success-number图，带平滑的“normalized Top-PW”攻击策略显著优于不平滑的攻击。在大多数情况下，该方法明显的优于“Top-PW”方法，尤其是当允许的登录次数较小时。这证明了平滑技术对于*honeywords*是非常重要的。当以Dodonew-tr为训练集时，对于[191]中的4种

方法，该攻击策略可以从Dodonew-ts的8,129,445个账户中，成功分辨出711,035+个真实口令。这表示，对于success-number指标，Juels-Rivest方法的脆弱性至少被低估了1352=(711035/($\mathcal{T}_2$/($k-1$)))倍。

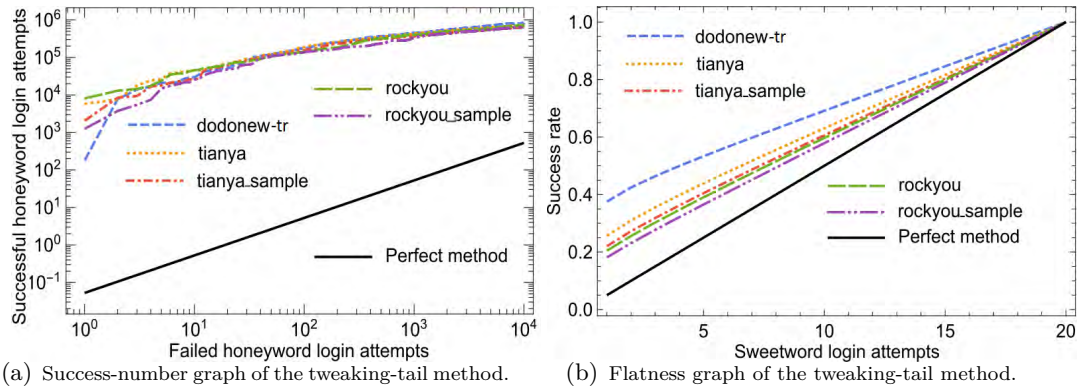
值得注意的一点是，当评估flatness安全指标时，上述的攻击策略对于同一个用户，会使用相同的顺序尝试honeywords。这是因为，步骤3中的“normalized Top-PW”策略，并没有影响每个sweetword列表中sweetword的相对顺序。因此，它们会产生同一个flatness图，见图4.4(e)。上述结果表明，这4种honeywords生成方法提供了相近的安全性，其中modeling syntax略好。在后续的实验中，它们也是相似的，因此下文仅以tweaking-tail为例。

对于每种攻击策略，攻击1、3或10次的结果很相近，如图4.4(a)∼4.4(d)。本质的原因是：(1)当每个用户只允许攻击一次(即$\mathcal{T}_1$=1)时，上述的攻击策略都能至少区分出615,664个真实口令；(2)即使每个用户允许攻击一次以上($\mathcal{T}_1$>1)，最多有$\mathcal{T}_2$个用户需要第二次攻击，那么相对于$\mathcal{T}_1$=1的情形，$\mathcal{A}$最多能额外分辨出$\mathcal{T}_2$个真实口令；(3)因为615,664$\gg\mathcal{T}_2$，所以615,664+$\mathcal{T}_2\approx$615,664。因此，以下本章评估success-number图时，只考虑$\mathcal{T}_1$=1的情形。注意，$\mathcal{T}_1$=1的条件对于评估flatness没有影响，这里总是允许对每个用户进行k次登录。

回忆一下，success-number测量的是，一个方法在最弱情况下抵抗攻击者的能力，即$\mathcal{A}$总是优先攻击最弱的账户。以图4.4(a)为例。由于123456是Dodonew-tr中最流行的口令(约占1.45%的比例)，所以$\mathcal{A}$会优先猜测所有用户中的123456为真口令。可能有人会觉得，在第一次登录失败之前，可以成功1.45%×8,129,445≈11万次。然而，这是误解，因为123456也可能是某个用户的honeyword，如果用户的口令服从模式123xxx 的话(如123123)。因此，攻击者$\mathcal{A}$的攻击123456时，很有可能在成功11万次前就已经发生多次失败登录了。如图4.4(a)，带平滑的“Norm Top-PW”攻击方式下$\mathcal{A}$的失败次数约为100，无平滑的“Norm Top-PW”攻击方式下约为1000。

4.3.3　进一步评估Juels-Rivest方法

上文评估了Juels-Rivest的4种主要方法抵抗2种攻击策略(包括不同的参数设置)的能力。所有的实验中，Dodonew-tr是训练集，Dodonew-ts是测试集，这意味着这两个集合都是来自于同一个分布，应该很相似。人们可能会考虑，如果测试集和训练集来自不同分布会如何。为回答这一问题，本章使用带平滑的“normalized top-PW”攻击策略，以Dodonew-ts为测试集，但是使用不同的训练集(Dodonew-tr, Tianya 和 Rockyou)进行攻击实验。为了解训练集大小对攻击效果的影响，本小节还使用了两个额外的训练集：Tianya_sample 和 Rockyou_sample。Tianya_sample是从Tianya 中随机抽取的与Dodonew-tr同等数



(a) Success-number graph of the tweaking-tail method. (b) Flatness graph of the tweaking-tail method.

图 4.5 研究不同训练集对攻击结果的影响：测试集为Dodonew-ts，攻击策略为“norm top-PW: smooth, 1t”。

量的样本，Rockyou_sample类似。

图4.5展示了针对tweaking-tail方法的攻击结果。从图4.5(a)中可以看出，当honeyword登录次数较小时(如小于300)，Tianya作为训练集最优；随着honeyword登录次数增加，Dodonew-tr作为训练集表现的最优。honeyword登录次数为10^4时，以Dodonew-tr为训练集的攻击者能分辨出820,703个真实口令，同时以Rockyou_sample为训练集的攻击者最差，但也能分辨出642,668个真实口令，1222倍于理想方法。图4.5(b)表明，即使训练集和测试集来自于两个不同的分布(即Tianya 和 Dodonew-ts)，“normalized top-PW”攻击策略只猜1次时也能到达26%的成功率，5倍于Juels-Rivest的最初预期；这5个训练集对于flatness指标有明确的优劣关系：Dodonew-tr>Tianya>Tianya_sample > Rockyou>Rockyou_sample。本章还对其它三种honeywords生成方法进行了实验，结果相似。总之，本节的结果表明：(1)训练集越接近测试集，攻击越有效；(2)训练集越大，攻击越有效；(3)在现实攻击下，Juels-Rivest的4种方法均是脆弱的，无法达到预期的安全性。

如前所述，本章目前只考虑了$\mathcal{A}_1$型的攻击者。如果攻击者有更强的现实能力(比如知道PII)，文献[191]中的方法很有可能更脆弱。另外，真实的口令集在不断泄漏(见[29, 225, 240, 259])，攻击者可以持续的改进她的训练集(即，第4.3.2节定义的List口令模型$P_{\mathcal{D}}(\cdot)$)，使得其尽可能接近测试集。实际上，许多网站(例如Anthem[229]，Yahoo[127]，Ubuntu论坛[260]和Phpbb[261]) 都已经不止一次泄漏过它们的口令集了。因此，本章认为，要评估一个honeywords生成算法，最好采用一个强大而现实的攻击者，并且使用最接近测试集的训练集来训练攻击者。根据这一点，在评估本章提出的honeyword方法时，会采用$\mathcal{A}_1 \sim \mathcal{A}_4$型的攻击者(见表4.6)，并且使用的测试集和训练集来自同一个分布(见节4.6)。

注意，在上述攻击中，sweetwords的概率均来自于List模型—一个已知的口令集。然而，每个泄漏口令集相对于整个口令空间都是非常小的。目前已知最

大的口令集是5亿的Yahoo[127]，它仍然远小于典型口令生成策略下的口令空间。比如，一个口令策略要求$len \in [8, 16]$，对应的口令空间大小为$10^{32} >> 10^9$。因此，如果使用更精确的概率模型(如PCFG[68]和Markov[34])，文献[191]中的4种方法会更脆弱。接下来我们将证实这一点。

4.3.4 Honeyindex的两个主要缺陷

为得到理想的honeywords生成方法，Imran[248]提出了honeyindex系统。Honeyindex直接使用其他用户的口令作为给定用户的honeywords。这样，honeywords的分布就与实际口令的分布相同。

对区分攻击是脆弱的。 Honeyindex系统中有一个严重缺陷。如果某个sweetindex si只包含在一个用户U_i的sweetindex列表中，那么si对应的口令必定是U_i的真实口令。这使得可以使用如下方法分辨真实口令：

1) 寻找只出现在一个用户的sweetindex列表中的孤立sweetindex。它对应的口令，必定是该用户的真实口令。
2) 删除上述用户的其它sweetindexes。这会出现新的孤立sweetindex。
3) 重复步骤1和2，直到没有孤立sweetindex。

注意，用户U_i真实口令对应的sweetindex，不可能包含在在U_i之前注册的用户的sweetindex列表中，只可能出现在在U_i之后注册的用户的sweetindex列表中。因此，上述分辨方法可以一直持续到所有用户的口令都被区分出来。

Imran[248] 意识到“新注册用户的口令可能不会被用作honeywords”，所以他建议对所有用户周期性的重新生成sweetindexes。但是，这会导致另一个问题：大部分用户不会周期性的更改他们的口令。所以，如果分辨攻击者拿到了两个不同时间点泄漏的口令文件，攻击者只需要比较同一用户的两个sweetword列表，同时包含在两个列表中的sweetword有很大可能是用户的真实口令。同时，很多网站(如[127, 240, 241])在泄漏事件发生数年之久后，才意识到被攻击。所以，攻击者有足够的时间多次窃取口令文件，直到网站发现这一点。

Honeyindex系统还有一个严重缺陷。口令是加盐存储在(sweetindex, password)表中。所以，很难保证用户的sweetwords互不相同。如果用户的某个honeyword与他的真实口令相同，那么，该honeyword在列表中的索引可能发送给honeychecker。这会造成错误的警报。为解决这一问题，honeywords需要与真实口令保持不同。当用户注册时，这一点可以做到，但是当sweetindex周期性重新生成时，就不可能做到了。这是因为，网站并不知道用户的明文口令，口令都是加盐哈希存储的。

计算成本高。 Honeyindex系统的存储成本低于honeyword系统，这是因为honeyindex只需要对每个用户存储1个哈希值，而honeyword每个用户要存储k

个哈希值。但是，这一优点是以高计算成本为代价的。

Honeywords系统对每个用户的sweetwords采用同一个盐进行哈希，用户登录时，网站只需使用该盐对用户的口令进行一次哈希，然后与列表中的k个sweetwords的哈希值比对即可。但是，honeyindex系统中的一个用户的sweetwords是用不同的盐哈希的(不同用户的真实口令会以不同的盐哈希存储)，所以网站最差情况下要做k次加盐哈希，平均情况下也要做$\frac{k}{2}$次。一般地，为安全地存储用户的口令，网站常被建议使用慢哈希函数(例如bcrypt和PBKDF2)。所以，认证时间会浪费在计算k次哈希上。综上，平均来说，honeyindex系统在登录时的计算成本，是honeyword的$\frac{k}{2}$倍。

小结。上述基于大规模真实数据的评估表明，Juels-Rivest的4种方法均无法实现预期的安全性：当每个帐户关联19个honeywords时(即[191]中推荐的$k = 20$)，$\mathcal{A}$只猜测一次，可以35.7%$^+$的概率将真实口令分辨出来，而不是预期的5%。下文将会展示，当$\mathcal{A}$获知了用户的一般个人信息时，这一概率将达到49.8%。

4.4 Honeywords攻击理论

为刻画攻击者$\mathcal{A}$从honeywords中分辨真实口令的最优策略，本节提出了一系列理论攻击模型。每个模型都基于不同的攻击者能力(即$\mathcal{A}$所掌握的不同信息)。特别的，我们首次考虑了利用受害者PII和利用用户注册顺序的攻击者，这样的攻击者是相当现实的。基于这些模型，可以利用真实口令集来设计有效的实验，对给定的honeywords生成算法评估其安全性(flatness)。

4.4.1 理论攻击模型

如节4.2.3所述，区分攻击者$\mathcal{A}$有两个目标：从一个给定用户U_i的sweetword列表中分辨出用户的真实口令；从一个给定的口令文件F(包含n个sweetword列表$\{\mathrm{SW}_1, \mathrm{SW}_2, ..., \mathrm{SW}_n\}$)中分辨出真实口令。由于$\mathcal{A}$对每个用户可以进行最多$\mathcal{T}_1$次失败尝试，对整个系统可以进行最多$\mathcal{T}_2$次失败尝试，为达到这两个目标，$\mathcal{A}$应优先尝试更有可能的sweetwords。本章下面将证明可达到这两个目标的最优攻击策略(框架)，并提出了一些新见解，来具体实现这一最优攻击策略。

定理 3 令 $sw_{i,j}$ (j=1, 2, $\cdots$, k) 表示“$\mathrm{sw}_{i,j}$ 是用户U_i 的 sweetword”这一事件, $\mathrm{sw}_{i,j}$表示“用户 U_i 选择 $\mathrm{sw}_{i,j}$ 作为自己的真实口令”这一事件。假设在事件$\mathrm{sw}_{i,j}$发生的条件下，$sw_{i,1}, sw_{i,2}, \cdots, sw_{i,j-1}, sw_{i,j+1}, \cdots, sw_{i,k}$是互相独立的[Φ]，那么有

[Φ]定理3中的假设并不与“$sw_{i,1}, sw_{i,2}, \cdots, sw_{i,j-1}, sw_{i,j+1}, \cdots, sw_{i,k}$依赖于 $\mathrm{sw}_{i,j}$”相矛盾。

$$\Pr(\mathrm{sw}_{i,j}|sw_{i,1},\cdots,sw_{i,k})=\frac{\Pr(\mathrm{sw}_{i,j})\prod_{l\neq j}\Pr(sw_{i,l}|\mathrm{sw}_{i,j})}{\sum_{t=1}^{k}\Pr(\mathrm{sw}_{i,t})\prod_{l\neq t}\Pr(sw_{i,l}|\mathrm{sw}_{i,t})}.$$

证明：由于$sw_{i,1},sw_{i,2},\cdots,sw_{i,j-1},sw_{i,j+1},\cdots,sw_{i,k}$在条件$\mathrm{sw}_{i,j}$下是互相独立的，所以$\Pr(sw_{i,1},\cdots,sw_{i,k}|\mathrm{sw}_{i,j})=\prod_{l\neq j}\Pr(sw_{i,l}|\mathrm{sw}_{i,j})$。再据贝叶斯理论可得

$$\begin{aligned}
&\Pr(\mathrm{sw}_{i,j}|sw_{i,1},\cdots,sw_{i,k})\\
=&\frac{\Pr(sw_{i,1},sw_{i,2},\cdots,sw_{i,j-1},\mathrm{sw}_{i,j},sw_{i,j+1},\cdots,sw_{i,k})}{\Pr(sw_{i,1},sw_{i,2},\cdots,sw_{i,k})}\\
=&\frac{\Pr(sw_{i,1},\cdots,sw_{i,k}|\mathrm{sw}_{i,j})\cdot\Pr(\mathrm{sw}_{i,j})}{\sum_{t=1}^{k}\Pr(sw_{i,1},\cdots,sw_{i,k}|\mathrm{sw}_{i,t})\cdot\Pr(\mathrm{sw}_{i,t})}\\
=&\frac{\prod_{l\neq j}\Pr(sw_{i,l}|\mathrm{sw}_{i,j})\cdot\Pr(\mathrm{sw}_{i,j})}{\sum_{t=1}^{k}\prod_{l\neq t}\Pr(sw_{i,l}|\mathrm{sw}_{i,t})\Pr(\mathrm{sw}_{i,t})}\\
=&\frac{\Pr(\mathrm{sw}_{i,j})\cdot\prod_{l\neq j}\Pr(sw_{i,l}|\mathrm{sw}_{i,j})}{\sum_{t=1}^{k}\Pr(\mathrm{sw}_{i,t})\cdot\prod_{l\neq t}\Pr(sw_{i,l}|\mathrm{sw}_{i,t})}.
\end{aligned}$$

■

这个定理表明，可以通过“分而治之”的方法来计算$\Pr(\mathrm{sw}_{i,j}|sw_{i,1},\cdots,sw_{i,k})$。具体来说，可以先计算$\Pr(\mathrm{sw}_{i,j})$(对任意$j\in\{1,2,\cdots,k\}$)和$\Pr(sw_{i,t}|\mathrm{sw}_{i,j})$(对任意$t\in\{1,2,\cdots,k\}$)，然后计算总体。幸运的是，$\Pr(\mathrm{sw}_{i,j})$可以使用多种概率模型进行计算(例如，直接使用某个泄漏口令集的频数，本章称之为List模型)，$\Pr(sw_{i,t}|\mathrm{sw}_{i,j})$可以通过分析honeywords生成方法来得到或消除(例如，对于tweaking-tail和modeling-syntax 方法，有$\Pr(sw_{i,l}|\mathrm{sw}_{i,j})=\Pr(sw_{i,l}|\mathrm{sw}_{i,t})$)。

定理 4 令 SW_i $(i=1,2,\cdots,n)$ 表示“SW_i 是用户 U_i 的sweetword列表”这一事件，其它符号与定理3相同。假设用户间选择口令是相互独立的，并且用户U_i的SW_i只依赖于U_i的真实口令$\mathrm{sw}_{i,j}$(以后会讨论依赖于U_i的*PII*的情况)，则

$$\Pr(\mathrm{sw}_{i,j}|SW_1,\cdots,SW_n)=\Pr(\mathrm{sw}_{i,j}|sw_{i,1},\cdots,sw_{i,k}),$$

证明：由于用户生成口令是独立的，并且SW_i只依赖于用户U_i的真实口令$\mathrm{sw}_{i,j}$，所以$\mathrm{SW}_1,\cdots,\mathrm{SW}_n$是相互独立的。由此可得

$$\begin{aligned}
&\Pr(\mathrm{sw}_{i,j}|SW_1,\cdots,SW_n)\\
=&\Pr(\mathrm{sw}_{i,j},\begin{pmatrix}sw_{1,1}&\cdots&sw_{1,k}\\ \vdots&\ddots&\vdots\\ sw_{n,1}&\cdots&sw_{n,k}\end{pmatrix})/\Pr(\begin{pmatrix}sw_{1,1}&\cdots&sw_{1,k}\\ \vdots&\ddots&\vdots\\ sw_{n,1}&\cdots&sw_{n,k}\end{pmatrix})\\
=&\frac{\Pr(\mathrm{sw}_{i,j},(sw_{i,1},\ldots,sw_{i,k}))\prod_{l\neq i}\Pr(sw_{l,1},\ldots,sw_{l,k})}{\prod_{l=1}^{n}\Pr(sw_{l,1},\ldots,sw_{l,k})}\\
=&\frac{\Pr(\mathrm{sw}_{i,j},(sw_{i,1},\ldots,sw_{i,k}))}{\Pr((sw_{i,1},\ldots,sw_{i,k}))}=\Pr(\mathrm{sw}_{i,j}|sw_{i,1},\ldots,sw_{i,k}).
\end{aligned}$$

■

这个定理揭示了，从$\mathrm{SW}_1,\cdots,\mathrm{SW}_n$中寻找最可能的口令，可以归结为先在每个$\mathrm{SW}_i$寻找最可能的口令$\mathrm{sw}_{i,j}$，再对这些$\mathrm{sw}_{i,j}$进行简单排序。从这一点来看，$\mathcal{A}$的两个目标本质上可以使用相同的攻击策略。

下面，本章将分析honeywords生成算法的性质。记$T(x)$为可以从sweetword x可以得到的sweetword空间。下面定义一个生成算法可能满足的4种性质：

1) $\mathrm{sw}_1 \in T(\mathrm{sw}_2) \implies T(\mathrm{sw}_1) = T(\mathrm{sw}_2)$。

这一性质说明，sweetword列表中的每个sweetword都可能是个生成者。这是个应该具有的性质。给定一个honeywords生成方法HW，如果$\mathrm{sw}_{i,j} \in \mathrm{SW}_i$不能生成$\mathrm{SW}_i$中的某些sweetwords，这就说明，$\mathrm{sw}_{i,j}$不是真实口令。对于这种的生成方法$HW$，$\mathcal{A}$不需要登录，就可以轻易的排除掉一些非真实口令，如$\mathrm{sw}_{i,j}$。

这一性质表明，$\forall t, T(\mathrm{sw}_{i,t}) = T(\mathrm{sw}_{i,1})$，并且$\forall t \forall l, \Pr(sw_{i,l}|\mathrm{sw}_{i,t}) \neq 0$。本章讨论的所有方法，都满足该性质。

2) $\forall \mathrm{sw}_3 \in T(\mathrm{sw}_1) = T(\mathrm{sw}_2), \Pr(sw_3|\mathrm{sw}_1) = \Pr(sw_3|\mathrm{sw}_2)$。

这一性质表示列表中的每个sweetword，都能被其它sweetword以同等的概率生成。该性质蕴含$\forall t,\ T(\mathrm{sw}_{i,t}) = T(\mathrm{sw}_{i,1})$。同样，本章讨论的所有方法，都满足该性质。

3) $\forall \mathrm{sw}_2,\ \mathrm{sw}_3 \in T(\mathrm{sw}_1), \Pr(sw_3|\mathrm{sw}_1) = \Pr(sw_2|\mathrm{sw}_1)$。

这一性质表明列表中的任何一个sweetword都能以同样的概率生成列表中的其它sweetwords。这蕴含了$\forall t \forall l, \Pr(\mathrm{sw}_{i,t}|\mathrm{sw}_{i,1}) = \Pr(\mathrm{sw}_{i,l}|\mathrm{sw}_{i,1})$。Juels-Rivest的前三种方法(见节4.3.1)都满足这一性质。

4) $\forall \mathrm{sw}_1, \mathrm{sw}_2, \mathrm{sw}_3 \in F, \Pr(sw_3|\mathrm{sw}_1) = \Pr(sw_3|\mathrm{sw}_2)$。其中$F$ 为口令文件。

这一性质是说，F中的每个sweetword都是独立于F中的其它sweetwords。这蕴含了honeywords是独立于真口令的。当与性质1结合时，它还蕴含了F中的sweetwords 都是来自于同一个空间—honeyword能生成的整个空间$\mathcal{D}_{hw}$。Honeyindex[248]以及本章提出的方法都满足这一性质。

情景 1。对于满足性质1~3的方法(如，第4.3.1节的前三种方法)，有$\forall i,j,l$, $\Pr(sw_{i,l}|\mathrm{sw}_{i,j}) = \Pr(sw_{i,l}|\mathrm{sw}_{i,t}) \neq 0$。那么

$$\begin{aligned} &\Pr(\mathrm{sw}_{i,j}|sw_{i,1},\cdots,sw_{i,k}) \\ =&\frac{\Pr(\mathrm{sw}_{i,j}) \cdot \prod_{l\neq j} \Pr(sw_{i,l}|\mathrm{sw}_{i,j})}{\sum_{t=1}^{k} \Pr(\mathrm{sw}_{i,t}) \cdot \prod_{l\neq t} \Pr(sw_{i,l}|\mathrm{sw}_{i,t})} \\ =&\frac{\Pr(\mathrm{sw}_{i,j})}{\sum_{t=1}^{k} \Pr(\mathrm{sw}_{i,t})}, \end{aligned} \tag{4.1}$$

这里 $\Pr(\mathrm{sw}_{i,j})$ 和 $\Pr(\mathrm{sw}_{i,t})$ 都可以通过各种口令概率模型(如，List、PCFG-

based[68] 和 Markov-based[34])来计算。

等式4.1从根本上解释了，为什么本章第4.3.2节中的“normalized top-PW”攻击策略是有效的，该策略对于这些生成方法是最优的。反过来说，第4.3.1节中的方法要想达到$\frac{1}{k}$-flat(即理想)的安全性，需要满足$\forall j,t,\ \Pr(\mathrm{sw}_{i,j}) = \Pr(\mathrm{sw}_{i,t})$。这意味着，列表$\mathrm{SW}_i$中的$k$个sweetwords被用户$U_i$选为真实口令的概率要相等。换句话说，给定一个真实口令PW_i，一个与真实口令相关的方法要能生成$k-1$个字符串(honeywords)，并且每个字符串的概率都等于$\Pr(\mathrm{PW}_i)$。这是根本无法实现的，因为用户选择的口令服从Zipf定律(见节2.4.3)：$p_r = C\cdot r^s - C\cdot(r-1)^s \approx C\cdot s\cdot r^{s-1}$，这里$p_r$表示第$r$流行口令的概率，并且常数$s\in[0.15,\ 0.30]$，$C\in[0.001,\ 0.01]$。$p_r$是$r$的单调递减函数，没有两个不同的字符串作为用户口令的概率是相同的。对于流行口令，这一问题尤其严重。这从根本上说明了，为解决与真实口令相关方法(*real-password-related methods*) 的缺陷，需要全新的、原则性的、系统性的设计方法。

情景 2。对于满足性质1、2、4的方法(例如，Honeyindex[248]和本章提出的方法)，都有$\forall \mathrm{sw}_1,\mathrm{sw}_2,\mathrm{sw}_3 \in \mathcal{D}_{hw},\ \Pr(sw_3|\mathrm{sw}_1) = \Pr(sw_3|\mathrm{sw}_2) = \Pr(sw_3|\mathcal{D}_{hw})$。为简便起见，将$\Pr(sw_3|\mathrm{sw}_1) = \Pr(sw_3|\mathrm{sw}_2)$ 记为 $\Pr_{\mathrm{HW}}(\mathrm{sw}_3)$，这意味着$\mathrm{sw}_3$作为sweetword的概率只依赖于honeyword 生成方法HW。类似，记用户选择字符串sw_1为真实口令的概率$\Pr(\mathrm{sw}_1)$为$\Pr_{\mathrm{PW}}(\mathrm{sw}_1)$。那么

$$
\begin{aligned}
&\Pr(\mathrm{sw}_{i,j}|sw_{i,1},\cdots,sw_{i,k})\\
=&\frac{\Pr(\mathrm{sw}_{i,j})\cdot\prod_{l\neq j}\Pr(sw_{i,l}|\mathrm{sw}_{i,j})}{\sum_{t=1}^{k}\Pr(\mathrm{sw}_{i,t})\cdot\prod_{l\neq t}\Pr(sw_{i,l}|\mathrm{sw}_{i,t})}\\
=&\frac{\Pr(\mathrm{sw}_{i,j})\prod_{l\neq j}\Pr(sw_{i,l}|\mathcal{D}_{hw})}{\sum_{t=1}^{k}\Pr(\mathrm{sw}_{i,t})\prod_{l\neq t}\Pr(sw_{i,l}|\mathcal{D}_{hw})}\\
=&\frac{\Pr(\mathrm{sw}_{i,j})\frac{\prod_{l}\Pr(sw_{i,l}|\mathcal{D}_{hw})}{\Pr(sw_{i,j}|\mathcal{D}_{hw})}}{\sum_{t=1}^{k}\Pr(\mathrm{sw}_{i,t})\frac{\prod_{l}\Pr(sw_{i,l}|\mathcal{D}_{hw})}{\Pr(sw_{i,t}|\mathcal{D}_{hw})}}\\
=&\frac{\frac{\Pr(\mathrm{sw}_{i,j})}{\Pr(sw_{i,j}|\mathcal{D}_{hw})}}{\sum_{t=1}^{k}\frac{\Pr(\mathrm{sw}_{i,t})}{\Pr(sw_{i,t}|\mathcal{D}_{hw})}} = \frac{\frac{\Pr_{\mathrm{PW}}(\mathrm{sw}_{i,j})}{\Pr_{\mathrm{HW}}(\mathrm{sw}_{i,j})}}{\sum_{t=1}^{k}\frac{\Pr_{\mathrm{PW}}(\mathrm{sw}_{i,t})}{\Pr_{\mathrm{HW}}(\mathrm{sw}_{i,t})}},
\end{aligned}
\tag{4.2}
$$

这里$\Pr_{\mathrm{PW}}(\mathrm{sw}_{i,j})\ (=\Pr(\mathrm{sw}_{i,j}))$ 和 $\Pr_{\mathrm{PW}}(\mathrm{sw}_{i,t})\ (=\Pr(\mathrm{sw}_{i,t}))$都可以通过多种口令概率模型(例如，List和 Markov-based[34])来计算；$\Pr_{\mathrm{HW}}(\mathrm{sw}_{i,j})$和$\Pr_{\mathrm{HW}}(\mathrm{sw}_{i,t})$可以通过直接查询给定的honeywords生成方法来获得。

式4.2给出了针对honeywords不依赖于用户真实口令(如Honeyindex[248]和本章方法)时的最优攻击策略。从另一方面来说，如果一个honeywords生成算法要到达理想的安全性(即$\frac{1}{k}$-flat)，它必须满足：$\forall j\forall t,\ \frac{\Pr_{\mathrm{PW}}(\mathrm{sw}_{i,t})}{\Pr_{\mathrm{HW}}(\mathrm{sw}_{i,t})} = \frac{\Pr_{\mathrm{PW}}(\mathrm{sw}_{i,j})}{\Pr_{\mathrm{HW}}(\mathrm{sw}_{i,j})}$。换句话说，生成方法的概率分布$\Pr_{\mathrm{HW}}(\cdot)$ 要正比于口令分布$\Pr_{\mathrm{PW}}(\cdot)$：$\forall x\in\mathcal{D}_{hw}$,

$\frac{\Pr_{\mathrm{PW}}(x)}{\Pr_{\mathrm{HW}}(x)}=c$，这里$D_{hw}$是honeyword空间，$c$是常量。一般地，因为$\mathcal{D}_{hw}=\mathcal{D}_{pw}$[①]，$\sum_{x\in\mathcal{D}_{hw}}\Pr_{\mathrm{HW}}(x)=1$ 并且 $\sum_{x\in\mathcal{D}_{pw}}\Pr_{\mathrm{PW}}(x)=1$, 所以$c=1$。

这也从另一方面说明，对于一个理想的honeyword方法，$\Pr_{\mathrm{HW}}(\cdot)$应和$\Pr_{\mathrm{PW}}(\cdot)$相等，而后者通常由底层的系统(例如语言、网站服务类型和口令生成策略)决定，可以采用多种口令概率模型(如List、PCFG-based[68]、Markov-based[34]和第三中的TarGuess)来近似计算。这意味着，这些口令概率模型也可以用来构造生成*honeyword*的生成方法(即$\Pr_{\mathrm{HW}}(\cdot)$)，我们将在下一节中展示如何使其成为现实。

4.4.2 其它三种攻击者类型

上文仅仅研究了$\mathcal{A}_1$能力的区分攻击者(见表4.6)$\mathcal{A}$的最优攻击策略。这种攻击者没有利用用户的PII 和用户的注册顺序。然而，如表4.5所示，大量的(38.33%∼51.43%)用户使用自己的PII构造口令；本节还将展示，用户的注册顺序也可以加以利用。下面先分别证明$\mathcal{A}_2$、$\mathcal{A}_3$ 和 $\mathcal{A}_4$型攻击者的最优攻击策略。

$\mathcal{A}_2$型攻击者。下文将展示，如果攻击者可以利用用户的个人信息，但 honeywords 生成方法不考虑的话，$\mathcal{A}_2$型攻击者将会获得比$\mathcal{A}_1$型更高的成功率。

定理 5 *令$sw_{i,j}$ (j=1, 2, ⋯, k) 表示“$\mathrm{sw}_{i,j}$ 是用户 U_i 的一个sweetword”这一事件, $\mathrm{sw}_{i,j}$ 表示“用户 U_i 选择 $\mathrm{sw}_{i,j}$ 作为自己的真实口令”这一事件, PII 表示“用户U_i 使用个人信息PII构造口令”这一事件。假设在事件($\mathrm{sw}_{i,j}$, PII) 下，$sw_{i,1}, sw_{i,2}, \cdots, sw_{i,j-1}, sw_{i,j+1}, \cdots, sw_{i,k}$是互相独立的，那么有*

$$\Pr(\mathrm{sw}_{i,j}|sw_{i,1},\cdots,sw_{i,k},\mathsf{PII})=\frac{\Pr(\mathrm{sw}_{i,j}|\mathsf{PII})\prod_{l\neq j}\Pr(sw_{i,l}|\mathrm{sw}_{i,j},\mathsf{PII})}{\sum_{t=1}^{k}\Pr(\mathrm{sw}_{i,t}|\mathsf{PII})\prod_{l\neq t}\Pr(sw_{i,l}|\mathrm{sw}_{i,t},\mathsf{PII})},$$

证明：由于在条件$\mathrm{sw}_{i,j}\cap\mathsf{PII}$下，$sw_{i,1}, sw_{i,2}, \cdots, sw_{i,j-1}, sw_{i,j+1}, \cdots, sw_{i,k}$是互相独立的，所以$\Pr(sw_{i,1},\cdots,sw_{i,k}|\mathrm{sw}_{i,j},\mathsf{PII})=\prod_{l\neq j}\Pr(sw_{i,l}|\mathrm{sw}_{i,j},\mathsf{PII})$。那么由贝叶斯理论可得，

$$\begin{aligned}
&\Pr(\mathrm{sw}_{i,j}|sw_{i,1},\cdots,sw_{i,k},\mathsf{PII})\\
=&\frac{\Pr(sw_{i,1},sw_{i,2},\cdots,sw_{i,j-1},\mathrm{sw}_{i,j},sw_{i,j+1},\cdots,sw_{i,k},\mathsf{PII})}{\Pr(sw_{i,1},sw_{i,2},\cdots,sw_{i,k},\mathsf{PII})}\\
=&\frac{\Pr(sw_{i,1},\cdots,sw_{i,k}|\mathrm{sw}_{i,j},\mathsf{PII})\cdot\Pr(\mathrm{sw}_{i,j}|\mathsf{PII})}{\sum_{t=1}^{k}\Pr(sw_{i,1},\cdots,sw_{i,k}|\mathrm{sw}_{i,t},\mathsf{PII})\cdot\Pr(\mathrm{sw}_{i,t}|\mathsf{PII})}\\
=&\frac{\prod_{l\neq j}\Pr(sw_{i,l}|\mathrm{sw}_{i,j},\mathsf{PII})\cdot\Pr(\mathrm{sw}_{i,j}|\mathsf{PII})}{\sum_{t=1}^{k}\prod_{l\neq t}\Pr(sw_{i,l}|\mathrm{sw}_{i,t},\mathsf{PII})\Pr(\mathrm{sw}_{i,t}|\mathsf{PII})}\\
=&\frac{\Pr(\mathrm{sw}_{i,j}|\mathsf{PII})\cdot\prod_{l\neq j}\Pr(sw_{i,l}|\mathrm{sw}_{i,j},\mathsf{PII})}{\sum_{t=1}^{k}\Pr(\mathrm{sw}_{i,t}|\mathsf{PII})\cdot\prod_{l\neq t}\Pr(sw_{i,l}|\mathrm{sw}_{i,t},\mathsf{PII})}.
\end{aligned}$$

■

[①]如果honeyword比$\mathcal{D}_{hw}$大的话，那么必定存在$x\in\mathcal{D}_{hw}$但是$x\notin\mathcal{D}_{pw}$。如果这样的x出现在一个sweetword列表中，$\mathcal{A}$可以轻易的排除它。因此，最好$\mathcal{D}_{hw}=\mathcal{D}_{pw}$。

类似定理4的证明，有

$$\Pr(\mathrm{sw}_{i,j}|SW_1,\cdots,SW_n,\mathsf{PII}) = \Pr(\mathrm{sw}_{i,j}|sw_{i,1},\ldots,sw_{i,k},\mathsf{PII}) \tag{4.3}$$

根据定理5, 如果攻击者 $\mathcal{A}$ 利用受害者的PII，但是honeywords生成方法不考虑用户的PII且满足性质1~3，那么式4.3可以化简为：

$$\Pr(\mathrm{sw}_{i,j}|sw_{i,1},\cdots,sw_{i,k},\mathsf{PII}) = \frac{\Pr(\mathrm{sw}_{i,j}|\mathsf{PII})}{\sum_{t=1}^{k}\Pr(\mathrm{sw}_{i,t}|\mathsf{PII})} = \frac{\Pr_{\mathrm{PW}}(\mathrm{sw}_{i,j}|\mathsf{PII})}{\sum_{t=1}^{k}\Pr_{\mathrm{PW}}(\mathrm{sw}_{i,t}|\mathsf{PII})}. \tag{4.4}$$

如第三章所述，$\Pr_{\mathrm{PW}}(\mathrm{sw}_{i,t}|\mathsf{PII})$比$\Pr_{\mathrm{PW}}(\mathrm{sw}_{i,t})$能更准确的刻画用户的口令分布。这意味着，$\mathcal{A}$有更大的优势。例如，100次猜测，TarGuess-I(即targeted-PCFG)模型比non-targeted PCFG模型能多攻击417%的口令。

对于满足性质1、2、4的honeywords生成方法，如果攻击者能利用受害者的PII，但是生成方法不考虑的话，式4.3可以化简为：

$$\Pr(\mathrm{sw}_{i,j}|sw_{i,1},\cdots,sw_{i,k},\mathsf{PII}) = \frac{\frac{\Pr_{\mathrm{PW}}(\mathrm{sw}_{i,j}|\mathsf{PII})}{\Pr_{\mathrm{HW}}(\mathrm{sw}_{i,j})}}{\sum_{t=1}^{k}\frac{\Pr_{\mathrm{PW}}(\mathrm{sw}_{i,t}|\mathsf{PII})}{\Pr_{\mathrm{HW}}(\mathrm{sw}_{i,t})}}. \tag{4.5}$$

式4.4和4.5都说明，如果攻击者考虑用户的PII，会提升她的攻击优势。本章在第4.4.3节展示了实验结果。

$\mathcal{A}_3$型攻击者。由于用户的注册顺序对与真实口令无关的honeywords生成方法是很重要的，所以第4.4.1节情景2中的$\mathcal{A}_3$型攻击者是受限的。更具体一点，对于式4.2，攻击者可以利用用户的注册顺序(例如，自适应的更新训练集)来更好的计算$\Pr_{\mathrm{HW}}(\cdot)$，记新的计算方法为$\Pr_{\widetilde{\mathrm{HW}}}(\cdot)$，那么有

$$\Pr(\mathrm{sw}_{i,j}|sw_{i,1},\cdots,sw_{i,k},\mathsf{Reg}) = \frac{\frac{\Pr_{\mathrm{PW}}(\mathrm{sw}_{i,j})}{\Pr_{\widetilde{\mathrm{HW}}}(\mathrm{sw}_{i,j})}}{\sum_{t=1}^{k}\frac{\Pr_{\mathrm{PW}}(\mathrm{sw}_{i,t})}{\Pr_{\widetilde{\mathrm{HW}}}(\mathrm{sw}_{i,t})}}. \tag{4.6}$$

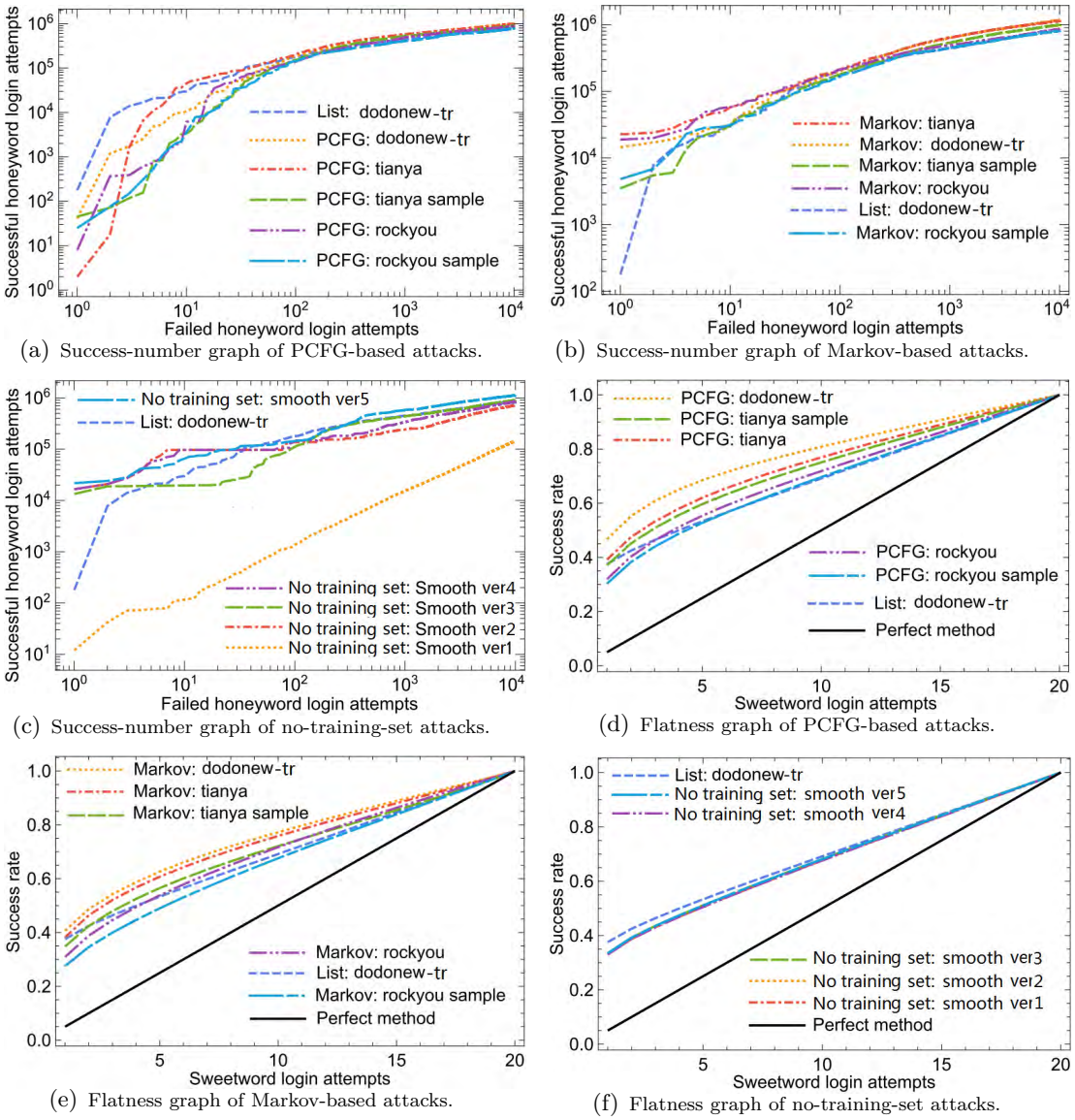
$\mathcal{A}_4$型的攻击者。当$\mathcal{A}_3$型攻击者也利用用户的PII时，通过联合式4.5和4.6可以获得更大的优势：

$$\Pr(\mathrm{sw}_{i,j}|sw_{i,1},\cdots,sw_{i,k},\mathsf{PII},\mathsf{Reg}) = \frac{\frac{\Pr_{\mathrm{PW}}(\mathrm{sw}_{i,j}|\mathsf{PII})}{\Pr_{\widetilde{\mathrm{HW}}}(\mathrm{sw}_{i,j})}}{\sum_{t=1}^{k}\frac{\Pr_{\mathrm{PW}}(\mathrm{sw}_{i,t}|\mathsf{PII})}{\Pr_{\widetilde{\mathrm{HW}}}(\mathrm{sw}_{i,t})}}. \tag{4.7}$$

式4.6和4.7都说明，攻击者可以利用用户的注册顺序显著的提高其优势，下文中的实验将会证实这一点。同时，这两个式也指出设计新方法的方向。

4.4.3　攻击理论的实现

本节展示如何利用上述攻击理论来设计更有效的攻击方案。不失一般性，这里仅以[191]中的tweaking-tail方法作为攻击目标。在此之前，我们先提出一个



(a) Success-number graph of PCFG-based attacks.
(b) Success-number graph of Markov-based attacks.
(c) Success-number graph of no-training-set attacks.
(d) Flatness graph of PCFG-based attacks.
(e) Flatness graph of Markov-based attacks.
(f) Flatness graph of no-training-set attacks.

图 4.6 针对tweaking-tail方法，基于不同的概率模型进行攻击，训练集为Dodonew-tr。每个概率模型都使用了5种不同的训练集，并且以训练集为Dodonew-tr的List模型作为基准。

不需要任何训练集的honeywords概率模型。

(1) No-train-set模型

该模型利用了一个事实：如果honeywords是均匀分布的(即$\Pr_{\mathrm{HW}}(\cdot|x) = \frac{1}{|T(x)|}$)，那么可以从sweetwords分布中计算出真实口令的分布。

由于每个sweetword列表都含有$k-1$个honeywords和1个真实口令，所以sweetword的分布$\Pr_{\mathrm{SW}}(\cdot)$可以被真实口令分布$\Pr_{\mathrm{PW}}(\cdot)$和honeyword分布$\Pr_{\mathrm{HW}}(\cdot|\cdot)$所决定，式如下：

$$\Pr_{\mathrm{SW}}(x) = \frac{1}{k}\Pr_{\mathrm{PW}}(x) + \frac{k-1}{k}\sum_{y}\Pr_{\mathrm{PW}}(y)\Pr_{\mathrm{HW}}(x|y).$$

因此有

$$\begin{aligned}\Pr_{\mathrm{PW}}(x) =& k\Pr_{\mathrm{SW}}(x) - (k-1)\sum_{y}\Pr_{\mathrm{PW}}(y)\Pr_{\mathrm{HW}}(x|y)\\ =& k\Pr_{\mathrm{SW}}(x) - (k-1)\Pr_{\mathrm{PW}}(T(x))\frac{1}{|T(x)|}\\ =& k\Pr_{\mathrm{SW}}(x) - (k-1)\Pr_{\mathrm{SW}}(T(x))\frac{1}{|T(x)|}.\end{aligned}$$

由于sweetword的概率(记为$\Pr_{\mathrm{SW}}(x)$)可以使用其频率$\frac{\mathrm{count}_{\mathrm{SW}}(x)}{\mathrm{count}_{\mathrm{SW}}}$进行估计，由此可得$\Pr_{\mathrm{PW}}(x)$的一个估计：

$$\widehat{\Pr}_{\mathrm{PW}}(x) = k\cdot\frac{\mathrm{count}_{\mathrm{SW}}(x)}{\mathrm{count}_{\mathrm{SW}}} - (k-1)\frac{\mathrm{count}_{\mathrm{SW}}(T(x))}{\mathrm{count}_{\mathrm{SW}}}\cdot\frac{1}{|T(x)|}.$$

本章发现，有些sweetwords的$\widehat{\Pr}_{\mathrm{PW}}(\cdot)$值小于0，所以需要一些平滑技术。本章测试了如下几种平滑方法，见图4.6(f)：

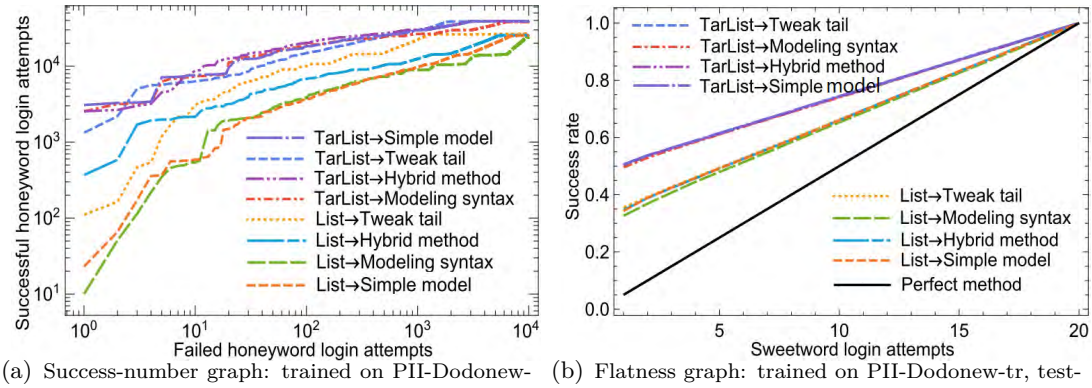
Version 1: $\widehat{\Pr}_{PWwithSmooth1} = \max\{0, \widehat{\Pr}_{\mathrm{PW}}(x)\}$;
Version 2: $\widehat{\Pr}_{PWwithSmooth2} = \max\{\frac{1}{\mathrm{count}_{\mathrm{SW}}}, \widehat{\Pr}_{\mathrm{PW}}(x)\}$;
Version 3: $\widehat{\Pr}_{PWwithSmooth3} = \max\{\frac{\mathrm{count}_{\mathrm{SW}}(x)}{\mathrm{count}_{\mathrm{SW}}}, \widehat{\Pr}_{\mathrm{PW}}(x)\}$;
Version 4: $\widehat{\Pr}_{PWwithSmooth4} = \max\{\frac{1}{\mathrm{count}_{\mathrm{PW}}}, \widehat{\Pr}_{\mathrm{PW}}(x)\}$;
Version 5: $\widehat{\Pr}_{PWwithSmooth5} = \frac{\mathrm{count}_{\mathrm{SW}}(x)}{\mathrm{count}_{\mathrm{SW}}}$.

(2) 攻击理论的现实应用

首先，由于定理3和4的普适性，它们可以很容易的应用于tweaking-tail方法。下一步将通过分析honeywords生成方法的性质，选择合适的攻击理论。由于tweaking-tail方法满足性质1~3，对于$\mathcal{A}_1$ 型的攻击者，式4.1是合适的；对于$\mathcal{A}_2$型攻击者，式4.4 是合适的。由于tweaking-tail生成方法只与用户的真实口令相关，注册顺序是没有利用价值的。因此，在这种情况下，$\mathcal{A}_3$ 和 $\mathcal{A}_4$ 型的攻击者不会分别比$\mathcal{A}_1$ 和 $\mathcal{A}_2$型的攻击者更有效。综上，式4.1 和4.4对攻击tweaking-tail方法是有帮助的。

根据式4.1，本章设计了一系列针对[191]中tweaking-tail方法的攻击实验(见图4.6)，来揭示不同口令攻击模型的优劣。攻击模型包括，PCFG[68], Markov[34]和本节提出的 No-training-set模型。对于Markov-based攻击，所有参数与图4.4(a)中的“Norm top-PW: smooth, 1t”保持一致，除了已知口令分布$\mathcal{D}$(例如，某个泄漏口令集的经验分布)被换成Markov攻击模型(本章使用文献[34]中 的backoff技 术)。 记Markov的 概 率 分 布 为$P_{\mathsf{Markov}(\mathcal{D})}(\cdot)$，那么式中的$P_{\mathcal{D}}(\cdot)$要替换成$P_{\mathsf{Markov}(\mathcal{D})}(\cdot)$。类似的，记PCFG-based口令模型的概率分布为$P_{\mathsf{PCFG}(\mathcal{D})}(\cdot)$，no-training-set的模型为$P_{\mathcal{F}}(\cdot)$。

从图4.6可知：(1)一般 PCFG-based 模型$P_{\mathsf{PCFG}(\mathcal{D})}(\cdot)$ 和 Markov-based 模



(a) Success-number graph: trained on PII-Dodonew-tr, tested on PII-Dodonew-ts.

(b) Flatness graph: trained on PII-Dodonew-tr, tested on PII-Dodonew-ts.

图 4.7 对比$\mathcal{A}_1$型攻击者(使用List模型)和$\mathcal{A}_2$型攻击者(使用TarList模型)的效率。

型$P_{\mathsf{Markov}(\mathcal{D})}(\cdot)$ 攻击效果好于List模型$P_{\mathcal{D}}(\cdot)$，当给定同样的测试集和训练集时；(2)tweaking-tail方法是0.47$^+$-flat(见图4.6(d))，比[191]中预期的弱9倍；(3)有趣的是，当采用第5种平滑方法时，本章no-training-set模型$P_{\mathcal{F}}(\cdot)$，非常接近 List 模型$P_{\mathcal{D}}(\cdot)$，这说明$\mathcal{A}$只使用给定的sweetword 文件F，无需额外的训练集，也能进行有效的攻击。

根据式4.4，本章设计了一系列实验，攻击[191]中的4种方法。结果显示，如果攻击者利用受害者的PII，但是生成方法不考虑用户的PII的话，$\mathcal{A}$可以有效的提升攻击成功率。如图4.7所示，当$\mathcal{T}_1$=10^4时，$\mathcal{A}$猜中的真实口令数可以增加46.7%~61.3%，并且ϵ-flatness方面，成功率可以增加42.3%~52.9%。更令人担忧的是，对于任意一种生成方法，利用PII的攻击者只尝试一次就能以超过49.8%的概率从19个honeywords区分出用户的真实口令(见图4.7(b)中的点(x=1, y=0.5))，而期望的成功率只有5%。

表 4.7 本章提出的honeywords生成方法及其评估方法

$\mathcal{A}$的类型(见表4.6)	本文提出的方法		最优的攻击 / 评估方法 (即计算第4.4 节中的概率)	
	使用的口令模型	解决的问题	$\Pr_{\mathrm{PW}}(\cdot)$ (or $\Pr_{\widetilde{\mathrm{PW}}}(\cdot)$)	$\Pr_{\mathrm{HW}}(\cdot)$ (or $\Pr_{\widetilde{\mathrm{HW}}}(\cdot)$)
$\mathcal{A}_1$型	List	+1 平滑	List; +1 平滑, $\frac{\Pr_{\mathrm{PW}}(\cdot)}{\Pr_{\mathrm{HW}}(\cdot)}$=1 平滑	与生成方法相同
$\mathcal{A}_2$型	TarList	+1 平滑	TarList; +1 平滑, $\frac{\Pr_{\mathrm{PW}}(\cdot)}{\Pr_{\mathrm{HW}}(\cdot)}$=1 平滑	与生成方法相同
$\mathcal{A}_3$型	$\frac{1}{3}$List+$\frac{1}{3}$Markov +$\frac{1}{3}$PCFG	+1 平滑; 内部训练; $\frac{\Pr_{\mathrm{PW}}(\cdot)}{\Pr_{\widetilde{\mathrm{HW}}}(\cdot)}$ >20 tweak tail	List; +1 平滑, $\frac{\Pr_{\mathrm{PW}}(\cdot)}{\Pr_{\widetilde{\mathrm{HW}}}(\cdot)}$=1 平滑	与生成方法相同
$\mathcal{A}_4$型	$\frac{1}{3}$TarList+$\frac{1}{3}$Tar-Markov+$\frac{1}{3}$TarPCFG	+1 平滑; 内部训练; $\frac{\Pr_{\mathrm{PW}}(\cdot)}{\Pr_{\widetilde{\mathrm{HW}}}(\cdot)}$ >20 tweak tail	TarList; +1 平滑, $\frac{\Pr_{\mathrm{PW}}(\cdot)}{\Pr_{\widetilde{\mathrm{HW}}}(\cdot)}$=1 平滑	与生成方法相同

小结。本章首次针对不同能力的攻击者，提出了一系列坚实的理论攻击模型。特别地，本章首次研究了利用受害者PII和利用用户注册顺序的攻击者。根据这些模型，可以使用现实口令集来设计有效的实验，来评估honeywords生成方法的安全性，这回答了[191]中一个遗留的一个公开问题“是否有好的实验方法来测量honeywords生成算法的平坦性？”这将使honeywords的评估从艺术走向科学，同时我们将在下一节讨论如何设计良好的honeywords生成方法。

4.5　新的honeyword生成方法

接下来，本章将展示如何构造新的honeywords生成方法。受到上述最佳“剑”(包括攻击理论和实验)的启发，本章打造相应的最佳“盾”—4种安全有效的honeword生成方法(基于各种主流的口令攻击概率模型，如漫步猜测模型[34, 68] 和第三章中的定向猜测模型)。然而，如何使用这些口令模型并不是显然的，需要一系列重要、新颖、创造性的努力，本章将通过一系列探索性研究来展示。除此以外，honeyword方法在实际部署中还遇到了多个之前未曾披露的新挑战，本章成功解决了这些问题。

4.5.1　新方法概述

文献[191]中4种方法在为一个用户生成honeywords时，都依赖于该用户的真实口令，这一设计方法存在根本性的局限。这种方法会暴露真实口令PW_i的特征(如长度和字符组成)，这将减少$\mathcal{A}$离线猜测的成本。一旦$\mathcal{A}$从口令哈希文件F 中恢复出了一个sweetword，那么她就知道了同一个sweetword列表中其它sweetwords的特征，从而大大减少猜测空间。更严重的是，这种方法本质上不能生成k-1个概率值为$\mathrm{Pr}(\mathrm{PW}_i)$的honeywords，这是因为用户选择的口令服从Zipf 定理(见节4.4.1的情景1)。因此，本章采用与真实口令无关的设计方法。

本章考虑了表4.6中的4种不同能力的区分攻击者，并设计了对应的最优攻击策略。我们的基本观点是，不同能力的攻击者，有不同的最优攻击策略；为抵御不同的最优攻击策略，需要设计不同的*honeywords*生成方法。这导致了表4.7中总结的4种方法。现存的概率模型(例如，PCFG-based[68]、Markov-based[34]和TarGuess-based)不能直接利用，需要在设计和评估过程中进行一系列调整。当$\mathcal{A}$利用用户注册顺序(例如社交论坛)时，这些调整是特别具有挑战性。表4.7中的设计及其内在逻辑将在下文中仔细阐述。

4.5.2　现有口令概率模型对比

本章主要关注，List-based、Markov-based[34]和PCFG-based[68]这三个口令概率模型，及它们的targeted版本。这些模型是当前的主流模型，并在口令研究中得到了广泛的应用和检验[142, 197, 262]。

第一个面临的问题是：在这么多候选口令概率模型中，哪个模型最合适？为回答这一问题，本章研究了每个模型的缺陷。一般，有两个因素影响machine-learning-based的口令模型的有效性：模型本身和训练集。为排除训练集的影响，如[71, 93]中的方法，本章随机的将Dodonew口令集切成相等的两份，一份作为训练集(即Dodonew-tr)，一份作为测试集(即Dodonew-ts)。根据[34]中的最新改进，本章实现了PCFG-based模型和Markov-based模型。更具

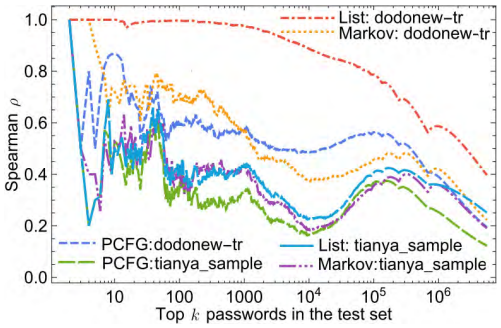


图 4.8 比较不同口令概率模型生成的猜测序列与测试集Dodonew-ts的差距。图上标注了每个口令模型的训练集。Tianya_sample与Dodonew-tr的大小相同。

表 4.8 不同口令模型下典型口令的$\frac{\Pr_{\mathrm{PW}}(\cdot)}{\Pr_{\mathrm{HW}}(\cdot)}$值(Dodonew-tr vs. Dodonew-ts)。深灰色的$\frac{\Pr_{\mathrm{PW}}(\cdot)}{\Pr_{\mathrm{HW}}(\cdot)}$值表示相应的模型预测该口令的概率表现最差，浅灰色表示对应的模型表现第二差。

典型口令	概率 $\Pr_{\mathrm{PW}}(\cdot)$	List	PCFG	Markov
123456	0.01443750	0.99	1.25	0.96
password	0.00044136	1.02	1.63	1.25
123qwe	0.00027111	0.95	46.35	1.39
1q2w3e4r	0.00011588	1.14	$6.57 \cdot 10^{10}$	1.18
147852369	0.00004293	1.07	0.92	107.42
110120130	0.00002337	0.84	0.86	5059.23
110011	0.00000886	0.77	1.05	41.06
password123	0.00000381	1.07	0.55	1.82
p@ssw0rd	0.00000221	0.90	$8.74 \cdot 10^{10}$	1.25
XX123456	0.00000160	13.00	1.39	4.58
34567890	0.00000148	12.53	6.87	6.92
123qwe123qwe	0.00000123	0.77	6662.66	27.13
Password123	0.00000037	0.60	2.02	5.31
iloveyou123456	0.00000025	0.67	0.59	0.51
123456abcdefg	0.00000012	0.09	7.98	0.31
520yong	0.00000011	0.09	0.51	0.44

体的说，PCFG-based模型的letter组件是来自于训练集(而不是额外的字典)，对于Markov-based模型，本章使用的是backoff技术。

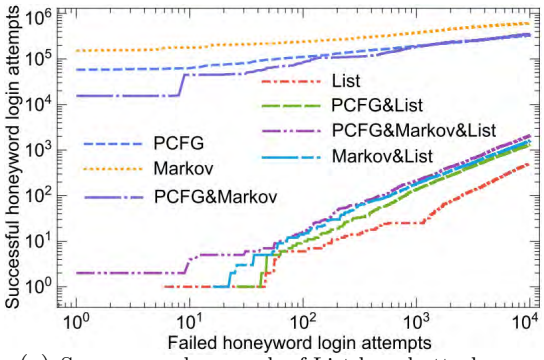
本章采用斯皮尔曼等级相关系数ρ来测量概率模型的猜测序列与测试集的距离(见第4.4.2节)。这引出了图4.8，表明：(1)在估计流行口令方面，List模型表现最好；(2)PCFG模型和Markov模型的曲线相互交错，说明各有优势；(3)训练集的选择对结果有重要影响，但对比较不同概率模型的优劣没有影响。注意，点(x,y)表示，某个模型的前x次攻击序列与测试集的前x流行口令序列之间的斯皮尔曼等级相关系数ρ为y。

表 4.9 不同口令概率模型下$\frac{\Pr_{\mathrm{PW}}(\cdot)}{\Pr_{\mathrm{HW}}(\cdot)}$最大的前$10^3$个口令的基本信息

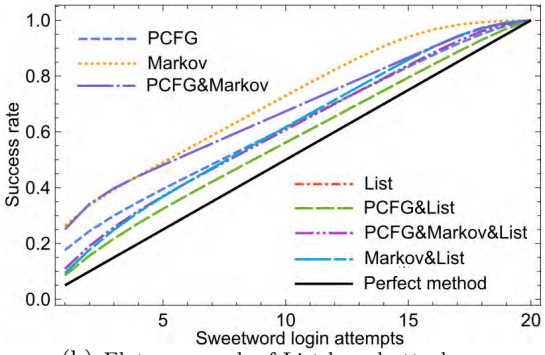
	List	Markov	PCFG
在前10^4个流行口令	**0.000**	**0.000**	**0.000**
不在训练集中	**1.000**	**0.993**	**0.998**
频数>2 (测试集中)	**1.000**	**0.006**	**0.005**
频数<10 (测试集中)	0.960	1.000	1.000
口令长度≥16	0.000	**0.906**	**1.000**
结构长度≥6	0.001	**0.995**	**0.576**
含PII语义信息	0.012	0.016	0.051
Email/网站地址	0.000	**0.442**	**0.695**

为更好的理解图4.8，我们提供了一个更微观的视角，来研究3种模型的区别，见表4.8。我们计算了典型口令x的$\frac{\Pr_{\mathrm{PW}}(x)}{\Pr_{\mathrm{HW}}(x)}$值，$\Pr_{\mathrm{PW}}(x)$直接来自测试集Dodonew-ts，而$\Pr_{\mathrm{HW}}(x)$来自于口令模型。可以推测：List模型估计流行口令的概率是最优的，PCFG擅长估计有相同结构的口令的概率，而Markov擅长短口令。这些说明，单个口令模型各有其优势和劣势，为抵御利用概率模型缺陷的攻击者，混合模型(例如，$\frac{1}{3}$List+$\frac{1}{3}$Markov+$\frac{1}{3}$PCFG)可能更为合适。

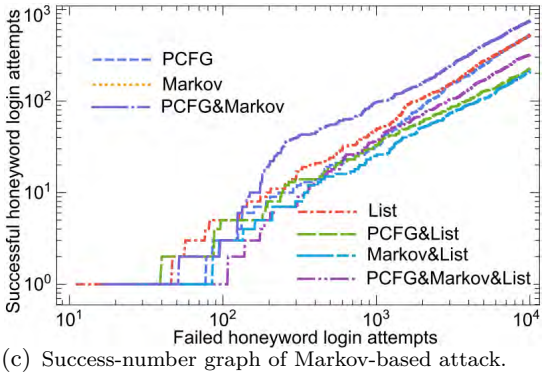
上述猜测在表4.9中得到了证实。本章评估了每个模型的$\frac{\Pr_{\mathrm{PW}}(\cdot)}{\Pr_{\mathrm{HW}}(\cdot)}$最大的



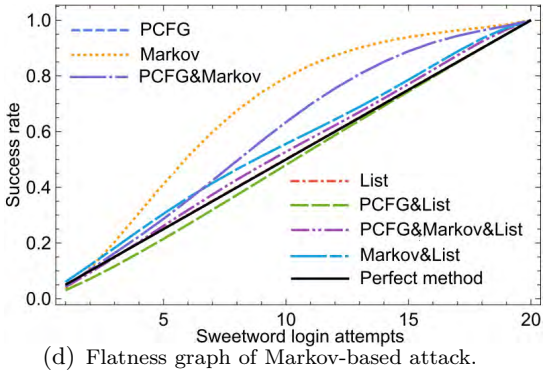
(a) Success-number graph of List-based attack.



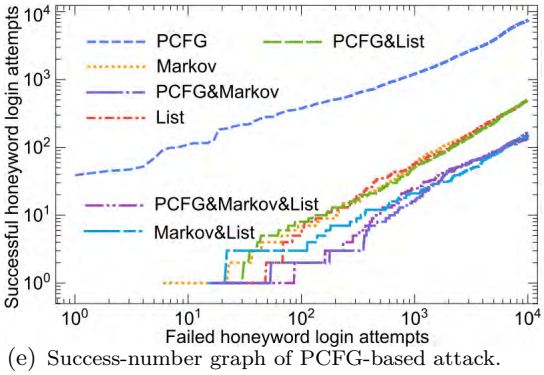
(b) Flatness graph of List-based attack.



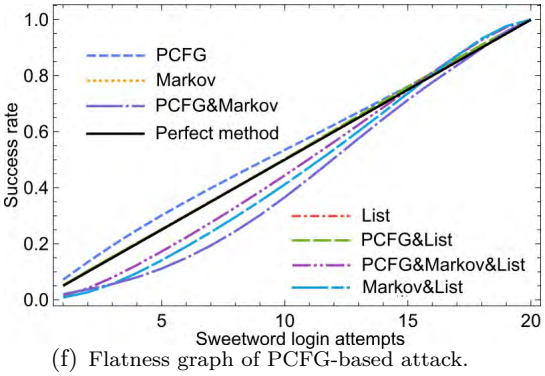
(c) Success-number graph of Markov-based attack.



(d) Flatness graph of Markov-based attack.



(e) Success-number graph of PCFG-based attack.



(f) Flatness graph of PCFG-based attack.

图 4.9　使用3种口令猜测模型攻击7种honeyword方法：攻击者类型为$\mathcal{A}_1$，训练集为Dodonew-tr，测试集为Dodonew-ts。List模型在所有指标中表现最优。

前10^3口令。根据本章第4.4节中的理论，这些口令会在$\mathcal{A}$的前1000次攻击内，因此它们是最弱的1000个口令。跟预期相同，所有的方法都不能很好的处理含有PII语义的口令和未出现在训练集中的口令。*这指出了，当面对$\mathcal{A}_2$型攻击者时，需要设计考虑PII的honeywords生成方法。*

表4.9还披露了一些超出预期的现象。对于所有模型，最弱的10^3个口令都不是流行口令，它们都不是最流行的10^4个口令[Φ]。观察第3 行和第4行，可以看出，List 模型不擅长处理频数在3到9之间的口令，而另外两个模型不擅长处理频数小于等于2的口令。这对设计混合模型(例如，$\frac{1}{3}$List+$\frac{1}{3}$Markov+$\frac{1}{3}$PCFG)有重要启示：对于一个用户提交的口令，如果$\frac{\mathrm{Pr_{PW}}(x)}{\mathrm{Pr_{HW}}(x)}$特别大，可以采用tweaking-

[Φ]本章使用节3.4.2中的方法来产生最流行的10^4个口令。

tail的方法来生成honeywords，因为这种方法对于非流行口令，能生成更平坦(flat)的honeywords。

4.5.3 一系列探索性实验

基于前述攻击理论(见节4.4.2的式4.2、4.5∼4.7)，我们接下来研究在何种口令概率模型下采用何种参数设置可最佳地抵御给定类型的攻击者。

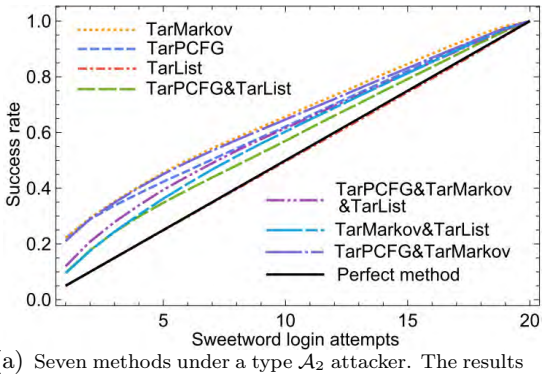
对于$\mathcal{A}_1$型攻击者。这种攻击者对应的是式4.2。由于共有3种口令模型(List、Markov和PCFG)，所以共有7种组合方法($=C_3^1+C_3^2+C_3^3$)来生成honeyword。如图4.11，记$\frac{1}{2}$Markov+$\frac{1}{2}$PCFG代表混合方法Markov&PCFG，$\frac{1}{3}$Markov+$\frac{1}{3}$PCFG+$\frac{1}{3}$List代表Markov&PCFG&List，其余类似。本章使用3种口令概率模型对这7种honeyword方法加以攻击。

结果表明，List作为生成方法，在success-number和flatness指标上，与理想方法相符。更具体的，不论使用何种模型来计算式4.2中的$\mathrm{Pr}_{\mathrm{PW}}(\cdot)$[Φ]，$\mathcal{A}$都只能分辨出约$526=10^4/19$的真实口令(见图4.9(a)，4.9(c)和4.9(e))；$\mathcal{A}$对20个sweetwords的列表进行一次攻击只能得到5%的成功率(见图4.9(b)，4.9(d)和4.9(f))。此外，List模型也是最有效的攻击模型，见图4.9(a)和4.9(b)。以上说明，对于$\mathcal{A}_1$型攻击者，应该使用List模型来生成honeywords。

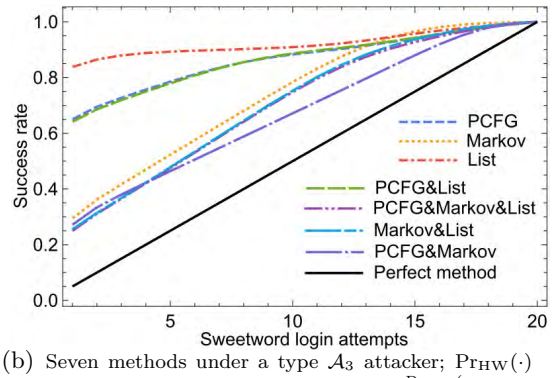
我们注意到，当$\mathcal{A}$使用List模型计算$\mathrm{Pr}_{\mathrm{PW}}(\cdot)$时，会出现很多sweetwords有很大的$\frac{\mathrm{Pr}_{\mathrm{PW}}(\cdot)}{\mathrm{Pr}_{\mathrm{HW}}(\cdot)}$值，但它们却不是真实口令。进一步仔细观察发现，这是由于“+1”平滑造成的(见节4.3.2步骤2′)：$\forall \mathrm{sw}_{i,j}\notin\mathcal{D}$，令$\Pr(\mathrm{sw}_{i,j})=\frac{1}{|\mathcal{D}|+1}$。这种平滑方法适合攻击流行口令，所以适合Juels-Rivest的方法。然而，对于本章的方法，弱sweetwords都是非流行口令(见节4.5.2)，所以平滑方法需要特别调整。对于极度罕见的口令，$\frac{1}{|\mathcal{D}|+1}$还是太大了，会造成$\frac{\mathrm{Pr}_{\mathrm{PW}}(\cdot)}{\mathrm{Pr}_{\mathrm{HW}}(\cdot)}$特别大，这会造成假阳性。因此，本章为$\mathcal{A}$提供了一个改进的平滑方法：如果$\mathrm{sw}_{i,j}\notin\mathcal{D}\wedge\frac{\mathrm{Pr}_{\mathrm{PW}}(\mathrm{sw}_{i,j})}{\mathrm{Pr}_{\mathrm{HW}}(\mathrm{sw}_{i,j})}>1$，就令$\frac{\mathrm{Pr}_{\mathrm{PW}}(\mathrm{sw}_{i,j})}{\mathrm{Pr}_{\mathrm{HW}}(\mathrm{sw}_{i,j})}=1$。这样就可以排出这些假阳性。

$\mathcal{A}_2$型攻击者。与$\mathcal{A}_1$型攻击者相比，$\mathcal{A}_2$可以进一步利用用户的PII，此时对应式4.5。因此，为安全起见，对应的生成方法需要能利用用户的PII来生成honeywords。这促使本章采用TarList方法来设计honeywords生成方法，来抵御$\mathcal{A}_2$型攻击者。这是因为，TarList方法很好的继承了List方法的优点(见前文)，并且能进一步处理用户的PII。与预期相同，图4.10(a)显示，最优的攻击者对于20个sweetwords一次猜测只有5%的成功率。并且，当$\mathcal{T}_2$=10^4时，攻击者$\mathcal{A}_2$只分辨出了531个真实口令，与理想方法的526($=10^4/19$)接近。由于空间限制，本章之后只给出$\mathcal{A}$的success-number(10^4次登录失败时)，不再给出整个图形。

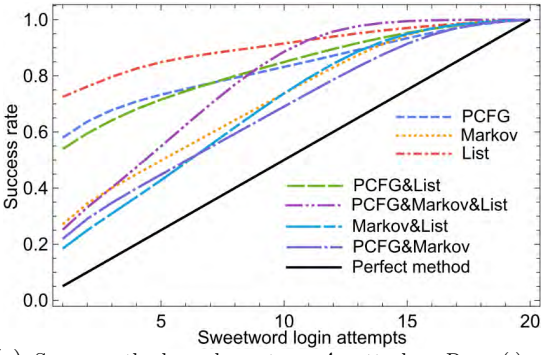
[Φ]注意，根据Kerchoffs原则，系统的honeywords生成方法应是公开信息。为保证效率，攻击者$\mathcal{A}$应使用系统的honeywords生成方法来计算$\mathrm{Pr}_{\mathrm{HW}}(\cdot)$。



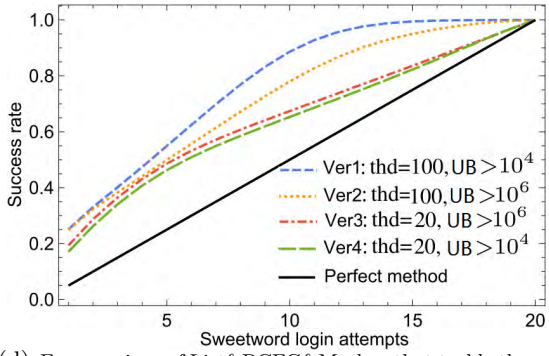
(a) Seven methods under a type $\mathcal{A}_2$ attacker. The results are desirable and the experiment succeeds.



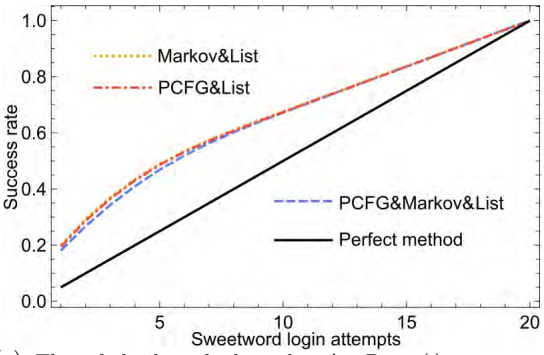
(b) Seven methods under a type $\mathcal{A}_3$ attacker; $\Pr_{\mathrm{HW}}(\cdot)$ uses an external training set, with no $\frac{\Pr_{\mathrm{PW}}(\cdot)}{\Pr_{\mathrm{HW}}(\cdot)}$ >thd smooth.



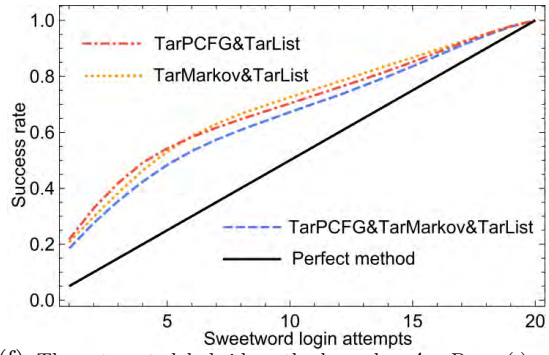
(c) Seven methods under a type $\mathcal{A}_3$ attacker; $\Pr_{\mathrm{HW}}(\cdot)$ uses an internal training set, with no $\frac{\Pr_{\mathrm{PW}}(\cdot)}{\Pr_{\mathrm{HW}}(\cdot)}$ check.



(d) Four versions of List&PCFG&Markov that tackle the $\frac{\Pr_{\mathrm{PW}}(\cdot)}{\Pr_{\mathrm{HW}}(\cdot)}$>thd smooth, UB=user base. Ver4 is the best.



(e) Three hybrid methods under $\mathcal{A}_3$; $\Pr_{\mathrm{HW}}(\cdot)$ uses an internal training set and $\frac{\Pr_{\mathrm{PW}}(\cdot)}{\Pr_{\mathrm{HW}}(\cdot)}$ >20 smooth.



(f) Three targeted hybrid methods under $\mathcal{A}_4$; $\Pr_{\mathrm{HW}}(\cdot)$ uses an internal training set and $\frac{\Pr_{\mathrm{PW}}(\cdot)}{\Pr_{\mathrm{HW}}(\cdot)}$ >20 smooth.

图 4.10 针对各种类型的攻击者的一系列探索性实验。训练集为Dodonew-tr，测试集为Dodonew-ts。对于$\mathcal{A}_2$型攻击者，TarList方法最好，对于$\mathcal{A}_3$型是List&PCFG&Markov，$\mathcal{A}_4$型是TarList&TarPCFG&TarMarkov。

$\mathcal{A}_3$型攻击者。相比于$\mathcal{A}_1$型攻击者，这种攻击者$\mathcal{A}_3$可以进一步利用用户的注册顺序，这对应式4.6。知道用户的注册顺序，$\mathcal{A}_3$现在可以分辨出哪些是流行口令，哪些不是(在给定的时间点上)。正如本章节4.5.2所显示的，List-based口令模型对非流行口令的保护很脆弱。例如，如果攻击者$\mathcal{A}_3$发现某个sweetword从未出现在之前用户的sweetword列表中，那么可以确定该sweetword必定是当前用户的真实口令。因此，List模型作为honeywords生成方法，并不适合$\mathcal{A}_3$型

的攻击者。幸运的是，PCFG-based和Markov-based的口令模型对非流行口令的处理比较好，虽然它们有自己的缺陷(见节4.5.2)。这促使本章采用混合模型$\frac{1}{3}$List+$\frac{1}{3}$Markov+$\frac{1}{3}$PCFG来抵御$\mathcal{A}_3$型的攻击者。

然而，在实现这一方法时，面临一些实际困难需要解决：(1)当用户数量较少时，能否使用额外的口令集(例如其它网站泄漏的口令集)来训练hoenyword生成方法？(2)如何处理$\frac{\Pr_{\mathrm{PW}}(\mathrm{sw}_{i,j})}{\Pr_{\mathrm{HW}}(\mathrm{sw}_{i,j})} \gg 1$的sweetwords？

例如，当一个网站在其生命周期开始时(例如，只有10^4个用户时)，就想要采用honeyword系统。一般地，对于PCFG模型和Markov模型，用户数小的训练集是不充分的(见[34, 105]和第三章)。因此，一个自然的想法是使用外部的口令集进行训练。然而，实验结果显示这个方法是不安全的(见图4.10(b))。这是因为额外的口令集是静态的，而网站的口令集是动态的，随着用户数量增加，这两个集合的差异会越来越大。$\mathcal{A}$可以利用这一事实。本章建议只使用内部训练集，即网站自身用户的口令。如图4.10(c)所示，使用内部训练集效果更好，但是依然没有达到理想的安全性。

幸运的是，如节4.5.1所发现的，$\frac{\Pr_{\mathrm{PW}}(x)}{\Pr_{\mathrm{HW}}(x)}$ 值大的sweetwords x大都是非流行口令。由于tweaking-tail对于非流行口令能生成很平坦(flat)的honeywords，所以自然的想法是当碰到这种口令时，切换成tweaking-tail方法：对于用户提交的口令x，如果$\frac{\Pr_{\mathrm{PW}}(x)}{\Pr_{\mathrm{HW}}(x)}$大于阈值$\mathsf{thd}$，就使用tweaking-tail方法生成honeywords。注意，当区分攻击者发现某个用户的hoenywords是用tweaking-tail方法生成的话，那么应该使用式4.4来进行攻击。现在剩下的问题就在于如何选择合适的阈值thd。经过一系列实验，本章发现thd=20时效果最好，见图4.10(d)。

当用户U_i注册时，网站首先使用混合方法$\frac{1}{3}$List+$\frac{1}{3}$Markov+$\frac{1}{3}$PCFG(内部训练集)生成$k-1$个honeywords，然后，将口令放入训练集中，动态更新honeywords生成方法。图4.10(e)显示，解决了这些实际问题后，混合方法抵抗$\mathcal{A}_3$型攻击者，可以达到0.178-flat。当$\mathcal{T}_2$=10^4时，$\mathcal{A}$仅能分辨出736个真实口令。这些结果表明，即使面对非常强大的攻击者，本章的方法也是很有效的。

$\mathcal{A}_4$型攻击者。与$\mathcal{A}_3$型攻击者相比，这种攻击者$\mathcal{A}_4$可以进一步利用用户的PII，对应于式4.7。这促使本章设计$\frac{1}{3}$TarList+$\frac{1}{3}$TarMarkov+$\frac{1}{3}$TarPCFG方法来抵御这种攻击者，原因与$\mathcal{A}_2$型攻击者相同。与预期相符，最强大的攻击者对20个sweetwords只猜一次，只能获得18.2%的成功率(见图4.10(f))，并且当$\mathcal{T}_2$=10^4时，仅能分辨出981个真实口令。

小结。本章重组了口令攻击模型，来生成平坦的honeywords。这一方法有重要意义，未来任何攻击模型的改进都可以很容易的合并到honeyword系统中。在将口令攻击模型具体实现到honeyword系统上的过程中，遇到了多个之前未曾披露的新挑战，本章经过一系列努力将其解决。这回答了[191]中遗留的一个公开

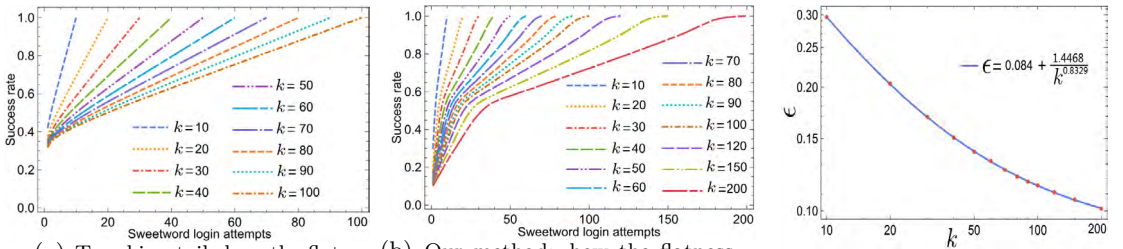
问题“潜在的攻击算法的口令模型(如[68])是否能容易的加以利用”。

4.6　评估结果

本节首先从参数k的角度，来评估所提生成方法的扩展性，然后通过两种攻击实验来评估其安全性。这两种攻击实验包括基于算法的自动化攻击实验和目前为止最为强大的攻击者—人类专家的人工攻击实验。

4.6.1　参数 k 的扩展性

显然，honeyword方法的安全性依赖于参数k，k是每个用户关联的sweetwords的个数。在Juels-Rivest的工作[191]中，k建议为20，这是因为他们相信当给定20个sweetwords时，攻击者猜中正确口令的概率为5%(即ϵ=0.05 flat)，这是可以接受的。然而，如本章所揭示的，当k=20时，对于[191] 中的主要方法，攻击者$\mathcal{A}$猜一次可以达到37.5%的成功率(见图4.4)，当$\mathcal{A}$进一步利用用户的PII时，这一数字可以达到50%(见图4.7)。



(a) Tweaking-tail: how the flatness curve varies with k

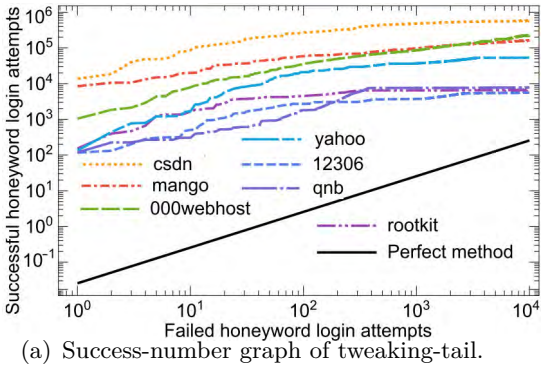
(b) Our method: how the flatness curve varies with k.

(c) Our method: ϵ follows a Zipf's law with k.

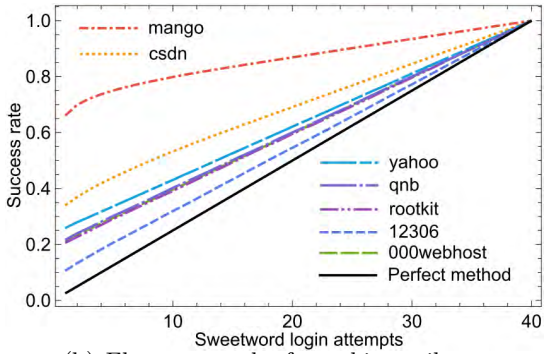
图 4.11　研究Flatness曲线随k的变化规律，训练集为Dodonew-tr，测试集为Dodonew-ts。这里仅以tweaking-tail(对应$\mathcal{A}_1$型攻击者)和$\frac{1}{3}$List+$\frac{1}{3}$Markov+$\frac{1}{3}$PCFG(对应$\mathcal{A}_3$型攻击者)为例。子图(c) 展示了ϵ (等于子图(b)中的$y|_{x=1}$)随k 的变化。

那么自然就要面对一个问题：如何设置k使得能获得理想的安全性(例如，0.05-flat)？换句话说，当k变化的时候，一个生成方法的安全性会如何变化。如图4.11(a)，0.05-flat的安全目标看上去是遥不可及的，而且存储太多sweetwords不止会增加存储空间，还会延迟登录时间。虽然本章提出的混合方法$\frac{1}{3}$List+$\frac{1}{3}$Markov+$\frac{1}{3}$PCFG(针对$\mathcal{A}_3$型攻击者)有更好的扩展性(见图4.11(b))，但当k=200 时，它也只增加到0.1-flat。值得注意的是，随着k的增加，ϵ减少得相当缓慢。有趣的是，本章发现 ϵ 和 k 有以下关系：$\epsilon = ak^c + b$，a, b, c是常数。图4.11(c)显示ϵ的下限为0.084 >0.05。这说明对于某些口令分布，0.05-flat的安全性无法实现。

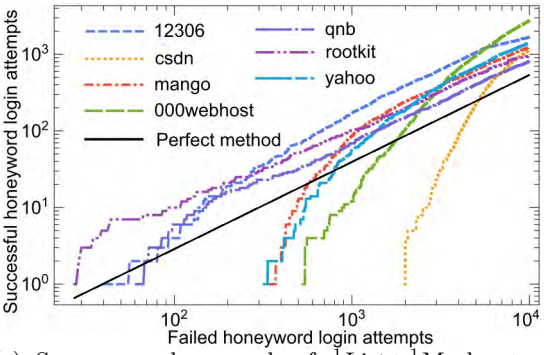
一方面，如图4.11(b)所示，本章的方法$\frac{1}{3}$ List+ $\frac{1}{3}$ Markov+ $\frac{1}{3}$ PCFG，在k=20时 可 以 使ϵ=0.20， k=30时ϵ=0.17， k=40时ϵ=0.15， k=50时ϵ=0.14，



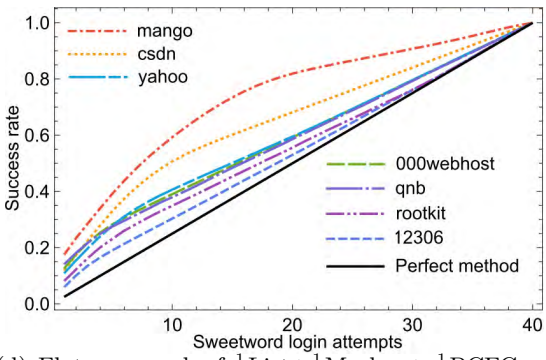
(a) Success-number graph of tweaking-tail.



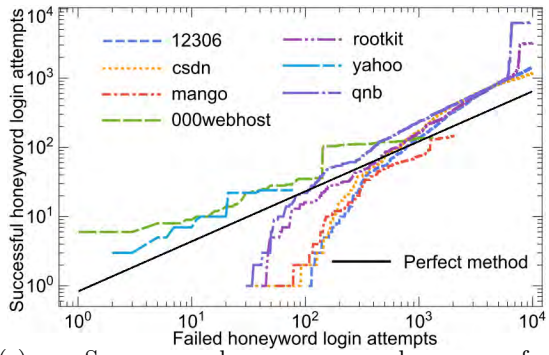
(b) Flatness graph of tweaking-tail.



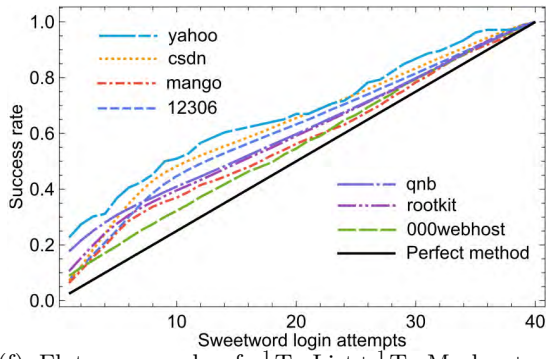
(c) Success-number graph of $\frac{1}{3}$List+$\frac{1}{3}$Markov+ $\frac{1}{3}$PCFG.



(d) Flatness graph of $\frac{1}{3}$List+$\frac{1}{3}$Markov+ $\frac{1}{3}$PCFG.



(e) Success-number graph of $\frac{1}{3}$TarList+$\frac{1}{3}$TarMarkov+ $\frac{1}{3}$TarPCFG



(f) Flatness graph of $\frac{1}{3}$TarList+$\frac{1}{3}$TarMarkov+ $\frac{1}{3}$TarPCFG

图 4.12 使用7个口令集对tweaking-tail方法[191](对应$\mathcal{A}_1$型攻击者)和本章的两种方法(分别对应$\mathcal{A}_3$型和$\mathcal{A}_4$型攻击者)进行评估。对每个口令集，50% 用于训练，另外50%用于测试。总的来说，即使面对最强的攻击者，本章的方法仍能达到预期的0.15-flat的安全性。

k=60时ϵ=0.135。本章还对其它混合方法进行了试验，结果是类似的。另一方面，采用honeyword系统的服务(网站)都是安全攸关的。因此，本章推荐k=40来确保一个可接受的安全性(即0.15-flat)，同时合理的平衡成本和效益。

4.6.2 实证实验评估结果

本章使用了7个大规模实际口令集，对Juels-Rivest的方法[191]和本章提出的方法进行对比评估。不失一般性，这里只用tweaking-tail方法进行说明。对于本章提出的方法，值得注意的是，List模型和TarList模型，分别面对$\mathcal{A}_1$型

攻击者和$\mathcal{A}_2$型攻击者时，总是能达到和理想方法一样的安全性。这是因为，最优的攻击者需要使用List模型$\Pr_{\mathcal{D}}(\cdot)$计算式4.1，并且服务器使用$\Pr_{\mathcal{D}'}(\cdot)$来生成honeywords，但是$D$只能接近$D'$，但是无法相等。图4.11和4.10 展示了这一点。因此，下面主要评估剩余的两个方法。

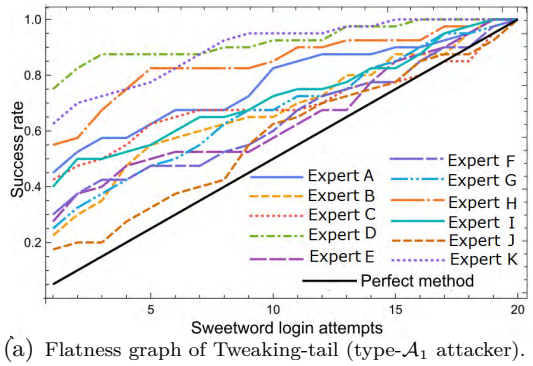
为避免过拟合，在探索性实验中所有使用过的口令集，本节均未使用。为公平起见，本章使用$\mathcal{A}_1$型攻击者攻击tweaking-tail，$\mathcal{A}_3$型攻击者攻击$\frac{1}{3}$List+$\frac{1}{3}$Markov+$\frac{1}{3}$PCFG，$\mathcal{A}_4$型攻击者攻击$\frac{1}{3}$TarList+$\frac{1}{3}$TarMarkov + $\frac{1}{3}$TarPCFG。这是因为每种方法都是针对特定的安全模型而设计的，只能在特定的模型下提供某种程度的安全性。图4.12显示对于绝大多数口令集，本章的方法均能达到0.15-flat，tweaking-tail为0.2^{+}-flat，而我们的方法均基于最强攻击者进行评估。特别地，本章方法产生的“低垂的果实”比Juels-Rivest的方法少多个量级，见图4.12(c)和4.12(e)。总之，*上述试验结果与探索性实验的结果相吻合，并且当k=40，本章提出的方法可实现0.15-flat的安全性。*

4.6.3　人工实验评估结果

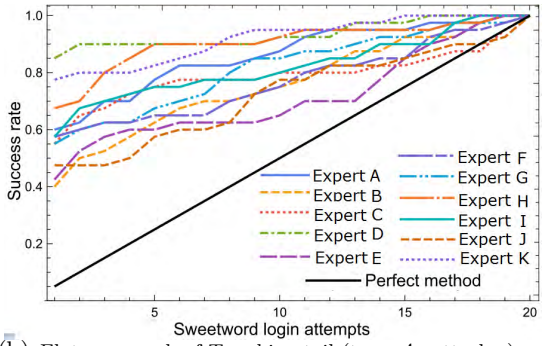
上文显示我们提出的方法可抵抗自动化的机器攻击者，但是是否能抵抗人类攻击者？这个是一个值得考虑的问题，因为口令(例如`bond007` 和`john1981`)中有很多语义信息可以容易地被人类识别，但是很难被自动化的机器攻击者识别。为回答这一问题，我们招募了11位正在参加“网络安全”课程学习的学生参与本节的评估；除此以外，他们未参与本章的其它工作。

实验设计。实验开始前，11位参与者被要求阅读honeywords相关的文献(即[191, 248, 249])，并被告知本章的4种honeywords生成方法。一个月之后，他们参与了一个相关知识测验，并且所有人都通过了该测验。他们的专业知识使他们能够理解所涉及到的12种攻击情景，知道如何利用攻击线索(生成方法弱点)将honeywords和真实口令区分开来。因此，他们是强大的攻击者。为激励他们，对于一个用户的含k个sweetwords的列表，第1次攻击就找到真实口令的参与者奖励1 元，第2 次找到的奖励$\frac{1}{2}$元，以此类推。另外，将参与者的收入排名，除了排最后两位，每人会有额外的奖励。本章的整个调研过程在两位可用性安全研究专家的指导下进行。

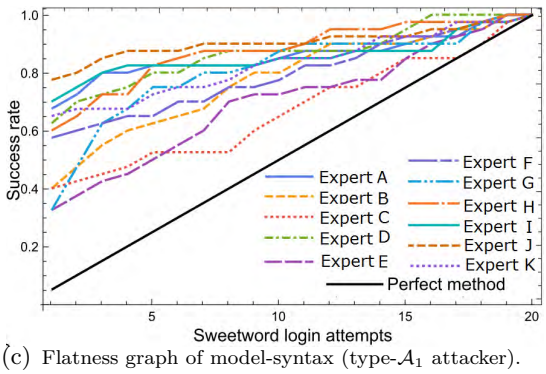
本实验选择在一个中国法定假期内进行，共持续6天。每天，上午完成一个攻击情景，下午完成另一个。对于每个情景，给予每个参与者40个sweetwords列表(对应40个目标用户)，每个列表包含20个sweetwords。因此，实验总共需要针对5280($=40\times2\times6\times11$)个目标用户生成sweetwords列表。我们最初试图在每个



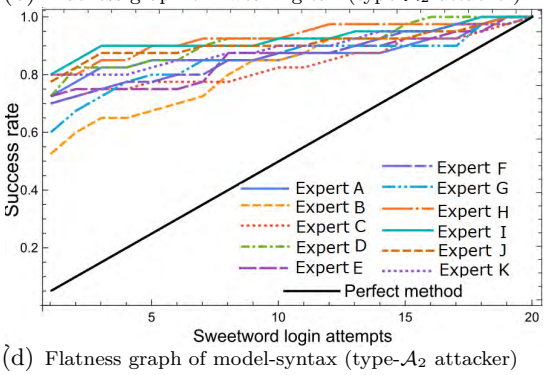
(a) Flatness graph of Tweaking-tail (type-$\mathcal{A}_1$ attacker).



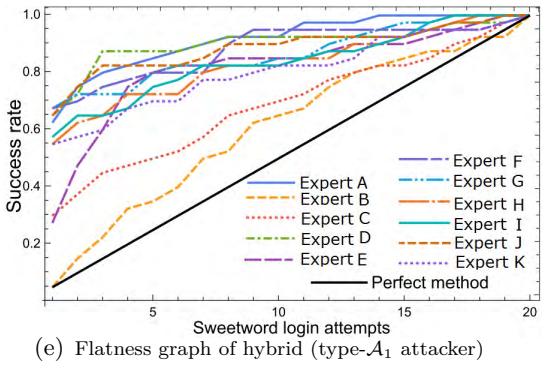
(b) Flatness graph of Tweaking-tail (type-$\mathcal{A}_2$ attacker).



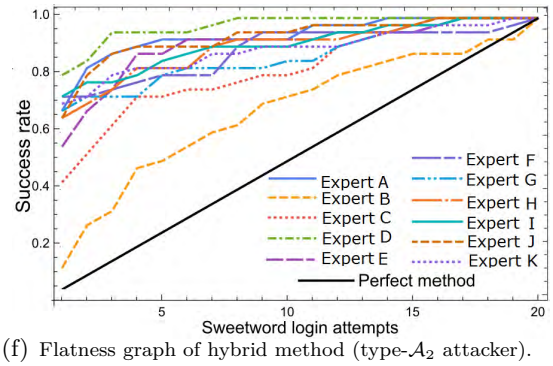
(c) Flatness graph of model-syntax (type-$\mathcal{A}_1$ attacker).



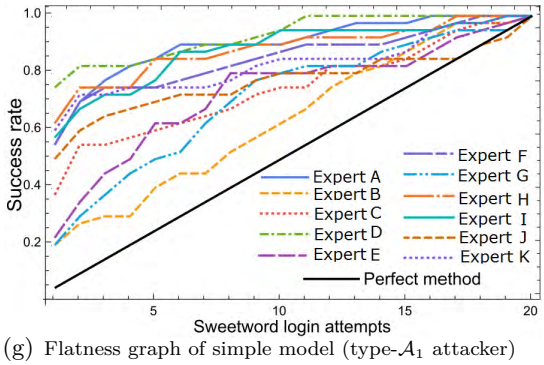
(d) Flatness graph of model-syntax (type-$\mathcal{A}_2$ attacker)



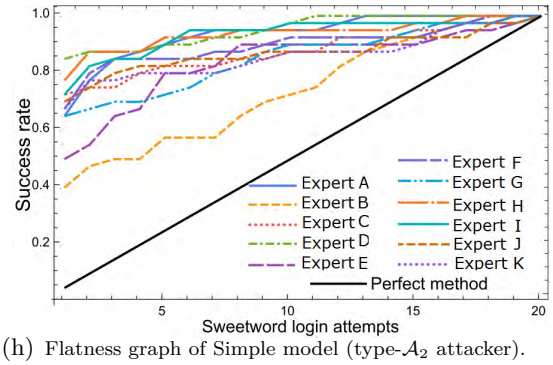
(e) Flatness graph of hybrid (type-$\mathcal{A}_1$ attacker)



(f) Flatness graph of hybrid method (type-$\mathcal{A}_2$ attacker).

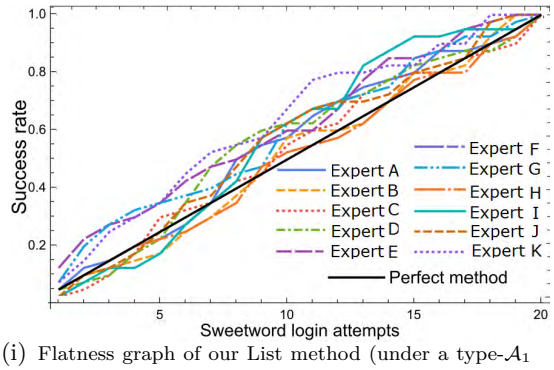


(g) Flatness graph of simple model (type-$\mathcal{A}_1$ attacker)

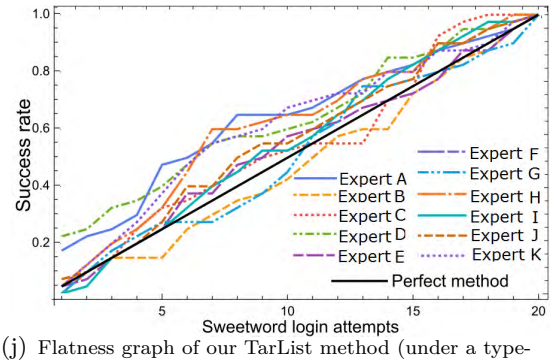


(h) Flatness graph of Simple model (type-$\mathcal{A}_2$ attacker).

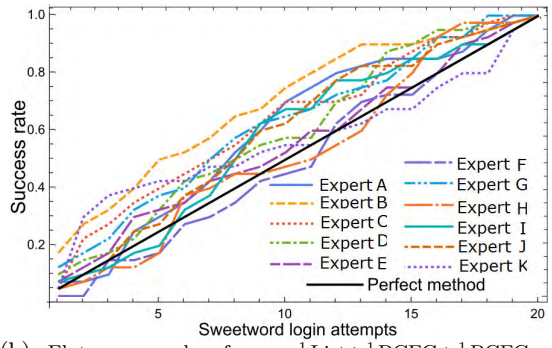
图 4.13 使用人类专家攻击者来评估Juels-Rivest的4个方法和本章提出的4 个方法。括号内表示的是攻击者的类型。对于Juels-Rivest 的方法，当已知用户的PII时，人类有特别强的分辨能力(见子图b, d, f 和 h)。但是对于本章提出的生成方法，它们没有表现出比自动化的机器攻击者有更大的优势。因此，本章提出的方法明显有更大的优势。



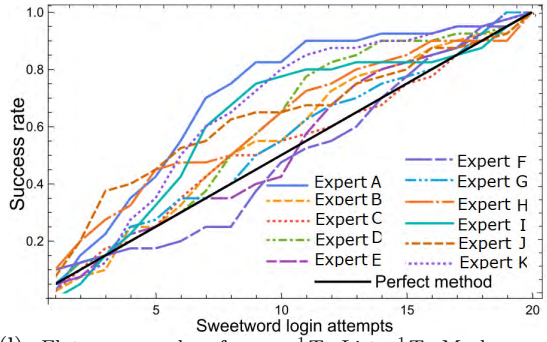
(i) Flatness graph of our List method (under a type-$\mathcal{A}_1$ attacker)



(j) Flatness graph of our TarList method (under a type-$\mathcal{A}_2$ attacker)



(k) Flatness graph of our $\frac{1}{3}$List+$\frac{1}{3}$PCFG+$\frac{1}{3}$PCFG method (under a type-$\mathcal{A}_3$ attacker)



(l) Flatness graph of our $\frac{1}{3}$TarList+$\frac{1}{3}$TarMarkov+$\frac{1}{3}$TarPCFG method (under a type-$\mathcal{A}_4$ attacker)

图 4.13　使用人类专家攻击者来评估Juels-Rivest的4个方法和本章提出的4个方法。括号内表示的是攻击者的类型。对于Juels-Rivest 的方法，当已知用户的PII时，人类有特别强的分辨能力(见子图b, d, f 和 h)。但是对于本章提出的生成方法，它们没有表现出比自动化的机器攻击者有更大的优势。因此，本章提出的方法明显有更大的优势。

sweetword列表中包含40个sweetwords，但是两位可用性安全专家建议，这样的话参与者的负担会过重。实验前，日程已经安排好，并且告知了每位参与者。在实验阶段，参与者需要到指定实验室来进行攻击实验。每人一台PC，他们可以查询10余个常见的口令集。基于伦理要求，禁止使用纸张和存储卡来记录这些口令集。每个攻击情景在honeywords生成方法或攻击者类型方面与其它情景有所不同。对于$\mathcal{A}_1$型攻击者，仅仅给予参与者40个sweetwords列表；对于$\mathcal{A}_2$型的攻击者，会额外提供用户最常见的PII(见表4.2)；对于$\mathcal{A}_3$型攻击者，sweetwords列表的顺序就是用户的注册顺序；对于$\mathcal{A}_4$，会提供$\mathcal{A}_2$型和$\mathcal{A}_3$型攻击者所有的全部信息。

实验结果。图4.13中的每个子图显示了对于给定的攻击情景，每个专家(用A到K标记)的flatness曲线。对于[191]中的4 种方法，$\mathcal{A}_1$型攻击者和$\mathcal{A}_2$型攻击者在成功率上有显著差别，这意味着对于已知PII的攻击者，这4种方法的安全性是非常脆弱的。对于本章提出的方法，这一差异较小。有趣的是，当参与者获得了PII时，他们表现的反而更差了(见图4.13(j)和4.13(l))。这可能是因

为本章提出的targeted模型能很好的利用用户的PII构造sweetwords，这些含有用户PII的sweetwords很好的迷惑了攻击者。综上所述，实验结果表明本章提出的方法能很好的抵御人类专家的攻击。

小结。基于11个大规模真实口令集，本章首次展示如何对honeywords生成方法进行科学合理的实证评估。此外，为评估Juels-Rivest的4种方法和本章新提出的4种方法能否抵御具有*语义感知能力*的人类攻击者，我们对11 位受过充分训练的honeywords专家进行了用户调查，这是为此目的而进行的首次人工实验。结果表明，本章的方法同时可以抵御自动化的机器攻击和人工的专家的攻击，克服了Juels-Rivest方法的缺陷。

4.7 本章小结

本章系统性地研究了如何*攻击*、*设计*和*评估*honeywords生成方法这一关键问题。基于大规模真实口令集的实验表明，Juels-Rivest提出的4种方法[191]不能有效的抵抗漫步和定向区分攻击者。本章首次对拥有不同能力的攻击者如何最优地攻击honeywords，进行了严格的理论建模分析，提出了一套honeywords最优攻击理论。基于这一基础性工作，我们提出了一系列基于口令概率模型的honeywords生成方法，并通过大规模实验和训练有素的人类专家攻击者显示其有效性。与此同时，本章较好地解决了[191]中遗留的4个公开问题。

考虑到当前口令文件泄漏的普遍性和其滞后发现所产生的巨大负面影响，实现主动、及时的泄露检测是非常重要而紧迫的，本章工作为有效解决这一问题迈出了重要的一步。可以预期的是，本章提出的理论和技术不仅适用于口令数据，它们很可能有助于在其它环境中生成和评估诱饵对象，尤其是当目标数据与口令数据有相似的特征时(比如服从Zipf定律的短文本)。此外，本章也遗留了一些有趣的问题有待下一步解决。比如，如何抵御针对honeyword系统的DoS攻击？当已知用户注册顺序时，如何最优地实现本章的理论攻击模型？

第五章　基于重用行为的口令强度评价方法

本章研究如何设计准确高效的口令强度评价方法。当前，几乎所有的主流网站都采用了某种口令强度评测方法，对用户提交的口令进行评测并反馈评测结果。口令强度评测方法的好坏不仅关系着众多网站的安全性，而且影响着数以亿计用户的体验。但是现有方法都假设用户在注册新帐户时，是从无到有生成口令。这不符合绝大多数用户会重用已有口令的实际。为证实这一点，我们首次针对中文用户进行了口令使用习惯的调研；进一步基于用户的口令重用行为，提出了一个基于模糊概率上下文无关文法(Fuzzy PCFG) 的口令强度评价算法fuzzyPSM；最后，基于大规模真实口令集的一系列实验显示了fuzzyPSM的有效性。本章是第三章中定向口令猜测理论的应用。

5.1　研究背景

众所周知，用户在注册新账户时倾向于选择易记的口令，但这样的口令往往是弱口令，会让他们的帐户处于危险状态。为解决这一问题，几乎每一个主流的网站都会执行某种口令生成策略，主要包括两个方面：一是设置口令构造规则，如口令长度限制和字符构成限制；二是执行口令强度评价器(Password strength meter, PSM)，用户口令必须达到一定的强度阈值，比如评测为“弱”则阻止用户使用(谷歌的PSM见图5.1)。1999 年，Adams[74]发现，当用户被迫服从不合情理、不友好的口令生成策略时，他们往往会尽力来规避这些策略。Proctor等[263]测量了在不同口令生成策略下，用户创建口令所花的时间、口令的易记性和抗攻击能力的差别，他们发现：更严格的策略虽然会使用户的口令更难破解，但是执行起来也更难，并且口令也不好记忆。后续的一些研究[263, 264]表明，如果口令生成策略设计不合理，一方面会降低用户工作效率，另一方面用户很有可能采用一些应对措施从而危害系统的安全性。

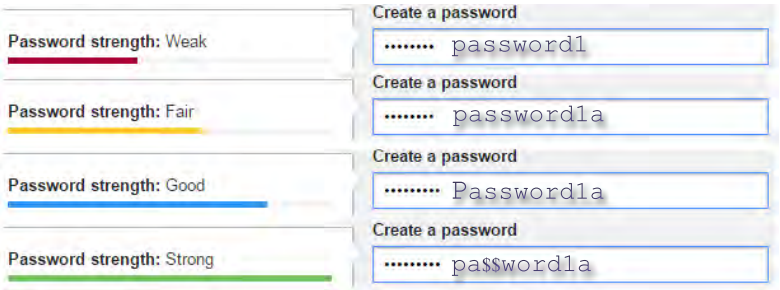


图 5.1　Gmail的口令强度评价器。

Ur等[110]2012年发现，只有对口令强度反馈准确的PSM才能显著提高用户的口令强度；2015年他们进一步指出有相当一部分用户对他们自己选择的口令的强度存在误判[88]，这主要是因为在工业界广泛使用的口令强度评价器[73, 121]常常给出不准确、误导性的反馈。除了图5.1中显示的Gmail的PSM，本文进一步做了试验：Yahoo的PSM评估“password”这个口令为“弱”，“password1”为“中等”，“password123”为“强”；World Wide Web Consortium(W3C)的PSM评估“password”为“非常弱(8%)”,“password1”为“弱(26%)”,“Password1”为“足够(46%)”，“Password123”为“强(67%)”。更多的不一致、不准确的评测案例可见表1.6。这些不准确的口令强度反馈会误导普通用户，使他们误认为在口令末尾添加数字、或将首字母大写会使原本的弱口令提升为强口令。

对口令进行准确的强度评估需要一个合适的度量标准。当前影响最为广泛的标准或许就是 NIST 电子认证指南(Electronic Authentication Guideline SP-800-63[37])中提出的一个基于熵的度量标准，记为NIST PSM。它是一个基于启发式规则的评估方法，比如既有大写又有特殊字符就给熵值+6。大多数知名网站(比如Yahoo、Gmail、Microsoft和Paypal)的PSM几乎都“完美地”[121]秉承了NIST PSM的思想。然而，Weir等[93]和Kelly 等[265]的工作表明，基于熵的PSM只能实现非常粗略的口令强度评估。

最近的研究(见[110, 265])表明，更有效的口令强度度量标准是“可猜测性”，它与口令的真实强度相关联，刻画了攻破一个账户口令所需的时间复杂度：猜测次数[104]。这种度量方式消除了对特定认证系统和潜在攻击者信息的依赖，只跟用户口令的复杂性有关。这一度量标准特别适用于给定一个口令集时比较不同猜测算法的优劣，或者给定一个猜测算法比较不同口令集的分布强弱。因此，基于“可猜测性”的度量标准在新近的研究[73, 105, 197]被广泛采用。

如表1.3所示，依据是否涉及了认证服务器，猜测攻击分为两种：在线猜测攻击与离线猜测攻击[82]；依据是否涉及了用户的个人信息，猜测攻击又可分为漫步猜测攻击与定向猜测攻击[73]。本文主要关注漫步猜测攻击和基于口令重用行为的定向猜测攻击，其它的攻击方式(如基于PII的定向猜测攻击、键盘记录[51]、肩窥[266])及其对策不在本章的讨论范围内。

虽然离线猜测攻击的问题能够通过加盐或者采用复杂的哈希(如bcrypt、PBKDF2)来缓解，但是一个攻击者的猜测效率可能比人们普遍认为的要高几个数量级，因为：(1)攻击者可以采用专用的硬件和软件来加速破解[243]；(2)明文口令集的频繁泄漏(如3200万RockYou[198]、3000万Tianya[28])，使攻击者对用户选择口令的习惯有更深层次的了解，有助于提升猜测效率。比如，基于PCFG[68]和Markov[34]模型的口令猜测算法可以使用这些泄漏的口令集作为训练集来模拟、重现真实世界中用户的行为。同样地，在线猜测攻击的威胁虽然

一定程度上通过异常登录检测[19]、限制尝试次数、锁定[267]等手段来缓解，但攻击者的能力同样可以很大程度上通过泄漏的口令集得到提升。如果进一步考虑能提供巨大计算、带宽能力的僵尸网络[268]，不管是在线还是离线攻击，攻击者可造成的破坏更是惊人的。

为刻画攻击者的猜测策略以更精确地评估口令强度，Castelluccia等[71]首次提出了一个自适应的Markov-based PSM。该PSM能够对用户选择的口令进行动态的反馈。与此同时，Houshmand和Aggarwal[70]提出了一个基于PCFG 的自适应的PSM，该PSM还能够为不满足强度要求的用户口令提供备选口令。然而，这两个PSM相互之间、与其它主流网站的PSM之间均没有进行公开对比，其有效性尚待检验。

2015年，Carnavalet和Mannan[121]调研了22个知名网站和口令管理器的PSM。结果表明，绝大多数PSM属于启发式设计；不同的PSM之间常常对同一个口令给出高度不一致的评估结果，甚至有些PSM评价为“弱”的口令会被另一些PSM标识为“强”。相对而言，这些PSM中效果稍好一些的是Dropbox网站的zxcvbn[269]与口令管理器KeePass的KeePSM[123]。

5.1.1　研究动机

我们调研了Alexa全球排名(基于网站流量)前120的网站，进一步结合文献[121]中对22个PSM的研究结果，发现学术界主流的PSM(即PCFG-based[68]和Markov-based[34])没有被工业界(如网站或口令管理器)所采用。我们认为，产生这种不理想现状的一个重要原因在于，迄今学术界先进的PSM与工业界的主流PSM还没有进行过真实数据上的实证对比。如果这些PSM孰优孰劣都无从知晓，何谈工业界应用?

更令人惊讶的是，学术界的这两个主流PSM到目前为止尚无公开明确的对比结果，两者谁优谁劣仍是一个未知数。Castelluccia等[104]认为，基于Markov模型的口令猜测算法比基于PCFG的猜测算法效率更高，因此推测Markov-based PSM会比PCFG-based PSM更准确。但是事实正好相反，我们的大规模实验表明PCFG-based PSM 整体上要优于基于Markov-based PSM，尤其当用户口令面临的主要威胁是在线猜测攻击时。

另一个不足是，来自学术界PSM都有一个隐式假定：(1)在PCFG-based PSM里，假定所有用户的口令都是通过字母段、数字段和特殊符号段的混合，全新构造的[70, 230]；(2)在Markov-based PSM里，所有用户口令都是通过Markov n-grams(状态机)全新构造的[71]。两种PSM都认为用户在注册新帐户时，口令都是从无到有全新构造的。事实并非如此。本章开展了一个中文用户口令使用习惯的调研，共有442个有效参与者：77.38%表示当注册新账户的时候，他们会重

用(或者简单地修改)旧口令；只有14.48%表示会从无到有构造一个全新的口令。我们的这两个数据与Das等[7]对224个英文用户的调研数据基本一致。因此，基于用户口令重用行为这一更合理假设来设计PSM是一个值得研究的课题。一个关键问题自然地产生了：当用户在向一个网站注册新帐户时，网站应当不知道用户的旧口令，如何基于用户口令重用行为来设计PSM？

最后，现有PSM的准确性只在英文口令集上得到了证实。更准确地说，Markov-based PSM[71]仅在Rockyou[198]与Phpbb[270]这两个数据集上实验过，PCFG-based PSM[70]仅在Rockyou数据集上实验过(只关注了可用性，没有考虑准确性)。一方面，PSM对一些敏感网站(如电子商务网站)的口令评价效果值得研究；另一方面，由于现阶段的PSM都是为评价英文用户的口令而设计的，它们是否适用非英文网站？截至目前，中国共有7.10亿网民，约占全球网民的五分之一，因此研究这些PSM在中文网站上的适用性具有重要的现实意义。

5.1.2 本章贡献

本章具有下述四项贡献：

- **用户调研**。为理解用户的口令使用习惯，我们进行了一个中文用户调研，共有442位有效参与者。该调研是迄今基于此目的进行的最大规模调研，也是首次针对中文用户口令的调研。本调研结果与此前Das等[7]针对英文用户的调研结果一致。本调研有助于理解普通用户是如何采用一系列变换规则来修改旧口令，有助于设计更符合实际假设的、更精确的PSM。
- **新的PSM**。在未知用户现有口令的情况下，为模拟用户的口令重用行为，我们使用一个相对不敏感网站泄漏的口令集A(相对较弱)作为基础字典，另一个相对敏感的口令集B作为训练字典。然后，学习用户从口令集A取口令然后修改为B中相似口令的变换规则和相应频率，得到一个模糊上下文无关文法fuzzy PCFG。在口令评价阶段，基于fuzzy PCFG对给定口令计算一个概率值作为此口令的强度。相比于传统的PCFG-based PSM，我们的PSM对口令中的L、D、S 段不敏感，故称之为fuzzyPSM。
- **系统性的评估**。我们开展了一系列的实验，使用了秩相关检验方法来测量五个主流PSM的效果：其中两个来自学术界(PCFG-based[70] 和Markov-based[71])，两个来自工业界(Zxcvbn[269]和KeePSM[123])，一个来自标准组织(即NIST PSM[37])。本章的一系列实验评估建立在11个大规模真实口令集上，共含9743万中英文用户口令，涵盖了各种主流的网络服务类型，是目前为止为评估PSM而使用的最大的数据集。
- **新的见解**。本章提出了一些新的见解，其中一些是预料之中的，另一些在预期之外。在大多数情况下，PCFG-based PSM要优于Markov-based

PSM，这与之前广泛接受的看法不同[71]。另外，如果训练集选择得当，为英文用户设计的PSM适于评测非英文用户口令。

5.2　预备知识

5.2.1　安全模型

如前文所述，口令强度评价方法的核心目标是刻画攻击者猜测给定口令所要花费的代价。由于不同攻击方式涉及的资源不同，为达到这个目的，我们必须知道攻击者会使用什么类型的攻击方式。本章主要关注漫步猜测攻击，利用个人信息的定向猜测攻击遗留为下一步工作。其它一些攻击方式(如键盘记录[51]、肩窥[266])由于与口令强度无关，因此也不在本文的讨论范围。

漫步猜测攻击者并不关心受害者是谁，他们的目标是攻破越多的帐号越好[5]。因此，攻击者的最佳策略就是优先尝试最有可能的口令(按照概率从大到小的顺序进行)。漫步猜测攻击又分为在线漫步猜测攻击和离线漫步猜测攻击：在线漫步猜测攻击里，攻击者必须与认证服务器交互来确定猜测口令的正确性；而进行离线漫步猜测攻击的前提是攻击者已经事先获得了口令的哈希值，这样攻击者不需要与认证服务器交互，只要将猜测的口令用服务器的哈希函数哈希后再与获得的口令哈希值对比，如此可以不断地进行本地猜测。如果目标账户的口令哈希值与猜测口令的哈希值相同，那么攻击者就找到了明文口令。

表1.3详细列出了不同攻击方式的差异，在线攻击和离线攻击最为重要的区别在于允许的猜测次数。在线猜测攻击由于需要与认证服务器交互，而服务器通常会采取一些安全机制(如可疑登录检测[19]、账户锁定[267]等)来防范在线猜测，因此采用在线攻击的攻击者在账户被锁定前能进行的猜测次数要远远少于离线猜测攻击的猜测次数。为了尽可能攻破更多的账户，攻击者就必须优先猜测那些最有可能的口令[5, 190]。离线猜测攻击者由于不受服务器的安全机制的限制，他可以进行任意多次的猜测，唯一的限制就是攻击者自身的计算资源和所能承受的时间。这也意味着离线猜测攻击者可能会尝试数以亿计的口令[197]：不仅会尝试流行的口令，还会尝试罕见的口令。但是为了提高猜测效率，攻击者仍然会优先猜测概率高的口令。

总而言之，漫步猜测攻击者总是会优先尝试概率更高的口令(更有可能被用户使用)，在猜测出目标口令之前所使用的猜测数(Guess number)较好地刻画了攻击者的代价。因此，这个猜测数被定义为目标口令的强度。

5.2.2　口令强度评价方法

给定一个口令pw，口令评价算法一般不会直接输出pw需要的猜测次数。一个等价的做法是，输出pw的熵(如NIST PSM[37])或者用户生成口令pw的概

率。获得口令的熵或者概率之后，就能对口令空间中所有的口令进行排序，因此pw对应的猜测次数也就随之确定了。

口令。口令的形式化定义首次由Ma等给出[34]。一个口令pw是由一串限定在字符集Σ上的字符串组成，口令的长度介于L_{min} 与L_{max} 之间。一个口令认证系统上能使用的口令集合为：

$$\Gamma = \bigcup_{l=L_{min}}^{L_{max}} \Sigma^l \,. \tag{5.1}$$

一般来说，字符集Σ可以是95个可打印ASCII字符的任意子集，但是由于这95个字符按照人类的习惯可以分为：数字、小写字母、大写字母和特殊字符，Σ一般是这四个分类的任意组合。在本文的攻击实验中，我们的字符集Σ取所有的95个可打印ASCII字符。

口令强度评价算法。一个口令强度评价算法是一个输入为字符集Σ上的口令pw，输出为用户生成pw的概率的函数$M(\cdot)$：

$$M : pw \to [0,1], \quad \forall\, pw \in \Gamma \,. \tag{5.2}$$

口令pw的强度越高，那么它的概率就越小。需注意的是，只有当函数$M(\cdot)$是一个基于概率模型的算法时，才有$\sum_{pw\in\Gamma} M(pw)$ =1。文献[70, 71]中的评价算法是基于概率模型的评价器，而zxcvbn、KeePSM和NIST PSM 不是。在现实中，一般会将口令评价算法的输出按值的大小分为几类，比如“弱”、“中”、“强”，或者“弱”、“一般”、“好”、“强”。口令评价器的可能是强制的，也可能是建议性的。前者要求一个口令pw只有在其概率$M(pw)$达到了预先定义的阈值时，才会被接受；后者仅是告知用户目前的口令强度是多少，尽管会有潜在的风险，弱口令仍然会被系统接受。

理想口令评价算法。一个概率型口令强度评价算法的终极目的就是再现目标系统的口令分布，理论上，一个理想口令评价算法应能实现的是：

$$M(pw) = p_{pw}, \quad \forall\, pw \in \Gamma \,. \tag{5.3}$$

这里p_{pw}为口令pw在分布$\mathcal{X}$中的真实概率，换句话说，理想口令评价算法能够非常理想的重现目标系统的口令分布。但是在实践中，得到一个口令的真实概率p_{pw}几乎是不可能的；幸运的是，对于那些出现频率超过4的大众口令而言，我们可以计算$p = f_{pw}/|\mathcal{DS}|$来近似逼近真实$p_{pw}$的值，$f_{pw}$ 为口令pw在数据集$\mathcal{DS}$中出现的频率，$\mathcal{DS}$是分布$\mathcal{X}$的一个样本。这就意味着我们能够从分布$\mathcal{X}$中取出非常大的样本数据集$\mathcal{DS}$，然后将该数据集按照概率p降序排列，出现频率很高的大众口令就能被认为是理想口令评价算法给出的概率，排列里口令的次序就是理想口令评价算法给出的所需猜测次数。

相应地，现实世界的口令强度评价方法(如[68]和[34])都可视为对理想口令评价算法的某种近似。我们可通过比较口令评价算法的输出来评估它们的好坏：1)对同一口令输出的概率的差异；2)对同一口令输出的猜测次数的差异。我们注意到，前者不容易有效测量，但当猜测次数是一个PSM的主要关注点是，它可以通过后者来很好地刻画。举例示之。假设有两个口令评价算法$M_1(\cdot)$与$M_2(\cdot)$，其中$M_1(\cdot)$是理想口令评价算法，$M_2(\cdot)$的输出如下:

$$M_2(pw_i) = \begin{cases} M_1(pw_1) + (M_1(pw_2) - M_1(pw_3))/2, \\ M_1(pw_2) - (M_1(pw_2) - M_1(pw_3))/2, \\ M_1(pw_i), \text{ for all } i \geq 3 \text{ and } pw_i \in \Gamma. \end{cases} \tag{5.4}$$

如果$M_1(pw_1) \geq M_1(pw_2) \geq M_1(pw_3) \geq \cdots$，那么就有$M_2(pw_1) \geq M_2(pw_2) \geq M_2(pw_3) \geq \cdots$。可以确认，$M_2(\cdot)$不仅是一个基于概率模型的评价算法，而且与理想评价算法$M_1(\cdot)$一样能够为同一口令输出相同的猜测次数。这也就相当于放松了相应要求从而产生了更为实用的理想评价算法。

实用的理想口令评价算法。如果一个口令评价算法能够与理想口令评价算法为同一口令输出相同的猜测次数，那么我们就说这个口令评价算法是实用的理想口令评价算法。形式化定义如下: 对于函数$M(\cdot)$，只要$p_{pw_i} \geq p_{pw_j}$，就有

$$M(pw_i) \geq M(pw_j), \forall\, pw_i, pw_j \in \Gamma\,. \tag{5.5}$$

从猜测次数这个角度上看，$M(\cdot)$即为一个理想口令评价器。

5.2.3　两个秩相关检验指标

根据以上定义，评测一个口令评价算法的准确性可以通过测量该口令评价算法与理想口令评价算法之间的“距离”。这个“距离”的测量可以由如下方法实现：对同一口令集，计算待测口令评价算法给出的序数与理想口令评价算法给的序数的秩相关系数。Spearman 相关性系数ρ[271]和Kendall相关性系数τ[272]是两个被广泛使用的秩相关检验指标。ρ和τ均是$[-1,1]$区间内的实数，1表示完全相同，0表示完全不相关，-1表示完全相反。

5.3　用户调研

为更准确地评估用户口令的强度，我们首先需要更好地理解用户的行为，比如：当注册新账户时，用户是如何创建新口令的？为此我们在问卷星上进行了一个用户调研，问卷地址：`http://www.sojump.com/jq/6443561.aspx`。为更好地与英文用户作比较，我们设计的问题都尽可能与Das等针对英文用户的

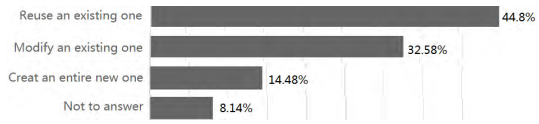


图 5.2 现在假设需要你为新的163邮箱账户设置密码，该密码你需要记住，并且邮箱会经常性的使用。你会采用哪项做法？

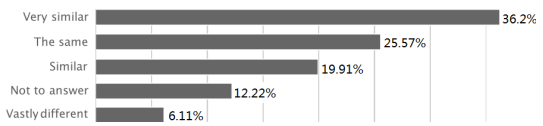


图 5.3 你将如何叙述你的新密码与以前其它网站密码的相似性呢？

调研[197]类似。

经过严格筛选，我们的问卷最后共有442个有效参与者。我们的调研设置了一个场景：假设你(参与者)正在创建一个以后会频繁使用的邮箱账户，以下相关问题中你的选择是什么？问卷的第一个问题是帮助我们理解用户的新口令与旧口令到底有多大关系。图5.2表明77.38%的参与者会重用或者简单修改一个旧口令，这个值与Das 等调研的77%[7]惊人的相似，虽然重用旧口令的中文用户比英文用户要少6.2%。

如果用户是修改一个旧口令，那么旧口令与新口令之间有多相似呢？由此，我们设计了相关的问题,如图5.3所示,我们可以看到“非常相似”和“一样”共占据了61.77%的比例，“相似”也有将近20%的比例。这表明，当创建一个将会频繁使用的新邮箱账户时，80%以上的用户将会向系统提交一个与其现有口令至少是相似的口令。这与之前人们普遍认为的用户会使用一个全新的口令不同。大部分用户的新账户口令会与旧账户口令相似，同时由于口令集不断地泄漏，攻击者有非常大的概率已经获得了用户的旧口令，因此新口令的强度一定与旧口令的相似度有关——旧口令做了多大的修改得到了新口令，这个想法促成了本文提出的fuzzyPCFG。

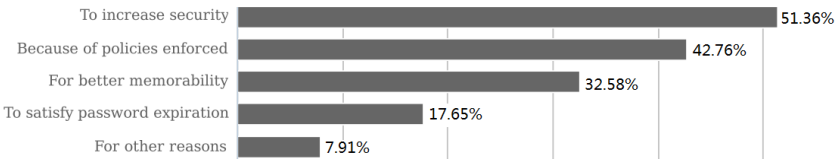


图 5.4 如果你确实是将另一网站的密码修改后在新网站使用，那么修改已有密码的主要原因是什么？(多选)。

图5.4解释了为什么用户要修改旧口令而不是直接使用它；51%为了提高新口令的安全强度，他们相信对口令做一些简单的修改能够极大的提高口令的强度——“我可是在它后面添加了‘!’来让它更安全!”[88]。作为对照，英文用户只有24%的比例会这么做。一个合理的解释可能是：相对于英文网站，更多知名度高的中文网站泄漏了用户口令，因此中文用户更关心他们口令的安全性。

用户修改他们口令的规则总结在图5.5，“连接”(比如:在旧口令首或尾添加数字或者字母)占了大多数，其次是“字母大写”和“字符跳换”。有些用户还会使用“子串移动”、“倒序”、“添加网站特有信息”等规则。我们的这些发现

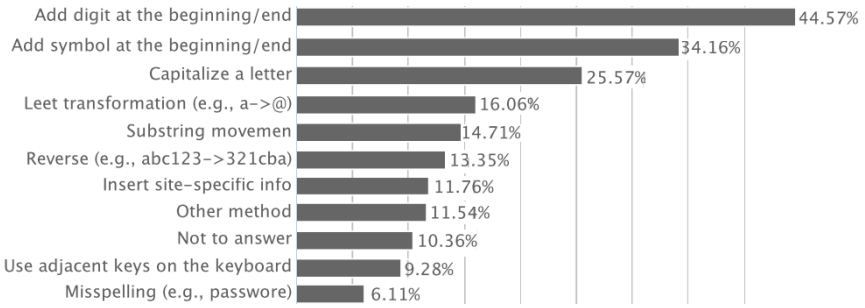


图 5.5 你是否曾经用过下面一些方法，来将一个网站的密码修改后在另一个网站使用呢？请选择您曾使用的方法(多选)。

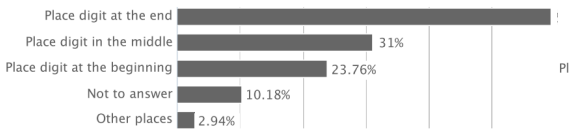


图 5.6 当你的密码需要至少包含一个数字时，你经常会把数字放在什么地方？(多选)。

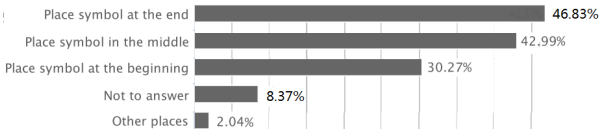


图 5.7 当你的密码需要至少包含一个特殊字符时，你经常会把特殊字符放在哪里？(多选)。

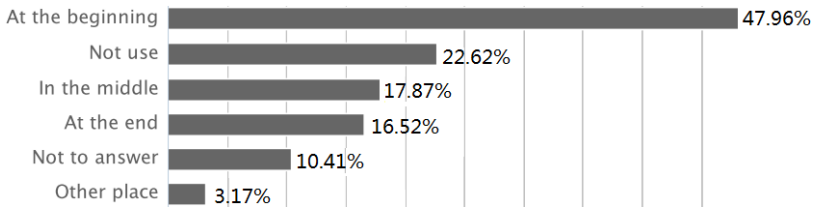


图 5.8 你的密码中使用大写字母吗？如果是，通常在哪里使用？(多选)

也与其英文调研一致。

接下来的几个问题集中询问了参与者他们会将这些变换规则运用在口令的哪些位置上。图5.6 和图5.7 告诉我们用户更乐于在旧口令末尾添加数字或字符，其次是中间、首部。图5.8 告诉我们47.96%的用户喜欢将旧口令的首字母大写，这也与英文的调研一致(44%)。

本文同时还分析了参与者的人口统计学信息。参与者中三分之二是男性，80.55%的参与者年龄介于18~34岁之间，15.67%的参与者年龄大于35岁。80.55%的参与者有学士学位或者本科在读，43.44%的参与者有硕士学位或者研究生在读。虽然参与者学历较高，但是相较而言，我们的调研参与者也比之前的英文调研更具多样性，他们93% 的调研参与者年龄介于18~34岁之间，92%至少有学士学位。相比普通大众，我们的参与者更年轻、受过更好地教育，更有安全意识和科学常识—这意味着，我们很可能低估了口令重用问题。

伦理问题和不足。据我们所知，本章的调研是目前第一个针对最大的互联网用户群体——中文用户。参与者匿名回答问题，并且我们展示的是综合集成之后的信息，避免了单个参与者信息的泄露。虽然我们的参与者比之前调

研(见[7, 80, 90])的参与者在数量上和多样性上更有优势，但是我们的调研也存在口令安全调研固有的不足：用户可能提供虚假信息，无法保证信息的有效性。尽管如此，我们的调研结果在很多方面(如口令重用的频率、插入/删除字符的位置)与英文用户的调研、与真实口令集的统计数据是一致的。

5.4 新方法fuzzyPSM

本节描述一个新的基于PCFG口令模型的口令强度评价算法fuzzyPSM。为解释为什么选择PCFG类的口令模型而不是Markov类的模型作为算法基石，我们首先对现有的PSM进行对比，理解它们各自的优劣。

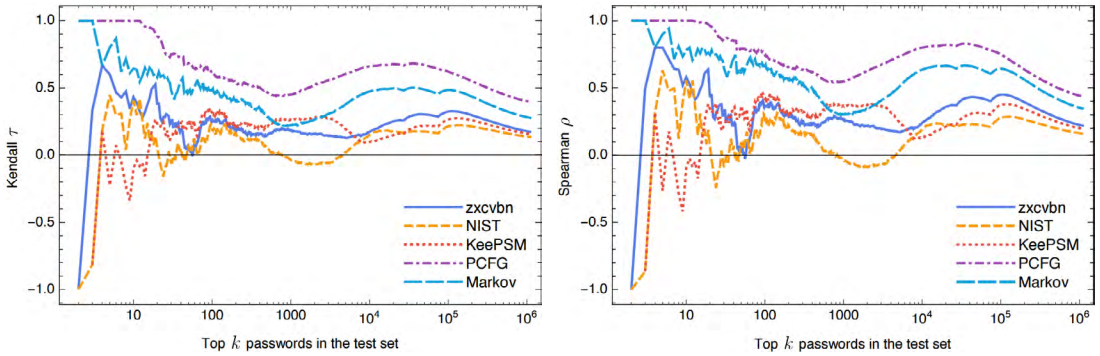
5.4.1 对比现有PSM

尽我们所知，学术界的两个主流 PSM (即PCFG-based PSM和Markov-based PSM)尚未曾有公开的对比分析。因为Markov-based口令猜测模型的破解效率优于PCFG-based猜测模型[34, 76, 197]，Castelluccia等[71]于是推测并声称Markov-based PSM要优于PCFG-based PSM。但是，我们的实验结果与Castelluccia 等的推测正好相反。

作为例证，我们使用这两个评价算法评估泄漏自`csdn.net`的643万口令。一般来说，基于机器学习的口令评价算法实验结果的优劣取决于两个因素：算法本身和选取的训练集。为了排除训练集的影响，与文献[71, 93]一样，我们随机将CSDN 的口令集分为 4 部分，我们使用第一部分来训练，第四部分做测试。由于这两个口令评价算法都没有开源实现，本章按照论文[70]实现了基于PCFG的口令评价算法，按照论文[71]实现了基于Markov模型的口令评价算法，并按文献[34]应用了最新的改进方法：基于PCFG的口令评价算法的字母段—本章从训练集里面学习提取，不使用字典；基于Markov模型的评价算法——使用了Backoff 平滑方法。为了确保正确实现了现有口令评价算法，我们首先使用所实现的算法生成猜测集，然后重复已经公开发表的攻击实验——我们的攻击结果与现有的结果[34, 76]一致。这表明我们无误地实现了现有算法。

接着，我们使用zxcvbn[269]评测测试集中的口令，然后将口令按该PSM给出的概率排序，记序列为s_1；计算s_1与理想PSM得到的序列s_0间的Kendall秩相关系数τ和Spearman秩相关系数ρ，以此测量zxcvbn与理想PSM间的“距离”。类似地，我们测量NIST PSM[37]、KeePSM[123]、PCFG-based和Markov-based PSM与理想PSM间的“距离”。Kendall与Spearman的秩相关系数的值域为$[-1,1]$，越接近1表明距离越小。图5.9(a) 给出了各现实PSM与理想PSM间的Kendall系数τ，图5.9(b) 给出了各现实PSM与理想PSM间的Kendall系数ρ。

Kendall与Spearman秩相关系数检验方法的结果都表明：(1)来自工业界的



(a) A comparison of real-world PSMs with an ideal PSM in terms of Kendall τ

(b) A comparison of real-world PSMs with an ideal PSM in terms of Spearman ρ

图 5.9 现有PSM与理想PSM之间的对比。

三个基于启发式规则的PSM，比来自学术界的两个基于机器学习技术的PSM表现要差；(2)基于PCFG的评价算法优于基于Markov模型的评价算法。前者不难理解，也正是普遍所预期的；但后者令人有些吃惊，它否定了人们的一个普遍预期—因为基于Markov模型的口令猜测算法要比基于PCFG的猜测算法破解效率更高[34, 104]，故基于Markov 模型的口令评价算法要优于基于PCFG的评价算法[71]。这一发现有些反直觉。

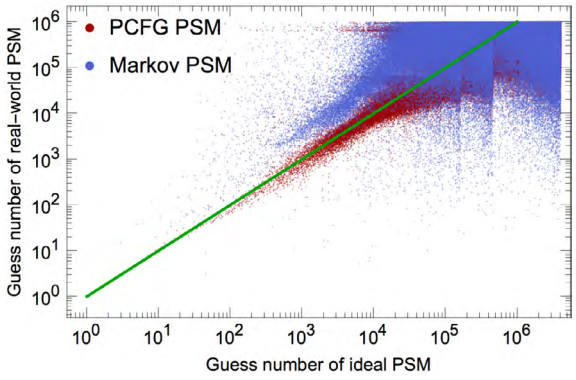


图 5.10 PCFG-based PSM与Markov-based PSM输出的对比。

5.4.2 理解PCFG的优越性

为解释这一看似矛盾的现象，本节从微观层面研究PCFG-based PSM与Markov-based PSM 输出结果的差异。图5.10中，每个红色的点代表一个来自测试集的口令，它的横纵坐标(x_i, y_i)中横坐标x_i代表理想口令评价算法认为该口令需要的猜测次数，纵坐标y_i代表基于PCFG的评价算法认为该口令需要的猜测次数。同样的，蓝色点的横坐标x_i与红色点的横坐标意义相同，纵坐标y_i表示基于Markov 模型的评价算法认为该口令需要的猜测次数。易知，一个PSM越好，那么它的数据点与图中绿线的重合度就应越高。图5.10表明，基

表 5.1 对一些弱口令不同评价器给出的序

Typical password	Training set	Ideal PSM	PCFG PSM	Markov PSM	fuzzyPSM
123qwe	10182	8527	771719	324	4764
123qwe123qwe	2778	3105	194671	236965	4505
password123	2315	2522	4219	6124	2397
Password123	27097	10170	29341	31009	23438
password	18	21	20	35	46
p@ssw0rd	76	259	337982	290	274

注：深灰色表明与理想评价器最接近，浅灰色次之

表 5.2 基于PCFG和Markov口令猜测模型的“无效”猜测数

	top 10^2		top 10^4		top 10^6		top 10^7	
	PCFG	Markov	PCFG	Markov	PCFG	Markov	PCFG	Markov
un-usable guesses	1	12	160	4509	372883	359401	9153626	8587614

于PCFG的口令评价算法比基于Markov模型的口令评价算法更优越，特别是在弱口令评价方面。

为了得到更令人信服的结论,我们在表5.1中总结了对一些典型弱口令不同评价算法给出的猜测次数。6个弱口令在原测试集里按照出现频率排的序(猜测次数)都比较小。同样，理想口令评价算法给出的序(该口令在训练集里按照出现频率排的序)与原测试集的序基本一致。从表中也可以得出，基于PCFG的评价算法要比基于Markov模型的评价算法要好，但是总的来说，本章提出的fuzzyPSM算法提供了最好的口令强度评估。

表5.2总结了不同攻击算法(基于PCFG或Markov模型)产生的“无效”猜测的数目。这里的“无效”是指被算法模型生成，但是没有在测试集里面出现过的口令。这种“无效”的猜测数目多少能够显示一个攻击模型的好坏：更少的“无效”猜测意味着更高效的攻击模型。我们能够看到，当猜测数比较小的时候，基于PCFG的猜测算法产生了更少的“无效”猜测，但是当猜测数大于10^6之后，情况发生了反转，基于Markov 模型的猜测算法产生更少的“无效”猜测。这也就解释了为什么基于Markov的猜测算法要优于基于PCFG的猜测算法—离线猜测时的允许的猜测数一般比较巨大。因此，我们能得出结论：基于PCFG的概率模型更适于评估口令强度，基于Markov概率模型的算法更适合于离线猜测口令。

5.4.3 fuzzyPSM算法

虽然基于PCFG的口令评价算法在5个现有的评价算法里面表现最优，但是它还是有它的缺陷。在PCFG模型里，一个口令被认为是由三种“段”组成：数字段、字母段、特殊字符段，每个段里分别由且仅由数字、字母、特殊字符组成。比如，口令“p@ssw0rd”由“p”，“@”，“ssw”，“0”和“rd”5个段组成，基础结构为$L_1S_1L_3D_1L_2$；口令“Password123”由“Password”、“123”2个段组成，基础结构为L_8D_3；口令“123qwe123qwe”由“123”，“qwe”，“123”，“qwe”4个段组成,基础结构为$D_3L_3D_3L_3$。口令“p@ssw0rd”的概率计算如

表 5.3 基础结构和各字段的概率

LHS	**RHS**	Probability
$\mathcal{S}\rightarrow$	B_6	0.4
$\mathcal{S}\rightarrow$	B_8	0.4
$\mathcal{S}\rightarrow$	$\mathsf{B}_8\mathsf{B}_3$	0.1
$\mathcal{S}\rightarrow$	$\mathsf{B}_6\mathsf{B}_6$	0.1
$\mathsf{B}_1\rightarrow$	`1`	0.8
$\mathsf{B}_1\rightarrow$	`a`	0.2
$\mathsf{B}_3\rightarrow$	`123`	1
$\mathsf{B}_6\rightarrow$	`123456`	0.7
$\mathsf{B}_6\rightarrow$	`123qwe`	0.2
$\mathsf{B}_6\rightarrow$	`dragon`	0.1
$\mathsf{B}_8\rightarrow$	`password`	0.85
$\mathsf{B}_8\rightarrow$	`p@ssword`	0.15

表 5.4 字符跳变表及其概率

	$L_1: a\leftrightarrow @$		$L_2: s\leftrightarrow \$$		$L_3: o\leftrightarrow 0$		$L_4: i\leftrightarrow 1$		$L_5: e\leftrightarrow 3$		$L_6: t\leftrightarrow 7$	
	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No
Probability	0.006	0.994	0.005	0.995	0.004	0.996	0.003	0.997	0.002	0.998	0.001	0.999

下：$P($“`p@ssw0rd`”$)= P(\mathsf{L}_1\mathsf{S}_1\mathsf{L}_3\mathsf{D}_1\mathsf{L}_2)\cdot P(\mathsf{L}_1\rightarrow \mathtt{p})\cdot P(\mathsf{S}_1\rightarrow \mathtt{@})\cdot P(\mathsf{L}_3\rightarrow \mathtt{ssw})\cdot P(\mathsf{S}_1\rightarrow \mathtt{0})\cdot P(\mathsf{L}_2\rightarrow \mathtt{rd})$。在Weir的原始论文[68]里，基础结构，数字段和特殊字符段的概率是从训练集里面学习出来的，字母段的概率是由外部的字典提供的。Ma[34]建议字母段的概率也应该从训练集里训练学习出来，这个改进可显著提高PCFG算法的效率，已被广泛使用[34, 105]，因此本章采用了这个方法。

通过上述PCFG-based PSM评估口令的例子，不难发现，该PSM是假定用户是从无到有全新地构造一个新口令——通过混合数字段、字母段和特殊字符段；“混合”其实是使用了“连接”这种变换规则。PCFG-based PSM的做法与两个基本的事实相矛盾：第一，相当大比例的用户不是全新构建口令，而是通过修改旧口令形成新口令；第二，只考虑了“连接”这一变换规则，其它的变换规则没有考虑，如“字母大写”、“字符跳变”。虽然有些PSM(如[37, 71, 123, 230, 269])考虑了第二个事实，但是尚无PSM考虑第一个事实。

那么如何才能将用户的旧口令考虑进去呢？本章使用了一个叫做“基础字典”的训练集$\mathcal{B}$，这个训练集是从较为不敏感的网站上泄漏出来的口令集——该数据集里的口令强度比较弱；同时使用另一个较为敏感的网站上泄漏的口令集作为训练字典$\mathcal{T}$，该字典里的口令较强。本章做了一个合理的假设：基础字典是用户的旧口令，训练字典是用户的新口令。从基础字典到训练字典，本章的算法可以学习到有哪些规则被应用到了新口令上，训练过程自动地产生了本章称之为模糊上下文无关文法的变换规则。

在训练阶段,我们在PCFG原有三个段的基础之上引入了新的段“基段”，用符号B表示，基段其实就是基础字典里的口令，与其它段一样，需要记录所有类型的段的概率。基础字典中一个长度为n的口令，我们将其标注为B_n。训练字典中的每一个口令都与基础字典中的口令进行最长前缀匹配，然后再采用变换规则“连接”、“字母大写”、“字符跳变”，最后剩下无法匹配的部分使用原始的PCFG方法匹配。

本章的上下文无关文法$G=(V,\Sigma,\mathcal{S},R)$正式定义如下：

1) $V=\{\mathcal{S},B_1,B_2,\cdots,B_k,\text{ Capitalize},\ L_1,L_2,\cdots,L_6\}$是一个有限集，$k$不超过允许的最长口令的长度；$L_1$~$L_6$为6种字符跳变，如表5.4所示。

2) $\Sigma=\mathcal{BS}=\{b_1,b_2,\cdots,b_N,\ \texttt{Yes},\ \texttt{No}\}$是一个有限集，且95个可打印ASCII字符也在$\Sigma$里。

3) $\mathcal{S}\in V$是起始字符。

4) R是一个形如$A\to\alpha$的生成规则集合，$A\in V$并且$\alpha\in V\cup\Sigma$(见表5.3和5.4)。

举个例子，训练字典中的一个口令“password123”的基础结构是B_{11}，因为基础字典$\mathcal{B}$中有该口令，那么该口令就只由一个段构成，不再涉及到其它的变换；口令“Password123”未包含于基础字典$\mathcal{B}$，因此就涉及到一个“字母大写”(p到P)的转换；口令“p@ssw0rd”不包含于基础字典,但是“p@ssword”包含于基础字典，因此涉及到一个“字符跳变”(o到0)。口令“123qwe123qwe”的基础结构是$\mathsf{B}_6\mathsf{B}_6$，因为通过最长前缀匹配可以匹配到B_6，之后就是一个“连接”操作。如果碰到一个口令“tyxdqd123”，该口令无法通过基础字典解析，那么就通过传统的PCFG来解析，其基础结构是$\mathsf{B}_6\mathsf{B}_3$。

在评估阶段，fuzzyPSM算法可以对每一个输入的口令输出其概率。举个例子，如果“连接”操作的概率为0.1，输入口令是“p@ssw0rd1”，其基础结构为$\mathsf{B}_8\mathsf{B}_1$，$\mathsf{B}_8\to$ p@ssword的概率为0.15，$\mathsf{L}_3\to$ Yes的概率为$P($“p@ssw0rd1”$)=P(\mathcal{S}\to\mathsf{B}_8\mathsf{B}_1)\cdot P(\mathsf{B}_8\to\texttt{p@ssword})\cdot P(\mathsf{B}_1\to\texttt{1})\cdot P(\text{Capitalize}\to\texttt{No})\cdot P(L_1\to\texttt{No})\cdot P(L_2\to\texttt{No})\cdot P(L_2\to\texttt{No})\cdot P(L_3\to\texttt{Yes})\ P(\text{Capitalize}\to\texttt{No})\cdot P(L_4\to\texttt{No})=0.1*0.15*0.8*0.97*0.994*0.995*0.995*0.004*0.97*0.997=0.00004431$。

随着时间的推移，不断有新用户注册，系统的口令分布也会随之发生变化，此时需要修正fuzzyPSM算法的训练结果，这一过程称为更新阶段。比如，如果新口令“p@ssw0rd123qwe”被系统接受了，那么“连接”操作的概率、基础结构B_6和B_8的概率、规则$\mathsf{L}_3\to$ Yes的概率也需要相应更新。更新操作使得fuzzy PCFG概率模型与真实口令分布更加接近，使其具有自适应性。

概率上下文无关文法(PCFG)就是每个文法规则都附带一个转换概率的CFG，且具有相同左部的文法规则的概率加和为1。通过上述描述易知，本章提出的文法也是一个概率上下文无关文法。当使用大规模真实口令集训练时，训练字典里80%以上的口令都是$\mathcal{S}\to B_m$这样的形式，而不是$\mathcal{S}\to B_mB_n$或更复杂的形式。与之相对的是，在PCFG-based PSM的基础结构表中超过50%都是$\mathsf{L}_m\mathsf{D}_n$或更复杂的形式。换句话说，我们的PCFG文法保持着与旧口令的密切联系，对数字段、字母段或者特殊字符段不敏感。因此，我们的PCFG文法是“模糊”的，称为Fuzzy PCFG。

表 5.5 11个真实口令集的基本信息

Dataset	Web service	Location	Language	Unique PWs	Total PWs
Tianya	Social forum	China	Chinese	12,898,437	30,901,241
Dodonew	Gaming&E-commerce	China	Chinese	10,135,260	16,258,891
CSDN	Programmer forum	China	Chinese	4,037,605	6,428,277
Zhenai	Dating site	China	Chinese	3,521,764	5,260,229
Weibo	Social forum	China	Chinese	2,828,618	4,730,662
Rockyou	Social forum	USA	English	14,326,970	32,581,870
Battlefield	Game site	USA	English	417,453	542,386
Yahoo	Web portal	USA	English	342,510	442,834
Phpbb	Programmer forum	USA	English	184,341	255,373
Singles.org	Christian dating	USA	English	12,233	16,248
Faithwriters	Christian writing	USA	English	8,346	9,708

表 5.6 不同口令集的字符组成信息

Dataset	^[a-z]+$	[a-z]	^[A-Z]+$	**[A-Z]**	^[A-Za-z]+$	[a-zA-Z]	^[0-9]+$	[0-9]	**Symbol only**	**^[a-zA-Z0-9]+$**	^[0-9]+[a-z]+$	^[a-zA-Z]+[0-9]$+	^[0-9]+[a-zA-Z]+$	**^[a-z]+1$**
Tianya	9.91%	34.63%	0.18%	2.96%	10.24%	35.66%	**63.77%**	89.49%	0.03%	98.08%	4.12%	15.73%	4.39%	0.12%
Dodonew	10.30%	66.32%	0.30%	3.67%	10.92%	69.05%	**30.76%**	88.52%	0.02%	98.33%	7.55%	45.74%	7.93%	1.40%
CSDN	11.64%	51.39%	0.47%	4.57%	12.35%	54.33%	**45.01%**	87.10%	0.03%	96.31%	5.88%	28.45%	6.46%	0.24%
Zhenai	6.41%	37.33%	0.24%	3.40%	6.74%	39.54%	**59.52%**	92.87%	0.02%	95.79%	5.24%	21.09%	5.69%	0.08%
Weibo	19.07%	44.77%	0.64%	3.66%	20.55%	46.71%	**53.04%**	78.78%	0.06%	97.79%	2.80%	18.74%	2.91%	1.24%
Rockyou	**41.71%**	80.58%	1.50%	5.94%	**44.07%**	83.89%	15.94%	54.04%	0.02%	96.25%	2.54%	30.18%	2.75%	4.55%
Battlefield	**32.11%**	89.71%	0.29%	9.60%	**34.01%**	90.69%	9.23%	65.49%	0.01%	98.06%	3.05%	39.58%	3.39%	5.08%
Yahoo	**33.09%**	92.83%	0.40%	8.51%	**34.64%**	94.06%	5.89%	64.74%	0.00%	97.15%	5.31%	41.85%	5.64%	4.80%
Phpbb	**50.18%**	86.18%	0.74%	7.70%	**53.07%**	87.83%	12.06%	46.14%	0.03%	98.34%	2.03%	20.94%	2.35%	2.33%
Singles.org	**60.21%**	87.84%	1.92%	8.14%	**65.82%**	90.42%	9.58%	34.06%	0.00%	99.79%	1.77%	19.68%	1.92%	2.73%
Faithwriters	**54.37%**	91.74%	1.16%	8.84%	**58.98%**	93.64%	6.36%	40.88%	0.00%	99.52%	2.37%	25.45%	2.73%	4.13%

需要指出的是，Fuzzy PCFG也存在一些不足。首先，它只训练学习三种口令修改方式：“连接”、“字母大写”和“字符跳变”，其它的修改操作，比如“子串移动”、“反转”都遗留为下一步工作；其次，对于“字母大写”这个操作，目前只考虑了基础口令的首字母大写操作；最后，对于“字符跳变”操作我们只考虑了6种最流行的跳变方式。尽管如此，fuzzyPSM算法展示了基于口令重用行为构造PSM的可能性，并且那些暂未考虑的修改操作加入到fuzzyPSM中只是技术细节上的问题。总之，fuzzyPSM能够更好地建模用户的口令构造行为，下面将显示它优于现有PSM，尤其适于评估弱口令。

5.4.4 实验设置和结果

本节首先给出我们的实验设置，包括使用的11个真实口令集，以及训练和测试场景配置；然后，我们采用节5.2.3中介绍的秩相关检验方法来评测fuzzyPSM与其它5个主流PSM的优劣。

数据集描述。为使我们的实验更充分，评测结果具有普适性，本章使用了11 个大规模真实口令集，包含9740万口令。如表5.5所示，大多数口令集都是因黑客团体攻陷网站，然后公布于网络；口令集涉及的网络服务类型多种多样，包含社交论坛、游戏网站、约会网站、网站门户和电子商务等；5个口令集来自中文用户，6个来自英文用户—世界上最大的两个网民群体。

口令特征。表5.6显示了各口令集的字符组成信息。值得注意的是：大部分英文用户的口令只由小写字母组成的，大部分中文用户的口令只由数字组成。为理解影响口令分布的因素，我们进一步描绘了不同口令集间共同口令的比例。图5.11说明，不同口令集之间相同口令所占的比例都低于60%，如果这两个口令集来自不同语言，那么共同口令比例会更低。

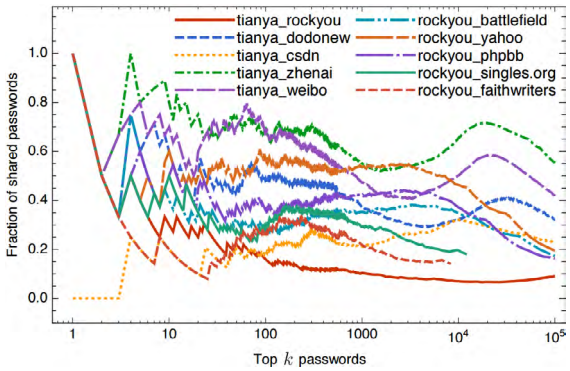


图 5.11 两个网站之间共同口令的比例。

表 5.7 训练和测试场景设置

Scenario	Base* dictionary	Training set(s)		Testing set, (each set is tested individually)
		Ideal case	Real-world	
English	Rockyou	1/4 testing set	Phpbb, 1/4 testing set	Yahoo,Battlefield,Singles,Faithwriters
Chinese	Tianya	1/4 testing set	Weibo, 1/4 testing set	Dodonew, CSDN, Zhenai
Cross-lan. #1	Rockyou	–	Phpbb, 1/4 Dodonew	Dodonew
Cross-lan. #2	Tianya	–	Weibo, 1/4 Yahoo	Yahoo

*基础字典只被fuzzyPCFG使用，“Cross-lan”指“Cross-language”。

数据集总结。我们的结果表明不同网站的口令分布差异巨大，这可能是由各种综合因素(语言、服务类型、文化、口令生成策略)导致的。这会给PSM训练集的选取带来困难：很难得到与目标网站口令分布完全相同的训练集，现实的做法是使训练集尽可能的接近目标网站。此外，因为不断有新用户注册，口令的分布必然会随时间变化，这就需要一个自适应的PSM。

实验设定。如前所述，口令的分布是受许多因素影响的，在这些因素里语言的影响最大。为了规避这些因素的影响，本章按照语言将训练集和测试集分成两部分，分别使用Rockyou和Tianya作为基础字典。为了有更好地评估效果，本章考虑了两种场景：理想场景和现实场景。在理想场景中，将测试集随机分为四等份，随机选其中2份分别用以训练和测试。这样做能够消除训练集选取不当带来的不利影响，评估结果只依赖于评价器本身。

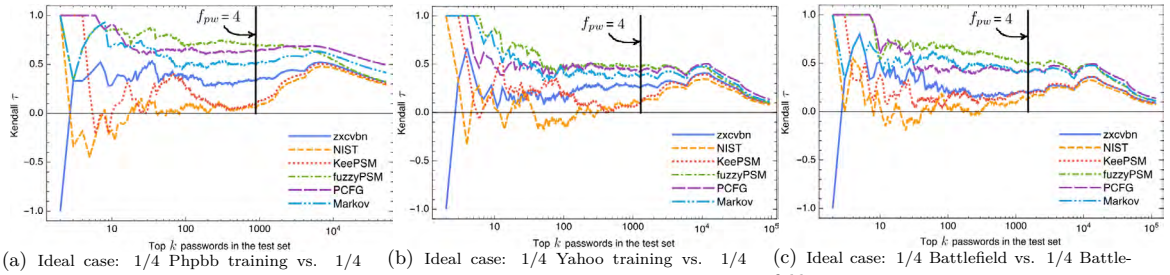
然而，理想场景在实现中并不实用，因为我们不能保证训练集与测试集来自同一口令分布。更现实的办法是，网站上线后，使用与目标网站有相同语言、类似服务、口令创建策略的泄漏口令集作为训练集来训练口令评价算法。之后，当一个用户提交的口令被系统接受，那么就将其插入训练集中，使口令评价算法得到动态的更新。不失一般性，这种场景可以如下模拟：口令评价算法使用测试集的四分之一和提供相似服务的网站泄漏的口令集作为训练集，然后用来评估测试集剩下的所有口令。我们进一步研究了PSM在语言不同的训练集与测试集下表现如何。表5.7总结了这些实验设置。Rockyou和Tianya的口令集在各组中强度最弱，因此被选为基础字典；Phpbb和Weibo的口令集具有中等强度，因此被选为训练集。

实验结果。本节比较了fuzzyPSM与现有5个主流PSM的性能：两个来自学术界(PCFG-based PSM和Markov-based PSM)，两个来自工业界(zxcvbn和KeePSM)，最后一个是来自标准组织(NIST PSM)。表5.7总结了模拟理想场景和现实场景的实验设置，各自分别产生9个(见图5.12(a)∼5.12(i))和7个(见图5.12(j)∼5.12(p))独立实验。基于Kendall系数和Spearman系数的评价结果基本类似，由于篇幅限制，这里只给出基于Kendall系数的评价结果。

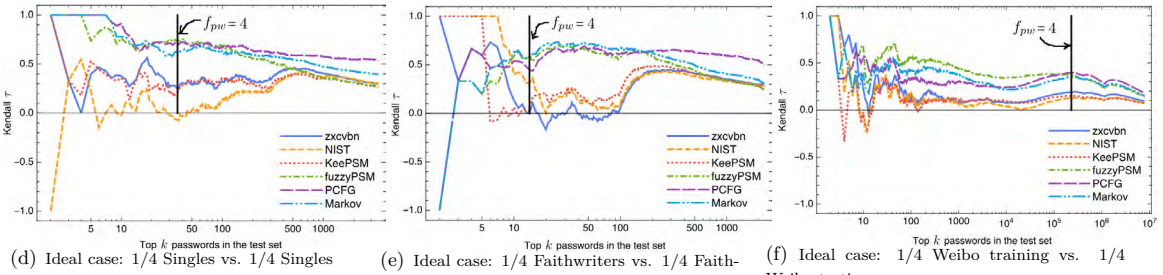
如图5.12所示，在大多数情况下，fuzzyPSM在评测$f_{pw} \geq 4$的口令时表现最佳，尤其是针对现实场景。各子图的横坐标表示测试集中频率排名前k的口令，纵坐标是Kendall系数τ，黑竖线表示左边是出现$f_{pw} \geq 4$的口令，右边是$f_{pw} \leq 4$的口令。根据节5.2中的理论，对于那些$f_{pw} \geq 4$ 的流行口令而言，我们可以通过计算$f_{pw}/|\mathcal{DS}|$来近似真实概率p_{pw}的值[5]，f_{pw}为口令pw在数据集$\mathcal{DS}$中的经验概率，$\mathcal{DS}$是分布$\mathcal{X}$的一个样本。对于那些$f_{pw} \leq 4$的口令而言，理想评价器不再有效，因此对于这些口令的秩相关系数比较没有实际意义。所以我们只关心该黑竖线左边的曲线。可以看到，fuzzyPSM无论在理想场景还是真实场景下，它与理想评价器的的“距离”最近，这表明fuzzyPSM表现最佳。但是在跨语言场景下，fuzzyPSM并没有比其它的算法更好，这显示了基础字典和训练集合理选取的重要性。

5.5　本章小结

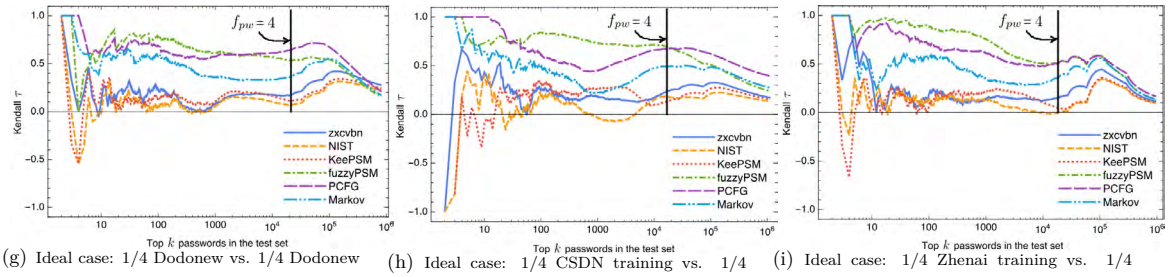
基于对用户口令构造真实行为的深刻理解，本章提出了一个简单而有效的口令强度评价方法fuzzyPSM。通过引入上下文无关文法，fuzzyPSM成功建模了用户的口令重用行为，这为模拟现实世界中用户构造口令的行为迈出了重要一步。一系列实验表明，fuzzyPSM 优于现有的主流PSM，尤其是在评测弱口令方面。最近的一些研究指出(见[7, 190])，准确评价并阻止弱口令、防御在线猜测应当是PSM的主要目标，这进一步显示fuzzyPSM将有广阔的应用前景。



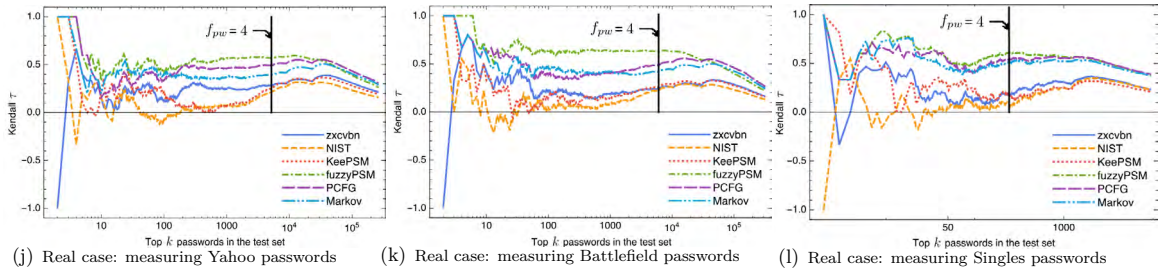
(a) Ideal case: 1/4 Phpbb training vs. 1/4 Phpbb testing (b) Ideal case: 1/4 Yahoo training vs. 1/4 Yahoo testing (c) Ideal case: 1/4 Battlefield vs. 1/4 Battlefield



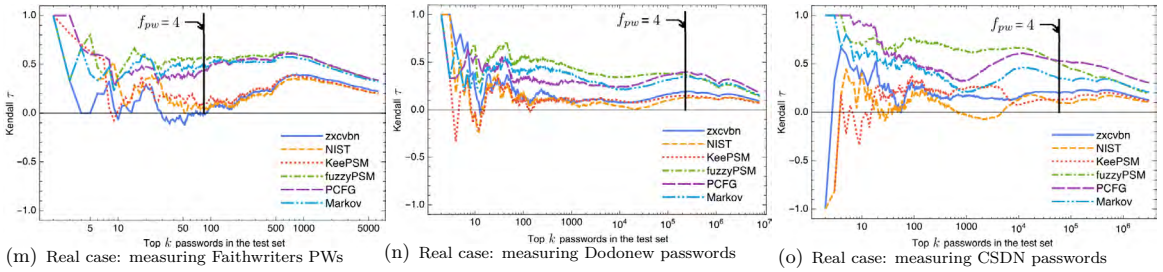
(d) Ideal case: 1/4 Singles vs. 1/4 Singles (e) Ideal case: 1/4 Faithwriters vs. 1/4 Faithwriters (f) Ideal case: 1/4 Weibo training vs. 1/4 Weibo testing



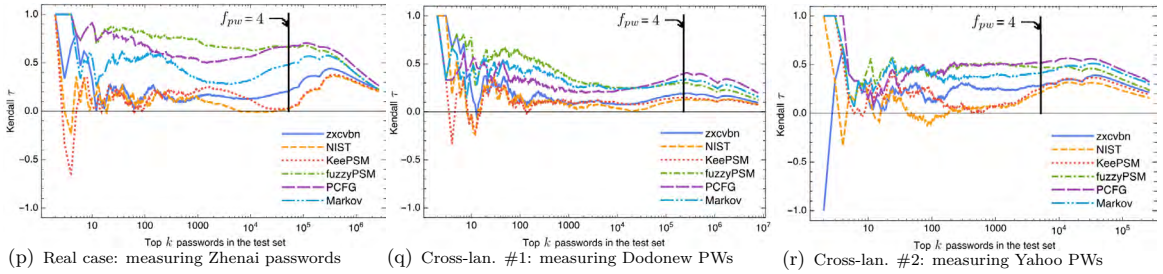
(g) Ideal case: 1/4 Dodonew vs. 1/4 Dodonew (h) Ideal case: 1/4 CSDN training vs. 1/4 CSDN testing (i) Ideal case: 1/4 Zhenai training vs. 1/4 Zhenai testing



(j) Real case: measuring Yahoo passwords (k) Real case: measuring Battlefield passwords (l) Real case: measuring Singles passwords



(m) Real case: measuring Faithwriters PWs (n) Real case: measuring Dodonew passwords (o) Real case: measuring CSDN passwords



(p) Real case: measuring Zhenai passwords (q) Cross-lan. #1: measuring Dodonew PWs (r) Cross-lan. #2: measuring Yahoo PWs

图 5.12 理想场景实验结果(a)～(i)；现实场景试验结果(j)～(p)；跨语言实验结果(q)(r)。

第六章　口令认证协议中匿名性的本质计算复杂度

在接下来三章中，我们研究传输阶段的口令安全—口令认证协议。由于基于口令的单因子认证协议已有充分的研究，本章主要关注如何设计和评估适于高安全需求环境的基于“口令+智能卡”的双因子认证协议。这种双因子协议应用最为广泛，它也是构造“口令+智能卡+生物特征”多因子协议的基石。

本章以传感网环境下的双因子协议为例，研究口令认证协议中实现匿名性的本质计算复杂度。用户隐私当前受到越来越多的关注，尤其是在资源受限应用环境下，如何确保用户匿名性(User anonymity)是设计双因子身份认证协议的核心难点之一。在身份认证协议中，用户匿名性主要包括两方面的内涵：一是身份保护(Identity protection)，二是用户行为不可追踪性(User untraceability)。实现用户匿名性主要有两条技术路线，一是基于公钥密码技术，二是仅采用对称密码技术。由于传感器节点和智能卡资源受限，计算需求相对较小的对称密码技术似乎显示出了更大的优势。学者们尝试各种各样巧妙的办法，仅基于对称密码技术来设计匿名身份认证协议，提出了数以百计的此类协议。遗憾的是，在非抗窜扰智能卡假设下，这些协议都陆续被指出无法实现所宣称的用户行为不可追踪性。目前，大批学者仍未放弃尝试。

基于我们过去分析和研究200多个匿名双因子身份认证协议的经验，发现那些存在匿名性问题的协议往往仅采用了对称密码技术，而实现了匿名性的协议往往采用了公钥密码技术。因此，我们猜测：在非抗窜扰智能卡假设下，采用公钥密码技术是实现匿名性的必要条件。基于Halevi–Krawczyk (1999)和Impagliazzo–Rudich (1989)这两个经典结果，我们成功证实了这一猜测。这意味着，无论设计如何“巧妙”，证明多么“严格”，所有那些仅采用对称密码技术的尝试都是徒劳。这为匿名双因子协议设计提供了一条基本原则。我们进一步显示，该原则具有普适性：不仅适于传感网环境下的双因子协议，而且适于各种具体应用环境下的双因子协议；不仅适于双因子协议，而且适于单因子口令协议、多因子协议。本章将为第七、八章奠定理论基础。

6.1　研究背景

随着微机电系统和无线网络技术的快速发展，无线传感器网络(Wireless sensor networks, WSNs)显示出巨大的发展潜力，当前已被广泛应用于军事、民用等领域(如战场环境监测、实时交通控制、工业过程控制和智能家居)。为增强网络扩展性，支持节点移动性，降低管理复杂性，延长网络生命周期，大规

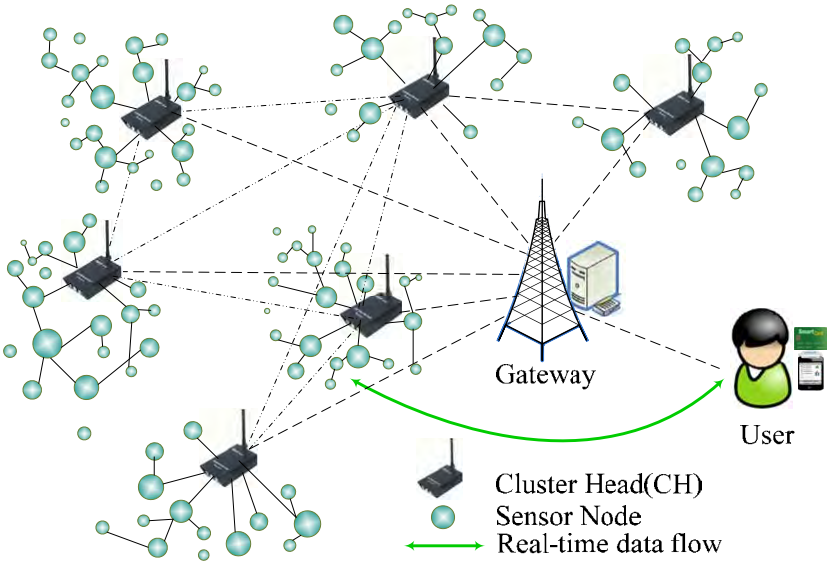


图 6.1 WSNs环境下实时数据的按需直接访问。

模 WSNs (见[273–275])一般采用了分层体系结构。故本章主要研究分层的大规模 WSNs 。在很多重要应用中，外部用户需要实时访问传感器节点上的数据，但由于大规模 WSNs 中的传感器节点和基站间的通信链路较长，若数据查询操作完全由基站执行，则不能确保执行效率，系统的扩展性也会受到影响[276, 277]。

为实现外部用户对传感器节点的实时数据的*按需直接访问*，如何保障敏感数据免受窃听、恶意修改、未经授权的访问等威胁成为随之产生的一个关键问题。一种自然的解决办法是采用身份认证机制，在用户被授予访问权限之前，由传感器节点验证用户身份。一方面，WSNs环境下实时数据应用多为高安全需求应用；另一方面，基于“口令+智能卡”的认证机制(简称双因子认证[278])具有简单高效、灵活方便、安全可靠等特点。因此，双因子认证成为保障WSNs环境下实时数据访问的最有前景的安全机制之一。

过去二十多年的研究表明，设计一个安全高效的针对普通环境的双因子认证协议并非易事[279–281]。设计安全高效的适于WSNs环境的双因子认证协议将更为困难，因为协议设计者面临着一个两难的境地：在资源极其受限的传感器节点上实现强认证、用户隐私保护和其它密码服务。一方面，由于传感器节点和智能卡的运算能力、存储空间、电池能量受限，应避免使用一些计算复杂度较高的密码操作/原语(如双线性对运算、群签名)。另一方面，WSNs通常部署在无人值守的环境中，且执行敏感任务(如健康监测、战场监测)。除传统的安全威胁外，WSNs环境下的认证协议存在更大的攻击面，面临更多后果严重的攻击(如危胁生命安全)。因此，WSNs 下的双因子认证协议应能防范各类已知的攻击(如仿冒攻击、重放攻击、离线口令猜测攻击)和 WSNs 环境下特有的攻击，

如网关旁路攻击和节点捕获攻击[282]。

除了安全性，在 WSNs 环境下用户隐私问题也备受关注。比如，当前 GEOSS[283] 和 NOPP[284] 等多个计划都在利用大规模 WSNs 来实现自适应监测地球-海洋-大气系统。从个体用户到大学、政府到商业公司都对这些WSNs收集的敏感数据感兴趣，例如GEOSS[283]涉及61个国家，NOPP[284]涉及 DARPA、国土安全部等。由于利益诉求不同，这些用户之间很难做到彼此相互信任，此时用户行为对外界来说就是敏感信息。比如，通过用户行为可以反推出用户兴趣。因此，迫切需要保护用户的数据访问隐私，比如某一用户访问了哪些敏感数据，对哪一类数据感兴趣，从哪一个节点获得数据等等。总之，为防止用户隐私信息(如个人偏好、登录历史、位置、身体状况、个人数据[171, 172])泄露和滥用，设计匿名双因子认证协议的需求日益增长。

需要指出的是，匿名性(Anonymity)是针对外部用户，而不是服务器而言。因计费、授权或审计的需要，以及为检测、记录和删除恶意用户，用户真实身份对所访问的服务器而言应是公开的[285]。此外，匿名性的含义依赖于具体的应用环境，不同的应用环境意味着相异的含义[286, 287]，例如用户名保护、用户行为不可追踪性、k-匿名和混合匿名性等。读者可参见文献[288]了解更多信息。对于匿名双因子认证协议，匿名性的基本含义是用户身份保护(Identity protection)，即攻击者无法从协议交互信息中获取目标用户的身份标识。

相较而言，更高层次的匿名性不仅要求实现用户名保护，而且要求攻击者无法分辨出两个会话是否源自同一用户，即实现用户不可追踪性(User untraceability)[289, 290]。实现用户不可追踪性的协议能够阻止攻击者关联同一用户的多次会话，防止追踪用户的位置、访问历史等。因此，绝大多数匿名认证协议(如[177, 282, 289, 291–296]) 中追求的匿名性即为用户不可追踪性。本章除特别说明，“用户匿名性”即代表用户不可追踪性。

6.1.1　相关工作

2009年，Das等[282]首次提出了能实现用户、网关节点和传感器节点三方相互认证的基于“口令+智能卡”双因子认证协议。该协议仅采用轻量的对称密码操作，便于授权用户通过移动设备(如掌上电脑、智能手机和笔记本电脑)访问WSNs。不幸的是，在协议提出不久，Nyang-Lee[297]、Khan-Alghath[292]、Chen 等[298]分别发现它易遭受离线口令猜测攻击、传感器节点捕获攻击、仿冒攻击和内部人员攻击。于是，一系列改进协议被提出，如[291, 292, 297–299]。然而，如文献[300–303]所指出的，这5个改进协议仍然无法实现宣称的安全性。

2011年，Fan等[304]指出上述的WSNs环境下双因子认证协议忽视了对用户

隐私的保护，于是提出了一个匿名双因子认证协议。该协议仅采用了轻量级的Hash函数和异或操作，适于资源受限的传感器网络环境。Fan等宣称所提出的协议能有效抵抗各类攻击并能实现用户匿名性。如本章将要指出的，尽管该协议有许多优良特性，并给出了形式化安全证明，但仍未实现用户不可追踪性，且对几种常见的攻击是脆弱的。

与此同时，Kumar等[293]也注意到以往的协议不是存在安全缺陷就是缺少基本功能，如双向验证、用户匿名性。因此，Kumar等设计了一个新的 WSNs 环境下实现隐私保护的双因子身份认证协议，宣称他们的协议可以抵抗所有已知攻击，并支持各种功能。2012年，Jiang等[294]分析发现， Kumar等协议既无法抵抗离线口令猜测攻击，也不能确保用户不可追踪性，于是提出了一个改进协议。然而，不难发现，Jiang等[294]协议易遭受异步攻击(见节7.4.2)，任一不诚实(被捕获)传感器节点可以通过改变最后一条协议信息使用户的智能卡失效，除非用户重新注册。

2012年，Das等[301]首次在匿名双因子协议中提出“节点动态增加”的概念，以实现在现有WSNs中动态增加新的传感器节点。为降低计算量，该协议仅采用轻量级对称密码操作，如Hash函数和对称加密算法。随后，Turkanovic和Holbl[305] 发现该协议无法实现所宣称的用户匿名性，于是提出了一个改进的匿名协议。不幸的是，Li等[306]指出Das等协议[301] 和Turkanovic-Holbl的协议[305]都易泄露服务器的长期会话密钥。

2013年，Xue等[295]提出一个基于临时凭证的轻量级双向认证和密钥协商协议。如上述协议[282, 291, 293, 294, 304]，该协议也企图仅采用Hash函数和异或操作实现用户匿名性。与现有协议相比，Xue等协议支持更多功能、具备更高的安全性，比如支持口令本地更新、抗网关旁路攻击等。尽管Xue等协议有许多可取之处，本章分析发现，该协议仍不能实现用户匿名性这一重要目标。

总的来说，当前所提出的众多协议之所以无法实现用户匿名性，很大程度上归因于缺少密码学中的不可能性结果(Impossibility results)：缺少对匿名性本质计算复杂度的研究。现有少数不可能性结果主要涉及的是一些密码原语和形式化证明方法[307]。例如，文献[308–310]给出了一些关于如何通过黑盒规约从一个原语构建一个新的原语的不可能性结果。在协议层面，不可能性结果更加罕见。1989 年， Impagliazzo 和 Rudich[311]提出仅基于对称密码技术不可能实现安全的密钥交换协议。1999 年， Halevi 和 Krawczyk[56]证明仅采用对称密码技术的PAKE协议无法抗离线口令猜测攻击。2000 年 Park 等[312]指出，为实现前向安全性，公钥加密技术不可或缺。2006 年，Nguyen[313]研究了PAKE协议和其它密码原语之间的关系，发现PAKE协议和公钥加密在黑盒归约中是不可比拟的。2011年， Madeline 和 Peeter[307]证实仅采用对称密码技术不足以构建永久性的

消息识别协议(Perennial message recognition protocols)。

在上述的理论研究中，Halevi-Krawczyk的工作[56]可能与本章最为接近，但文献[56]针对的是PAKE协议，且关注点主要是安全性。2009 年 Zeng 等[314] 指出，假设攻击者是合法的内部用户，可以从自己的智能卡中提取秘密参数，则文献[315–317]中提出的双因子认证协议无法实现用户匿名性。据我们所知，文献[314]是双因子认证文献中唯一关注用户匿名性的工作。遗憾的是，Zeng等将此前众多协议无法实现匿名性的原因归咎于表层的协议“结构性错误(structural mistakes)”，而忽略了更深层次的原因。尽我们所知，“在基于口令的双因子认证协议中，匿名性的本质计算复杂度是什么”这一关键理论问题鲜有研究。

6.1.2　研究动机

近期，大量研究致力于设计 WSNs 环境下实用的匿名双因子认证协议(如[181, 292–295, 297, 303, 304, 318–323])，然而到目前为止，这些协议没有一个能够实现用户匿名性这一“珍贵”属性。这些协议的一个共性是企图仅采用对称密码原语(如Hash函数、对称加密和异或操作等)，基于非抗窜扰智能卡假设，来实现用户匿名性。因为对称密码技术相对公钥密码技术来说更轻量，它们选取前者的动机十分明显，但为什么要基于非抗窜扰智能卡假设呢？

最近研究结果表明，普通智能卡内的安全参数信息可通过先进的边信道攻击技术(例如差分功率分析)[324–326]、软件攻击(针对通过软件实现的智能卡如Java卡)[327]和逆向工程技术[165]提取出来。但是，边信道攻击需要特殊的设备和平台，故只有当攻击者$\mathcal{A}$可较长时间持有目标用户的智能卡时，$\mathcal{A}$才能提取数据，否则认为智能卡是安全的。比如当智能卡丢失或被盗时，$\mathcal{A}$可以提取出卡内数据；当用户在消费终端刷卡消费时，由于用户在现场，并且时间较短，$\mathcal{A}$没有条件进行边信道攻击，故智能卡参数此时是安全的。这意味着，智能卡的非抗窜扰假设是条件性的。如图1.12所示，自2004年以来这一条件性非抗窜扰智能卡假设已被广泛接受，成为双因子认证协议的基本假设。

匿名双因子认证协议一再的设计失误引出一个基础性的问题：在(条件)非抗窜扰智能卡假设下，能否仅基于对称密码原语设计适于 WSNs 环境的匿名双因子认证协议？除了上文提到的 WSNs 下双因子协议，还有许多其它应用环境下的双因子协议(最近的一些协议见[177, 178, 180, 328–332]; 更全面信息请见图6.2)力求仅采用对称密码原语，来实现高效的匿名口令认证协议。本章的目标正是证明，在(条件)非抗窜扰智能卡假设下，这种协议设计策略在本质上是不可行的。

进一步，如果公钥加密技术是实现用户匿名性的基本组件，那么如何在可接受的效率下，设计安全的匿名双因子认证协议？

6.1.3 本章贡献

本章首次对上述两个关键问题进行了探索，并给出了确定性答案，主要贡献如下：

- 分析了两个典型的无线传感器网络(WSNs)环境下基于“口令+智能卡”的双因子认证协议，即 Fan等协议和Xue等协议，揭示在 WSNs 环境下设计匿名双因子认证协议的难点和挑战。
- 基于 Halevi-Krawczyk (1999)的工作和 Impagliazzo-Rudich (1989)的工作，提出了双因子认证协议的形式化安全模型，严格证明了实现用户匿名性的一般性原则，为节6.1.2中的第一个问题提供了否定性答案。
- 显示我们的匿名性实现原则具有普适性。尽管本章主要讨论的是 WSNs 环境下的用户匿名性问题，我们进一步显示本章提出的原则适用于各类应用环境，比如单服务器架构、多服务器环境、全球移动网络、代理移动 IPv6 网络、卫星网络、移动云和其它环境。
- 讨论实现匿名性原则的可行方案。借鉴移动漫游网络环境下的匿名双因子认证协议实现机制，基于公钥密码原语的实验数据，我们指出：适当的公钥加密算法配以合适的填充机制，既实现了可证明安全性，又保证了可接受的效率。这是实现匿名性的可行方案，回答了节6.1.2中的第二个问题。

6.2 对 Fan 等协议的分析

2011年， Fan 等[304]提出了一个适于 WSNs 环境的抗DoS攻击的高效双因子认证协议。Fan 等宣称他们的协议相较于现有协议有更多优点，比如能实现双向认证、用户匿名性、口令本地更新。然而，经本文作者分析发现，Fan 等协议仍然存在用户匿名性问题，并且对智能卡丢失攻击和内部攻击是脆弱的。本章主要关注其隐私相关缺陷，其它缺陷不作讨论。

6.2.1 Fan 等协议回顾

Fan 等协议包含四个参与者，即用户(U_i), 基站(BS), 主节点(MN_j) 和传感器节点 (SN)。基站BS通常作为网关节点，仅参与注册阶段。Fan等协议包含三个阶段：注册、登录、认证。协议中所使用的符号及其含义如表6.1所示。

1) **注册阶段** 在这一阶段执行之前，假设每个主节点 MN_j 和基站 BS 间已共享秘密参数 Y_j，并已为每个传感器节点和主节点分配长期共享密钥S_k。当新用户U_i要访问传感器节点的数据时，首先要完成如下过程：

Step R1. 用户U_i 选择用户名 ID_i 和口令 PW_i。

Step R2. $U_i \Rightarrow BS : \{ID_i, PW_i\}$。

表 6.1 符号定义

符号	含义
U_i	i^{th} 用户
BS	基站
MN_j	j^{th} 主节点
SN	传感器节点
$\mathcal{A}$	攻击者
ID_i	用户 U_i 的标识符
PW_i	用户 U_i 的口令
X	BS主密钥
Y_j	MN和BS的共享密钥
N_{ij}	j^{th} 主节点的i^{th} 用户凭证
$\oplus$	按位异或操作
$\|$	比特连接操作
$E(\cdot)/D(\cdot)$	对称密码加密/解密运算
$h(\cdot)$	安全散列函数
$\rightarrow$	普通信道
$\Rightarrow$	安全信道

Step R3. BS 收到 U_i 的注册请求, 选取随机数 R_i 并计算 $RID_i = h(R_i \| ID_{BS}) \oplus ID_i \oplus ID_{BS}$, 其中，$ID_{BS}$ 表示 BS的ID。然后，BS 计算 $A_i = h(X) \oplus h^2(ID_i \| PW_i)$ 和 $V_i = h^3(ID_i \| PW_i)$。

Step R4. BS 在相关的主节点上为每个用户建立数据表项，保存用户凭证 N_{ij} (例如：主节点ID，用户访问权限、有效期)和 $B_{ij} = h(N_{ij} \| Y_j) \oplus h(ID_i \| PW_i)$。$BS$ 在用户数据库中存储 $\{RID_i, R_i\}$。值得一提的是，验证表项有利于实现细粒度的访问控制，限制用户登录具体的传感器节点。BS 将参数 $\{h(\cdot), RID_i, V_i, A_i\}$ 存储在智能卡中。

Step R5. $BS \Rightarrow U_i$: 智能卡和访问控制权限表$\{N_{ij}, B_{ij}\}$。

2) **登录阶段** 当 U_i 要直接访问传感器节点的实时数据时，执行以下步骤：

Step L1. U_i 插入智能卡读卡器，输入 ID_i^* 和 PW_i^*。

Step L2. 智能卡计算 $V_i^* = h^3(ID_i^* \| PW_i^*)$ ，验证 V_i^* 是否等于存储的 V_i。若相等，表明 $ID_i^* = ID_i$，$PW_i^* = PW_i$；反之，说明 U_i 输入的口令不正确，智能卡拒绝本次会话。

Step L3. U_i 选取将要登录的主节点，智能卡读取该节点相关的 N_{ij} 和 B_{ij}，计算 $TK_i = h((B_{ij} \oplus h(ID_i \| PW_i)) \| T) = h(h(N_{ij} \| Y_j) \| T)$，$SID_i = RID_i \oplus h((A_i \oplus h^2(ID_i \| PW_i)) \| T) = RID_i \oplus h(h(X) \| T)$，其中，$T$ 表示客户端当前时间戳。

Step L4. 智能卡计算 $C_1 = SID_i \oplus TK_i$ 和 $C_2 = h(TK_i \| SID_i \| N_{ij} \| T)$。

Step L5. $U_i \rightarrow MN_j : \{N_{ij}, C_1, C_2, T\}$。

Step L6. 智能卡删除 ID_i， PW_i， SID_i， $h(N_{ij} \| Y_j)$， $h(ID_i \| PW_i)$，和 $h^2(ID_i \| PW_i)$ ，记录时间戳 T。

3) **认证阶段** 该阶段旨在完成U_i，MN_j 和 SN 之间的双向认证，同时在U_i 和 SN 之间建立会话密钥。由于该阶段与本章分析无关，在此省略。

6.2.2 Fan等协议匿名性缺陷

一方面，攻击者可以通过多种手段，如分析协议交互的消息、查询用户访问的服务或资源[172, 333]，来获得用户的敏感个人信息(如爱好、生活方式、社交圈和住址等)。另一方面，个人、组织和政府对隐私问题的关注度持续上升。因此，保护用户隐私的密码协议引起广泛兴趣。在无线和移动环境中，未经授权的实体可以利用用户的静态身份追踪用户的当前位置和访问历史[175, 316]，故用户匿名性是双因子认证协议的重要属性。

如前文所述，用户匿名性内涵多种多样，不同应用环境下的匿名性内涵往往差异巨大[288]。对于身份认证协议来说，用户匿名性主要包含两个属性，基本属性和高级属性：(1)用户名保护，即攻击者通过分析协议交互消息无法获取真实用户名；(2) 用户不可追踪性，即攻击者不仅无法识别用户名，也不能分辨两次会话是否来自同一用户。后者已被广泛接受，成为大多数认证协议的重要设计目标(见[181, 282, 291–295, 319])。

为实现用户匿名性，一种常见的解决办法是采用 Das 等提出的“动态ID技术”[334]，将真实用户名隐藏在(随会话)动态变化的伪ID中。通过此种方式，仅认证服务器可获知用户ID，而其他实体通过侦听信道中消息，无法获取任何有用的个人信息。采用这种技术的协议称为“动态ID协议”或隐私保护协议[335]，Fan 等协议隶属于这一类型。但是，如下面攻击所显示的，Fan等协议并未实现所宣称的用户匿名性。

假设某一主节点MN_m不可信，与恶意用户U_m 相互勾结，则可破坏任意用户U_i 的不可追踪性。恶意用户 U_m 利用边信道攻击技术，或逆向工程技术[165, 324–326]从自己的智能卡中提取 A_m。目前，大多数针对双因子认证协议的安全性分析都假设智能卡是(条件)非抗窜扰的：如果攻击者获得U_i智能卡足够长时间(几个小时或更长时间)，则智能卡内秘密信息可以被攻击者提取[175, 278, 300]。这一假设在 Fan 等协议中也被明确提及。当 A_m 已知，U_m 根据自己的ID_m 和 PW_m 可计算 $h(X) = A_m \oplus h^2(ID_m \| PW_m)$。已知 $h(X)$， U_m 和 MN_m，U_m 可以通过以下步骤获取任意合法用户 U_i 的敏感信息：

Step 1. MN_m 收到 U_i 登录信息 $\{N_{im}, C_1, C_2, T\}$后，计算 $TK_i = h(h(N_{im} \|$

$Y_m) \| T)$，其中 Y_m 由MN_m 和BS 共享；

Step 2. MN_m 计算$SID_i = C_1 \oplus TK_i$；

Step 3. U_m 从智能卡内提取A_m，根据自己的ID_m，PW_m 计算 $h(X) = A_m \oplus h^2(ID_m \| PW_m)$ ；

Step 4. MN_m 与 U_m 勾结，计算 $RID_i = SID_i \oplus h(h(X) \| T)$。

不难看出，RID_i 是静态的，并且与 U_i 唯一相关，即 RID_i可以被视为 U_i的ID。因此，攻击者MN_m 可以利用此信息查询和追踪用户U_i，破坏用户隐私。这违背了协议[304] 声称的“除BS以外的其他参与者无法获知用户ID”。

为成功实施上述攻击，需要有不可信的主节点MN_m与恶意用户U_m相互串通，实施共谋。注意，这在实际应用中是现实的，因为用户和主节点是不可信实体，可能会是恶意的、好奇的。因此，假设WSNs中存在MN_m与U_m相互串通的安全威胁是现实的，这一点已被广泛接受(见[277, 336, 337])。尽管存在所有主节点和用户都可信的可能，但本章提出的潜在安全威胁应引起协议设计者和应用者的重视。此外，上述攻击无需用户智能卡内秘密参数。更确切地说，攻击者U_m只需要提取自己智能卡内的参数$h(X)$。这一假设的难度远低于提取受害者U_i智能卡内信息。综上所述，攻击者通过监测通信信道，提取自己智能卡内秘密参数，即可发动上述匿名性(不可追踪性)攻击。因此，本节提出的匿名性共谋攻击简单有效，是WSNs环境下匿名双因子认证协议面临的现实挑战。

6.3　对 Xue 等协议的分析

Fan 等协议[304]中，攻击者相互串通来破坏用户匿名性，而对于 Xue 等协议[295]，本章提出一个新的攻击策略。

本章简要回顾 Xue 等[295]在2012年提出的基于临时凭证的双因子认证协议。与早期协议(如[282, 293, 301])相比，Xue等协议实现了更多安全性和可用性属性，比如，双向认证、抗网关旁路攻击，口令本地更新等。总体而言，Xue 等协议功能完善、简单可靠，在WSNs中具有一定应用潜力。尽管如此，本节指出 Xue 等协议仍不能实现用户匿名性。

6.3.1　Xue等协议回顾

Xue 等协议[295] 涉及三个参与者，即用户(U_i)，网关节点(GWN)和传感器节点 (S_j)。与 Fan 等协议[304]不同，在[295]中，GWN不仅参与注册阶段，还参与U_i和S_j的认证阶段。Xue 等协议包含三个阶段：注册、登录、认证和会话密钥协商。

1) **注册阶段** 假设注册之前，U_i的用户名ID_i和口令$H(PW_i)$已在GWN存储；每个传感器节点保存有预定义的身份标识SID_j和口令$H(PW_j)$；GWN存储传感器节点的口令$H(PW_j)$。注册阶段分为用户注册和传感器注册。

(a) 用户注册阶段

U_i 读取当前时间戳 TS_1，计算 $VI_i = H(TS_1 \| H(PW_i))$。

Step RU1. $U_i \rightarrow GWN$: $\{ID_i, TS_1, VI_i\}$。

Step RU2. GWN 接收到来自 U_i 的注册请求后，首先检查 TS_1 的有效性。然后，根据ID_i 从数据库中提取 $H(PW_i)$ ，计算 $VI_i^* = H(TS_1 \| H(PW_i))$，比较 $VI_i^* \stackrel{?}{=} VI_i$。若不相等, GWN 拒绝注册请求。

Step RU3. GWN 计算 $P_i = H(ID_i \| TE_i)$， $TC_i = H(K_{GWN-U}) \| P_i \| TE_i$ 和 $PTC_i = TC_i \oplus H(PW_i)$，其中，$TE_i$是由$GWN$或可信第三方(TTP)设定的超时时间阈值，$K_{GWN-U}$ 代表GWN的私钥，TC_i为GWN 为U_i颁发的临时身份凭据。GWN 将 $\{H(\cdot), ID_i, H(H(PW_i)), TE_i, PTC_i\}$ 写入智能卡。

Step RU4. $GWN \rightarrow U_i$: $\{H(\cdot), ID_i, H(H(PW_i)), TE_i, TC_i\}$。

(b) 传感器节点注册阶段

这一阶段与本章讨论无关，在此省略。

2) **登录阶段** U_i 构造登录请求，执行如下操作。

Step L1. U_i 插入智能卡读卡器，输入 ID_i^* 和 PW_i^*。

Step L2. 智能卡通过比较存储的ID_i，$h(PW_i)$，验证输入的ID_i^*，$h(PW_i^*)$ 的有效性。若都相等，表明U_i 是合法用户，然后，智能卡计算 $TC_i = PTC_i \oplus H(PW_i)$。

Step L3. U_i 读取当前时间戳 TS_4，选取随机数 K_i，计算 $DID_i = ID_i \oplus H(TC_i \| TS_4)$，$C_i = H(H(ID_i \| \ TS_4) \oplus TC_i)$ 和 $PKS_i = K_i \oplus H(TC_i \| TS_4 \| \text{ø000Æ})$。注意，$H(TC_i \| TS_4 \| \text{Æ000Æ})$ 不同于 $H(TC_i \| TS_4)$，这得益于Hash函数抗碰撞特性。

Step L4. $U_i \rightarrow GWN$: $\{DID_i, C_i, PKS_i, TS_4, TE_i, P_i\}$。

3) **认证和密钥协商阶段** 这一阶段旨在实现U_i和 GWN的双向认证，协商U_i 和S_j的会话密钥。由于该阶段与后面的讨论无关，故不在此详述。

6.3.2 Xue等协议匿名性缺陷

文献[295]沿用了经典的Dolev-Yao攻击者模型[338]，即攻击者$\mathcal{A}$ 完全控制用户 U_i、网关节点GWN和传感器节点S_j之间的通信信道，可以任意拦截、阻断、删除、插入或更改流经公开信道中的消息。尽管攻击者$\mathcal{A}$ 的能力强大，但不是

万能的，$\mathcal{A}$ 不能破解Hash函数、分组加密、公钥加密等密码原语。另外，Xue等协议中明确假设智能卡可以被窜改。然而，在下面针对Xue 等协议安全性的讨论中，我们发现仅根据第1项假设(即$\mathcal{A}$完全控制通信信道)，$\mathcal{A}$ 就可以成功破坏用户匿名性：

Step 1. 截获U_i发送的登录请求信息：$\{DID_i, C_i, PKS_i, TS_4, TE_i, P_i\}$；

Step 2. 猜测用户名为ID_i^*；

Step 3. 计算 $P_i^* = h(ID_i^* \| TE_i)$，其中 TE_i 在Step 1中获得；

Step 4. 验证$P_i^* = P_i$是否成立。如果等式成立，则ID_i^*猜测正确；否则重复Steps $2 \sim 4$。

上述攻击没有涉及高代价的计算或高难度的技术(如能耗分析或逆向工程技术)，$\mathcal{A}$ 只需要在协议运行时窃听一次会话消息，然后离线猜测用户名即可。令$|\mathcal{D}_{id}|$表示用户身份标识空间大小，上述流程的时间复杂度为$\mathcal{O}(|\mathcal{D}_{id}| * T_H)$，$T_H$代表Hash操作时间。因此，上述攻击代价与$|\mathcal{D}_{id}|$呈线性关系。

6.3.3　攻击效率分析

为评估上述攻击的有效性，本章利用公开的密码函数库MIRACL[339]来测量密码原语操作时间。MIRACL 是基于C/C++ 的大数运算库。我们在多台具有不同计算能力的PC上进行实验，为保证统计结果的健壮性，每种密码原语操作重复一千次后取均值。表6.2列出相关密码原语操作及实验数据。

表 6.2 相关操作在普通PC上的计算时间评估

Experimental platform (Laptops)	Hash operation T_H (SHA-1)	Other lightweight operations (e.g., XOR)
Intel T5870 2.00 GHz	2.437 µs	0.011 µs
Intel i5-3210 2.50 GHz	1.106 µs	0.009 µs
Intel i3-530 2.93 GHz	0.815 µs	0.008 µs

在实际中，$|\mathcal{D}_{id}|$ 一般非常有限，例如 $|\mathcal{D}_{id}| \leq |\mathcal{D}_{pw}| \leq 10^{6}$[5, 104, 340]，故上述攻击能够在多项式时间内完成。值得一提的是，该用户匿名性攻击中，攻击者只需监测通信信道，不涉及任何特殊的高难度技术(如边信道分析)。因此，上述攻击切实可行，攻击者可以在一台普通的PC上几秒钟内猜测出用户名。

6.4　匿名双因子认证协议的公钥技术原则

由于传感器节点和智能卡在运算能力、存储空间和电池能量方面资源受限，如何设计兼顾安全、效率和功能需求的匿名双因子认证协议是一项巨大的挑战。通常看似良好的设计动机，往往却产生非预期的结果。在缺乏必要的设计原则(准则)进行指导的情况下，协议设计者只能靠经验尽可能保证协议是完备

的和安全的，这一过程更似艺术而非科学。这很好地解释了，为什么现实中大多数复杂的密码协议最终都被指出存在这样或那样的安全缺陷[341–343]。

众所周知，基于数字证书的身份认证协议对 WSNs 来说过于庞大和复杂；基于公钥密码技术(比如模幂运算、椭圆曲线和双线性对)的传统认证协议(比如为互联网应用设计的协议[60, 344])往往也是计算密集型，能耗高，不适于资源受限的WSNs 环境。于是，近年来WSNs环境下双因子认证协议设计走向另一个极端：学者们尝试仅采用非公钥技术(例如，Hash 函数、对称加密、异或运算、MAC 操作和伪随机函数)来设计匿名协议，并宣称在实现隐私保护和安全性的同时，降低了计算复杂度、通信成本和存储开销。数以百计的此类协议被提出，典型的见[181, 282, 291–293, 295, 298, 301, 303, 304, 318–323])。如本文将指出的，这些尝试都是徒劳。为解释这些协议失败的根本原因，克服这一长期存在的问题，下面我们提出并证明一个普适的匿名认证协议设计原则，为将来设计安全高效匿名协议提供基本指导。

6.4.1 公钥技术原则

本文作者分析了300余个近年来提出的双因子认证协议(包括通用环境下的200余个，无线通信环境下的50余个，WSNs环境下的70多个)，一些公开发表的相关成果见[279, 290, 345–348]。我们发现，在非抗窜扰智能卡假设下，采用公钥密码技术的协议不一定能实现用户匿名性，但所有未采用公钥密码技术的协议都，无一例外地，无法实现用户匿名性。这意味着，那些仅基于对称密码原语却声称能保护用户隐私的协议都无法实现所宣称的目标，近期典型的失效协议如[176–178, 181, 295, 318–322, 329, 331, 349–353]。我们猜测：*在非抗窜扰智能卡假设下，公钥密码技术是实现用名匿名性的基本组件*。基于Halevi-Krawczyk[56]和 Impagliazzo-Rudich[311]这两项工作，采用反证法和是归约法，我们成功证实了这一猜测。

1) Halevi-Krawczyk 单因子口令认证协议模型

在文献[56]中，Halevi-Krawczyk提出了一个基于口令的认证密钥交换(PAKE)协议，并给出了此类协议(即单因子口令)的形式化安全模型。该模型建立在标准的分布式计算模型上(即 Dolev-Yao 模型)[338]，侧重用户到服务器的单向认证。用户口令往往来自有限的空间$\mathcal{D}$(见第三章)，并且服从严重偏态的Zipf分布(见第二章)，攻击者$\mathcal{A}$总是可以从这一有限的空间$\mathcal{D}$中选取一个口令作为猜测，然后忠实地执行协议，实施一次在线仿冒攻击。现实中，一般通过帐号锁定、登录速率限制或异常登录检测[19]等手段来抵御这种在线仿冒攻击，因此$\mathcal{A}$无法进行大量的登录尝试，可进行的猜测次数m十分有限(通常$m \leq 10^4$)。

由于在线仿冒攻击不可避免，对于一个口令认证协议$\mathcal{P}$来说，如果该攻击是$\mathcal{A}$获取用户口令、实现仿冒的最佳途径，那么协议$\mathcal{P}$就是安全的。

Halevi-Krawczyk[56]的基于口令的单向认证安全模型采用了密码学中的通行做法，即通过一个“概率游戏”来刻画，主要包括游戏参与者、游戏规则、攻击者能力，以及安全概念。具体如下：

定义 1 (参与者) 协议参与者为用户U，服务器S和攻击者$\mathcal{A}$。

定义 2 (游戏规则) 本游戏以安全参数k和口令空间$\mathcal{D}$为参数，执行过程如下：

— *设置阶段:* 服务器S选取并公开身份标识ID_S，选择加密密钥，若采用公钥加密技术，则公开公钥；U选取任意字符串ID_i 作为用户名，从口令空间 $\mathcal{D}$ 中随机均匀抽取口令PW_i[Φ]，U 秘密向服务器发送口令PW_i。
攻击者$\mathcal{A}$可在任意时刻向服务器S 注册，$\mathcal{A}$ 选取任意用户名AID_i 和口令APW_i (确保$AID_i \neq ID_i$)，公开AID_i，向服务器S发送APW_i。

— *运行阶段:* 攻击者$\mathcal{A}$ 完全控制U和S之间的通信信道。换句话说，U和S发送和接收的每条消息都经过攻击者$\mathcal{A}$。$\mathcal{A}$ 可以任意拦截、阻断、删除、插入或更改通信信道中的消息。$\mathcal{A}$ 也可以向U发送特殊“提示”消息，来发起会话。最终由$\mathcal{A}$ 决定游戏何时停止。

定义 3 (攻击者能力) 攻击者$\mathcal{A}$ 的能力通过上述游戏规则进行模拟并通过计算可行性(如概率多项式时间PPT)来衡量。总结如下：

(i) $\mathcal{A}$ 可以控制公开信道中用户U 和服务器S之间交互的消息。

(ii) $\mathcal{A}$ 可以离线穷举口令空间 $\mathcal{D}$ 中的所有元素。

(iii) $\mathcal{A}$ 可以创建或注册新用户，即$\mathcal{A}$ 可以是合法但恶意的用户。

然而，攻击者$\mathcal{A}$ 的能力是有限的，不能破解加密、Hash等密码原语。由于攻击者计算能力虽然强大但受限，因此称之为可行性攻击者。

为便于理解，在给出安全模型的定义(即定义10)之前，本节先给出一些预备的术语定义。

定义 4 (输出对) 每个认证会话具有唯一会话标识符sid，每个参与者需记录与认证的安全性相关的事件，即每次会话结束时，用户U 输出(S, sid)；服务器S输出(U, sid)。若以sid为标识的一次会话结束后，用户U未通过服务器S的认证，则S输出 $(U, sid, \perp)$。不失一般性，sid 被设置为实体所发送的和接收的消息的按序连接(Ordered concatenation of protocol transcripts)。

[Φ]Halevi-Krawczyk[56]指出，这里假设口令服从均匀分布仅仅是为了表达方便，任何其它分布假设都是可行的。幸运的是，我们在第二章中发现口令服从Zipf分布，从此口令没有必要、也不应再作“其它分布假设”。

定义 5 (正确性) 在以sid为标识的一次会话中，如果U和S之间的所有通信消息都未被修改，在协议结束时S 和U分别输出(U, sid)和(S, sid)，即S和U均已接受且拥有相同的会话密钥，则称$\mathcal{P}$为语法正确性的协议。

定义 6 (仿冒) 在一次会话中，若服务器S输出(U, sid)，而目标用户U从未输出(S, sid)，则称发生成功仿冒事件；若服务器先输出$(U\text{Æ}, sid)$，又输出(U', sid)，则称发生成功重放事件；若服务器输出$(U, sid, \perp)$，则称发生登录失败事件。

定义 7 ((ℓ, m)-win) (ℓ, m)-run是指攻击者$\mathcal{A}$至多执行m次主动仿冒尝试，而U最多输出ℓ对(S, sid)。若在一次(ℓ, m)-run中，攻击者$\mathcal{A}$ 至少实现一次成功仿冒，则称$\mathcal{A}$实现(ℓ, m)-win。

定义 8 (安全 $U2S$ 口令认证) 设 $\epsilon(\cdot, \cdot, \cdot)$ 为正实函数，$\mathcal{P}$ 为语法正确的单向认证协议(即用户到服务器认证，简称$U2S$)。若对任意可行性攻击者$\mathcal{A}$，任意有限口令空间$\mathcal{D}$，任意足够大的安全参数k，任意多项式ℓ 和任意整数$m < |\mathcal{D}|$，满足

$$\Pr[(\ell, m)-\text{win}] \leq \frac{m}{|\mathcal{D}|} + \epsilon(k, \ell, m)$$

则称$\mathcal{P}$确保$U2S$认证达到 ϵ 的安全性。其中，ℓ 为用户输出对的上限，m为攻击者$\mathcal{A}$被允许的在线仿冒尝试次数上限。

需要指出的是，在定义 10 中，口令被假设为服从均匀分布。实际上，根据第二章的发现，用户生成的口令较完美地服从Zipf分布，一种与均匀分布迥异的分布。根据节2.6.1的大规模实验结果，基于均匀分布假设的口令认证协议的安全性表达函数会低估真实攻击者的优势 $2 \sim 4$ 个数量级，而基于Zipf分布假设的安全性表达函数与真实攻击者的优势间最大误差为0.48%~4.59%(均值1.84%)。因此，接下来我们将采用口令的Zipf分布假设。

2) 本文双因子认证协议安全模型

本节在 Halevi-Krawczyk[56]的单因子口令认证协议安全模型基础上，定义基于口令的双因子认证协议安全模型。与文献[56]类似，为简单起见，本节同样仅关注单向身份认证。不难看出，更通用的安全概念(如双向认证)可以采用同样的形式化方法进行建模。我们仅考虑最一般的客户–服务器架构，即系统有多个用户U和但仅存在单个服务器S。本节提出的模型可以很容易地扩展到除U和S外还包含其他实体的特定应用环境中。比如在无线传感器网络中[282]，主节点和传感器节点二者一起可被视为本节的服务器端；在移动网络中[316]，内部代理和外部代理二者一起可被视为本节的服务器端。

本节的安全游戏采用与 Halevi-Krawczyk 安全游戏相同的参与者。两个模型间主要的区别在于：(1) 设置阶段，S 分发给U 一个带有参数的智能卡；(2)运行阶段，攻击者$\mathcal{A}$可以分析出智能卡内参数信息，该智能卡可能是用户丢失或$\mathcal{A}$窃取的，也可能是$\mathcal{A}$自身拥有的合法智能卡。

定义 9 (双因子认证的攻击者) 根据我们的游戏规则，攻击者具有定义 3 中 $i \sim iii$ 的能力和以下能力：

(iv) 一旦$\mathcal{A}$ 通过某种手段获得较长时间访问智能卡的权限(如拾到或窃取)，$\mathcal{A}$ 可以分析出智能卡内秘密参数信息。显然，如果$\mathcal{A}$也是合法用户，这意味着$\mathcal{A}$可以提取自己智能卡内的参数信息。

如文献[278]和节7.2.2所论证的，在双因子认证协议中假设$\mathcal{A}$具有 $i \sim iv$ 的能力是合理的。该假设目前已获得普遍认可(见[181, 279, 301, 303, 354, 355])。

单因子口令认证是构成双因子认证的基石，故在 Halevi-Krawczyk 安全模型中定义的术语(如“输出对”、“正确性”、“仿冒”、(ℓ,m)-win)可以直接迁移到双因子认证安全模型的定义中。基于上述游戏规则和基本术语，本节将给出安全双因子认证的定义。很明显，如果$\mathcal{A}$同时获得了用户U的口令和智能卡，没有任何办法可以阻碍$\mathcal{A}$仿冒用户U。任何双因子协议都不能抵御这种平凡攻击。

因此，我们主要考虑剩余三种情况：(1)用户U的口令和智能卡都是安全的，则协议也是安全的；(2) $\mathcal{A}$ 获得U的口令但没有U的智能卡，因智能卡内存有高信息熵的秘密参数，则$\mathcal{A}$对协议不构成实际威胁；(3)$\mathcal{A}$获得用户U的智能卡但没有U的口令，此时唯一的防线是U的低信息熵的口令，双因子认证退化为传统的单因子认证，协议受到极大威胁。这是$\mathcal{A}$破坏协议的最佳时机。因此，非正式地，我们称一个双因子认证协议$\mathcal{T}$是安全的，如果$\mathcal{A}$在情况3下仍无法获得不可忽略的概率成功猜测出U的口令。这就解释了为什么本节安全模型中安全性的概念与 Halevi-Krawczyk 模型一致。为处理隐私保护协议，本节也形式化定义了用户匿名性概念。

定义 10 (安全 $U2S$ 双因子认证) 设 $\epsilon(\cdot,\cdot,\cdot)$ 为正实函数， $\mathcal{T}$ 为语法正确的单向双因子认证协议(即用户到服务器，简称$U2S$)。若对任意可行性攻击者$\mathcal{A}$，任意有限的服从$Zipf$分布的口令空间$\mathcal{D}$，任意足够大的安全参数k，任意多项式ℓ和任意整数$m < |\mathcal{D}|$，满足

$$\Pr[(\ell,m)-\text{win}] \leq C' \cdot m^{s'} + \epsilon(k,\ell,m)$$

则称协议 $\mathcal{T}$ 可确保 $U2S$ 认证达到 ϵ 安全性。其中，ℓ 为用户输出上限，m为攻击者$\mathcal{A}$被允许的在线仿冒尝试次数上限，C'和s'为口令空间$\mathcal{D}$的$Zipf$分布参数。

定义 11 (用户匿名性/不可追踪性) 由于用户名通常为用户自行选择的易于记忆的字符串，往往遵循一定的格式，且相较口令而言缺乏保护(比如明文显示、用户名列表公开可得)。当$N \geq 2$时，$\mathcal{A}$可以穷举用户名$\mathcal{D}_{id} = \{ID_1, ID_2, \cdots, ID_N\}$。即使如此，匿名认证协议$\mathcal{T}$需要确保$\mathcal{A}$不能从协议交互消息中推断出：(1)该登录会话(比如会话sid_j)由哪个用户参与；(2)某一特定用户(比如用户U_i)是否参与了两个(或任意数量)给定会话。

3) 公钥技术原则的证明

在 Havelli-Crawczyk 工作之前，Impagliazzo和Rudich[311]设计了一个有趣的安全模型，该模型假设“仅公钥加密技术可用”，然后研究某种特定类型的协议是否可在该模型中存在。Impagliazzo-Rudich 模型与标准分布式计算模型(即Dolev-Yao 模型[338])有两个不同点：

- 允许攻击者访问一个NP-完全问题预言机，即给定公钥，攻击者可以在多项式时间找到对应的私钥，所有公钥密码原语将不复存在。
- 授予所有实体访问随机函数$f(\cdot)$的权限，该函数将$\{0,1\}^*$中的每个字符串映射成一个均匀分布的k-bit字符串。更具体地说，一些密码原语(如抗碰撞Hash、MAC和对称加密算法)可使用$f(\cdot)$来实现。

接着，Impagliazzo-Rudich 证明在该安全模型中不存在安全的(广义的)密钥交换协议：

引理 1 (Impagliazzo-Rudich 1989) 任意$\epsilon'(k) \leq \frac{1}{2^k}$, *Impagliazzo-Rudich* 模型中不存在安全上界为$\epsilon'(k)$的密钥交换协议。其中，k 为系统安全参数，$\epsilon'(k) = \epsilon(k,1,1)$。

根据这一引理，可以得出以下结论：

定理 6 基于(条件性)非抗窜扰智能卡假设，对任意$\epsilon(k,\ell,m) \leq \ell \cdot m \cdot \epsilon'(k) \leq \frac{\ell m}{2^k}$，不存在仅使用对称密码原语的安全上界达$\epsilon(k,\ell,m)$的双因子认证协议。其中，$k$为系统安全参数，$\ell$为用户输出对上限，$m$为攻击者$\mathcal{A}$主动仿冒尝试次数上限。

证明 采用反证法。基本思想：给定一个仅基于对称密码原语的双因子认证协议$\mathcal{T}$，假设它实现了匿名性，然后我们用它来构建一个安全的密钥交换协议$\mathcal{E}$。这意味着，协议$\mathcal{E}$仅基于对称密码原语。然而，引理 1 表明$\mathcal{E}$无法仅建立在对称密码原语上，这就产生了矛盾。

为得到矛盾，我们假设攻击者具有 $i \sim iv$ 能力(之后称为“我们的攻击者模型”，也是双因子协议中广泛采用的模型)，存在仅基于对称密码原语的匿名双因子认证协议$\mathcal{T}$。注意，我们的攻击者模型采用了非抗窜扰智能卡假设，即

假设存储在智能卡内的安全参数可以通过边信道攻击或逆向工程技术提取出来[324–327]。一旦智能卡因子的安全性被破坏，协议$\mathcal{T}$的安全性完全依赖于口令因子的安全。因此，协议$\mathcal{T}$即退化为传统的单因子口令认证协议，称为协议$\mathcal{P}$。

实际上，口令协议$\mathcal{P}$背后的攻击者模型就是 Impagliazzo-Rudich 模型，因为$\mathcal{P}$ 仅基于对称密码原语，与 Impagliazzo-Rudich 模型相一致。由于协议$\mathcal{T}$在我们的攻击者模型中被假设为实现了匿名性的双因子认证协议，因此协议$\mathcal{P}$为 Impagliazzo-Rudich 模型下实现了匿名性的单因子口令认证协议。这意味着，对于协议$\mathcal{P}$的任意两个会话，$\mathcal{A}$不能分辨出这两个会话是否来自同一用户。

接下来我们显示，在 Impagliazzo-Rudich 模型中，可以基于口令协议$\mathcal{P}$ 来构造安全上界为$\epsilon'(k)$的单因子(即口令)密钥交换协议。首先，改变协议$\mathcal{P}$中会话密钥生成方式。一次会话(即sid_j)运行中，每个参与者(即U_i和S)可以交换 1 比特信息b_j，其中 $b_j = h(sid_j \| ID_i \| ID_S) \bmod 2$， $h(\cdot)$ 为抗碰撞散列函数，$h(\cdot)$可由 Impagliazzo-Rudich 模型中定义的随机函数$f(\cdot)$来实例化得到。协议成功运行后，由于U_i 和S 已知sid_j，以及ID_i和ID_S之间的正确关系，因此U_i和S可以计算出相同的比特b_j。因此，通过简单的重复运行协议l次(如 $l \geq \log_2 k$ 比特)，然后令会话密钥$sk = \overbrace{b_j \ldots b_m \ldots b_t}^{l \text{ bits}}$，则协议$\mathcal{P}$实现了会话密钥的交换。其中，$b_m$和$b_t$ 分别为会话 sid_m 和 sid_t 交换的1比特信息(与b_j计算方法类似)。我们称此密钥交换协议为 $\mathcal{E}$。

在密钥交换协议$\mathcal{E}$中，首先可以明显看出 l 次会话后，U_i和S 确实成功交换了相同的比特串 sk。接下来，我们考虑一个增强的攻击者$\mathcal{A}^+$，既具备攻击者$\mathcal{A}$的所有能力，也能够分辨两次会话是否来自同一(但未知ID)用户。那么，$\mathcal{A}^+$ 将有$\frac{1}{|\mathcal{D}_{id}|}(\gg \epsilon'(k))$ 的概率正确计算出协议$\mathcal{E}$输出的比特串sk，步骤如下：(1)选取目标用户U_i参与的 l 次会话，即$sid_j, \ldots, sid_m, \ldots sid_t$；(2)从用户的身份标识空间$\mathcal{D}_{id}$ 中猜测目标用户的用户名ID'_i；(3)计算$sk' = b'_j \| \ldots \| b'_m \| \ldots \| b'_t$，其中，$b'_j = h(sid_j \| \ ID'_i \| ID_S) \bmod 2$，$b'_m = h(sid_m \| ID'_i \| ID_S) \bmod 2$，$b'_t = h(sid_t \| ID'_i \| ID_S) \bmod 2$。对任意攻击者($\mathcal{A}$或$\mathcal{A}^+$)，为成功计算(而不是猜测)出$sk$，必须正确选取目标用户$U_i$参与的 l 次会话。因为$h(\cdot)$是抗碰撞散列函数，若 l 次会话选取不正确，则攻击者不可能计算成功。不难看出，对增强攻击者$\mathcal{A}^+$来说，上述方法是计算密钥交换协议$\mathcal{E}$输出的sk'的最简单有效方法。

再者，由于协议$\mathcal{P}$能实现用户不可追踪性，则$\mathcal{A}$不能分辨出两个会话是否来自同一(未知)用户。即给定顺序的 $x(x \geq l)$ 次会话，$\mathcal{A}$正确选取包含U_i的l次会话的可能性不大于$1/\mathbf{C}_x^l$。其中，$\mathbf{C}_x^l$ 为从x次会话中取l 次会话的组合数。此外，$\mathcal{A}$正确计算会话密钥sk的概率$\mathrm{Adv}_{\mathcal{P}}^{\mathrm{sk}}(\mathcal{A})$ 不大于$1/\mathbf{C}_x^l$。总之，当$x \gg l$(比如$x = 2l, l \geq k = 160$)时，$\mathrm{Adv}_{\mathcal{P}}^{\mathrm{sk}}(\mathcal{A}) \leq \frac{1}{2^l} \leq \epsilon'(k)$。因此，协议$\mathcal{P}$可以用来构造达

到 $\epsilon'(k)$ 安全性的密钥交换协议$\mathcal{E}$，尽管性能不尽人意。

另一方面，根据引理1，在 Impagliazzo-Rudich 模型下不存在仅采用对称密码技术(换句话说，排除所有公钥密码技术)的安全性达$\epsilon'(k)$的密钥交换协议，协议 $\mathcal{E}$ 也不例外。这样矛盾就产生了。原命题假设协议 $\mathcal{P}$ 具有用户匿名性，而通过协议$\mathcal{P}$我们可构造安全的协议$\mathcal{E}$，因此原命题不成立。进一步回溯，双因子协议$\mathcal{T}$具有匿名性的假设也不成立。证明完成。 □

我们称上述原则为*匿名性的公钥技术原则*。根据这一原则，可以很容易发现，除前文讨论过的 Xue 等协议[295] 和 Fan 等协议[304] 外，所有企图仅采用轻量级对称密码原语(如Hash函数、MAC、分组密码和异或操作)实现匿名性的 WSNs 环境下双因子认证协议(如[181, 293–295, 297, 303, 304, 318–322, 322, 323, 356])，都无法实现所宣称的匿名性。尽作者所知，这些协议中的很大一部分近期才在线公开，除本文外还没有其它第三方进行过安全性分析。其中的一些，如文献[181, 303, 304, 318, 320, 321, 323, 356] 还提供了形式化安全证明，但由于违反上述协议设计基本原则，从本质上无法实现匿名性。

6.4.2 原则的普适性

值得一提的是，我们的上述原则对各类应用环境下的匿名双因子认证协议具有普遍适用性，比如单服务器架构[174, 350]、多服务器环境[177, 335]、传统移动网络[175, 329]、移动 IPv6 网络[353]、全球移动网络[357, 358]、卫星网络[331]等。主要原因在于，我们在证明这一原则时所采用的攻击者模型(即双因子认证攻击者模型，参阅定义9)可以直接迁移到其它的应用环境中。更具体地说，如果基于相同的攻击者能力假设，实现相同的隐私保护目标(即用户匿名性和不可追踪性)，无论协议是为哪种特定应用环境所设计的，那么协议均应采用相同的基础密码工具和具有相同的本质计算复杂度(即采用对称密码技术或公钥技术)。

如图 6.2 中虚线所标示的协议，它们仅采用了对称密码技术，但却宣称能实现用户匿名性，包括已经过充分分析的[173–178, 316, 329–332, 335, 350, 351, 353, 359–363] 和新近提出的[181, 318–321, 356]。根据我们的公钥技术原则，这些协议都存在问题。由于空间所限，许多重要协议未能包含在图 6.2 中。接下来，我们以 Chuang 等协议[353]为例，展示上述原则的普适性。感兴趣的读者可以将新近提出的协议(例如[181, 318–321, 356])作为案例，去分析验证这些协议是否无法实现匿名性。

Chuang等协议的回顾。 2013年，Chuang等[353]提出了一个匿名双因子认证协议，宣称在移动代理 IPv6 环境下能保护所有移动用户和其他网络实体，如移动接入网关(MAG)、本地移动锚(LMA)和认证、授权、审计(AAA) 服务器，

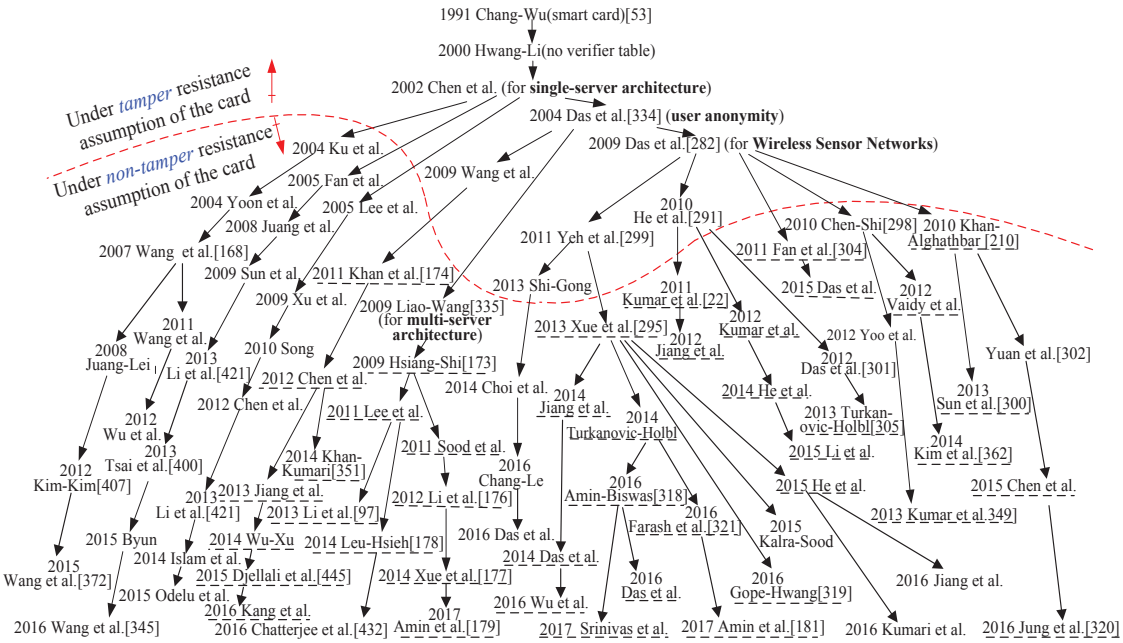


图 6.2 各类应用环境下匿名双因子认证协议的一个简要历史。

免受恶意攻击者的攻击。该协议采用与 Fan 等协议类似的符号(参见表6.1)，包含三个阶段：注册、认证和口令更新阶段。注册阶段，移动用户MN 向 AAA 服务器提交用户名 ID_{MN}，口令 PW_{MN}，AAA 服务器将 $\{ID_{MN}, c_1 = h(ID_{MN}\|sv), c_2 = h(PW_{MN}) \oplus c_1, c_3 = E_{PSK}(ID_{AAA}\|sv),\ c_4 = h(ID_{AAA}\|sv), c_5 = h(sv), h(\cdot)\}$ 写入智能卡，然后将智能卡通过安全信道发往MN。其中，ID_{AAA} 和 sv 分别表示服务器身份标识和长期密钥。认证阶段，MN插入智能卡读卡器，输入ID_{MN} 和PW_{MN}。智能卡验证ID_{MN} 和PW_{MN}，选取随机数N_1，计算 $AID_{MN} = ID_{MN} \oplus h(c_5\|N_1),\ Auth_{MN} = h(c_1\|N_1)$, 向 LMA 发送登录请求消息 $\{AID_{MN},\ c_3, E_{c_4}(Auth_{MN}\|N_1)\}$。协议后续部分与本章讨论无关，在此省略。

分析 Chuang 等协议。假设合法但不诚实用户$MN_{\mathcal{A}}$已在 AAA 注册。其他用户(包括受害者MN)与$MN_{\mathcal{A}}$一样，也已在 AAA 注册。$MN_{\mathcal{A}}$ 可以提取自己智能卡中参数 $\{c_4, c_5\}$。注意，在所有用户(包括MN)的智能卡内参数c_4 和 c_5 是相同的。随着边信道攻击技术和逆向工程技术[165, 324–327, 364]的不断进展，假设$MN_{\mathcal{A}}$可提取出智能卡内秘密参数是合理的。实际上，从2004年起，该假设已被广泛应用于分析双因子认证协议的安全性[278, 365, 366]。已知c_4 和 c_5，$MN_{\mathcal{A}}$ 可以通过以下步骤获得 MN 的用户名 ID_{MN}:

Step 1. 截获MN发送的登录请求消息 $\{AID_{MN},\ c_3, E_{c_4}(Auth_{MN}\|N_1)\}$;

Step 2. 解密 $E_{c_4}(Auth_{MN}\|N_1)$ 获得 $Auth_{MN}\|N_1$，其中， c_4 从$MN_{\mathcal{A}}$ 的智能卡中提取，与MN 智能卡中c_4 值相同。

Step 3. 计算 $ID_{MN} = AID_{MN} \oplus h(c_5 \| N_1)$，其中， c_5 可从 $MN_{\mathcal{A}}$ 的智能卡中提取，与MN 智能卡中c_5 值相等。

上述攻击表明，一旦攻击者$MN_{\mathcal{A}}$分析出自己智能卡内秘密参数，$MN_{\mathcal{A}}$可以简单通过窃听公开(无线)信道中的通信信息，来获得任意合法用户的用户名ID_{MN}。值得注意的是，$MN_{\mathcal{A}}$ 只需提取自己智能卡内的秘密参数，无需获得受害者的智能卡，她甚至可以请求专业组织(实验室)援助。从这一点来看，上述攻击非常现实可行。显然，如移动用户MN的用户名 ID_{MN} 泄露，则$\mathcal{A}$可追踪MN的活动，如活动轨迹，生活方式，甚至当前位置[367]。这就解释了为什么用户匿名性是移动计算环境下身份认证协议的重要属性。总之，上述攻击说明了 $\mathcal{A}$ 破坏 Chuang 等协议[353]用户隐私的现实性，同时显示了“匿名性的公钥技术原则”的普适性。

6.5 潜在的实现方案

上一节证明了匿名性的公钥技术原则，并显示其普适性，本节进一步讨论如何实现这一原则。由于对称密码技术不足以实现用户匿名性(不可追踪性)，可行的选择只能诉诸公钥技术。关键问题是，如何将公钥密码原语融合到传统的基于对称密码技术的协议中？

6.5.1 预装载伪IDs池

为实现用户匿名性，一个比较直观的想法是，在用户智能卡中预加载一个由服务器私钥签名的伪身份池(Pesudonym ID pool)，用户每次登录服务器时使用一个新的伪ID，伪ID不重复使用。然而，这种方法只适合用户拥有大存储容量的设备[368]，而不适合资源受限的智能卡。更重要的是，一旦受害者移动设备(智能卡)丢失，所有的伪 ID 将泄露，危及整个系统。

6.5.2 群签名和环签名

事实上，早在1991年文献[369]率先通过引入群签名技术来实现用户匿名性。群签名方案允许任一群成员代表整个群进行匿名签名，而不会泄露自己的身份，群签名的验证者无法分辨出群中真实签名者。用户匿名性也可以通过环签名技术实现[370]。在环签名方案中，签名者可以用环中其他成员的公钥签名而无需其他成员同意，生成的签名可以使验证者(包括群管理员)相信该消息确实来自环内某一成员，但验证者不知道谁是真正的签名者。

群签名和环签名采取本质上同样的策略确保用户匿名性，即：将成员的真实身份隐藏在一个群体之中。二者的主要区别是：(1)环签名的组织结构更自

由；(2)环签名对任意验证者(包括管理员)而言都是匿名的，而群签名中验证者无法分辨真实的签名者，但管理员可以确定谁是签名者。换句话说，每种签名方式对匿名身份认证都各有优势。因为需要公钥基础设施(PKI)为每个成员配备一个证书，故群签名和环签名都不适于双因子身份认证协议。现有环签名或群签名方案的计算复杂度，一般随群或环大小的增加而线性增长。当用户数目较大时，这两种方案明显不适于匿名双因子认证。

6.5.3 IND-CCA2公钥加密技术

幸运的是，近期几个协议[344, 354, 366, 371–373]采用不同的公钥加密技术实现了匿名性，同时保持了安全性和效率。其中，文献[354, 366, 371, 372]实现了可证明安全性。虽然这六个协议分别为不同应用环境设计(包括WSNs)，但它们都有一些共同点：即服务器拥有公私钥对，用户智能卡中存储服务器公钥，用户记忆一个(弱)口令，用户登录时用服务器的公钥加密传输真实用户名。通过这种方式，使用户身份保护和不可追踪性这两个匿名性目标建立在**NP**难问题之上，如大合数因子分解问题(IFP)、离散对数问题(DLP)、计算 Diffie-Hellman 问题(CDHP)，以及它们的变种如二次剩余问题(QRP)、双线性计算 Diffie-Hellman 问题(BCDHP)。但是，这些协议都没有明确解释为何选用计算复杂度相对较高的公钥技术，而不是轻量级的对称密码原语。幸运的是，在节6.4 中我们终于回答了这一关键问题。

表 6.3 典型公钥密码原语操作在传感器节点和智能卡上的运算时间(单位：秒)

Experimental platform	RSA Enc/Decryption ($\|n\|=1024, e=2^{16}+1$)	Modular Exponentiation ($\|n\|=512$)	Point Multiplication ($E(\mathbb{F}_p), \|p\|=160$)	Tate Pairing ($E(\mathbb{F}_p), \|p\|=1024$)
Mica2 ATmega128L 8-bit 4MHz	0.862/22.029	2.679	1.616	45.632
MicaZ ATmega128 8-bit 8MHz	0.430/10.990	5.370	0.810	22.765
Philips HiPerSmart 32-bit 36MHz	0.006/ 0.140	0.068	0.011	0.290

此外，这些工作[344, 354, 366, 371–373]对协议设计过程中各种策略选择背后的原理探讨不够充分。若将研究重点向设计原理上倾斜一下可能会更好，而不是仅仅宣称“这样那样协议就实现了用户匿名性(such and such a protocol achieves user anonymity)”。为进一步说明这一点，本节采用文献[366]中匿名认证协议作为示例。该协议由Zhou和Xu在2011年针对全球移动网络漫游服务而提出。Zhou-Xu首先提出协议，之后给出形式化(复杂却很常规)安全证明，但没有给出该协议如何(和为什么)可以实现用户匿名性的基本思想。这里我们填补这一空白。在 Zhou-Xu 协议中，用户U_i的伪用户名SID_i采用了Elgamal算法的一个变形进行加密：令$SID_i = ID_i \oplus h(D \parallel n_i)$，其中，$D = B^a \bmod p$，$a$ 为U_i选取的随机数，$B(= g^b \bmod p)$ 为内部服务器的公钥，g 表示阶为p 的循环群的生成

元。U_i向服务器发送登录请求 $\{SID_i, n_i, A = g^a \bmod p, V = h(C\|D), ID_H\}$，其中，$C$表示主服务器分发给$U_i$的静态凭证，$ID_H$ 表示主服务器身份标识。

如果没有错的话，Zhou-Xu协议[366]的设计原理如下：基于CDH难解性问题，如果给定三元组(g, g^a, g^b)，只有U_i和主服务器可以计算 $D = g^{ab}$，从伪用户名SID_i中恢复出真实用户名 $ID_i = SID_i \oplus h(D \| n_i)$。同样，只有$U_i$和主服务器可以计算用户的动态凭证 V 来仿冒U_i。由于D具有随会话的动态可变性，所有登录请求消息中可以被用来推导出U_i的特定信息的元素都是会话可变的，因此可以保障用户不可追踪性。文献[354, 372] 为单服务器架构设计的协议和文献[344]为多服务器架构设计的协议采用了同样的策略，即通过公钥密码算法在会话消息中隐藏用户名和长期凭证。

这使我们想起了Fujisaki和Okamoto[374]在1999年的先驱性工作。我们意识到，上述协议[344, 354, 366, 371, 372]的做法实际上等同于采用了适应性选择密文攻击不可区分(IND-CCA2)的公钥加密算法。更确切地说，通过采用抗碰撞的Hash函数和Fujisaki-Okamoto[374]提出的消息填充机制，上述协议[344, 354, 366, 371, 372]中应用的公钥加密算法(即Elgamal)可成功升级为随机预言机(ROM)模型[375]或真实-随机模型(ROR模型)[57]下达到 IND-CCA2 安全性的加密算法。这就实现了安全加密的概念。

至于Son等[373]2012年提出的协议，它也可以实现ROM模型下可证明安全性。其实现匿名性的底层原理类似于文献[344, 354, 366, 371, 372]，不同之处在于Son等协议采用了Rabin加密算法和Boneh[376]提出的消息填充机制。由于在用户端仅需要执行一次模乘运算，该协议相比[344, 354, 366, 371, 372](至少需要2-3次模幂运算)更高效，更适于移动应用环境。

在ROM模型下安全的协议，并不意味着该协议在标准模型(即真实世界)中也是安全的。文献[377]讨论了ROM模型的局限。尽管如此，在(条件)非抗窜扰智能卡假设下，不需要借助笨重的PKI，仅利用现有的公钥加密算法就可以构造高效的匿名双因子身份认证协议，并在ROM模型下实现可证明安全性，这仍是鼓舞人心的。只要“随机预言机(ROM)模型是验证协议安全性的有效工具”，ROM 模型下的可证安全性远比纯启发式分析下的安全性更加可靠。

此外，尽管这些协议(见[344, 354, 366, 371–373])最初都不是为 WSNs 环境而设计，它们提供了非常有力的证据表明可以利用相同的密码原语实现 WSNs 环境下安全高效的匿名双因子认证协议。表 6.3 总结了文献[378–381]的研究结果，证实了在资源受限的传感器和智能卡上应用公钥操作(如RSA加密、椭圆曲线点乘和双线性对)的计算可行性。

6.6　本章小结

本章研究了基于口令的双因子认证协议实现匿名性的本质计算复杂度问题：在(条件)非抗窜扰智能卡假设下，能否仅基于轻量级对称密码技术构造匿名双因子身份认证协议？为解决这一基础性问题，本章首先以两个WSNs 下匿名双因子认证协议为例，显示实现匿名性的难点和挑战；然后，基于Halevi–Krawczyk (1999)和Impagliazzo–Rudich (1989)这两个经典结果，在广泛接受的攻击者模型下，我们提出并成功证明了一个协议设计原则：在(条件)非抗窜扰智能卡假设下，公钥密码技术是实现匿名性的基本组件；接着，显示这一协议设计原则具有普适性，适于双因子协议的各类应用环境。针对“公钥技术对实现用户匿名性不可或缺，但智能卡(和传感器节点)是资源受限设备”这一矛盾，我们进一步探讨了可行的解决方案，指出现有公钥加密算法结合适当的消息填充机制有望较好地平衡这一矛盾。

尽我们所知，本章是探索匿名双因子认证协议设计基本原理的首次尝试。我们的“公钥技术原则”揭示了过去匿名双因子认证协议设计过程中一再失败的根本原因，有助于协议设计者和安全工程师理解保护用户隐私所需的密码学本质计算复杂度，为未来设计安全高效的匿名双因子认证协议提供重要指导。下一步值得研究的课题是，如何根据本章提出的原则和方法，设计适于移动互联网、全球移动网络、卫星网络等环境下的安全高效的匿名双因子认证协议。

第七章　基于口令的双因子认证协议评价指标体系

协议评价指标体系是衡量一个协议优劣的准绳，一般包括安全模型和评价标准两个方面。安全模型是对攻击者能力的现实假设，是分析评价标准中各具体评价指标是否实现的基石；评价标准是对协议应实现的一系列目标的客观规范，一般包含安全目标(如抗智能卡丢失攻击)、功能属性(如第六章中的用户匿名性)和效率三个维度，并且缺一不可。这两者都严重依赖于具体的应用环境。

对于双因子协议来说，除了效率维度应主要评测计算量、存储量、通信量三方面已达成共识外，其它两个维度的具体评价内容目前仍未形成统一的认识。近十年来虽有多个协议评价标准(见[151, 170, 278])被提出，也涌现了数以百计的双因子协议(如[289–294, 354, 382–386])，但绝大多数情况下，协议设计者们并未采用这些第三方标准作为衡量尺度，而是为自己的协议自定义一套评价标准。这往往导致协议评价不够客观，虽被宣称“安全高效”，但实际上却漏洞频出，使双因子协议研究进入“攻击→ 改进→ 攻击→ 改进”的怪圈(见图1.12)。

在这个怪圈里，虽然学者们投入了大量的精力和努力来设计新协议，但实质性研究进展却很少，很大程度上是在原地踏步[170, 345, 387]。产生这种现状的重要原因是，关于协议评价指标体系自身的研究严重匮乏。本章发现：(1)当前仅有的几个评价标准(见[170, 278, 388–390])都存在严重缺陷(如冗余和歧义)，不适于应用；(2)一些研究(如[170, 389, 390])仅关注评价标准而忽略了与之相辅相成的安全模型，在定义具体评价指标时，未明确安全模型，使指标的含义不明确。为此，我们明确地定义了一个迄今最严格的攻击者模型，并在综合现有各评价标准优点的基础上，提出一个含有 12 项指标的新标准，这二者共同构成一个新的协议评价指标体系。最后，通过对 67 个典型双因子协议的评估，显示新评价指标体系的有效性。本章将为第八章奠定理论基础。

7.1　研究背景

身份认证是保障信息系统安全的第一道防线。在众多的身份认证方法中，基于口令的身份认证(PAKE)由于其简单易用、方便扩展、低价兼容的特点，获得了广泛的应用[391, 392]。在此类认证协议中(其中著名的如SRP[393]，KOY[21]，J-PAKE[22])，每个用户拥有一个易记忆的低熵口令，而认证服务器需存储一个口令相关的验证表项。当用户登录时，检查此验证表项以证实用户身份的合法性。正是这一口令验证表项成为基于口令的单因子认证机制的一个根本缺陷：一旦认证服务器被攻陷，所有用户的口令将面临泄露的严峻危胁。

受记忆能力和安全意识的局限，普通用户选择的口令服从严重偏态的Zipf分布(见第二章)，并且往往与个人信息紧密相关(见第三章)。因此，即使服务器采用了加盐方式存储口令，攻击者也可以使用先进的基于机器学习的破解算法(如[34, 197])和常见的硬件(如GPU)以极高成功率恢复出明文来(见[36])。在Password'12上，Gosney[35]展示一台含有25 个GPU的PC，可以每秒钟离线猜测3500次经传统哈希函数MD5处理过的口令密文。网站即使采用了专用的口令哈希函数(如bcrypt和PBKDF2)，也仅仅一定程度上降低了攻击者的猜测速率[244]：服务器为验证用户需要增加与攻击者相同倍数的运算成本，但攻击者更可能配备了更好的专用口令破解设备。

近年来，数以亿计的用户口令遭遇泄露已不再是新闻。近期知名的口令泄露事件包括：Yahoo (5亿)[242]，MySpace (3.6亿)[240]，Adobe (1.5 亿)，LinkedIn (1.2亿)，Evernote (5000万)，Anthem (4000 万)[229]，000webhost (1500万)[394]等。在过去 5 年里，一些服务商(如Anthem[229]和Phpbb[261])甚至不止一次被攻陷，用户口令多次遭遇泄露。更糟糕的是，如第五章所证实的，用户倾向于在多个服务中重用(或稍作修改)已有口令。一旦某个服务出现口令泄露，用户在其它服务中的帐户也会受到严重危胁，这被称为口令重用的“多米诺效应”。

为解决服务器被攻陷导致的口令文件泄露问题，最近一些学者提出在PAKE协议中引入门限密码学技术[247]，即把口令文件和用户数据分布在多个服务器上。因此，当攻击者$\mathcal{A}$攻陷的服务器数量未达到预定阈值时，$\mathcal{A}$ 无法猜测口令。第四章中也提出了通过honeywords 技术来及时检测口令文件的泄露，并阻止$\mathcal{A}$从口令文件中找出真口令。但是，这两项技术在本质上无法解决另一问题—用户端口令泄露(如肩窥、键盘记录和网络钓鱼)。为防止用户端口令泄露，文献[395]提出了抗泄露口令系统(LRPS)的概念，该系统要求用户间接地输入口令，这带来可用性问题。最近，文献[396]指出，为实现安全性和可用性的合理平衡，LRPS方案必须在用户端采用可信设备来消除口令泄露威胁。

由于门限技术、Honeywords技术和LRPS方案的局限性，第四种解决办法的地位得到增强—引入智能卡作为“第二道防线”，即采用基于“口令+智能卡”认证方式，也称为“双因子认证”。这种认证方式在20多年前由Chang等首次提出[53]，现已被广泛应用于各种安全攸关的应用中，如电子商务、电子银行和电子医疗，并且也是三因子认证的基石[60, 214]。如图7.1所示，此类认证协议主要包含两个参与者，用户U和服务器S[278]。用户注册阶段，用户U向服务器S提交自己选择的个人数据(如用户名和口令)，然后S 颁发给U一个包含安全参数的智能卡。此后，U和S通过登录阶段实现相互认证。此外，用户可能需要定期更新口令，涉及到口令更新阶段。注意，利用短消息服务(SMS) 作为第二认证信道的情形，因为协议需额外涉及电信服务商，不在本章的研究范围。

双因子认证协议最核心的安全目标是实现“真正双因子安全性”[279]，即当且仅当攻击者同时拥有智能卡和相应正确的口令时才能登录服务器。换句话说，一个实现了“真正双因子安全性”的协议应满足以下要求：(1)攻击者拥有用户智能卡(且已分析出智能卡内参数)，仍无法通过离线字典攻击猜测出用户口令或仿冒用户；(2)攻击者已知用户口令，但无用户智能卡，仍不能仿冒用户。除了双因子安全性，一个安全的协议还应能抵抗其它各种主动攻击和被动攻击[151, 397]，例如窃取验证项攻击、拒绝服务攻击、反射攻击和平行会话攻击等。除了安全目标，一个实用的协议还应保持较高的效率并实现一些重要属性[170, 398]，如口令友好性、会话密钥协商和用户匿名性。

7.1.1　研究动机

2012年以前，学者们提出了数以百计的双因子认证协议(知名的如[221, 278, 289, 399, 400])，随之一些协议评价标准也陆续面世(如[278, 388–390])。然而，绝大多数情况下，协议设计者们在评估新协议时，很少采用第三方评价标准，往往自行定义一套评价标准，强调新协议的先进、优越之处，却(可能无意识地)忽略其不足之处。由于这些新提出的协议缺乏全面系统的评估，协议之间也缺乏客观公平的对比，导致了很多宣称“安全高效”的协议在提出不久即被攻破，宣称“更安全”的协议却常常与原协议一样脆弱甚至更脆弱(详见文献[345])。为解决这一严峻问题，2012年，Madhusudhan和Mittal[170]在现有评价标准研究成果(见[278, 388–390])的基础上，提出了一个新的评价标准，包含9个安全目标和10个理想属性。他们指出，现有的双因子协议都存在这样或那样的问题，尚没有一个可完全实现他们所提出的19个评价指标。

Madhusudhan-Mittal评价标准提出后，学者们又陆续提出了大量的新协议(如[221, 354, 385, 386, 400, 401])来尝试满足这一标准。不幸的是，这些新协议随后都被指出无法实现宣称的目标(见[345, 397, 402–404])。这种研究模式再次陷入了难以令人满意的“攻击→ 改进→ 攻击→ 改进”的怪圈(见图1.12)。在这个怪圈中，协议设计者们投入大量的精力试图通过改进现有协议来实现一个理想的协议，而协议分析者们则希望通过揭示新协议存在的具体缺陷来助于修补之。虽然在协议设计和分析这两方面投入了大量的研究精力，但实质性进展却很少。与之相对的是，关于协议评价指标体系的研究投入却严重不足。

一方面，鲜有研究关注当前这些协议评价标准(见[151, 170, 278, 388–390])的有效性，缺少对这些标准之间进行对比和综合分析。比如，迄今没有研究关注一个基本的问题：这些协议的失效到底是因为协议设计者们某方面的具体设计不当，还是因为Madhusudhan-Mittal评价标准本身存在某些内在的局限，使得我们无法设计出“一个理想的协议”？换句话说，在现有密码学工具下，构建一

个满足文献[170]中列出的所有19个评价指标的理想协议是否“可能”？尽我们所知，在现有的协议评价标准中，Madhusudhan-Mittal标准[170]最明确、全面和系统。如本章将要指出的，Madhusudhan-Mittal标准的一些评价指标定义仍不清晰，指标间存在冗余和歧义，从而导致可操作性差。比如，仅仅智能卡丢失攻击，我们发现至少有8种可能的攻击情形，无法抵抗其中任意一种就会导致无法实现最核心的双因子安全性，但是此前的评价标准都没有考虑这一点。

另一方面，协议评价标准的研究离不开对所基于的具体敌手安全危胁模型的讨论。也就是说，协议评价标准无法离开具体敌手模型而单独存在[151, 279]，对这一点认识的偏差正是过去众多协议评价标准(见[170, 389, 390])失效的重要原因。举例示之。假设一个双因子协议$\mathcal{T}$，部署在两个不同的环境中，分别对应敌手模型$\mathcal{A}_1$和模型$\mathcal{A}_2$。其中模型$\mathcal{A}_1$假设智能卡内参数可被攻击者分析出来，模型$\mathcal{A}_2$假设智能卡是抗窜扰的(亦即智能卡无法被攻陷)。对于一个双因子认证协议评价标准SE，SE中的指标SR4(即抗智能卡丢失攻击)是任何双因子协议都应当满足的一项基本安全目标。不难发现，一个协议(如[174, 405])可能在敌手模型$\mathcal{A}_1$下无法实现SR4，但在敌手模型$\mathcal{A}_2$下则可成功实现SR4。可见，离开具体的敌手模型($\mathcal{A}_1$或$\mathcal{A}_2$)来讨论协议评价标准缺乏实际意义。

如果这两方面的基础性问题没有解决，我们只能在原地踏步：尽管学术界投入了巨大的研究力量，数以百计的新协议不断被提出(随后又被证实不安全)，整个领域的研究却无实质性进展。

7.1.2 本章贡献

本章通过研究攻击者模型和理解现有评价标准的不足，为打破“break-fix-break-fix”循环迈出实质性一步。我们明确地定义一个迄今最为严格但现实的攻击者模型，并提出一个包含12项独立指标的评价标准，这二者共同构成一个系统化的协议评价体系。我们的评价体系虽然可能不是完美的，但可以预期，这一体系及其明确的定义将为后续研究打下坚实基础。总之，本章贡献如下：

- 分析了近期提出的两个典型协议，揭示在实现双因子协议评价指标时面临的挑战和微妙之处。这两个协议都声称能够抵抗各种已知的攻击并实现了众多理想属性，然而分析结果表明，它们都无法满足一些重要的设计目标。值得一提的是，我们确定了Tsai等形式化安全证明中存在的根本失误之处，并突出了两个重要的现实威胁：智能卡丢失攻击和异步攻击。其中，后一攻击专门针对匿名双因子协议，一批新近提出的协议(如[289, 358, 390, 406, 407])无法抵抗这一攻击。
- 探讨了Madhusudhan-Mittal标准的各项指标间关系，指出其中存在的歧义和冗余，揭示在设计双因子协议时必须考虑的内在冲突和不可避免的平

衡。我们的结果表明，在当前广泛接受的攻击者模型下，三项评价指标仅在现有密码工具下无法同时实现。这意味着数以百计宣称同时实现了这些指标的协议本质上是不可行的。

- 提出了一个新的双因子认证协议评价指标体系，包括一个现实的攻击者模型和一个系统全面的评价标准。对比分析表明，本章攻击者模型比现有模型更为严格；新评价标准克服了现有标准的缺陷，内涵更全面，各指标定义也更具体明确。最后，基于 67 个典型双因子协议的评估测试结果，显示了我们新指标体系的有效性和实用性。
- 思考了形式化方法在打破“攻击-修复-攻击-修复”这一怪圈中可能承担的角色及其不足，指出传统启发式分析技术在确保复杂密码协议的安全性和理想属性方面仍具有重要作用。

7.2　系统架构、攻击者模型和评价标准

本节描述系统架构，定义一个严格但现实的攻击者模型，并介绍Madhusudhan-Mittal评价标准[170]。

7.2.1　系统架构

与文献[279, 397]一样，本章主要关注基于“口令+智能卡”的这种应用最为广泛的双因子认证协议(见图7.1)。协议的参与者包括一组用户和一个远程服务器。通常，这类协议包含三个基本阶段(即注册、认证和口令更新)，以及一些辅助阶段如用户移除和智能卡撤销等[398]。在注册阶段，用户向服务器提交一些个人信息，服务器颁发用户一张智能卡。智能卡中会含有一些公开的和敏感的安全参数，这些参数在随后的认证阶段会用到。注册阶段只执行一次，直到用户重新注册。注册阶段完成后，用户可通过认证阶段访问服务器。认证阶段可按需多次执行。如节7.1提到的，一个真正的双因子协议可确保：用户只有同时拥有智能卡和相应正确的口令才能成功登录。在口令更新阶段，用户可在本地或在线与服务器交互来更新口令(及智能卡中的信息)。为移除恶意用户和撤销失效的智能卡，一个实用的协议还可能包含用户移除和智能卡撤销阶段。

图 7.1 基于口令的双因子认证协议。

7.2.2 攻击者模型

在传统的口令认证密钥交换协议(PAKE)中，一般假设攻击者$\mathcal{A}$完全控制通信双方的通信信道[56, 375]，即可以任意窃听、拦截、插入、删除和修改公开信道中的消息。为刻画前向安全性这一概念，允许$\mathcal{A}$攻陷一方实体并获得其长期密钥。此外，$\mathcal{A}$可能获得过期会话密钥(例如因未成功擦除)。

尽管这些假设对PAKE协议是合理的，但对双因子认证来说，它们不足以刻画攻击者的现实能力。最新研究表明，智能卡内秘密参数信息可以通过边信道攻击攻击分析出来，如能耗分析[324, 408]、软件漏洞攻击(如针对软件支撑的Java卡)[364]和逆向工程[165]等。如果智能卡内敏感信息泄露，原本安全的协议将不再安全，易遭受离线口令猜测攻击(见[400, 409])和仿冒攻击(见[150, 410])。因此，在非抗窜扰智能卡假设下设计和分析协议更为可取。2012年，Wang等[151]首次探讨了存在恶意读卡器的情形，指出恶意读卡器也会破坏协议安全性—窃取用户输入的口令。一旦读卡器被攻击者控制(如读卡器被植入病毒/木马)，用户输入的口令会被攻击者窃取。

我们限制$\mathcal{A}$“先通过恶意读卡器获取用户输入的口令，然后通过某种方式获取(如窃取或拾到)用户的智能卡并分析出卡内秘密参数”这一平凡攻击情形。否则，这一情形将使$\mathcal{A}$有能力攻破任一双因子协议。这一限制符合“极端攻击者原则”[411]：对攻击者的唯一限制应是使其不具备平凡攻击的能力，一个强健安全的协议应能抵御这种极端攻击者。在现实中，这一限制是合理的：(1)当用户使用恶意读卡器时，由于用户在现场，$\mathcal{A}$无法实施往往需要特殊工具和平台支持的边信道攻击而不被用户察觉；(2)用户使用恶意读卡器的时间一般较短(如几分钟)，而边信道攻击往往需要较长时间(如数小时甚至更长)。

这意味着，针对智能卡的“非抗窜扰假设”是条件性的，即仅当$\mathcal{A}$持有智能卡的时间足够长时，才能分析出卡内秘密参数，在其它情况下(比如用户将智能卡插入到恶意读卡器)智能卡仍然是抗窜扰的。但是，当在恶意读卡器使用USB存储卡时，用户口令和USB卡内数据可轻易同时被$\mathcal{A}$获取。这显示了智能卡相比USB存储卡的核心优势：在恶意读卡器存在的情况下，基于智能卡的双因子协议将比基于USB存储卡的双因子协议更安全，即使假设智能卡是非抗窜扰的。这很好地解释了，为什么绝大多数安全攸关的服务选用智能卡，而不是价格更低廉的USB存储卡。

上述分析也否定了文献[278, 365]中过于保守的假设：“在协议中只将智能卡视为一个嵌入了微处理器的、可按需执行操作的存储卡”，“不考虑智能卡所能支持的任何特殊的安全特性”。由于基于存储卡的协议在不可信终端上操作是彻底不安全的，那些采用上述过于保守假设的智能卡认证协议(如[278, 365])也是彻底不安全的，在不可信终端上无法提供真正双因子安全性。因此，本章提出

的“条件”非抗窜扰智能卡假设比文献[278, 365]的假设更为合理。

表 7.1 攻击者能力

C-01	The adversary $\mathcal{A}$ can enumerate offline all the items in the Cartesian product $\mathcal{D}_{id} * \mathcal{D}_{pw}$ within polynomial time, where $\mathcal{D}_{pw}$ and $\mathcal{D}_{id}$ denote the password space and the identity space, respectively.
C-02	The adversary $\mathcal{A}$ has the capability of somehow learning the victim's identity when evaluating security strength (but not privacy provisions) of the protocol.
C-1	The adversary $\mathcal{A}$ is in full control of the communication channel between the protocol participants.
C-2	The adversary $\mathcal{A}$ may either (i) learn the password of a legitimate user via malicious card reader, or (ii) extract the sensitive parameters in the card memory by side-channel attacks, but cannot achieve both.
C-3	The adversary $\mathcal{A}$ can learn the previous session key(s).
C-4	The adversary $\mathcal{A}$ can learn the server's long-time private key(s) only when evaluating the resistance to the system's eventual failure (e.g., forward secrecy).

为提高用户友好性，协议一般都允许用户在注册阶段自行选择用户名(有时可能需要满足既定格式)。用户为方便记忆，通常倾向于选择低信息熵的用户名，导致用户名空间$\mathcal{D}_{id}$十分受限：$|\mathcal{D}_{id}| \leq |\mathcal{D}_{pw}| \approx 10^{6}$[5]。因此，假设$\mathcal{A}$可在多项式时间内离线穷举笛卡儿空间$\mathcal{D}_{id} * \mathcal{D}_{pw}$是合理的。相比之下，很多用户匿名协议(如[410, 412])明确假设$\mathcal{A}$不能同时猜测出正确的ID和PW。这类协议在本章安全模型下将不再安全。

表7.1总结了我们的攻击者的能力。我们关于攻击者的工作建立在文献[151, 278, 413]的基础之上，但提出了新的见解，是少数几个明确定义了攻击者能力的工作之一。由于一个协议只能在某个(些)特定安全模型下是“安全”的，离开一个明确的安全模型，来公平地评估协议的安全性几乎是不可能的。文献[151]中，Wang提出了三个安全模型，称为Type I~III。其中，Type III是现有安全模型中最为严格的一个，它主要包含三项假设：

(1) 假设$\mathcal{A}$完全控制通信信道，这与表7.1中的C-1 相一致；
(2) 假设智能卡是非抗窜扰的，$\mathcal{A}$也可以利用恶意读卡器窃取用户口令，但二者不能同时实现，这与表7.1中的C-2相一致；
(3) 假设智能卡无计数器保护，即$\mathcal{A}$可以利用恶意读卡器向智能卡发起大量查询请求，以获取有用信息。

对于假设3，其现实意义不大。这是因为在假设2下，假设3成立与否都与协议安全性没有关系。一方面，如果在智能卡运行之前不对口令的正确性进行验证，则$\mathcal{A}$ 只能通过在线与远程服务器交互来获得有用信息(除假设2中提到的智能卡内存储的参数信息)，而服务器可执行很多成熟的措施来阻止$\mathcal{A}$的在线请求，如账号锁定和异常登录检测。另一方面，如果在智能卡运行之前对口

令的正确性进行验证而无计数器保护，$\mathcal{A}$可以在恶意读卡器上大量尝试输入潜在口令，从而猜测出正确的口令，并且可根据假设2可以分析出智能卡内参数，而这是Type III 模型明确禁止的。因此，我们的安全模型不考虑假设3。如文献[412]，我们简单假设智能卡存在计数保护，即一旦查询次数超过某一阈值，智能卡将被锁定一段时间(如GSM SIM卡V2 或后续版本具有此功能)。

综合来看，本章提出的模型与文献[151]中的Type III模型最为接近，关键区别在于Type III 模型中$\mathcal{A}$不具备C-3和C-4的能力。因此Type III模型将无法处理一些重要的安全概念，如前向安全性和抗已知密钥攻击等。与Li-Lee模型[413]和Yang等模型[278]相比，本章模型明确考虑了存在恶意读卡器的情形，并且允许$\mathcal{A}$具备C-3和C-4的能力。

此外，本章模型假设$\mathcal{A}$能够在多项式时间内离线穷举笛卡儿空间$\mathcal{D}_{id} * \mathcal{D}_{pw}$中的$(ID, PW)$对，这使得它可以在匿名协议中刻画一些特殊的安全概念，如抗离线(ID, PW)对猜测攻击(如协议[410, 412]无法抵抗此攻击)和抗不可检测在线口令猜测攻击(如协议[414]无法抵抗此攻击)。文献[151, 278]虽然隐式地假设了C-01，但它们都未考虑用户匿名性，而只有当考虑用户匿名性时，将C-01从C-00中分离出来才有意义。总之，我们的安全模型不仅涵盖了现有模型里的合理假设，还包含了新的现实假设(即$\mathcal{A}$的计算能力虽强大但并非无所不能)，因此更严格、实用。特别地，即使在这样一个严格的安全模型下，如第八章所示，我们仍可以设计出安全高效的协议。这显示了我们模型的实用性。

7.2.3 Madhusudhan-Mittal 评估标准

2012年，Madhusudhan和Mittal[170]指出之前的协议评价标准(如[388, 389])存在歧义和冗余，并提出一个新的标准来评估协议的优劣。Madhusudhan-Mittal 标准是对之前评价标准的精化，包含9个安全目标(见表1.8)和10个理想属性(见表1.9)，读者可参见[170]来了解每个指标的具体定义。本节我们只揭示该标准的一些微妙之处，其指标间的内在关系将在节7.6中进行分析。

一个协议只有满足Madhusudhan-Mittal 标准中全部 9 个安全目标和 10 个理想属性，才能称之为“理想的”协议。Madhusudhan和Mittal[170]在提出上述标准后，分析了6个近期提出的协议，发现没有一个协议能够完全满足他们的19个指标。因此，他们得出结论：设计一个理想的协议仍是一个公开问题。

Madhusudhan-Mittal评价标准优于其它标准的原因有二：(1)其安全目标基于非抗窜扰智能卡假设，如果考虑到越来越现实的边信道攻击，这一假设显然是可取的。(2)安全目标和理想属性的分离让他们的标准更明确、具体，因此更具可操作性，有助于协议设计者们对协议进行系统性的评估。因此，本章使用Madhusudhan-Mittal的标准[170]作为现有评价标准(如[278, 388–390])的代表。

为便于理解，这里阐明Madhusudhan-Mittal指标定义中两个微妙之处，而它们在原文献[170]中没有明确说明。第一，只有在讨论安全目标SR6时，才假设$\mathcal{A}$可获得受害用户的智能卡，而在讨论其它安全目标(如SR2和SR4)时应假设$\mathcal{A}$未获得智能卡。第二，在基于口令的身份认证协议中，用户匿名性(即DA8)通常包含两个层次的含义(见第六章和[289, 415])：身份标识保护和用户行为不可追踪性。这两个含义都是针对公开信道上的攻击者而定义，而不是针对服务器而言，因为服务器需要证实用户的合法性，获取用户的真实身份以便实现审计、计费等管理目的。这意味着，当服务器已被攻陷时，用户匿名性将不再是一个有价值的设计目标。

7.3　对Tsai等协议的分析

2013年，Tsai等[400]指出Li等协议[289]存在严重的安全漏洞，并提出了一个新的协议。他们声称新协议实现了用户的不可追踪性，并能抵抗常见的攻击。然而，我们将指出，在Tsai等[400]自己的假设下，他们的协议无法提供真正的双因子安全性，而这是任何双因子协议都应实现的最核心目标。此外，由于忽视了可用性和安全性之间内在的平衡，此协议也无法抵抗智能卡丢失攻击。

7.3.1　回顾Tsai等协议

本章简要地描述Tsai等[400]在2013年提出的协议。他们的协议有两个版本：一个不涉及生物认证因子，另一个涉及。实质上，涉及生物因子的版本是一个三因子协议，然而Tsai等并没有讨论当引入第三种认证因子(即生物特征)后带来的安全问题和挑战。基于Huang等的工作[15]，不难发现Tsai等三因子版本的协议缺乏生物特征识别误差容忍和存在不可信终端攻击等问题。此外，双因子版本协议是三因子版本协议的基石。因此，我们主要关注其双因子版本协议。该协议包含五个阶段：参数产生、注册、预计算、登录和口令更新。为便于表示，表7.2列出了一些直观的符号及其定义。

1) **参数生成阶段** 该协议基于椭圆曲线密码系统(ECC)。首先，服务器S在有限域$\mathbb{F}_p$上选择一条椭圆曲线E_p，设G为其上一个阶为n的点，私钥为x。随后，S计算$P_S = x \times P$作为公钥，选择两个哈希函数$h(\cdot): \{0,1\}^* \to \{0,1\}^k$和$h_1(\cdot): \{0,1\}^* \to \{0,1\}^{k'}$，其中$|k| = |x|$，$|k'| \geq |k|$，例如$|k| = |x| = 128$比特，$|k'| = 256$比特。最后，$S$公开安全参数$\{p, E_p, G, P_S, n, h(\cdot), h_1(\cdot)\}$。

2) **注册阶段** 当U_i向服务器S注册时，需要执行如下操作：

(1) U_i选择身份标识ID_i、口令PW_i和随机数$b \in_R \mathbb{Z}_p^*$。

(2) $U_i \Rightarrow S: \{ID_i, h(PW_i \| b)\}$。

(3) 一旦收到来自U_i的注册消息，S计算$V = h(ID_i \| x) \oplus h(PW_i \| b)$。

表 7.2 符号定义

符号	定义
U_i	i^{th} user
S	remote server
ID_i	identity of user U_i
CID_i	dynamic identity of user U_i
PW_i	password of user U_i
x	the secret key of remote server S
O	the point at infinity
P	base point of the elliptic curve E_p where $n \cdot P = O$
$h(\cdot), h_1(\cdot)$	common collision free one-way hash functions
$\oplus$	the bitwise XOR operation
$\|$	the string concatenation operation
$\rightarrow$	a common communication channel
$\Rightarrow$	a secure communication channel

(4) $S \Rightarrow U_i$: 安全参数V。

(5) 收到V后，U_i向智能卡中输入b和V。

3) **预计算阶段** 智能卡选择$N_{C1} \in_R \mathbb{Z}_p^*$，计算$e = N_{C1} \times P$和$c = N_{C1} \times P_S$，然后存储$\{e, c, N_{C1}\}$。本阶段仅在$U_i$和$S$已成功地建立了会话密钥之后才会被执行。通过这样预计算，在下次用户登录时，智能卡无须重新计算以节省时间。

4) **登录阶段** 当U_i访问S时，将执行如下操作：

(1) U_i计算$h(ID_i \| x) = V \oplus h(PW_i \| b)$，然后选择一个随机数$N_{C2}$，并计算$C_1 = (ID_i \| (h(ID_i \| x) \oplus N_{C2})) \oplus h_1(c)$。

(2) $U_i \rightarrow S : \{C_1, e\}$。

(3) S计算$ID_i \| (h(ID_i \| x) \oplus N_{C2}) = C_1 \oplus h_1(x \times e)$来得到$ID_i$ 和 $h(ID_i \| x) \oplus N_{C2}$。然后，$S$计算$N_{C2} = h(ID_i \| x) \oplus N_{C2} \oplus h(ID_i \| x)$，选择$N_S \in_R \mathbb{Z}_p^*$，并计算$C_2 = N_S \times P$， $SK = h(e \| C_2 \| N_S \times e)$ 和 $C_3 = h(h(ID_i \| x) \ \| N_{C2} \| C_2 \| e \| SK)$。

(4) $S \rightarrow U_i : \{C_2, C_3\}$。

(5) 一旦收到来自S的响应，U_i计算$SK = h(e \| C_2 \| N_{C1} \times C_2) = h(e \| C_2 \| N_{C1} \times N_S \times P)$和 $C_3' = h(h(ID_i \| x) \| N_{C2} \| C_2 \| e \| SK)$。然后，$U_i$验证收到的$C_3$是否等于$C_3'$。如果不相等，$U_i$拒绝此次通信。否则，$U_i$计算$C_4 = h(h(ID_i \| x) \| SK \| N_{C2} \| C_2 \| e)$。

(6) $U_i \rightarrow S : \{C_4\}$。

(7) S首先计算$C_4' = h(h(ID_i \| x) \| SK \| N_{C2} \| C_2 \| e)$，然后比较$C_4'$是否等于收到的$C_4$。如果不相等，$S$拒绝$U_i$的登录请求。否则，$S$授权$U_i$的请求。

5) **口令更新阶段** 这一阶段实现用户在本地自由更新口令。用户U_i将智能卡插入读卡器，输入旧口令PW_i 和新口令PW_i^{new}。然后，智能卡计算 $V^{new} = V \oplus h(PW_i \| b) \oplus h(PW_i^{new} \| b) = h(ID_i \| x) \oplus h(PW_i^{new} \| b)$，并用$V^{new}$代替$V$。

7.3.2　对Tsai等协议的分析

Tsai等提出了5种攻击者模型，在每种模型下给出了协议的形式化安全证明。不难看出，他们的5种攻击者模型全部包含在我们的模型中(见节7.2.2)。虽然他们的协议和其它协议相比，有许多可取之处，如实现了用户不可追踪性和可证明安全性，且具有较高的效率，但它远不是一个实用的“理想”协议。本章我们将指出，在Tsai等自己的假设下，该协议无法抵抗智能卡丢失攻击(即SR6)，并且存在智能卡撤销问题(DA9)。自然地，一个矛盾现象产生了: *为什么已被形式化证明安全的协议，后来又被发现不安全？*我们将找出Tsai等在形式化安全证明中的失误之处，来解释这一矛盾。

1) **智能卡丢失攻击 I**。 一个能够抵抗智能卡丢失攻击(即SR6)的协议应该确保，即使用户的智能卡丢失(或被盗)，$\mathcal{A}$依然无法轻易地更改用户的口令、利用口令猜测攻击猜出用户口令或仿冒用户。那么，用户丢失智能卡的实际可能性有多大呢？最近一项关于生活中双因子系统的可用性研究[416]表明，用户确实容易令智能卡脱离自己的监管：在为期6个星期的研究期间，54%的用户至少有一次把智能卡忘在了读卡器上。因此，SR6是任何实用的协议应实现的基本目标。不幸的是，过去的研究(见[279, 397])表明，实现这个目标并非易事。下面，我们将显示Tsai等的尝试也是徒劳的—一旦攻击者$\mathcal{A}$获得用户的智能卡，$\mathcal{A}$就能够通过离线口令猜测攻击获取用户的口令。

假设攻击者$\mathcal{A}$通过某种方式(如窃取或拾到)较长时间(如数小时)拥有了用户U_i的智能卡，并自行(或求助于专业的实验室)通过实施边信道攻击[165, 326, 417]分析出智能卡内秘密参数信息$\{e, c, N_{C1}, V, b\}$。然后，$\mathcal{A}$在U_i未察觉的情况下，把已被攻破的卡还给U_i。一旦U_i使用此卡登录，$\mathcal{A}$通过截获U_i的登录请求$\{C_1, e, C_2, C_3\}$，就可以通过下面的方法离线猜测出PW_i：

Step 1. 从字典空间$\mathcal{D}_{pw}$中选取PW_i^*作为用户的口令。
Step 2. 计算$h(ID_i\|x)^* = V \oplus h(PW_i^*\|b)$，其中$V, b$从$U_i$的智能卡中得到。
Step 3. 通过计算$ID_i\|\ (h(ID_i\|x) \oplus N_{C2}) = C_1 \oplus h(c)$，得到$h(ID_i\|x) \oplus N_{C2}$，其中$c$从$U_i$的智能卡中得到，$C_1$通过截获登录请求消息得到。
Step 4. 计算$N_{C2}^* = (h(ID_i\|x) \oplus N_{C2}) \oplus h(ID_i\|x)^*$，$SK = h(e\|C_2\|N_{C1} \times C_2)$和 $C_3^* = h(h(ID_i\|x)^*\|\ N_{C2}^*\|\ C_2\|e\|SK)$。
Step 5. 验证计算出的C_3^*是否等于截获的C_3，从而判断PW_i^*的正确性。
Step 6. 重复步骤1 ~ 5，直到正确的PW_i被猜测出来。

在上面的攻击中，我们做了两个假设：(1) $\mathcal{A}$能够获得并攻陷用户的智能卡；(2) $\mathcal{A}$可以归还该已被攻陷的卡，且不被发现。第一条假设在现有研究中已广泛采用(如[150, 151, 278])，而且Tsai等[400]中也明确地采用了这一假设。第二

条假设在近期才被关注，确实是合理的[279, 397]。[Φ]这里我们给出生活中一个典型的场景。用户U_i下午下班后将智能卡遗忘在自己的工位上(或未上锁的抽屉里)，$\mathcal{A}$(U_i的一位恶意同事)取走U_i 的卡并在晚上实施(自行或借助专业机构)边信道攻击获取卡内参数。在U_i 第二天上班前，$\mathcal{A}$将卡归还原位。

用$|\mathcal{D}_{pw}|$ 表示口令空间的大小，则上述攻击的时间复杂度为$\mathcal{O}(|\mathcal{D}_{pw}|*(T_P+4T_H+T_X))$。其中，$T_P$是ECC点乘的运算时间，$T_H$是哈希函数的运算时间，$T_X$是按位异或运算的时间。由于$\mathcal{A}$恢复$U_i$口令的时间是关于$\mathcal{D}_{pw}$的一个线性函数，因此我们的攻击是十分高效的。

为更好地理解上述攻击的效率，我们进一步采用C/C++大数运算密码库MIRACL [339]在普通PC上运行了相关密码原语操作，并给出各操作所花费的时间(见表7.3)。在现实中，用户为方便记忆通常会选用低熵的短字符串作为口令，因此口令空间非常有限：$|\mathcal{D}_{pw}| \leq 10^6$ [5]。综上，$\mathcal{A}$在普通PC上执行上述攻击只需要几秒钟。

表 7.3 在普通 PC 机上相关密码原语操作的计算时间

实验凭他 (common PCs)	ECC点乘操作 T_P(ECC sect163r1 [419])	模指数操作 $T_E(\|n\|=512)$	对称解密操作 T_S(AES-128)	哈希操作 T_H(SHA-1)	其它轻量级操作(比如：XOR)
Intel T5870 2.00 GHz	1.226 ms	2.573 ms	2.049 µs	2.580 µs	0.011 µs
Intel E5500 2.80 GHz	0.617 ms	1.348 ms	0.572 µs	0.753 µs	0.009 µs
Intel i3-530 2.93 GHz	0.508 ms	1.169 ms	0.541 µs	0.693 µs	0.008 µs

我们的攻击表明，一旦智能卡因子被攻破了，剩余的口令因子也可被离线猜测出来，从而整个系统崩溃。这意味着，Tsai等的协议并不是一个真正的双因子协议，但原协议(即Li协议[289])却不存在这一缺陷。Tsai等协议失效的主要原因在于预计算阶段，这一阶段使$\mathcal{A}$不仅有机会获得智能卡中长期参数，还得到了特定会话相关信息(如参数c)。为解决这一问题，一个直观的办法是去除预计算阶段，直接在登录阶段计算e和c。但是这会降低协议的效率。我们也注意到，Li等协议[289]采用了类似的预计算技术，虽可抵抗上面的攻击，但存在另一安全问题—异步攻击(一种拒绝服务攻击，见节7.4.2)。自然地，一个有趣的问题产生了：是否存在一个安全的、采用预计算技术的匿名认证协议？这一问题超出了我们的研究范围，我们将其遗留为一个公开问题。

2) **智能卡丢失攻击II**。 为提高用户友好性，像文献[174, 177, 278, 398]一样，Tsai等协议实现了“用户口令本地自由更新”这一特性(即强DA2)—口令更新在本地进行，无须与远程服务器交互。但Tsai等协议引入了一个不安全因素：在更新口令之前，未对用户旧口令的真实性进行验证。如果$\mathcal{A}$设法获得了对U_i的智能卡的临时访问的机会(如节7.3.2所讨论的，这是一个非常实际的假设)，$\mathcal{A}$能

[Φ]Huang等[397]分析了Juang等[418]协议，指出如果$\mathcal{A}$在登录阶段之前获得了智能卡中的信息，则“$\mathcal{A}$可以计算出会话密钥”。这意味着，他们隐含地假设了$\mathcal{A}$能在未被发现的情况下，归还已破解的卡。

够轻易地按如下步骤更新U_i的口令：

Step 1. $\mathcal{A}$把用户U_i的智能卡插入读卡器，发起口令更新请求。

Step 2. $\mathcal{A}$提交一个随机字符串R作为U_i的原始口令，和一个新字符串PW_i^{new}作为新口令。

Step 3. 智能卡计算$V^{new} = V \oplus h(R\|b) \oplus h(PW_i^{new}\|b)$，并用$V^{new}$替代$V$。

即使卡被归还，一旦智能卡内V的值被更新，合法用户U_i此后将无法成功登录S。因为$V^{new} \oplus h(PW_i\|b) \neq h(ID_i\|x)$，此后$U_i$的登录请求都将通不过服务器$S$的验证，将被$S$拒绝。这意味着，一旦智能卡丢失，拒绝服务攻击可轻易实施。因此，Tsai等协议无法实现SR6。

实际上，这一设计缺陷也可能导致另一个十分实际和麻烦的可用性问题：如果一个合法用户在口令更新过程中意外输错了当前(旧)的口令，参数V将被更新为不可预测的随机值。从此，智能卡将完全不可用，除非用户向S重新注册。

尽管这两个可用性缺陷看似很简单、不值得讨论，但它们非常现实，而且无法通过小的协议改动来很好地修正它们。为实现强DA2的同时满足SR6，在更新智能卡中存储的参数V的值之前，需要验证原口令的正确性。因此，除了V，还应在智能卡中存储一些额外的参数，但这可能引入新的安全问题，如离线猜测攻击和仿冒攻击。Nam等[420]也发现了这一微秒之处，但他们没有尝试给出解决方案，把它遗留为公开问题。大多数后续工作要么忽视这一微秒之处(如[174, 278, 390, 398, 405])，要么选择放弃实现强DA2属性(如[150, 289, 421])，而那些宣称既支持口令本地自由更新(即强DA2属性)又能抵抗智能卡丢失攻击II的少数协议(如[177, 382, 422])，都被指出易遭智能卡丢失攻击I。

为更深入地理解这个问题，这里我们给出一个具体的例子。假设一个额外参数$A_i = h(ID_i \,\|\, h(PW_i))$存储在智能卡中。当$U_i$想更新口令时，必须先输入身份$ID_i^*$和口令$PW_i^*$，然后智能卡验证 $h(ID_i^* \,\|\, h(PW_i^*))$是否等于存储的$A_i$。不难看出，这会引入离线口令猜测攻击：根据攻击者能力假设C-01，$\mathcal{A}$能够枚举出所有的(ID_i, PW_i)对，一旦获得参数A_i，$\mathcal{A}$可通过离线的方式找出唯一正确的(ID_i, PW_i)对。

然而，如果令参数$A_i = h(\hbar(ID_i) \oplus \hbar(PW_i))$，则有$\frac{|\mathcal{D}_{id}|*|\mathcal{D}_{pw}|}{2^8} \approx 2^{32}$个候选$(ID_i, PW_i)$对满足$A_i = h(\hbar(ID_i) \oplus \hbar(PW_i))$，其中$|\mathcal{D}_{id}| = |\mathcal{D}_{pw}| = 10^{6}$[5]，$\hbar(\cdot)$是一个特殊的哈希函数 $\{0,1\}^* \to \{0,1,\ldots,255\}$，$|\mathcal{D}_{id}|$和$|\mathcal{D}_{pw}|$代表身份标识空间和口令空间的大小。针对这$\frac{|\mathcal{D}_{id}|*|\mathcal{D}_{pw}|}{2^8} \approx 2^{32}$个候选$(ID_i, PW_i)$对，$\mathcal{A}$必须逐一通过与服务器$S$在线交互的方式，来确定唯一正确的$(ID_i, PW_i)$对。针对$\mathcal{A}$的在线猜测攻击，$S$有众多成熟的手段来阻止之。

即使$\mathcal{A}$已知ID_i，仍然存在$\frac{|\mathcal{D}_{pw}|}{2^8} \approx 2^{12}$个$PW_i$候选项，这些候选项也只能通

过在线猜测攻击的方式来确定。通过参数A_i的这种新的计算方式，可有效阻止$\mathcal{A}$获得正确的PW_i，称这种新型计算方式的A_i为“模糊验证因子(Fuzzy verifier)”[423]。可能有人会想，如果U_i无意中输入了一个错误的(ID_i^*, PW_i^*)对，恰好满足$h(\hbar(ID_i^*) \oplus \hbar(PW_i^*)) = A_i$，但是 $(ID_i^*, PW_i^*) \neq (ID_i, PW_i)$，会怎样呢？实际上，这种情况发生的概率只有$\frac{1}{256}$。如果进一步要求用户在更新口令时输入新旧口令各两次，且$\hbar(\cdot)$是一个随机预言机的话，则这个概率会降到$\frac{1}{256^2}$。“模糊验证因子”的一个明显的“副产品”就是，能够在用户登录时及时检测出用户名/口令错误输入的情况(即实现DA10)。

因此，只有对Tsai等协议进行大幅改动才能消除上面提到的各种问题。我们推测，指标DA2、DA10和SR6之间存在不可避免的平衡，相关证据将在节7.6.4给出。“模糊验证因子”技术看起来是解决这一问题的一个不错选择，但是它的实际有效性只能通过真实数据集来实验验证。幸运的是，近期发生的一系列大规模口令泄露事件(如5亿Yahoo口令[127]和600万CSDN口令[28])为这项实验提供了充足的素材，具体的实验将在节8.2.2给出。

3) **智能卡撤销问题**。为实现指标DA10，Tsai等协议在服务器端没有存储与用户(或智能卡)相关的任何数据。当然，这样确实能够实现DA10(“服务器端无口令相关验证表项”)，但过犹不及也是不可取的。由于没有任何智能卡相关的信息存储在服务器端，服务器无法区分可信的卡和不可信的卡，这意味着协议无法撤销不可信(或过期)的智能卡。

此外，这也导致Tsai等协议可修复性差。当U_i发现其智能卡丢失或者核心参数$h(ID_i \| x) = V \oplus h(PW_i \| b)$已被攻击者分析出来时，即使$U_i$想重新在服务器上注册，服务器也无法轻易应对这种情况。如节7.3.2所述，一旦用户的卡丢失，$\mathcal{A}$可获得$h(ID_i \| x)$。这种情况下，即使U_i发现自己的卡已经不在自己掌控的范围内，U_i在S重新注册也无法阻止仿冒攻击。由于$h(ID_i \| x)$由U_i的身份ID_i和S的长期密钥x构成，所以除非ID_i或者x能够被修改为一个新的值，否则S无法更新U_i对应的$h(ID_i \| x)$。一方面，x是S的长期密钥，往往与所有的注册用户相关，而不是仅为某一用户服务，所以仅为恢复U_i的安全性而改变x的值是低效且不现实的。另一方面，因为在大部分场景中ID_i是唯一的定值，且在很多应用系统中ID_i会与U_i绑定，故更改ID_i也不现实。

4) **Tsai等协议在形式化安全证明中的失误**。一般而言，一个实现了会话密钥语义安全性(或简称为“语义安全性”，“AKE安全性”)的双因子协议能够提供基本的双因子安全性保障：在非抗窜扰智能卡假设下(即表7.1中的C-2i，或者[400]中的智能卡丢失情形)，即使智能卡丢失并被攻破，仍可抵抗仿冒攻击和离线口令猜测攻击。Tsai等协议虽然宣称实现了语义安全性，但如前文所示，一旦智能卡因子被攻破，口令因子也随之可被离线猜测出来。

这自然产生了矛盾：一个已被证明为安全的协议为何现在又变得不安全了？为回答这一有趣的问题，我们仔细研究了Tsai等协议的形式化安全证明的归约过程，并成功找出他们推理证明中存在的根本失误。

Tsai等[400]采用的形式化安全模型建立在Xu等工作[150]之上，而后者本质上是Bellare等[375]提出的随机预言机模型(ROM)的一个变体。在ROM模型下，语义安全性证明的基本原理如下：(1)将所有哈希函数都建模为预言机，这些预言机为每个新查询输出一个随机值，但确保相同的查询拥有相同的输出；(2)假设$\mathcal{A}$能够攻破目标协议$\mathcal{P}$的语义安全性；(3)利用$\mathcal{A}$来构建解决协议中密码原语(如大合数因子分解问题、计算性Diffie-Hellman问题和判定性Diffie-Hellman问题)的算法，如果$\mathcal{A}$成功攻破了协议$\mathcal{P}$的语义安全性，那么至少有一个密码原语将被解决。由于这些原语被广泛认为是困难的，无法在概率多项式时间(PPT)内被解决，产生矛盾，因此协议$\mathcal{P}$是安全的(即实现了语义安全性)。

Tsai等采用了与文献[150, 423]类似的方法来证明协议的语义安全性。证明由一系列的攻击游戏G_n $(n = 0, 1, \cdots, 5)$构成，从真实攻击G_0开始，结束于游戏G_5，且在G_5中$\mathcal{A}$的优势是0。接着，在任意两个连续游戏之间，给出$\mathcal{A}$的优势差异的上界。这样，也就最终限定了$\mathcal{A}$攻击原始协议的优势。尽管Tsai等定义了一系列游戏，但是，他们的安全归约缺乏严谨性，只是进行了简单粗糙的偷梁换柱！更具体地说，他们把只适用于双因子认证协议的安全模型运用到他们三因子版本的协议上，来证明其协议的安全性。由于没有采用恰当的安全模型(它是安全归约的基础)，证明必然会失效。

Tsai等将其安全模型分为五种情形(参见[400]中的节III)，然而没有一种情形与生物因子相关。众所周知，生物特征 (例如指纹和虹膜)容易遭受各种现实攻击，并不是一个不存在威胁的安全的“黑盒子”[15, 214, 424]。这意味着， Tsai等安全模型不适合用来分析“口令+智能卡+生物特征”三因子协议，仅适合分析“口令+智能卡”双因子协议。遗憾的是，Tsai等并没有利用这一模型来分析他们双因子版本的协议。他们五个攻击情形中的第二个考虑了智能卡丢失的情况。在这一情形下，$\mathcal{A}$本质上具有能力C-1&C-2-ii(见表7.1)，即$\mathcal{A}$完全控制通信信道，且可分析出受害者智能卡内参数。正是在这一攻击场景下，我们已经证实他们的双因子版本协议无法实现真正的双因子安全性，这也意味着他们的三因子版本协议也至少无法实现真正的三因子安全性—一旦智能卡因子和生物特征因子被破获，$\mathcal{A}$就能够通过离线猜测的方式得到口令因子。这表明，Tsai等双因子和三因子版本的协议都存在严重安全缺陷。

7.4　对Li协议的分析

在上述分析的协议中，用户不可追踪性是通过使用公钥技术来实现的，即

为每个会话都生成相异的伪随机身份标识。相对而言，本节讨论的协议采用了一个完全不同的方法：每个实体在认证对方后，更新用户的会话可变伪身份标识。虽然这一方法确实实现了用户不可追踪性，但我们下面将指出，它是不实用的：会引发严重的问题以致于大大降低协议的可用性。此外，Li协议也无法实现双因子协议中最核心的安全目标—双因子安全性。

7.4.1 回顾Li协议

2013年，Li[421]提出了两个基于ECC的双因子认证协议，其中一个实现了用户匿名性，另一个未实现。本章只关注前者。这一协议含有四个阶段：注册、认证、口令变更、用户移除。为清楚起见，除表7.2列出的一些符号定义外，表7.4列出了一些本节将使用的其它符号定义。

表 7.4 在Li协议中额外用到的符号

符号	描述
ID_A	用户A的身份
PW_A	用户A的口令
d_S，U_S	远程服务器S的私钥和公钥，其中$U_S = d_S \cdot G$
K_x	密钥，其中$K = PW_A \cdot r_A \cdot U_s = (K_x, K_y)$
$E_{K_x}(\cdot)/D_{K_x}(\cdot)$	对称加密/解密，密钥为K_x

1) **注册阶段**。这一阶段涉及如下操作：

(1) 用户A选择身份标识ID_A，口令PW_A和$r_A \in_R \mathbb{Z}_n^*$，计算$U_A = PW_A \cdot r_A \cdot G$;

(2) $A \Rightarrow S$: $\{ID_A, U_A\}$。

(3) 一旦收到注册消息，服务器S为A选择一个伪身份IND_A, 并在数据库中创建一个条目(IND_A, U_A, *status-bit*)，其中*status-bit*表明A的状态。进一步讲，当A成功登录S时，*status-bit*设为1，否则设为0。

(4) $S \Rightarrow A$: 一张包含参数$\{IND_A, G, h(\cdot), E_{K_x}(\cdot)/D_{K_x}(\cdot)\}$的智能卡。

(5) 收到智能卡后，用户A输入r_A。这意味着此后卡中存储有$\{IND_A, G, h(\cdot), E_{K_x}(\cdot)/D_{K_x}(\cdot), r_A\}$。

2) **认证阶段**。当A欲登录S时，执行下面的步骤：

(1) A将智能卡插入读卡器，输入口令PW_A。

(2) 智能卡从存储的参数中获取r_A，生成$r'_A \in_R \mathbb{Z}_n^*$，计算$R_A = r_A \cdot U_S = r_A \cdot d_S \cdot G$, $W_A = r_A \cdot r_A \cdot PW_A \cdot G$，$U'_A = PW_A \cdot r'_A \cdot G$和$K = PW_A \cdot r_A \cdot U_s = (K_x, K_y)$。

(3) $A \rightarrow S$: $\{IND_A, E_{K_x}(IND_A, R_A, W_A, U'_A)\}$。

(4) 收到登录请求后，S计算解密密钥K_x：$K = d_S \cdot U_A = PW_A \cdot r_A \cdot d_s \cdot G = (K_x, K_y)$，并解密$E_{K_x}(IND_A, R_A, W_A, U'_A)$得到$\{IND_A, R_A, W_A, U'_A\}$。然后，$S$分别对比解密的$IND_A$和收到的$IND_A$，以及$\hat{e}(d_S \cdot R_A, U_A)$ 和$\hat{e}(W_A, U_S)$。如果任一对比不相等，则S拒绝A的请求。否则，S 通过下面的等式来验证A的合法性：

$$\hat{e}(d_S \cdot R_A, U_A) = \hat{e}(d_S \cdot r_A \cdot G, r_A \cdot pw_A \cdot G) = \hat{e}(G,G)^{r_A \cdot r_A \cdot pw_A \cdot d_S}$$
$$\hat{e}(W_A, U_S) = \hat{e}(r_A \cdot r_A \cdot pw_A \cdot G, d_S \cdot G) = \hat{e}(G,G)^{r_A \cdot r_A \cdot pw_A \cdot d_S}$$

(5) S为A生成新的身份标识IND_A，选择$r_S \in_R \mathbb{Z}_n^*$，计算$W_S = r_S \cdot U_S = r_S \cdot d_S \cdot G$和会话密钥$sk = r_S \cdot d_S \cdot W_A$。

(6) $S \to A : \{W_A + W_S, h(W_S \| U'_A \| sk \| IND'_A), E_{sk}(IND'_A)\}$。

(7) A通过将$(W_A + W_S)$减去W_A得到W_S，计算$sk = r_A \cdot r_A \cdot PW_A \cdot W_S$，并用$sk$解密$E_{sk}(IND'_A)$得到$IND'_A$。$A$检验$\{W_S \| U'_A \| sk \| IND'_A\}$的哈希结果是否等于接收到的$H(W_S \| U'_A \| sk \| IND'_A)$。如果相等，$A$确信$S$是可信的，并用$\{r'_A, IND'_A\}$代替$\{r_A, IND_A\}$。

(8) $A \to S : \{IND_A, h(sk \| IND'_A)\}$。

(9) S检验$\{sk \| IND'_A\}$的哈希结果是否等于接收到的$h(sk \| IND'_A)$。如果相等，S接受A的登录请求，并用$\{IND'_A, U'_A\}$ 代替$\{IND_A, U_A\}$。

3) **口令更新阶段**。口令更新阶段用以方便用户自由地更新他们的口令(但非本地)。当A想更新口令时，首先需通过上面的认证阶段来确保其输入的口令是正确的，然后才能更新口令。

4) **用户撤销阶段**。如果用户A的有效期到期，服务器S可以删除后端数据库中的(IND_A, U_A)来撤销用户。此后，A将不能再使用IND_A和U_A来登录S。

7.4.2　对Li协议的分析

Li[421]指出Islam-Biswas协议[425]存在严重安全缺陷，如未实现双因子安全性和无法抵抗窃取验证表项攻击。为消除这些缺陷，Li提出了一个改进的协议。除了因ECC 带来的高效性，Li还宣称(并启发式地论证了)新协议能够提供强健的安全性(即满足SR1∼SR7)，且实现了五个有吸引力的属性(即满足DA1∼DA5)。因此，Li协议显示出较好的应用潜力。但经过详细分析后，我们发现Li协议离实际应用还有很长的距离—它缺乏可用性，且在非抗窜扰智能卡假设下仍无法实现双因子安全性。

1) **智能卡丢失攻击**。双因子认证协议最基本的安全目标是双因子安全性，即攻击者获得用户的口令因子或者智能卡因子中任一个，仍无法仿冒合法用户。正如[279, 397] 中指出的，到目前为止，鲜有协议实现了这一“珍贵”目标。再一次，Li的尝试[421]也以失效告终：我们将展示拥有用户智能卡的攻击者如何通过自动化流程恢复出用户的口令。

假设攻击者$\mathcal{A}$以某种方式获得(窃取或拾到)用户A的智能卡，并通过边信道攻击[165, 326, 417]自己(或借助专业机构)分析出卡秘密信息$\{IND_A, r_A, U_S\}$，随

后$\mathcal{A}$在A没有意识到的情况下归还该已被攻破的卡。注意，如我们在节7.2.1和7.3.2中讨论的，这些假设是现实的。一旦当用户A使用被攻破的智能卡登录时，$\mathcal{A}$首先截获A的登录请求$\{IND_A, E_{K_x}(IND_A, R_A, W_A, U'_A)\}$，然后通过下面的方式可离线猜测出$PW_A$：

Step 1. 猜测PW_A为PW_A^*。
Step 2. 计算$K^* = PW_A^* \cdot r_A \cdot U_s = (K_x^*, K_y^*)$，其中$r_A, U_s$来自$A$的智能卡。
Step 3. 用K_x^*解密之前截获的$E_{K_x}(IND_A, R_A, W_A, U'_A)$ 来得到IND_A^* 。
Step 4. 通过判断IND_A^*是否等于IND_A，来检验PW_A^*的正确性。
Step 5. 重复步骤1 ~ 4直至猜测出正确的PW_A。

令$|\mathcal{D}_{pw}|$表示口令空间$\mathcal{D}_{pw}$的大小，则上述攻击的时间复杂度为$\mathcal{O}(|\mathcal{D}_{pw}| * (2T_P + T_S))$，其中$T_P$是ECC点乘运行的时间，$T_S$是对称解密的运行时间。换句话说，$\mathcal{A}$恢复$U_i$口令的时间是关于$|\mathcal{D}_{pw}|$的线性函数，因此，上述攻击是高效的。此外，在实际中，用户通常会选择易记忆的弱口令，因此口令空间 $\mathcal{D}_{pw}$的规模一般非常有限，如$|\mathcal{D}_{pw}| \leq 10^6$ [5]。进一步根据表7.3中的运行时间，$\mathcal{A}$ 只需几秒钟就能在普通PC上完成上述攻击流程。

我们的攻击表明，一旦智能卡因子被攻破，剩下的口令因子也可离线猜测出来。自此，A除了重新向服务器注册，再没有其它的方法能够阻止$\mathcal{A}$仿冒A来访问系统服务/资源。这意味着，Li协议并不是一个真正的双因子协议，无法提供比原协议(即Islam-Bswas协议[425])更高的安全性。

2) **异步攻击**。 为实现用户不可追踪性，一些匿名认证协议(如[398, 400, 423])尝试在每个登录会话中采用密码学技术 (如公钥加密算法)为用户生成新的伪身份标识，剩余一些匿名认证协议(如[289, 390, 406])未采用密码学技术，而是采用一种完全不同的策略：在当前登录会话中，为用户新生成下次登录请求中将使用的伪身份标识。显然，在后一种策略中，用户新的伪身份标识(将在下一次登录请求中使用到)需要被存储在用户端的某处。并且，为在协议的下一次运行中正确识别该用户，服务器端也需在当前会话结束前，存储用户的这一新生成的伪身份标识。因此，用户和服务器间针对用户新生成的伪身份标识的正确同步，对协议之后的成功运行至关重要。但是，确保正确的同步并非易事。下面我们将显示，$\mathcal{A}$总是能够以某种方式破坏Li协议中的这种同步，使得用户从此无法正常登录。这意味着，后一种策略(即采用同步机制)是不可行的。

举例示之。假设用户A已经执行了认证阶段中的步骤7(见节7.4.1)，并按照协议规定把$\{IND_A, h(sk\|IND'_A)\}$发给了S。这意味着，A已经用$\{r'_A, IND'_A\}$替换了智能卡中存储的$\{r_A, IND_A\}$。在消息$\{IND_A, h(sk\|IND'_A)\}$达到S之前，$\mathcal{A}$拦截这个消息并把它修改为$\{IND_A, X\}$，其中X是一个随机值。在认证阶段中的步

骤8，S将发现$X \neq h(sk\|IND'_A)$，因而必定会拒绝A 的登录请求，并拒绝在后端数据库中把$\{U_A, IND_A\}$更新为$\{U'_A, IND'_A\}$。因此，A和S之间用户伪身份标识的一致性被打破了。之后，A会在登录请求中发送IND'_A给S，但S存储的却是旧的IND_A，故S总会因为$IND'_A \neq IND_A$而拒绝A。

上述攻击(如图7.2)是现实可行的，因为$\mathcal{A}$只需要修改一条协议交互消息(即A到S的第三条消息)，就可完全破坏$\mathcal{A}$和S间的“同步”。这一攻击不涉及任何其它复杂或高代价的操作，比如逆向工程和能耗分析。另外，不难看出，除了修改协议消息，$\mathcal{A}$也可简单地通过阻止协议消息来达到同样的目的。

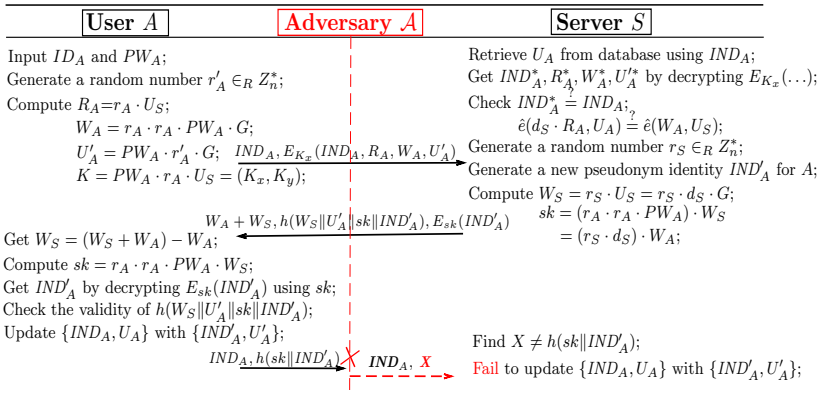


图 7.2 针对 Li 协议的异步攻击。

需注意的是，虽然这个攻击看上去非常简单，但是仅仅通过微小的协议改动无法消除该攻击。可能有人会想，如果在$A \to S$的第三条消息之后，增加$S \to A$的第四条消息“ack”，且只有在A收到“ack”之后，才$\{r_A, IND_A\}$更新为$\{r'_A, IND'_A\}$，这样上面的攻击就无法成功了。不可否认，这没有错。但是，如果$\mathcal{A}$现在简单地阻止这条“ack”消息，用户A会一直等待实际上永远不会到来的“ack”消息，不会把$\{r_A, IND_A\}$更新为$\{r'_A, IND'_A\}$。而另一方面，服务器S在发送“ack”之前就已经把$\{U_A, IND_A\}$ 更新为了$\{U'_A, IND'_A\}$。这样，用户和服务器之间的“同步”也被破坏了。同理，所有类似试图通过添加新的协议流来克服Li协议的这一缺陷都会失效。

另一种可能的办法是在智能卡(和/或服务器端)同时存储旧的和新的两个伪身份标识。如果新的伪身份标识无法使用(即出现了不同步的情况)，就转而使用旧的伪身份标识。遗憾的是，这一解决方法会带来新的问题。其中一个突出的是，一旦$\mathcal{A}$盲目地阻塞信道中的会话，同一个用户参与的会话就会出现同样的旧身份标识，这就违反了用户不可追踪性。

备注 2。我们分析了300多个双因子协议，其中包括100多个匿名双因子协议(我们近期的一些公开成果可见 [279, 345–347])，发现所有那些运用了类似Li协议[421]中的同步机制来维持用户身份标识一致性的协议，都无一例外地存在异步

攻击问题。存在这一问题的典型协议包括[289, 358, 390, 406, 407]。上述攻击表明，这些协议缺乏完备性。不幸的是，如[343, 426]所指出的，广泛应用的形式化方法(如ROM模型，BAN逻辑)都无法刻画并检测协议中的这类结构性缺陷，因此如何确保协议的完备性仍是一个公开问题。

7.5 “可证明安全”方法的角色

最早的数十年里，安全协议的进展主要靠试错：当一个新协议提出来后，学术界试图分析其缺陷并进行修复。结果表明，很多协议往往在提出多年后才被指出存在这样或那样的安全问题。最典型的例子就是，著名的Needham-Schroeder认证协议在提出17年后才被发现存在严重漏洞[427]。这种令人不满意的状况直到 20 世纪 80 年代初期才因Goldwasser和Micali[428]的开创性工作而有所改善，随后一系列有影响力的工作也陆续面世[375, 429]。这些工作表明：在广泛接受的基于复杂度的困难问题假设下(例如CDH难解性)，安全性是可以被“证明”的。他们的方法称为“可证明安全”方法，一个通过了这种方法检验的协议即认为实现了“可证明安全性”。这种方法在消除启发式攻击间的冗余方面特别有效。比如，表1.8中列出的大部分面向攻击的目标可以简化为两种形式的目标，即语义安全性和双向认证[423, 430]。此外，通过形式化模型定义的安全目标，在描述协议能够提供的安全性方面比启发式模型更加精确。因此，可证明安全方法已经成为分析和评估一个协议，尤其复杂密码协议的安全性的不可或缺工具。

但是，可证明安全有它自己的局限。一般来说，为协议提供“可证明安全”涉及五个步骤：(1)定义攻击者模型；(2)声明安全目标；(3)陈述密码原语假设；(4)描述协议；(5)归约证明。因此，一个协议只有在某种安全模型下，基于给定的密码原语假设，才能实现可证明安全目标，而不是仅仅声称这样或那样，协议就能实现可证明安全。身份认证协议过去二十年的研究历程告诉我们，如果采用了一个不正确的安全模型而遗漏攻击者某项现实能力(如[186, 431])，在归约过程中出现错误(如[150, 432])，或者是归约不够紧致(见节2.6.1)，那么安全证明的结果往往会失去实际意义。依据我们对300余个双因子认证协议的分析经验，那些提供了形式化证明的协议(如[150, 382, 390, 398])大多数在提出后不久，就被发现存在严重的安全问题，未实现所“证明”的目标。节7.3.2对Tsai等协议的“智能卡丢失攻击I”也很好地表明：采用形式化(但不实际)的安全模型，并在该模型下设计一个“可证明安全”的协议，这本质上无法确保协议的真正安全性。由于形式化方法很容易被误用，安全归约证明过程也往往非常复杂、晦涩难懂、容易出错，所以在对复杂密码协议(如双因子认证协议)进行形式化证明时应格外小心。

还应注意的是，很多攻击场景难以在形式化安全模型中刻画出来。比如，

尽我们所知，Tsai等协议中的“智能卡丢失攻击II”(见节7.3.2)和Li协议中的“异步攻击”(见节7.4.2)就难以在ROM或BAN模型中刻画出来。由于这些现实的攻击难以在当前的安全模型中刻画出来，这就要求协议设计者们对这些攻击应有清醒的认识。此外，如[426, 433]所述，即使采用的安全模型是合适的，协议安全证明的正确性在很大程度上还依赖于证明者的攻击经验。因此，“可证明安全更像是艺术而非科学”[434, 435]。最后，即使采用了准确的安全模型，并且正确地完成了安全归约证明过程，可证明安全方法也无法用来分析或者确保协议的一些功能属性。所有的这些，都突出显示了*传统启发式分析技术*在确保复杂密码协议的安全性和理想属性方面的重要作用。

7.6　Madhusudhan-Mittal评价标准的不足

本节我们首先回顾和总结实现用户匿名的可行方法，随后解释Madhusudhan-Mittal 标准中一些指标的模糊、微妙之处，为研究它们的内在关系打下基础。进一步，我们以前文分析的两个协议为案例，在现有一些密码分析结果[347, 397, 402, 436]的基础之上，指出仅采用密码学工具，构建一个满足Madhusudhan-Mittal[170]所有评价指标的“理想的”匿名双因子认证协议是不可行的，这为 Huang等遗留的公开问题[397]提供了一个否定的答案。

7.6.1　实现用户匿名的方式

直观地，有两种常用的方法来实现“动态ID技术”：(1)利用密码技术，如文献[174, 405]中的对称密码操作，[384, 400] 中的CCA-2公钥加密，[369, 370]中的环/群签名，或[437]中的属性基签名；(2)利用一些非密码技术(如文献[368]中的预装载伪ID 池，或[289, 421]中的同步机制)。但是，不难看出，在资源受限的智能卡上预加载大量伪ID池的想法实际上是不可行的；如节7.4所讨论的，采用同步机制的匿名协议容易出现严重的可用性问题。

如此看来，既然非密码机制不可行，密码机制则是实现双因子认证中用户匿名性的唯一可选办法。由于环/群签名通常需要公钥基础设施(PKI)的支持，且它们的计算复杂度随着环/群的大小线性增长，所以它们都不适于双因子认证协议；基于属性的签名方案虽可以摆脱笨重的PKI，但往往涉及开销巨大的双线性对操作，不适于在计算能力受限智能卡上应用。此外，在非抗窜扰智能卡假设下，如何安全地保存用户端的签名密钥是一个难题。

尽我们所知，上述分析很好地解释了为什么大部分匿名双因子协议(如[174, 177, 334, 400, 402, 405, 421, 423, 438])倾向于采用对称密码原语(例如哈希函数、XOR操作、对称加密) 或一些相对轻量级的公钥操作(例如小指数模幂运算和椭圆曲线点乘运算)。相应地，我们进一步将这些匿名认证协议分为两类：

(1)基于对称密码技术的匿名协议；(2)基于公钥技术的匿名协议。称前一种协议支持属性“DA5-SymmetricKey”，后一种协议支持属性“DA5-PublicKey”。在文献[347]中，我们发现这两个属性与安全目标SR6(即抗智能卡丢失攻击)间存在内在(但隐晦)的关系，在后文中我们将进一步阐述这一点。

7.6.2 指标内涵存在歧义

属性DA1要求服务器S不能存储和口令相关的验证数据，这样即使S被攻陷，用户口令也不会泄露。自Yang和Shieh[439]在1999年首次提出支持DA1的协议后，DA1就成为双因子协议中最基本的设计目标之一[388, 389]。如节7.3中Tsai等方案，大量试图实现DA1 的协议(如[334, 405, 422, 438])只在服务器端存储了用来验证用户的长期密钥(针对系统所有用户)，除此之外，未存储与用户相关的任何其它数据。另外一些协议(如节7.4中分析的协议和[174, 390, 423])也实现了DA1，但却在服务器端存储了一些和安全无关的用户特定的信息，比如$\{ID_i, T_{reg}\}$，其中ID_i是用户身份，T_{reg}是用户的注册时间。我们称前一种协议支持属性“DA1-Strong”，后一种协议支持属性“DA1-Weak”。

在口令认证协议中，除了能够自由选择口令(DA2)，另一个推荐做法是允许用户定期更新自己的口令。因此，如节7.2.1所述，口令更新阶段已经成为绝大多数双因子协议的一个基本阶段。该阶段允许用户在本地或者通过与服务器交互的方式来修改口令。如节7.3.2所述，一个支持用户本地修改口令但不验证旧口令正确性的协议，易遭受智能卡丢失攻击II(即未提供SR6)；一个支持用户在本地修改口令且明确检验旧口令正确性的协议，也会遭受同样的安全威胁。为便于说明，我们称前一个协议是具有属性“DA2-Local-Secure”，后者具有属性“DA2-Local-Insecure”。此外，对于支持用户自由选择口令，但不支持本地修改口令的协议，我们称之具有属性DA2-Interactive。

除了属性DA1和DA2，我们发现，另外两个设计目标也需要进一步阐释，即DA8(用户匿名性)和SR6(抗智能卡丢失攻击)。更确切地说，因为用户匿名性有两个不同层次的含义(见第六章)，DA8也应相应拆分成DA8-Weak和DA8-Strong。DA8-Weak是指用户身份标识保护，即用户身份标识不能通过协议交互的消息分析出来，部分协议(如[418, 440])只能实现这一层次的匿名性；高级的含义是用户行为不可追踪性，即用户身份标识和用户行为都不能通过协议交互的消息分析出来，部分协议(如[400, 441])可实现这种高层次的隐私概念。

基于过去分析300多个双因子协议的经验，我们发现在评估安全目标SR6时，应该至少考虑8种不同类型的智能卡丢失攻击(Type I∼Type VIII)。如表7.5所示，这8种攻击类型各自利用了完全不同的攻击策略，都是非常现实的威胁。注意，表7.5给出的攻击中，每类都至多与服务器在线交互一次。这意味着，攻击

表 7.5 智能卡丢失攻击(SR6)的分类†

Smart-card-loss attack types		Extract card	Return extracted card	# of online interactions	Weaknesses exploited	Goals breached	Typical reference
Passive	Type I	No	Yes	0	No verification when PW change	Usability	Sec. 7.3.2
	Type II	Yes	No	0	Definite verifier for PW change	2FS	Sec. 3.2.2 of [345]
	Type III	Yes	No	0	Partition of non-group PWs	2FS	Sec. 3.5 of [398]
	Type IV	Yes	No	0	Protocol flow 1	2FS	Sec. 3.2.1 of [345]
	Type V	Yes	Yes	0	Protocol flow 2, pre-computation	2FS	Sec. 3.2 and 3.4 of [397]
	Type VI	Yes	Yes	0	Protocol flow 1, pre-computation	2FS	Sec. 7.3.2
Hybrid	Type VII	Yes	No	1 with S	Protocol flow 2	2FS	Sec. 5.2.2 of [345]
	Type VIII	Yes	Yes	1 with U	Protocol flow 3	2FS	Sec. 4.2 of [279]

†PW = passwords; 2FS= Two-factor security.

的绝大部分工作都是以离线的方式完成，不会受到系统安全机制(例如异常登录检测和帐户锁定[19])的限制。此外，有些攻击(例如Types I，II和IV)比较通用，而有些攻击(例如TypeIII)则只对那些不恰当地将群元素与口令(或口令的哈希)进行运算的协议有效。如果协议设计者忽视了其中任意一种威胁，那么提出的协议很有可能无法实现SR6。同理，协议分析者/安全工程师在评价协议时，如果忽视它们中的任意一个也会得到不可靠的结论，导致不恰当的选择。

小结。上述四个解释使得Madhusudhan-Mittal评价标准更加具体明确，大大增强了其可操作性。这不仅显示了该标准的缺陷，而且为我们新标准的提出奠定了基础。尽我们所知，我们首次对智能卡丢失攻击(SR6)进行了分类。考虑到“目前的关键难题是如何在非抗窜扰智能卡条件下，实现真正的双因子安全性”[279]，以及表7.5中列出的8种智能卡丢失攻击中有7种可能会破坏协议的双因子安全性，我们对攻击者行为的深入理解是解决“目前的关键难题”的重要一步。

7.6.3　指标间存在冗余

除了上述歧义，我们还注意到SR6与DA4、SR9与DA3之间存在冗余。如果协议能实现 SR6，则意味着获得受害者智能卡的$\mathcal{A}$无法轻易修改智能卡的口令、进行离线口令猜测或仿冒用户。这意味着，在这种情况下，系统安全依赖于用户口令的安全性。因此，实现SR6也表明认证过程是与用户口令相关的。所以DA4完全包含在SR6中。同理，DA3完全包含在SR9中。但这两个冗余并不像上面介绍的歧义那样，严重地影响Madhusudhan-Mittal标准的可用性。

7.6.4　指标间存在内在冲突

本章主要关注评价指标之间最基本的三种关系：共生、互斥和派生，分别表示为∞、$\otimes$和$\triangleright$。更确切地说， DAi和SRj是共生关系(即DAi∞SRj)：只要协议能实现二者中的任一属性，则意味着两种属性可同时实现。 DAi和SRj是互斥关系(即DAi$\otimes$SRj)：当且仅当协议只能实现这两个属性中的至多一个。DAi和SRj是派生关系，当且仅当下面2种情况至少成立1个：(1)如果协议能满足DAi，那么也能满足SRj，即有DAi$\triangleright$SRj；(2) 如果协议能够支持SRj，那么也能支持DAi，即

表 7.6 Madhusudhan-Mittal评价标准中各指标之间的关系

Design Goals	DA1: No password verifier table	DA2: Password friendliness	DA3: No password reveal	DA4: Password dependent	DA5: Mutual authentication	DA6: Session key agreement	DA7: Forward secrecy	DA8: User anonymity	DA9: Smart card revocation	DA10: timely typo detection	SR1: Resist to DoS attack	SR2: Resist impersonation attack	SR3: Resist parallel session attack	SR4: Resist password guessing attack	SR5: Resist replay attack	SR6: Resist smart card loss attack	SR7: Resist stolen-verifier attack	SR8: Resist reflection attack	SR9: Resist insider attack
DA1-Strong	*	*	*	*	*	*	*	*	⊗	*	*	*	*	*	*	*	▷	*	*
DA1-Weak	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	▷	*	*
DA2-Local-Secure	*	*	*	*	*	*	*	*	*	∞	*	*	*	*	*	⊗	*	*	*
DA2-Local-Insecure	*	*	*	*	*	*	*	*	*	⊗	*	*	*	*	*	⊗	*	*	*
DA2-Interactive	*	*	*	*	*	*	*	*	*	⊗	*	*	*	*	*	◁	*	*	*
DA3	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	◁
DA4	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	◁	*	*	*
DA5-SymmetricKey	*	*	*	*	*	*	⊗	⊗	*	*	*	▷	▷	*	▷	⊗	*	▷	*
DA5-PublicKey	*	*	*	*	*	*	*	*	*	*	*	▷	▷	*	▷	*	*	▷	*
DA6	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
DA7	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
DA8	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
DA9	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
DA10	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	⊗	*	*	*

Note: X∗Y means the relation between X and Y is unknown (probably independent); X∞Y means a symbiotic relation between X and Y; X▷Y means Y is implied by X; X⊗Y means a mutually exclusive relation between X and Y.

有SRj▷DAi。表7.6总结了我们的研究结果，具体如下：

1) DA1-Strong⊗DA9。协议支持DA1-Strong意味着服务器端没有存储任何用户特定(或者智能卡特定)的信息。但是，如果智能卡授权的期限已到，服务器(例如协议[400, 405, 422])如何区分过期的卡和有效的卡呢？尽我们所知，服务器S无法实现这个目标。有人可能会想，对那些过期的卡，S可以在卡里加入其有效期限的信息以及该信息的数字签名，然后通过验证签名，S就可以分辨出有效的卡和无效的卡；对那些已撤销的卡，S可以像传统的证书颁发机构一样维护一个撤销列表。但是，这样的话就要求S存储一个撤销列表，这违背了在S上不存储任何用户特定数据这一目标(即DA1-Strong)。

2) DA1-Strong▷SR7。既然协议在服务器上没有存储任何用户相关验证表项，自然可防止窃取验证项攻击。反之，一个可抵御窃取验证项攻击的协议可能支持DA1-Weak，但不一定支持DA1-Strong。

3) DA1-Weak▷SR7。既然协议在服务器上没有存储安全相关的验证表项，自然可防止窃取验证项攻击。反之，一个不会受到窃取验证项攻击的协议可能支持DA1-Strong，但是不一定支持DA1-Weak。

4) DA2-Local-Secure∞DA10。实现DA2-Local-Secure协议支持用户在本地安全地更新口令，这意味着智能卡中要存储一些和口令相关的验证项(例如[422]中的$A_i = h(ID_i \| PW_i)^{PW_i} \bmod p$，或者$\{r, N = h(r\|x) \times h(PW_i)$，[382]中的$Y = h(ID_i \| h(r\|x))\}$)，这些口令相关的验证项可被用来检测用户在登录时是否意外地键入了错误的口令。

5) DA2-Local-Secure⊗SR6。实现DA2-Local-Secure的协议支持用户在本地

安全地更新口令，这意味着智能卡中要存储一些和口令相关的验证项(例如[422]中的$A_i = h(ID_i \| PW_i)^{PW_i} \bmod p$，或者$\{r, N = h(r\|x) \times h(PW_i)$，[382]中的$Y = h(ID_i \| h(r\|x))\}$)。并且，一旦攻击者得到了智能卡并分析出这些口令相关的验证项，那么该验证项可被用来验证猜测口令的正确与否。

6) DA2-Local-Insecure⊗DA10。这一关系能够直接从DA2-Local-Secure ∞ DA10中推导出来。

7) DA2-Local-Insecure⊗SR6。实现DA2-Local-Insecure的协议支持用户在本地更新口令，但是在智能卡里没有存储与口令相关的验证项。显然，如节7.3.2所述，$\mathcal{A}$可以轻易地修改用户的口令。另一方面，协议能实现SR6意味着即使$\mathcal{A}$ 获得了智能卡也无法轻易更改口令，这表明协议不支持DA2-local-insecure。

8) DA2-Interactive⊗DA10。实现DA2-Interactive的协议(如[289, 421])要求用户通过与服务器的交互来修改口令。这意味着，智能卡中没有存储与口令相关的验证项。没有验证项，智能卡就无法检验用户在登录时是否意外键入了错误的口令。这表明，支持DA2-Interactive的协议将不支持DA10，反之亦然。

9) DA5▷SR2, SR3, SR5 和SR8。支持双向认证的协议(即DA5)能够阻止攻击者仿冒用户或者服务器。因而排除了仿冒攻击、平行会话攻击、重放攻击和反射攻击的可能性。否则，就违背了DA5。另一方面，实现 SR2(或者SR3、SR5和SR8)并不意味着实现了DA5。比如一个实现了SR8的协议(如[150])，可能依旧受到仿冒攻击，不能实现DA5和SR2。

10) DA5-SymmetricKey⊗DA7, DA8 and SR6。实现DA5-SymmetricKey的协议表明通过对称密钥技术来实现双向认证。在表7.1中 C-2 假设下，根据第六章，没有采用公钥密码技术的协议本质上都无法实现DA8；根据文献[347]，没有采用公钥密码技术的协议本质上都不能提供DA7和SR6。

11) DA10⊗SR6。协议支持DA10(例如[402, 422])，意味着智能卡能够检测用户在登录时是否意外输入了错误的口令。因此，智能卡中应该存储一个和口令相关的验证项。这样的话，$\mathcal{A}$就能够利用这一验证项实施智能卡丢失攻击。

12) SR6▷DA4。根据[170]中给出的定义，DA4完全包含在SR6中：SR6和DA4 考虑的都是用户丢失智能卡的情况，但是SR6要求$\mathcal{A}$无法实施仿冒攻击、离线猜测攻击或者轻易地改变口令(参见7.3.2)；然而支持DA4的协议只能保证$\mathcal{A}$只有在知道口令的情况下才能仿冒用户(但$\mathcal{A}$可能会实施其它恶意操作，如更改口令)。举例示之：[405]中的协议无法实现DA4和SR6；但是[174, 177, 438]中的协议实现了DA4，却未实现SR6。

从表7.6中可以看出，“DA2-*”和某些指标之间总是互斥(由⊗表示)。更确切地，DA2-Local-Secure 和 DA2-Local-Insecure都与SR6互斥，而DA2-Interactive则与DA10互斥。这意味着无论用户如何修改口令(即本地的或者交互式的)，

SR6和DA10都将无法同时实现。这表明，在现有密码学工具下，不大可能设计一个满足Madhusudhan-Mittal评估标准[170]中所有指标的“理想”协议。幸运的是，通过在协议中引入第四章中的Honeywords这一系统安全技术，我们在第八章成功实现了这三者的合理平衡。

7.7 我们的评价标准

如上节所述，现有标准都存在这样或那样的缺陷，难以实用。为此，本节提出一个全面、明确的新的评价标准。

7.7.1 具体的12项评价指标

基于分析300余个双因子协议的经验和文献[151, 170, 278]的研究结果，借鉴文献[391]中的迭代方法对评价标准进行细化，从双因子认证协议需满足的用户友好性和安全性方面，本节提出12个独立的评价指标：

C1. **无口令验证表项**：服务器端无需存储用户口令或口令相关的数据；

C2. **口令友好性**：用户口令方便记忆，且可以在本地自由更新；

C3. **无口令泄露**：用户口令应无法被服务器内部管理人员获知；

C4. **抗智能卡丢失攻击**：协议免受智能卡丢失攻击表明，即使攻击者$\mathcal{A}$获取了受害者的智能卡，卡内秘密参数被分析出来，$\mathcal{A}$也无法更改智能卡口令、离线猜测用户口令或仿冒用户。协议应至少抵御表7.5中的8种子类型抗智能卡丢失攻击。

C5. **抗各种已知攻击**：协议可以抵抗各种基本的和复杂的攻击，包括离线口令猜测攻击、重放攻击、平行会话攻击、异步攻击、窃取验证项攻击、仿冒攻击、密钥控制攻击、已知密钥共享攻击、已知密钥攻击；

C6. **可修复性**：协议允许智能卡重用，具备良好的可修复性，即用户可以在不改变用户名的情况下重用智能卡；

C7. **密钥协商**：客户端和服务器端应能在认证阶段协商一个用于后续安全通信的会话密钥；

C8. **无时钟同步**：协议不受时钟同步和网络时延的影响，即服务器无需维护与智能卡同步的时钟，反之亦然；

C9. **检测错误口令输入**：协议及时检测用户登录时是否键入了错误口令；

C10. **双向认证**：用户和服务器可验证彼此的真实性；

C11. **用户匿名性**：协议保护用户身份标识，且实现用户行为不可跟踪性；

C12. **前向安全性**: 即使一方或多方长期私钥被$\mathcal{A}$获取，那些先前会话中建立的会话密钥的机密性仍不受危胁。

值得注意的是，C4涉及的是$\mathcal{A}$已获得受害者智能卡的攻击场景，而C5考虑的是在$\mathcal{A}$未获得受害者智能卡时的攻击场景。C4不仅考虑了传统的智能卡丢失攻击场景(参见[170])，而且考虑了新近提出的攻击场景(如文献 [397]中的利用服务器作为预言机的攻击和文献[279]中的归还智能卡攻击)，应至少抵御表7.5中的8种子类攻击场景。C5建立在PAKE 协议应防御的安全威胁[56, 375]之上，还考虑了与会话密钥相关的安全概念[148]，以及双因子认证环境中出现的新攻击方式(如窃取验证项攻击)。

新标准不仅消除了现有标准中的冗余和歧义，而且其明确、具体的特点使之便于实施协议评估。以 Madhusudhan-Mittal 标准[170]为例，其中的“SR9：内部人员攻击”和“G3：无口令泄露”本质上具有相同的含义，而“G4：口令独立”包含在“SR6：智能卡丢失攻击”中，因为如果协议不能确保口令独立，则必然易遭受智能卡丢失攻击。除此之外，文献[170]未考虑广泛接受(见[278, 398])的重要属性“口令自由更新”。可以验证，Madhusudhan-Mittal 标准的指标集[170]是新标准指标集的子集。

比Madhusudhan-Mittal 标准更早的标准(见[278, 388–390])也存在众多缺陷(如歧义或过于宽泛)。文献[388–390]中的标准包含一项关于性能的指标：“协议必须是高效、实用的”。然而，协议的高效性和实用性包含太多方面，过于宽泛以致难以明确测量。协议的效率取决于具体的软硬件实现环境，而实用性是一个很宽泛的综合目标且很大程度上与目标应用相关，因此我们的标准明确不考虑“高效性和实用性”这一条。这将使之更明确、更易判定。除了这一性能相关指标，文献[388, 390]中的指标均包含在我们的新指标集中。虽然Yang等标准[278]不含性能相关指标，但它仅由4个指标组成(C2-C5)，显然不够全面系统。

最后，必须承认我们的标准不是完美的。也许明天会有更好的标准被提出，其实任何这样的标准都可以随着对双因子协议理解的深入而不断扩展、优化。我们主要关注过去文献中曾广泛讨论的指标，抵制了创造新指标的诱惑，因为过去很少或未曾被关注的指标，可以认为是无关紧要的。尽管我们的标准不是完美的，相比早期的标准，它更具体、更全面、更易判定，有助于更好地理解当前和未来的双因子协议的优劣。这对公平客观地评估协议是至关重要的，同时也有助于设计出实现“可用性-安全性”更佳平衡的协议。

7.7.2　评价指标体系有效性的讨论

为显示我们指标体系的全面性和实用性，我们对67个典型双因子协议进行了评估：通过检验节7.7.1中提出的12项指标是否在节7.2.2提出的安全模型下得到满足。特别地，在这67个协议中，有4个设计于2004年之前，它们均假设智能卡是抗窜扰的。不出所料，这4个协议在新安全模型下表现不佳(见表7.7.7)，而

越近期的协议通常表现得更好。在我们的评估框架下，存在一种明显的趋势：越新的协议表现越好(如[221, 372, 403, 440–442])。这符合事物发展规律。

然而，如果利用现有的评估标准(即[170, 278, 388, 398])对这67 个协议进行评估，这一趋势并不明显。更具体地说，文献[278]中的标准只包括我们标准的C2-C5项，因此不会发现2010年提出的协议与2015年提出的协议之间的差异；文献[388, 398]的标准关注“协议效率”，且未明确定义安全模型，这将使得这两个标准可操作性差；文献[170]的标准由于缺乏“口令自由更新”属性，无法揭示极为关键的“安全性——可用性”冲突(稍后讨论)。总之，这些揭示的趋势一定程度上显示了我们评价指标体系的有效性。

从微观角度来看，每一项指标至少有15个协议可以满足，同时至少有7个协议未满足。这表明12个指标中每一项都是不可或缺的。此外，不存在可以实现所有12个指标的协议——Odelu等[441]在2015年提出的协议唯一可实现11个指标。这表明了我们标准的全面性，也凸显了投入更多的研究以设计更好的协议的必要性。仔细分析之后，我们发现Odelu等协议与最新的协议(如[221, 440, 442])存在同样的“安全性–可用性”冲突：不能同时实现C2(或C9)和C4。这显示了从C5中分离出C4 的必要性，揭示文献[278]中标准的不足。

需要指出的是，对表7.7.7中的67个协议的选择，主要综合考虑了时间代表性、技术路线代表性和指标实现代表性这三方面，尤其是那些处于双因子协议发展历程中重要节点的协议(见图1.12)。因此，在表7.7.7中评估了的协议，我们并不认为它们一定会比未包含在该表中的其它协议表现更优，仅因为它们具有一定的代表性。此外，我们主要关注单服务器架构的协议，因为：(1)不同的架构/环境可能涉及差异较大的安全模型，因此在单一安全模型下几乎不可能进行公平对比；(2) 基于单服务器架构的协议是构造其它更复杂架构/环境(如WSNs[282]和移动网络[443])下协议的基石。

最后，过去五年中本文作者分析了300多个双因子协议，其中包括170多个单服务器架构(见[279, 345, 347])，100多个多服务器架构(参见[183, 348])，60多个移动网络(见[290, 357])和50多个无线传感器网络(见节6.4.2)。根据我们过去的协议分析经验，可以相信：(1)当标识一个协议无法实现某指标时，这基本确凿无误；(2) 当一个协议被标识为能够实现某指标时，这很可能无误，因为我们在分析协议时有可能会遗漏潜在的攻击场景。比如，在评估一个协议是否可以抗智能卡丢失攻击(即指标C4)时，至少要考虑8种不同的攻击场景(见表7.5)，而这仅仅是一个协议的完整评估工作的冰山一角。这说明了公平合理评价一个复杂密码协议面临的巨大挑战。即使采用了一个较为完善的协议评价指标体系，完成如表7.7.7 的评估工作，在难度和工作量方面都是巨大的。

表 7.7.7 利用本章提出的评价指标体系测量67个双因子协议

Scheme	Year	Ref.	No verifier table (C1)	Password friendly (C2)	No password exposure (C3)	No smart card loss problem (C4)	Resistance to known attacks (C5)	Sound repairability (C6)	Provision of key agreement (C7)	No clock synchronization (C8)	Timely typo detection (C9)	Mutual authentication (C10)	User anonymity (C11)	Forward secrecy (C12)
Lu *et al.*	2016	[385]	✓	✓	✓	·	·	✓	✓	✓	✓	✓	·	✓
Xie *et al.*	2016	[444]	✓	✓	✓	·	✓	·	✓	·	·	✓	·	✓
Islam *et al.*	2016	[386]	✓	✓	✓	·	✓	·	✓	✓	✓	✓	✓	✓
Muhaya	2015	[403]	✓	✓	✓	·	✓	·	✓	✓	✓	✓	✓	·
Xu-Wu	2015	[442]	✓	·	✓	✓	·	✓	✓	✓	·	✓	·	✓
Mishra *et al.*	2015	[440]	✓	✓	✓	·	✓	✓	✓	·	✓	✓	✓	·
Odelu *et al.*	2015	[441]	✓	✓	✓	·	✓	✓	✓	✓	·	✓	✓	✓
Byun	2015	[221]	·	·	✓	✓	✓	✓	✓	✓	·	✓	✓	✓
Wang *et al.*	2015	[372]	✓	·	✓	✓	✓	✓	✓	·	✓	✓	✓	·
Truong *et al.*	2015	[383]	✓	✓	·	·	✓	·	✓	✓	·	✓	·	✓
Djellali *et al.*	2015	[445]	✓	✓	·	·	✓	·	✓	✓	·	✓	✓	·
Mishra *et al.*	2015	[446]	✓	·	✓	·	✓	✓	✓	·	✓	✓	·	✓
Wu *et al.*	2015	[447]	·	✓	✓	·	·	·	✓	✓	·	✓	✓	✓
Chaudry *et al.*	2015	[401]	✓	✓	✓	·	·	✓	✓	·	✓	✓	·	·
Kumari *et al.*	2014	[363]	✓	✓	✓	·	✓	·	✓	✓	✓	✓	·	·
Chen *et al.*	2014	[409]	✓	·	·	·	✓	·	✓	·	·	✓	·	·
Tsai *et al.*	2013	[400]	✓	✓	✓	·	✓	·	✓	✓	·	✓	✓	✓
Li *et al.*	2013	[422]	✓	✓	·	·	✓	·	✓	·	✓	✓	·	✓
Kumari-Khan	2013	[406]	✓	·	·	·	✓	·	✓	·	·	✓	·	·
Sun-Cao	2013	[448]	✓	·	✓	·	✓	✓	✓	·	·	✓	✓	·
Lee-Liu	2013	[449]	✓	✓	✓	·	✓	·	✓	✓	·	✓	✓	·
Islam-Biswas	2013	[425]	·	·	·	✓	·	✓	✓	·	·	✓	·	✓
Li-Zhang	2013	[450]	✓	·	·	✓	·	·	✓	·	·	✓	✓	✓
Chang et al.	2013	[438]	✓	·	·	·	·	·	✓	·	·	✓	·	·
Li	2013	[421]	✓	·	✓	·	·	✓	✓	✓	·	✓	✓	✓
Kim-Kim	2012	[407]	✓	·	·	✓	·	✓	✓	✓	·	✓	✓	·
Ramasamy et al.	2012	[451]	✓	✓	·	✓	·	·	·	·	·	·	·	·
Wu *et al.*	2012	[398]	✓	✓	✓	·	✓	✓	✓	✓	·	✓	✓	✓
Zhu	2012	[452]	✓	✓	✓	·	✓	✓	·	✓	·	✓	·	·
Wang-Ma	2012	[453]	✓	✓	✓	·	✓	✓	✓	·	✓	✓	✓	✓
Hsieh-Leu	2012	[454]	✓	·	·	·	✓	✓	·	·	·	✓	·	·
Wen-Li	2012	[352]	✓	·	·	·	·	✓	✓	·	·	✓	·	·
Wang *et al.*	2012	[412]	✓	✓	✓	·	✓	·	✓	✓	✓	✓	✓	✓
He *et al.*	2012	[455]	✓	·	✓	✓	✓	·	✓	·	·	✓	·	·
Wang *et al.*	2011	[390]	✓	✓	·	·	·	✓	✓	✓	·	✓	✓	✓
Chen *et al.*	2011	[456]	✓	✓	✓	·	·	·	✓	·	·	✓	·	·
Khan *et al.*	2011	[174]	✓	✓	✓	·	✓	✓	✓	·	✓	✓	·	·
Kim *et al.*	2011	[457]	✓	·	·	✓	✓	✓	✓	·	·	✓	·	·
Awasthi *et al.*	2011	[458]	✓	·	·	·	·	·	·	·	·	✓	·	·
Li-Lee	2011	[413]	✓	·	✓	·	·	✓	✓	✓	·	✓	·	·
Sood *et al.*	2011	[459]	✓	✓	·	·	·	✓	✓	·	✓	·	·	·
Li *et al.*	2010	[289]	✓	✓	✓	✓	·	✓	✓	✓	·	✓	·	·
Tsai *et al.*	2010	[460]	✓	✓	·	·	✓	·	✓	·	·	✓	✓	·
Song	2010	[461]	✓	·	·	·	✓	·	✓	·	·	✓	·	·
Sood *et al.*	2010	[462]	✓	✓	·	✓	✓	·	✓	·	·	·	·	·
Yeh *et al.*	2010	[382]	✓	✓	·	·	✓	✓	✓	·	·	✓	·	·
Horng *et al.*	2010	[463]	✓	·	✓	✓	·	·	✓	✓	·	✓	✓	✓
Sun *et al.*	2009	[365]	✓	✓	✓	·	✓	✓	✓	·	·	✓	·	✓
Hsiang-Shi	2009	[169]	✓	✓	·	·	✓	✓	·	·	·	✓	·	·
Xu *et al.*	2009	[150]	✓	·	·	·	·	·	✓	·	·	✓	·	✓
Kim-Chung	2009	[464]	✓	✓	·	·	·	·	·	·	✓	✓	·	·
Chung *et al.*	2009	[465]	✓	·	✓	✓	✓	✓	✓	✓	·	✓	·	✓
Ramasamy	2009	[466]	✓	·	·	✓	✓	✓	·	·	·	✓	·	·
Yang *et al.*	2008	[278]	✓	✓	✓	·	✓	·	✓	✓	·	✓	·	✓
Juang *et al.*	2008	[418]	✓	·	✓	✓	·	✓	✓	✓	·	✓	·	·
Chen *et al.*	2008	[467]	✓	·	✓	·	·	·	·	✓	·	✓	·	·
Wang *et al.*	2007	[168]	·	✓	✓	·	·	·	✓	·	✓	✓	·	·
Wang *et al.*	2007	[468]	✓	·	✓	·	·	✓	✓	✓	·	✓	·	·
Liao *et al.*	2006	[388]	✓	✓	✓	·	·	·	✓	·	·	✓	·	✓
Lee et al.	2005	[469]	✓	·	·	·	·	·	·	·	·	✓	·	·
Fan et al.	2005	[399]	✓	·	·	✓	✓	✓	·	✓	·	✓	·	·
Lu-Cao	2005	[470]	✓	·	·	✓	✓	·	·	·	·	·	·	·
Yoon *et al.*	2004	[167]	✓	✓	✓	·	·	✓	·	·	·	✓	·	·
Ku *et al.*	2004	[166]	✓	✓	✓	·	·	✓	·	·	·	✓	·	·
Wu-Chieu	2003	[161]	✓	✓	·	·	·	·	·	·	·	·	·	·
Awasthi-Lal	2003	[162]	·	·	·	·	·	·	·	·	·	·	·	·
Sun	2000	[160]	·	·	·	·	·	·	·	·	·	·	·	·
Hwang-Li	2000	[471]	·	·	·	·	·	·	·	·	·	·	·	·

小结。对比结果表明，节7.2.2中我们的攻击者模型迄今最为严格，而本节我们的标准比现有标准更具体、更简明、更全面。由于任何协议都是在某个给定安全模型上才满足某种指标 [279, 375]，故本文将这二者结合起来共同构成一个系统性的协议评估框架，即形成一个*评价体系*。可以预期，这一评价体系将具有长期的科学价值。通过对67个典型的协议进行评估，我们证实了这一评价体系的有效性。需要指出的是，我们的每一个评价指标及其定义都是在评估这些协议过程中反复精炼和重构的结果；一旦当所有评价指标稳定后，又重新对这67个协议统一进行最终评估。评估结果显示，每项标准都至少有15个协议满足，但也至少有7个协议不满足，这说明每项指标都是不可或缺的。另一方面，每个协议都至少有一个未实现的指标，这表明本文标准的全面性，也意味着需要更多的努力来设计更好的协议。

7.8 本章小结

本章系统性地研究了如何公平合理地评价一个基于口令的双因子认证协议。我们首先尝试回答一个基本问题：*是否有可能构建一个满足*Madhusudhan-Mittal*标准中所有指标的理想协议*？以Tsai等协议和Li协议为示例，我们揭示了设计此类协议的难点和微妙之处；指出Madhusudhan-Mittal标准存在的歧义和冗余，一些评价指标间存在本质冲突，这为前述问题提供了否定的答案。最为根本的，我们发现一个支持用户口令本地更新(DA2)的协议无法实现“SR6: 抗智能卡丢失攻击”，而一个不支持DA2的协议则无法提供属性“DA10:口令错误输入及时检测”。这表明在可用性和安全性间存在不可避免的平衡，而现有协议极少考虑到这一点，故产生顾此失彼的情况。

接着，根据“在给定安全模型下评估某项安全目标才有意义”这一基本思想，基于我们分析300多个双因子协议的经验，本章提出了一个新的协议评价指标体系，包括一个迄今最为严格(但现实)的安全模型和一套完善的协议评价指标。通过对67个典型的协议进行评估，我们证实了这一评价体系的有效性。特别地，我们的每一项评价指标及其定义都是在评估这些协议过程中反复精炼和重构的结果。我们的这项工作为未来公平合理评价双因子协议提供了一个基准，为该领域研究摆脱“攻击→ 改进→ 攻击→ 改进”的怪圈迈出了坚实一步。

本章还遗留了两个重要的问题有待研究：(1)如何利用最近泄露的大规模真实口令数据集(如1600万Dodonew口令和600万CSDN口令)，来评估节7.3.2中“模糊验证因子”这一方法的有效性和可行性？(2)如何设计一个可实现本章所提12个指标的协议？下一章我们将解决这两个问题。

第八章　基于口令的双因子认证协议设计与分析

本章研究如何应用前述章节中的口令Zipf分布理论、Honeywords理论和口令协议中匿名性的本质计算复杂度等口令安全理论，来设计新的安全高效的口令认证协议，并基于第七章中提出的评价指标体系对协议进行评估。本章首次将系统安全中的Honeywords技术引入口令安全协议设计，提出将“Honeywords”技术与“模糊验证因子”技术相结合的思想，使双因子认证协议可及时识别攻击者的在线猜测行为，在满足可用性指标的同时，实现超越传统上限的安全性。在修正的随机预言机模型下，证明了协议的安全性。特别地，我们的新协议在节7.2.2的安全模型下，实现了节7.7.1的全部 12 项评价指标，解决了 Huang 等[397] 2014年初提出的公开问题。

8.1　新协议描述

本节提出一个简单、强健、高效的基于口令双因子认证协议。本章协议建立在文献[412]中的双因子协议之上。为克服[412]中协议无法同时实现C2和C4的缺陷，本章将第四章中的Honeywords思想引入口令认证协议设计中，并通过将“honeywords”与“fuzzy-verifier”技术相结合，使新协议可及时检测攻击者通过利用受害者智能卡的参数发起的在线口令猜测攻击，实现了超越传统上限的安全性。新协议包含四个阶段：注册、登录、认证和口令更新。本章所采用的符号定义同表7.2。

8.1.1　注册阶段

协议定义在阶为q的循环群$\mathbb{G}=<g>$上，其中素数q的长度为$\ell \geq 512$ 比特，群$\mathbb{G}$可以是有限域群，也可以是椭圆曲线群。本章假设群$\mathbb{G}$为$\mathbb{Z}_p^*$的素阶子群，其中，$\mathbb{Z}_p^*=1,2,\cdots,p-1$，$p$为素数，$q|p-1$。Hash 函数$\{0,1\}^* \rightarrow \{0,1\}^{l_i}$表示为$\mathcal{H}_i(\cdot)$，其中，$l_i$为Hash函数输出的比特长度(如$l_i=160$)，$i=0,1,2,3$。整数$n_0$为中等大小，如$2^4 \leq n_0 \leq 2^8$，用来决定抗在线猜测池 (ID,PW) 的容量(与后文的模糊验证因子A_i相关)。$(x, y=g^x \bmod p)$分别表示服务器 S 的私钥和公钥，x 由S 安全存储，而y存储在每个用户的智能卡中。注册过程如下：

Step R1. 用户U_i 选择身份标识 ID_i，口令PW_i和随机字符串 b。

Step R2. $U_i \Rightarrow S:\{ID_i, \mathcal{H}_0(b\|PW_i)\}$。

Step R3. 服务器S 在时间T接收到用户U_i的注册请求后，S首先选择随机数a_i并

计算$A_i = \mathcal{H}_0((\mathcal{H}_0(ID_i) \oplus \mathcal{H}_0(b\|PW_i))\ \texttt{mod}\ n_0)$。$S$检查$U_i$是否是已注册的用户。如果$U_i$是初次注册，$S$在帐户数据库中为$U_i$创建一条新的表项，并存储$\{ID_i, T_{reg}{=}T, a_i,$ Honey_List=NULL$\}$；否则，S仅将T_{reg}的值更新为T，a_i更新为新选择的a_i，设置Honey_List为NULL。S计算$N_i = \mathcal{H}_0(b\|PW_i) \oplus \mathcal{H}_0(x\|ID_i\|T_{reg})$。

Step R4. $S \Rightarrow U_i$: 带参数$\{N_i, A_i, A_i \oplus a_i,\ q, g,\ y, n_0,\ \mathcal{H}_0(\cdot), \cdots, \mathcal{H}_3(\cdot)\}$的智能卡。

Step R5. 当用户U_i收到智能卡SC后，U_i先激活智能卡，再将随机字符串b写入智能卡中。注意，SC要求U_i输入两次b，以确保输入的正确性。

注意，在第R1步U_i可以将选择的随机字符串b写在纸上，在R5步之后可以销毁该纸张。这样，U_i在任何时刻都无需记住b，而b也可以尽可能随机以满足C3：无口令泄漏(见节7.7)。协议只需要T_{reg}有足够的信息熵以抗猜测即可。在某些情况下时戳T可能只有64位或者更短，这时不妨令$T_{reg} = T \parallel X$，其中X为大随机数。如后文所示，S保存Honey_List和a_i是为了实现对“智能卡被攻陷”事件的及时检测。

8.1.2 登录阶段

Step L1. 用户U_i插入智能卡SC，输入ID_i^*, PW_i^*。

Step L2. SC计算$A_i^* = \mathcal{H}_0((\mathcal{H}_0(ID_i^*) \oplus \mathcal{H}_0(b\|PW_i^*))\ \texttt{mod}\ n_0)$，通过比较$A_i^*$是否与存储的$A_i$相等来验证$ID_i^*$和$PW_i^*$的正确性。若不等，则终止会话。

Step L3. SC选择随机数u，计算$C_1 = g^u\ \texttt{mod}\ p$, $Y_1 = y^u\ \texttt{mod}\ p$, $k = \mathcal{H}_0(x\|ID_i\|T_{reg}) = N_i \oplus \mathcal{H}_0(b\|PW_i^*)$, $a_i{=}(A_i \oplus a_i) \oplus A_i$, $CID_i = ID_i^* \oplus \mathcal{H}_0(C_1\|Y_1)$, $CAK_i = (a_i\|k) \oplus \mathcal{H}_0(Y_1\|C_1)$和$M_i = \mathcal{H}_0(Y_1\|k\|CID_i\|CAK_i)$。

Step L4. $U_i \rightarrow S : \{C_1, CID_i, CAK_i, M_i\}$。

注意，计算CAK_i时$\mathcal{H}_0(\cdot)$的输入顺序与计算CID_i时$\mathcal{H}_0(\cdot)$的输入顺序正好相反。这是为了防止信息泄漏，否则$\mathcal{A}$可通过$CID_i \oplus CAK_i$，从而恢复出$a_i\|k$。实际上，我们也可以令$CID_i = (ID_i^*\|a_i\|k) \oplus \mathcal{H}_0(C_1\|Y_1)$，不用再计算$CAK_i$，但这改变了[412]中原协议的结构。为尽量减少对[412]中原协议的改变，我们单独计算了CAK_i，这样CAK_i的作用更明确，有利于显示如何将“honeywords”思想引入到传统协议中。在登录请求中使用验证项M_i来抵抗攻击者的能力C-01，即抵抗文献[397, 436]中提出的一种新的交互式智能卡丢失攻击。

8.1.3 认证阶段

服务器S接收到登录请求信息$\{C_1, CID_i, CAK_i, M_i\}$后，执行如下步骤：

Step V1. S用私钥x计算$Y_1 = (C_1)^x\ \texttt{mod}\ p$，然后，$S$获取$ID_i = CID_i \oplus \mathcal{H}_0(C_1\|Y_1)$，并检查$ID_i$的有效性。如果$ID_i$无效，则终

止会话；否则，S 继续执行下一步。

Step V2. S 计算 $k=\mathcal{H}_0(x\|ID_i\|T_{reg})$和 $M_i^*=\mathcal{H}_0(Y_1\ \|k\|CID_i\|CAK_i)$，其中 T_{reg} 为从ID_i 的帐户数据库中提取得到。如果 $M_i^*\neq M_i$，终止会话。

Step V3. S 解析 $a_i'\|k'=CAK_i\oplus\mathcal{H}_0(C_1\|Y_1)$，并验证$a_i'$是否与存储的$a_i$相等。若相等，表明$A_i$的值正确；否则，$S$ 拒绝登录请求。S 比较解析出的k'是否等于直接计算的k。如果相等，S继续执行后续步骤。如果不等，则S已知 $a_i'=a_i$，但是$k'\neq k$，这表明U_i的智能卡有$1-\frac{1}{2^{n_0}}$的概率被腐化。相应地，S执行以下二者之一：(1)当`Honey_List`中的条目小于m_0(比如$m_0=10$)时，如果k'已在`Honey_List`中，则S终止会话，否则S将k'写入`Honey_List`；(2)当`Honey_List` 中的条目达到m_0 时，挂起(Suspend)U_i的智能卡，直到U_i重新注册。

Step V4. S 生成随机数 v 并计算临时密钥 $K_S=(C_1)^v\ \mathtt{mod}\ p$, $C_2=g^v\ \mathtt{mod}\ p$ ，$C_3=\mathcal{H}_1(ID_i\|ID_S\|Y_1\|C_2\|\ k\|K_S)$。

Step V5. $S\rightarrow U_i:\{C_2,C_3\}$。

Step V6. 收到服务器S的响应信息，智能卡计算 $K_U=(C_2)^u\ \mathtt{mod}\ p$, $C_3^*=\mathcal{H}_1(ID_i\|ID_S\|Y_1\|\ C_2\|k\|K_U)$，并比较计算的 C_3^* 与接收的C_3是否一致。如果一致，则U_i成功认证S，并计算 $C_4=\mathcal{H}_2(ID_i\|ID_S\|Y_1\|C_2\|k\|K_U)$。

Step V7. $U_i\rightarrow S:\{C_4\}$。

Step V8. 接收到$\{C_4\}$后，S首先计算$C_4^*=\mathcal{H}_2(ID_i\|ID_S\|Y_1\|C_2\|k\|K_S)$，并验证$C_4^*$是否等于接收的$C_4$。若相等，$S$成功认证$U_i$，并接受$U_i$的登录请求；否则，终止会话。

Step V9. 用户 U_i 和服务器 S 协商会话密钥$sk_U=\mathcal{H}_3(ID_i\|ID_S\|Y_1\|C_2\|k\|K_U)=\mathcal{H}_3(ID_i\|ID_S\|Y_1\|C_2\|\ k\|K_S)=sk_S$，用于后续安全通信。

8.1.4　口令更新阶段

为实现用户友好性(满足C2)，同时保证安全性(满足C4)和通信效率，该阶段在本地执行，无需与服务器交互，执行过程如下：

Step P1. 用户U_i 插入智能卡，输入ID_i和相应的口令PW_i。

Step P2. 智能卡计算$A_i^*=\mathcal{H}_0((\mathcal{H}_0(ID_i)\oplus\mathcal{H}_0(b\|\ PW_i))\ \mathtt{mod}\ n_0)$ ，比较A_i^*与存储的A_i是否相等，如果相等，则意味着ID_i和PW_i正确的概率为$\frac{n_0-1}{n_0}(\approx\frac{99.61}{100}$，$n_0$=$2^8)$ 。

Step P3. 用户输入新的口令 PW_i^{new}，智能卡计算 $N_i^{new}=N_i\oplus\mathcal{H}_0(b\|PW_i)\oplus\mathcal{H}_0(b\|PW_i^{new})$, $A_i^{new}=\mathcal{H}_0((\mathcal{H}_0(ID_i)\oplus\mathcal{H}_0(b\|PW_i^{new})\ \mathtt{mod}\ n_0)$。然后，智能卡将$N_i$, A_i和$a_i\oplus A_i$分别更新为N_i^{new}, A_i^{new}和$a_i\oplus A_i^{new}$。

8.2 协议设计的基本原理

本节详述新协议的基本设计思想，论证“honeywords”与“fuzzy-verifier”相结合可实现“一石二鸟”：不仅解决了长期存在的“安全性 vs. 可用性”困扰，而且实现了超越传统安全上界的安全性。

8.2.1 基本思想

为实现最重要的目标—真正双因子安全性，协议在智能卡中存储长期密钥$k=\mathcal{H}_0(x\|ID_i\|T_{reg})$，其中，$x$为服务器主密钥。一方面，由于服务器需要计算出$U_i$的长期密钥$k$，故需要知道$U_i$的用户名$ID_i$和注册时间$T_{reg}$。为此，服务器需维护用户注册表项$\{ID_i,T_{reg}\}$。这并不违反C1。另一方面，$k$受口令保护，即使智能卡内参数$N_i$被分析出来，仍然不会泄露$k$。然而，$U_i$自己可以通过计算$k=N_i\oplus\mathcal{H}_0(b\|PW_i^*)$获取$k$，其中$N_i$和$b$存储在智能卡中。

为实现“口令本地自由更新”(即C2)，同时避免智能卡丢失所产生的问题(即C4)，在更新智能卡的参数N_i前需要验证口令的正确性。为此，如节7.3.2所述，除了N_i，智能卡内还需存储其它参数，这会引入新的脆弱性，比如离线口令猜测和用户仿冒。为更好地展示微妙之处，假设智能卡中存储了安全参数$A_i=\mathcal{H}_0(ID_i\|\mathcal{H}_0(PW_i))$。当$U_i$修改口令时，首先提交用户名$ID_i^*$和口令$PW_i^*$，智能卡验证$\mathcal{H}_0(ID_i^*\|\mathcal{H}_0(PW_i^*))$是否等于存储的$A_i$。显然，一旦攻击者$\mathcal{A}$提取了安全参数$A_i$，可以很容易在离线方式下猜测出正确的$(ID_i,PW_i)$，这将导致智能卡易遭受离线口令猜测攻击，最终导致C4无法实现。类似A_i的不安全计算方式在文献[177, 410, 412, 422]中皆有使用。根据表 7.1中攻击者的能力C-2(ii)，一旦A_i被提取，$\mathcal{A}$很容易获得正确的 (ID_i,PW_i) 。

然而，如果A_i的计算方式为$A_i=\mathcal{H}_0(\ \mathcal{H}_0(ID_i)\oplus\mathcal{H}_0(PW_i))\ \texttt{mod}\ n_0)$，可以确信，当$|\mathcal{D}_{id}|=|\mathcal{D}|=10^6$ [5, 34]，$n_0=2^8$时，存在$\frac{|\mathcal{D}_{id}|*|\mathcal{D}|}{n_0}\approx 2^{32}$ 个备选(ID,PW)对来阻止攻击者$\mathcal{A}$猜测出正确的(ID_i,PW_i)。其中，$|\mathcal{D}_{id}|$和$|\mathcal{D}|$分别表示用户标识空间和口令空间。即使具备C-01的能力(已知用户名)，因为存在$\frac{|\mathcal{D}|}{n_0}\approx 2^{12}$ 个备选口令(见节8.2.2)，$\mathcal{A}$仍会被挫败。为进一步从2^{12}个备选口令中排除掉不正确口令，$\mathcal{A}$只能通过与服务器交互的方式实施在线猜测攻击。

我们引入的“honeywords”技术(即本文中的`Honey_List`)可以及时检测并阻止这一攻击。在节8.2.3中，我们显示即使$\mathcal{A}$的在线猜测尝试次数低至10，它仍会以$1-\frac{1}{2^{80}}$的概率被成功检测出来；在节8.3中，我们将严格证明，这是$\mathcal{A}$获取口令、破坏协议的最佳攻击策略。这样，我们有效地阻止了攻击者$\mathcal{A}$获取(ID,PW)。我们称A_i的这种新型计算方式为“模糊验证因子”(fuzzy verifier)。注意，为及时检测到用户智能卡被攻陷事件的发生，服务器S需要存储A_i，但是要想同时实现C2和C4，并且抗异步攻击(一种拒绝服务攻击，见节7.4.2)，服

务器端不直接存储A_i，而是存储与 A_i 相对应的随机数a_i，同时智能卡中存储A_i和$a_i \oplus A_i$。

“模糊验证”的一个明显“副作用”就是用来实现口令错误输入的及时检测(C9)。协议能实现 C9 表明一旦U_i输入错误的(ID_i^*, PW_i^*)，可被及时检测出来，这就避免了产生无效的时延，在避免增加用户端计算和通信代价的同时，提高了用户体验。

我们注意到，一旦攻击者$\mathcal{A}$获得了智能卡的访问权，也会利用这一“安全性 vs. 可用性”平衡参数A_i。具体来说，$\mathcal{A}$可试图更新U_i的口令，然后：(I)尝试登录系统；或(II)归还智能卡。针对情况I，尽管$\mathcal{A}$以某种方法猜测出用户口令PW_i为PW_i^r，且满足$\mathcal{H}_0((\mathcal{H}_0(ID_i) \oplus \mathcal{H}_0(b \parallel PW_i^r)) \bmod n_0) = A_i$，但只有$\frac{1}{2^{12}}$的概率使得$PW_i^r = PW_i$ (见节8.2)。另一方面，当且仅当$PW_i = PW_i^r$时，$\mathcal{A}$才能恢复出正确的$k = \mathcal{H}_0(x \| ID_i \| T_{reg}) = N_i \oplus \mathcal{H}_0(b \| PW_i^r)$，并成功登录。这是因为$S$将在V2步验证$k$的正确性，伪造的 k 不会通过验证，同时“honeywords”也会对 k 进行验证(见节8.2.3)。针对情况II，这将造成拒绝服务攻击(DoS)。假设更新口令连续失败次数的阈值为每天5次，由于$\mathcal{H}_0(\cdots)$的输出值是随机的(见[375])，因此$\mathcal{A}$仅有$\frac{1.95}{100}(\approx \frac{5}{n_0} = \frac{5}{2^8})$的成功概率。这表明DoS攻击者不会轻易成功，即使攻击26天，成功概率也只有50% 。纵使DoS攻击成功，造成的危害也有限，U_i可以通过重新注册恢复智能卡。

综上所述，DoS攻击和在线口令猜测攻击是攻击者$\mathcal{A}$对我们新协议的最大威胁。幸运的是，$\mathcal{A}$从前者获益很少，而后者可以被及时检测。因此，本文协议可实现超越传统上限的安全性，即优于$q_{send}/|\mathcal{D}| + \epsilon$。

为避免口令泄露(C3)，本协议要求用户向服务器 S 提交$\mathcal{H}_0(b \| PW_i)$ 而非PW_i，其中b是S未知的随机数；为实现C6，S 的数据库中存储与U_i相关的表项(ID_i, T_{reg})，当U_i重用智能卡时，只需要更新T_{reg}；为避免时钟同步问题(C8)，本协议采用随机数机制代替时间戳机制以保证消息的新鲜性；为实现用户匿名性(C11)，用户的真实身份标识ID_i隐藏在会话可变的伪随机身份标识CID_i中，C11的形式化证明与文献[372]中定理2的证明类似；为实现前向安全性(C12)，协议采用DH密钥交换技术；C4 和 C10 将在节8.3中严格证明。

8.2.2　“Fuzzy-verifier”有效性分析

本节基于大规模真实口令集，来检验我们提出的“模糊验证因子”$A_i = \mathcal{H}_0((\mathcal{H}_0(ID_i) \oplus \mathcal{H}_0(b \| PW_i)) \bmod n_0)$的有效性。

假设$\mathcal{A}$已获取U_i的智能卡，并提取了参数A_i，$\mathcal{A}$ 可以将猜测空间从$\mathcal{D}$降为$\frac{|\mathcal{D}|}{n_0}$。下面我们进一步证实：$|\mathcal{D}|/n_0$实际上足够大。我们采用带频率的大规模真实口令集来很好地模拟口令空间$\mathcal{D}$。$\mathcal{A}$是聪明的，在进行在线猜测时总会优先尝

表 8.2.1 猜测熵(GE)在 $n_0 = 2^8$ 个子口令池的分布情况。通过模糊验证因子A_i将口令空间$\mathcal{D}$切分为n_0个子口令池。

Real-life password distributions	Percentage of password pools with GE $\geq 2^{12}$=($2^{20}/n_0$)
Rockyou_Top1Million	0.00%
Tianya_Top1Million	10.16%
CSDN_Top1Million	10.54%
Dodonew_Top1Million	14.45%
Rockyou_Top2Million	84.77%
Tianya_Top2Million	96.09%
CSDN_Top2Million	97.66%
Dodonew_Top2Million	98.83%
Rockyou_Top$x$Million($x \geq 3$)	99.61%
Tianya_Top$x$Million($x \geq 3$)	100.00%
CSDN_Top$x$Million($x \geq 3$)	100.00%
Dodonew_Top$x$Million($x \geq 3$)	100.00%

试最可能的口令(即概率最大的口令)[5]，$\mathcal{A}$的这种攻击策略可以很好地采用猜测熵(GE)这一安全概念来刻画：

$$G(\mathcal{D}) = \sum_{i=1}^{|\mathcal{D}|} p_i \cdot i$$

不失一般性，$\mathcal{D}$中的每个口令 pw_i假设概率为p_i，且$p_1 \geq p_2 \geq p_3 \geq \cdots$。不难看出，$G(\mathcal{D})$为猜测出正确口令$PW_i$的尝试次数的期望。

本章采用4个公开的数据集: 3200万 Rockyou[198]，3000万Tianya，1600万 Dodonew 和 600 万 CSDN[28]。第一个数据集源自流行游戏网站 Rockyou.com，2009年12月被黑客泄露；后三个数据集源自中国三个知名的网站，2011年12月被黑客泄露。为便于描述，设$n_0 = 2^8$。为提高实验的有效性，我们不采用整个口令数据集来模拟用户口令空间$\mathcal{D}$，而是使用其中最易被猜测的部分，即流行口令部分。原因有两点：(1) $\mathcal{A}$关心成本 vs. 收益，一般不会尝试成本高但收益低的口令，即那些非流行口令；(2)如果口令集中最脆弱的部分都可以保证足够大的猜测熵，那么其它部分自然达到目标。

如表8.2.1所示，当数据集的大小不低于300万时，每个子口令池的猜测熵GE确实不低于2^{12} (n_0=2^8)。这表明，即使攻击者$\mathcal{A}$(以某种方式)获得了参数A_i，仍存在2^{12}个备选口令，这会有效阻碍在线猜测攻击。说明当用户数据集大于300万时，“模糊验证因子”确实有效，可以通过调整n_0来适应较小规模的数据集。从猜测熵GE的角度来看，Dodonew是这4个数据集中最强健的。这得到以下有用的启示：对于那些拥有与Dodonew类似口令生成环境(如服务类型、口令生成策略、语言)的服务，为实现猜测熵 GE$\geq 2^{12}$，用户基数达到200万即可；对于那些拥有与Rockyou类似口令生成环境的网络服务，为实现猜测熵 GE$\geq 2^{12}$，用户基数需达到300万以上。

实验结果表明，A_i可有效划分$\mathcal{A}$的猜测空间，能预留足够多的备选口令，使协议利用“honeywords”阻止$\mathcal{A}$进行在线猜测成为可能，最终将 $\mathcal{A}$ 的猜测优势大大降低(见节8.2.3)。

8.2.3 “Fuzzy-verifier+Honeywords”有效性分析

为提供“口令本地安全更新”，U_i在智能卡中存储“模糊验证因子”A_i=$\mathcal{H}_0((\mathcal{H}_0(ID_i) \oplus \mathcal{H}_0(b\|PW_i))$ mod $n_0)$。这也使得离线口令猜测空间从$|\mathcal{D}|$降为$\frac{|\mathcal{D}|}{n_0}$。上一节证实了本章提出的“模糊验证因子”的有效性。可以看出，若S可以及时检测出智能卡中的参数被恶意提取这一事件(表示为Ext)，并及时暂停服务被腐化的智能卡，仍然可以阻止$\mathcal{A}$以较高优势从$\frac{|\mathcal{D}|}{n_0}$个备选口令中在线猜测出正确的$PW_i$。幸运的是，后文将展示“honeywords”(Honey_List) 和“fuzzy-verifier”(A_i)相结合，可使协议能够在$\mathcal{A}$进行在线猜测10次后，以高达$1-\frac{1}{2^{80}}$的精确度检测出事件Ext。

表 8.2.2 各个真实口令集中Top-x口令所占的比重

Datasets (leaked year)	Language	Total PWs	Unique PWs	Top 1	Top 3	Top 10	Top 10^2	Top 10^3	Top 10^4
Rockyou (2009)	English	32,581,870	14,326,970	0.89%	1.37%	2.05%	4.55%	11.30%	22.31%
Phpbb (2009)	English	255,373	184,341	1.04%	1.80%	2.79%	5.70%	12.89%	27.44%
Singles.org (2010)	English	16,248	12,233	1.36%	2.10%	3.40%	9.19%	26.35%	86.26%
Faithwriters (2009)	English	9,708	8,346	0.55%	1.03%	2.17%	7.76%	24.33%	100.00%
Tianya (2011)	Chinese	30,901,241	12,898,437	3.98%	5.59%	7.43%	11.50%	16.04%	25.78%
Dodonew (2011)	Chinese	16,258,891	10,135,260	1.45%	2.15%	3.28%	5.60%	8.59%	13.62%
Csdn (2011)	Chinese	6,428,277	4,037,605	3.66%	8.15%	10.44%	13.26%	16.54%	23.91%
Sina weibo (2011)	Chinese	4,730,662	2,828,618	3.74%	4.99%	7.17%	10.24%	13.96%	21.49%
Gmail (2014)	hybrid	4,926,650	3,132,028	0.97%	1.43%	2.07%	3.88%	8.65%	17.76%
Mail.ru (2014)	Russian	4,932,688	2,949,616	1.82%	3.06%	4.05%	6.37%	9.94%	17.40%
Yandex.ru (2014)	Russian	1,261,809	717,202	3.10%	4.98%	7.66%	12.26%	17.43%	26.53%
Average above	–	9,300,311	4,657,332	**2.05%**	**3.33%**	**4.78%**	**8.21%**	**15.09%**	28.25%
Uniform distribution	–	1000,000	1000,000	0.0001%	0.0003%	0.0010%	0.01%	0.10%	1.00%

在认证阶段V3步，一旦服务器S发现登录请求中的A_i正确，但k不正确，则有 $1-\frac{1}{n_0}$ 概率确认Ext事件发生。此外，当这种登录事件发生m_0次时，如果指定S撤销U_i，则S有 $1-(\frac{1}{n_0})^{m_0}$ (当$m_0=10$，$n_0=2^8$时，为$1-\frac{1}{2^{80}}$)的概率确信Ext事件发生。m_0和n_0的选择主要受以下条件限制：

R1. $\Pr[\mathsf{Err\_change}]=\frac{1}{n_0}$ 应尽可能小，其中，Err_change表示U_i在更改口令时，误输入错误口令，却令$\mathcal{H}_0((\mathcal{H}_0(ID_i) \oplus \mathcal{H}_0(b \,\|\, PW_i^*))$ mod $n_0)$=A_i，不知不觉将PW_i替换成PW_i^*。

R2. $\Pr[\mathsf{Err\_detect}]=\frac{1}{(n_0)^{m_0}}$ 应尽可能小，其中，Err_detect表示S错误地撤销U_i的智能卡。即S确信Ext事件发生，而实际上，U_i的智能卡并没有被腐化。

R3. $\Pr[\mathsf{Succ\_Ext}]=C' \cdot m_0^{s'}$ 应尽可能小，其中，Succ_Ext 表示 Ext 事件发生，且$\mathcal{A}$通过与S交互，成功获得PW_i。

观察到下面两点非常关键:(1)R1和R2要求m_0和n_0尽可能大，而R3要求m_0和n_0 尽可能小，因此需要平衡；(2)$\Pr[\mathsf{Err\_detect}]$ 随m_0增大，呈指数方式递减，$\Pr[\mathsf{Succ\_Ext}]$ 随m_0增大，呈线性方式递增，这说明可以及时检测Ext事件，

同时，通过设置m_0足够小(如$m_0 \in [3,20]$)，可以限制$\mathcal{A}$的猜测优势为最小值。

本文建议设置$n_0 = 2^8$，$m_0 = 10$。如上所述，设置$n_0 = 2^8$是可接受的。由于误停合法用户的智能卡代价高昂，因此Pr[Err_detect]应可忽略。当$m_0 \geq 10$时，

$$\Pr[\mathsf{Err\_detect}] = \frac{1}{(n_0)^{m_0}} \leq \frac{1}{(2^8)^{10}}$$

这表明Ext事件可以以$1 - \frac{1}{2^{80}}$的概率被检测出。另一方面，当设置$m_0 \leq 10$ 时，

$$\Pr[\mathsf{Succ\_Ext}] = C' \cdot m_0^{s'} \approx \sum_{j=1}^{m_0} p_i \leq \sum_{j=1}^{10} p_i = 7.43\%$$

这里我们以3100万Tianya口令集作为具体示例，其CDF-Zipf参数C'=0.062239，s'=0.155478 (见表2.6)。

表8.2.2总结了11个真实口令分布下$\mathcal{A}$的最佳猜测优势，猜测次数分别为 $m_0 = 1, 3, 10, 10^2$ 和 10^4。因此本文建议设置m_0=10。需要强调的是，m_0和n_0的值可以根据系统不同的安全需求进行调整。

表8.2.2显示，只需极少的在线猜测次数，$\mathcal{A}$就可获得相当大的优势。比如，只需执行100次在线猜测(按照最优猜测顺序)，$\mathcal{A}$的平均优势就达到8.21%。这与传统理论中的最优安全表达函数(即$q_{send}/|\mathcal{D}| + \epsilon$，其中$\mathcal{D}$被假设为均匀分布，$q_{send}$表示$\mathcal{A}$参与的在线猜测次数，见[221, 398, 472])预测的结果大不相同。由于一般情况下假设$\mathcal{D}$=10^6[5]，q_{send}为$10^2 \sim 0^4$(见第三章)，则$q_{send}/|\mathcal{D}|$理论上应约为0.01%~1.00%。然而，$\mathcal{A}$ 的实际优势为8.21%~28.25%，远不是可以忽略的风险。如表8.2.2最后两行所示，“Uniform”模型和实际情况下$\mathcal{A}$的在线猜测优势有2~4个数量级的差异。如果考虑第三章中的定向在线猜测威胁，再加上用户口令会随着认证因子的增加而变弱[473]这一特点，这一差距会更大。

所有这些都表明，由于现实中口令分布严重偏态，传统的最优安全上限表达函数很大程度上低估了$\mathcal{A}$的优势，会传递一种虚假的安全感。非常有必要防止$\mathcal{A}$尝试中等次数的在线猜测，将$\mathcal{A}$的优势限制在可接受范围内。幸运的是，“honeywords + fuzzy-veifier”组合为解决这一问题提供了有效途径。

综上所述，在$\mathcal{A}$触发“智能卡被腐化”的警报之前，只有m_0($m_0 \in$ [3,20])次在线猜测机会，而在传统的认证协议中，这个数字通常为数百甚至上千。并且，服务器仅有$\frac{1}{(n_0)^{m_0}}$的概率发生误报。由于对一个安全的口令认证协议，在线猜测就是$\mathcal{A}$可发起的最佳攻击，因此本章的协议可达到超越传统最优安全上限(即$q_{send}/|\mathcal{D}| + \epsilon$)的安全性，即$\Pr[\mathsf{Succ_Ext}] = C' \cdot m_0^{s'} \approx \sum_{j=1}^{m_0} p_i \leq \sum_{j=1}^{10} p_i$。这有效地缓解了人类有限的记忆力带来的安全性问题：为方便记忆而使用弱口令。

8.2.4 “Fuzzy-verifier+Honeywords”普适性分析

本节显示“Fuzzy-verifier+Honeywords”方法具有普适性，可以很方便地应用到现有的单服务器架构[278]和多服务器环境[186]下的双因子认证方案中。具体来说，在用户智能卡中保存一个模糊验证因子(如A_i)，以实现“口令本地安全更新”(C2)；在认证服务器(或多服务器环境下的控制服务器)的`Honey_List`中存储“honeywords”，以实现抗智能卡丢失攻击(C4)；在登录请求消息中，用户额外向服务器S发送加密后的核心长期密钥(如CAK_i)，如果发生“$\mathcal{A}$利用模糊验证因子作为预言机缩减口令空间”的事件Ext，则S将从加密后的核心长期密钥中解析一个honeyword(如k')，否则S将解析出正确的核心长期密钥。最核心地，“honeywords”使系统能够精确检测出事件Ext，及时阻止$\mathcal{A}$的在线猜测行为(比如通过登录频率限制、锁定帐户等)。

由于本章协议建立在文献[412]中双因子协议之上，而[412] 中协议无法同时实现C2和C4，故本章协议可看作一个典型的利用“honeywords+fuzzy-verifiers”技术升级[412]中协议的例证。我们进一步以Tsai等[400]和Xue等[177]的两个典型方案作为案例，证实本文的方法可以有效融入到任意的单服务器或多服务器环境下的双因子认证协议中。同时，也以Odelu等方案[186]为例，说明该方法适应于三因子方案。由于这些协议的改进方法与前文类似，且受篇幅所限，这里不再赘述。因此，先前很多长期受“C2 vs. C4”两难困扰的协议(如[60, 278, 365, 398])，都可以采用这一方法解决。

小结。理论和实验结果表明，将“honeywords”和“fuzzy-verifier”技术相结合可以有效平衡可用性属性“本地口令更新”(C2)和安全目标“无智能卡丢失问题”(C4)。这解决了Huang等[214]提出的公开问题：“是否存在安全的基于智能卡的口令认证协议，且在口令更新阶段无需与服务器交互？”

8.3 形式化安全分析

我们的协议在ROM模型下，基于CDH难解性假设，可实现可证明安全。

8.3.1 形式化安全模型

本文借鉴 BPR2000 这一单因子口令认证协议ROM安全模型[375]，攻击者与协议参与者之间的交互通过预言机查询来实现，以此精确刻画基于“口令+智能卡”双因子认证协议中攻击者能力。与Xu等[150]和Wu等[398]的工作类似，本文不直接沿用 BPR2000 安全模型，而采用 Bresson 等[430]的改进模型，可以用来定义双因子协议下的特殊安全目标(如抗智能卡丢失攻击)。

通信模型。双因子认证协议是一个由用户和服务器参与的两方协议，用户和服

务器间通过初始化过程事先共享一些私密参数，协议的目标是用户和服务器通过交互的方式完成对彼此身份合法性的确认，并协商一个会话密钥来保护后续通信。用户和服务器之间的通信信道由攻击者完全控制。

参与者。双因子认证协议 $\mathcal{P}$ 中的参与者为用户 $U \in \mathsf{User}$ 和服务器 $S \in \mathsf{Server}$，且 User 和 Server不相交。每个参与者由许多实例组成，被模拟成一组预言机，它们可并发执行协议$\mathcal{P}$。U^i 和 $S^j, i, j \in \mathbb{Z}$ 分别代表用户和服务器的实例，可统一表示为$I \in \mathsf{User} \cup \mathsf{Server}$。

长期密钥。参与者在注册阶段协商长期密钥和公共参数(若需要的话)。在基于公钥的密码协议中，服务器S拥有公私钥对(pub_{key}, pri_{key})；在基于对称密码的密码协议中，服务器S拥有对称密钥sym_{key}。每个用户$U \in \mathsf{User}$具有身份标识ID_U，且从Zipf分布的口令空间$\mathcal{D}$中选取口令PW_U，口令空间大小$|\mathcal{D}|$是独立于系统安全参数的常数。服务器S保存$<ID_U, TPW_U>_{U \in \mathrm{User}}$，其中$TPW_U$是$<ID_U, PW_U>$和$pri_{key}/sym_{key}$的单向映射。另外，服务器$S$将用户相关的一些个人数据和必要的公共参数写入智能卡，并将卡颁发给用户U。

查询。攻击者$\mathcal{A}$的能力通过与协议参与者之间的交互来模拟。$\mathcal{A}$可以执行的查询类型有5种：

- Execute(U^i, S^j): 这一查询向$\mathcal{A}$输出协议正常运行时各参与方间按序收发的消息，模拟$\mathcal{A}$实施被动攻击(监听信道)的能力(对应表7.1中的C-1)；
- Send(I, m): 这一查询向$\mathcal{A}$输出I收到消息m后的响应消息，模拟$\mathcal{A}$实施主动攻击的能力，对应表7.1中的C-1。其中，Send$(U^i,$ Start$)$可引起协议的启动，发起一个会话，$\mathcal{A}$将接收到U^i的登录请求。
- Test(I): 如果实例I已经生成了会话密钥sk，则随机选择一个比特c，若$c = 1$，向$\mathcal{A}$输出sk；若$c = 0$，向$\mathcal{A}$输出一个与sk同样长度的随机串。如果实例I还没有生成会话密钥，则返回$\perp$。这个查询只对“新鲜”的会话执行一次，对非“新鲜”的会话无效。这一查询不是为了模拟$\mathcal{A}$的攻击能力，而是为了刻画sk的语义安全性。
- Reveal(I): 若实例I已生成会话密钥sk，且I和其伙伴未被执行Test查询，则向$\mathcal{A}$输出sk；否则，向$\mathcal{A}$输出$\perp$。这一查询用来刻画“已知密钥攻击”这一安全概念，模拟$\mathcal{A}$对过期会话密钥的利用，对应表7.1中的C-3。
- Corrupt(I, a): 这一查询模拟$\mathcal{A}$腐化用户和服务器的能力：
  - 若$I = U, a = 1$，则向$\mathcal{A}$返回U的口令PW_U；如若$I = U, a = 2$，则向$\mathcal{A}$返回智能卡内保存的秘密参数信息。$\mathcal{A}$只能运行Corrupt$(U, 1)$和Corrupt$(U, 2)$二者中的一个。这对应表7.1中的C-2。

- 若$I=S,a=1$，则向$\mathcal{A}$返回S的私钥pri_{key}(或对称密钥pri_{key})和S的后台数据库中存储的$<ID_U, TPW_U>_{U\in \text{User}}$。这对应表7.1中的C-4。

上面5个预言机查询全面地模拟了表7.1所示的攻击者的能力，可用以精确刻画各种已知的攻击，如仿冒攻击、智能卡丢失攻击、窃取验证项攻击、离线口令猜测攻击、已知密钥攻击、前向安全性和被动侦听。本章主要关注安全性，关于用户匿名性的形式化定义，感兴趣的读者可以参见节6.4.1；关于用户匿名性的形式化证明过程与作者在文献[372]中的过程基本类似，不再赘述。

伙伴关系。ROM模型下广泛接受的做法是通过会话标识sid来定义伙伴关系。称实例U^i和实例S^j为一对伙伴关系，仅当它们满足三个条件：①U^i和S^j均已接受；②U^i的会话标识符sid_U^i和S^j 的会话标识符sid_S^j满足$sid_U^i=sid_S^j$；③U^i的伙伴标识符$pid_U^i=S$，S^j的伙伴标识符$pid_S^j=U$。一般地，如文献[21]所示，可令sid为U^i(或S^j)所有按序收发的协议交互消息的连接。伙伴关系的定义对下面的“新鲜性”的形式化定义是必不可少的。

新鲜性。新鲜性是定义协议安全性和刻画$\mathcal{A}$不应通过平凡攻击获取会话密钥这一事实的关键概念。称实例I是新鲜的，仅当I满足条件：① I 已接受且拥有sk；②I和其伙伴都未被$\mathcal{A}$实施Reveal查询；③当I为用户实例时，I被执行Corrupt查询的次数不超过1；④当I为服务器实例时，I未曾被执行Corrupt查询。

完备性。如果$pid_U^i=S$，$pid_S^j=U$ S^j，并且U^i和S^j均已接受，则有$sk_U^i=sk_S^j$。

认证。认证协议的核心目标就是抵御$\mathcal{A}$仿冒U或S。用$\mathsf{Adv}_{\mathcal{P}}^{auth}(\mathcal{A})$表示$\mathcal{A}$在攻击协议$\mathcal{P}$时，成功仿冒实例$U^i$或$S^j$的概率。换句话说，协议$\mathcal{P}$的一次运行中，$S^j$(或$U^i$)计算出会话密钥$sk$，但“$sk$不是和意定中的实例$U^i$(或$S^j$)协调而来”的概率为$\mathsf{Adv}_{\mathcal{P}}^{auth}(\mathcal{A})$。

现在，需定义什么是安全的双因子认证协议。为简单起见，这里不考虑前向安全性。对任一基于口令的单因子认证(PAKE)协议来说，在线猜测攻击(在每次会话中尝试猜测一个口令)不可避免。因此，针对一个安全的PAKE协议，攻击者可采取的最优攻击策略应是在线猜测[21, 56]。相对而言，这一攻击容易被服务器检测和防御(如失败登录频率限制)。

针对基于(条件)非抗窜扰智能卡假设的双因子认证协议，在线口令猜测攻击也应是攻击者$\mathcal{A}$仿冒参与者的最优策略：(1)如果口令泄漏，但是智能卡仍然安全，由于$\mathcal{A}$不能获得智能卡内存储的高信息熵的秘密参数，所以很难成功；(2) 如果智能卡内参数泄漏，但口令仍然安全，$\mathcal{A}$至少可以在每次与服务器交互中验证一个猜测的口令，即为在线口令猜测攻击。注意，攻击者执行Execute查询的实例不计入在线攻击次数。这引出下面定义：对最多执行q_{send}次在线猜测攻击的任一概率多项式时间(PPT)攻击者$\mathcal{A}$，都存在可忽略函数$\epsilon(\cdot)$，满足：

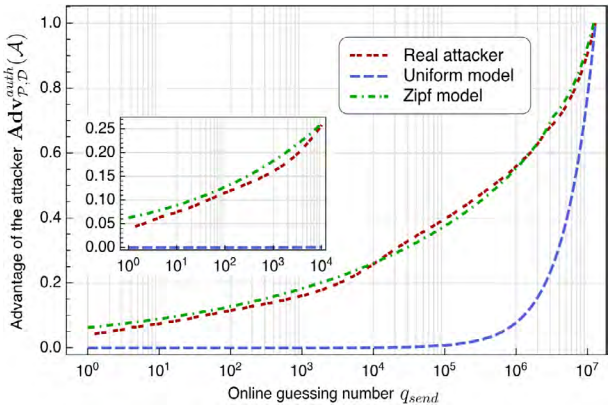


图 8.1 在允许q_{send}次在线猜测的情况下，三种攻击者的优势对比：真实攻击者、基于均匀分布的攻击者[221, 278, 398, 472]和基于CDF-Zipf模型的攻击者。基于CDF-Zipf模型的攻击者优势曲线与真实攻击者几乎重合，表现最优。

$$\mathsf{Adv}^{auth}_{\mathcal{P},\mathcal{D}}(\mathcal{A}) \leq C' \cdot q^{s'}_{send} + \epsilon(\ell)$$

则称双因子认证协议$\mathcal{P}$可实现双向认证。其中，$\mathcal{D}$ 为服从 Zipf 分布的口令空间(见节2.4)，C'和s'为 Zipf 参数，ℓ为系统安全参数。比如，如果目标网站是一个中文社交论坛，可采用Tianya的CDF-Zipf模型参数(见表2.6)，即C'=0.062239，$s' = 0.155478$。这一结果要求Execute查询(即被动攻击)对$\mathcal{A}$是无价值的。

为更好地理解口令均匀(Uniform)分布假设下的攻击者优势表达函数(即$q_{send}/|\mathcal{D}| + \epsilon$，读者可参阅 [221, 278, 398, 472])和 Zipf 分布假设下的攻击者优势表达函数(即$C' \cdot q^{s'}_{send} + \epsilon(\ell)$)的优劣，这里以Tianya 3000万口令为例进行说明。图8.1显示了实际攻击者优势和Uniform模型、Zipf 分布模型下预测的攻击者优势之间的差距。由于$|\mathcal{D}_{\text{tianya}}| = 12.6 * 10^6$，通常$q_{send} < 10^4$，则 Uniform 模型预测的攻击者优势应小于0.1%。然而，攻击者的实际优势接近25.78%。幸运的是，我们的 Zipf 模型预测的攻击者优势接近实际攻击者的优势，大约为26.06%。如图8.1和2.6所示，当查询次数$q_{send} < 10^7$时，Uniform模型总是低估实际攻击者$\mathcal{A}$的优势。相反，Zipf 模型在查询次数小于10^4时，略高估实际攻击者的优势。当$10^4 < q_{send} < 10^6$ 时，我们的 Zipf 模型略低估攻击者的优势。以上分析表明，我们的 Zipf 模型可精确刻画$\mathcal{A}$的在线猜测优势，无论猜测次数q_{send}的多寡。

语义安全性。除了“认证”—证实对方身份，认证协议的另一个主要目标就是在会话结束时生成双方共享的会话密钥，用以加密保护后续通信，因此需要确保会话密钥的机密性。会话密钥的语义安全性保障的是，一个外部攻击者应不能够多项式时间内将真实的会话密钥和与之等长的随机串区分开来。在针对协

议$\mathcal{P}$的一次攻击会话中，$\mathcal{A}$可实施多项式次Execute查询，Reveal查询和Send查询，也可对新鲜实例实施一次Test查询。在攻击会话结束时，$\mathcal{A}$猜测Test查询中比特c的值为c'。若$c'=c$，则称$\mathcal{A}$攻击成功，定义这一猜测成功事件为**Succ**。相应地，$\mathcal{A}$攻破$\mathcal{P}$的语义安全性的优势为

$$\mathsf{Adv}_{\mathcal{P}}^{ake}(\mathcal{A})=2\Pr[\mathbf{Succ}(\mathcal{A})]-1=2\Pr[c'=c]-1$$

其中，概率空间定义在$\mathcal{A}$所有的实例和所有的掷币上。

前面关于“认证”这一概念的讨论也引出以下定义：对最多执行q_{send}次在线猜测攻击的任一概率多项式时间攻击者$\mathcal{A}$，都存在可忽略函数$\epsilon(\cdot)$，满足：

$$\mathsf{Adv}_{\mathcal{P},\mathcal{D}}^{ake}(\mathcal{A})\leq C'\cdot q_{send}^{s'}+\epsilon(\ell)$$

则称双因子认证协议$\mathcal{P}$是语义安全的。其中，上式中的参数含义与“认证”定义中的参数一致。这一结果表明$\mathcal{A}$执行Execute查询(即被动攻击)是无价值的。

8.3.2　安全性证明

首先，我们通过形式化定义注册阶段和攻击者可访问的预言机，来实现对协议的形式化描述。在注册阶段获得安全参数ℓ之前，`Init` 算法首先运行$\mathcal{G}$算法生成阶为q的群$\mathbb{G}$ ，且$|q|=\ell$。然后，`Init` 算法生成群$\mathbb{G}$的生成元g，4个抗碰撞Hash函数$\mathcal{H}_i:\{0,1\}^*\rightarrow\{0,1\}^{l_i}$ $(i=0,1,2,3)$，服务器S 长期的公/私钥对$(x,y=g^x)$。用户U_i从一个服从 Zipf 分布的空间$\mathcal{D}$中随机选取口令PW_i，$\mathcal{D}$的空间大小为$|\mathcal{D}|$。当用户U_i向服务器S 注册时，S 将用户特定的安全参数$\{N_i,\ A_i,\ A_i\oplus a_i\}$ 和其它公共参数写入U_i的智能卡中。其中，N_i和A_i由用户口令PW_i和服务器私钥x共同作用产生。关于预言机Execute，Reveal，Corrupt和Test的形式化定义见图8.3.2。

在给出具体证明之前，先回顾形式化安全证明所依赖的计算假设。

计算 Diffie–Hellman (CDH)假设。令$\mathbb{G}$表示一个阶为q的有限循环群，g表示它的生成元，用乘法表示操作。群$\mathbb{G}$的一个(t,ϵ)-CDH的攻击者是一个运行时间至多为t的PPT算法Δ，满足：

$$\mathsf{Adv}_{g,\mathbb{G}}^{\mathrm{CDH}}(\Delta)=\Pr_{x,y}[\Delta(g^x,g^y)=g^{xy}]\geq\epsilon$$
$$\mathsf{Adv}_{g,\mathbb{G}}^{\mathrm{CDH}}(t)=\max_{\Delta}\{\mathsf{Adv}_{g,\mathbb{G}}^{\mathrm{CDH}}(\Delta)\}$$

其中，概率基于随机选取的x和y，CDH假设表明对任意不太大的t/ϵ，有$\mathsf{Adv}_{g,\mathbb{G}}^{\mathrm{CDH}}(t)\leq\epsilon$。

定理 7 令$\mathbb{G}$为循环群，$\mathcal{D}$为服从 *Zipf* 分布的口令空间。n_0表示“安全性–可用性”平衡参数。令$\mathcal{P}$表示节8.1中提出的认证协议。令$\mathcal{A}$表示时间上限为t的欲破坏语义安全性的*PPT*攻击者，执行至多$q_{send}(\leq m_0=10)$次 *Send* 查询，q_{exe}次 *Execution* 查询，q_h次随机预言机查询。我们有：

$$\begin{aligned}\mathit{Adv}^{\mathrm{ake}}_{\mathcal{P},\mathcal{D}}(\mathcal{A}) &= 2\Pr[\mathrm{Succ}_7]-1+2(\Pr[\mathrm{Succ}_0]-\Pr[\mathrm{Succ}_7])\\ &\leq C'\cdot q_{send}^{s'}+12q_h\mathit{Adv}^{\mathrm{CDH}}_{\mathcal{P}}(t+(q_{send}+q_{exe}\\ &+1)\cdot\tau_e)+\frac{q_h^2+8q_{send}}{2^l}+\frac{(q_{send}+q_{exe})^2}{p},\end{aligned}$$

本章使用*3023.4*万 *Tianya* 口令分布的*CDF-Zipf*模型参数，其中C'=*0.062239*；s'=*0.155478*；$n_0=2^8$*(*见表*2.6)*；τ_e表示群$\mathbb{G}$ 中模幂运算的时间，$l=min\{l_i\},i=0,1,2,3$.

证明 为简单起见，下文的证明不考虑前向安全性。令$\mathcal{A}$表示针对协议$\mathcal{P}$语义安全性的PPT攻击者。主要思想是，利用$\mathcal{A}$构造针对每个底层密码原语(如Hash和CDH难解性)的PPT敌手，如果$\mathcal{A}$可成功破坏语义安全性，那么至少有一个PPT敌手可成功攻破底层的某个密码原语的安全性，产生矛盾。我们定义一系列混合游戏$G_n(n=0,1,\ldots,8)$来证明定理7：从真正的攻击G_0开始，以G_8结束(此时$\mathcal{A}$ 的优势是0)，并限制$\mathcal{A}$在两个连续游戏间的优势差异。详细证明如下。

在每个游戏G_n (n=0,1, . . . , 8)中，定义下面4个事件：

- $\mathbf{Succ_n}$：表示$\mathcal{A}$在Test查询中输出的猜测比特c'等于测试比特c；
- $\mathbf{AskPara_n}$：$\mathcal{A}$ 通过对 $b\|PW_i$ 或 $x\|ID_i\|T_{reg}$ 执行 Hash 查询$\mathcal{H}_0$，计算出(而非猜测)用户U 的核心秘密参数k；
- $\mathbf{AskAuth_n}$：事件$\mathbf{AskPara_n}$发生，并且$\mathcal{A}$对 $ID_i\|ID_S\|Y_1\|C_2\|k\|K$ 执行 Hash 查询$\mathcal{H}_1$或$\mathcal{H}_2$，其中，$K=K_U$或K_S；
- $\mathbf{AskH_n}$：$\mathcal{A}$ 对 $ID_i\|ID_S\|Y_1\|C_2\|k\|K$ 执行 Hash查询$\mathcal{H}_i$ $(i=1,2,3)$，其中，$K=K_U$或K_S。

Game $\mathbf{G_0}$: 该游戏模拟ROM下$\mathcal{A}$对真实协议的攻击，根据定义有

$$\mathrm{Adv}^{\mathrm{ake}}_{\mathcal{P}}(\mathcal{A})=2\Pr[\mathrm{Succ}_0]-1. \tag{8.1}$$

Game $\mathbf{G_1}$: 这一游戏中，协议中所有 Hash 预言机和将在$\mathbf{G_7}$中使用到的$\mathcal{H}_i$(i=0,1,2,3)通过维护一个 Hash 列表$\Lambda_{\mathcal{H}}$(和$\Lambda_{\mathcal{A}}$列表，记录由攻击者直接实施的 Hash 查询)来模拟，如图8.3.2的上部。根据协议描述，我们模拟所有的Send、Execute、Reveal、Corrupt 和Test查询实例，如图8.3.2的下部。

For a hash-query $\mathcal{H}_i(q)$ or $\mathcal{H}_i'(q)$ (with $i \in \{0,1,2,3\}$), such that a record (i,q,r) appears in $\Lambda_{\mathcal{H}}$, the answer is r. Otherwise the answer r is defined according to the following rule:

▶**Rule** $\mathcal{H}^{(1)}$ — Choose a random element $r \in \{0,1\}^{l_1}$.

The record (i,q,r) is added to $\Lambda_{\mathcal{H}}$. If the query is directly asked by the adversary, one adds (i,q,r) to Λ.

We answer to the Send-queries to the client as follows:

— A Send (U^i, Start) -query is processed according to the following rule:

▶**Rule U1**$^{(1)}$ — Choose $\theta \in_R \mathbb{Z}_p^*$ and compute $C_1 = g^\theta$, $Y_1 = y^\theta$, $k = N_i \oplus \mathcal{H}_0(b \| PW_i)$, $M_i = \mathcal{H}_0(Y_1 \| k \| CAK_i \| CID_i), CAK_i = (a_i \| k) \oplus \mathcal{H}_0(Y_1 \| C_1)$, and $CID_i = ID_i \oplus \mathcal{H}_0(C_1 \| Y_1)$.

Then the query is answered with (C_1, M_i, CID_i, CAK_i), and the client instance goes to an expected state.

— If the client instance U^i is in an expected state, a query Send $(U^i, (C_2, C_3))$ is processed by computing the session key and producing an authenticator. We apply the following rules:

▶**Rule U2**$^{(1)}$ — Compute $K_U = C_2{}^\theta$ and $C_3^* = \mathcal{H}_1(ID_i \| ID_S \| Y_1 \| C_2 \| k \| K_U)$.
Reject if the computed C_3^* is not equal to the received C_3. Otherwise moves on.

▶**Rule U3**$^{(1)}$ — Compute the authenticator $C_4 = \mathcal{H}_2(ID_i \| ID_S \| Y_1 \| C_2 \| k \| K_U)$ and the session key $sk_U = \mathcal{H}_3(ID_i \| ID_S \| Y_1 \| C_2 \| k \| K_U)$.

Finally the query is answered with C_4, the client instance accepts and terminates. Our simulation also adds $((C_1, M_i, CID_i, CAK_i), (C_2, C_3), C_4)$ to Λ_{tr}. The variable Λ_{tr} keeps track of the exchanged messages.

We answer to the Send-queries to the server as follows:

— A Send $(S^j, (C_1, M_i, CID_i, CAK_i))$-query is processed according to the following rule:

▶**Rule S1**$^{(1)}$ — Compute $Y = C_1{}^x$ and $ID_i = CID_i \oplus \mathcal{H}_0(C_1 \| Y_1)$, and rejects ID_i is not valid.
Reject if the computed $M_i^* = \mathcal{H}_0(Y_1 \| k \| CID_i \| CAK_i)$ is not equal to the received M_i.
Compute $k = \mathcal{H}_0(x \| ID_i \| T_{reg})$ and $a_i' \| k' = CAK_i \oplus \mathcal{H}_0(Y_1 \| C_1)$.
Reject if $A_i' \neq$ stored A_i;
Suspend U_i if $|Honey\_list| \geq m_0 \wedge (k' \neq k) \wedge (a_i' = a_i)$;
Insert k' into $Honey\_list$ if $|Honey\_list| < m_0$.

▶**Rule S2**$^{(1)}$ — Choose a random exponent $\varphi \in \phi_q^*$;
Compute $K_S = C_1^\varphi$, $C_2 = g^\varphi$, and $C_3 = \mathcal{H}_1(ID_i \| ID_S \| Y_1 \| C_2 \| k \| K_S)$.

Finally, the query is answered with (C_2, C_3) and the server instance goes to an expected state.

— If the server instance S^j is in an expected state, a query Send (S^j, C_4) is processed according to the following rules:

▶**Rule S3**$^{(1)}$ — Compute $C_4^* = \mathcal{H}_2(ID_i \| ID_S \| Y_1 \| C_2 \| k \| K_S)$, and check whether $C_4^* = C_4$.
If the equality does not hold, the server instance terminates without acceptance.
If equality holds, the server instance accepts and goes on, applying the following rule:

▶**Rule S4**$^{(1)}$ — Compute the session key $sk_S = \mathcal{H}_3(ID_i \| ID_S \| Y_1 \| C_2 \| k \| K_S)$.

Finally, the server instance terminates.

An Execute (U^i, S^j)-query is processed using a successive simulations of the Send-queries: $(U, (C_1, M_i, CID_i)) \leftarrow \text{Send}(U^i, \text{Start})$, $(C_2, C_3) \leftarrow \text{Send}(S^j, (C_1, M_i, CID_i, CAK_i))$ and $C_4 \leftarrow \text{Send}(U^i, (C_2, C_3))$, and outputting the transcript $((C_1, M_i, CID_i, CAK_i), (C_2, C_3), C_4)$.

A Corrupt(U, a)-query returns the password PW_i if a=1; returns $\{N_i, A_i, b\}$ if a=2. (Since we do not consider forward secrecy in this proof, no Corrupt(S, 1)-query occurs.

A Reveal(I)-query returns the session key (sk_U or sk_S) computed by the instance I (if the latter has accepted).

A Test(I)-query first gets sk from Reveal(I), and flips a coin c. If c = 1, we return the value of the session key sk, otherwise we return a random value drawn from $\{0,1\}^{l_3}$.

图 8.3.2 本章协议相关的ROM查询的模拟。

显然，本游戏中$\mathcal{A}$的优势与$\mathcal{A}$攻击真实协议是不可区分的，故

$$\left|\Pr[\text{Succ}_1] - \Pr[\text{Succ}_0]\right| = 0 \tag{8.2}$$

Game $\mathbf{G_2}$: 本游戏去掉$\mathbf{G_1}$中一些以可忽略概率发生的碰撞：

- 协议交互消息$((C_1, M_i, CAK_i, CID_i), (C_2, C_3), C_4)$ 的碰撞。由于$\mathcal{A}$只能充当协议通信的一方，协议任意一次运行中至少有一个实体是诚实的(为合法的U或S)，故$C_1 = g^u \bmod p$和$C_2 = g^v \bmod p$中至少有一个是由诚实实体按原协议计算而来，故C_1和C_2中至少有一个服从随机分布；
- Hash 预言机输出的碰撞。

这些碰撞的概率根据生日悖论原理可得:

$$\Big|\Pr[\mathrm{Succ}_2]-\Pr[\mathrm{Succ}_1]\Big| \leq \frac{(q_{send}+q_{exe})^2}{2p}+\frac{q_h^2}{2^{l+1}} \tag{8.3}$$

其中 $l=min\{l_i\}, i=0,1,2,3.$

Game $\mathbf{G_3}$: 本游戏中，如果攻击者幸运地猜测出正确的认证元C_3和C_4(即未经查询对应的$\mathcal{H}_1$或$\mathcal{H}_2$)，则终止游戏。我们通过检查列表$\Lambda_{\mathcal{A}}$来实现这一策略：如果记录$(1, ID_i\|ID_S\|Y_1\|\ C_2\|k\ \|K_U, C_3) \notin \Lambda_{\mathcal{A}}$，但用户$U_i$接受，则意味着$\mathcal{A}$幸运地猜测出认证元$C_3$，则终止协议。类似地，如果$(2, ID_i\|ID_S\|Y_1\|\ C_2\|k\ \|K_U, C_4) \notin \Lambda_{\mathcal{A}}$，但用户$S$接受，则意味着$\mathcal{A}$幸运地猜测出认证元$C_4$，则终止协议的执行。由于Hash输出碰撞的可能性在$\mathbf{G_2}$中已排除，因此除非合法认证元(由$\mathcal{A}$猜测而来)被拒绝, $\mathbf{G_3}$和$\mathbf{G_2}$是不可区分的，可得

$$\Big|\Pr[\mathrm{Succ}_3]-\Pr[\mathrm{Succ}_2]\Big| \leq \frac{q_{send}}{2^l} \tag{8.4}$$

Game $\mathbf{G_4}$: 本游戏中，如果攻击者幸运地猜测出安全参数k(即未经相应查询)，且成功地仿冒用户或服务器，则终止游戏。我们通过修改参与者处理查询的方式来达到此目的，规则如下：

1. **Rule U3$^{(4)}$** – 在$\Lambda_{\mathcal{A}}$列表中查询记录$(0, *\|ID_i\|*, k)$，若不存在，则终止游戏；否则，计算$C_4=\mathcal{H}_2(ID_i\|ID_S\|Y_1\|\ C_2\|k\ \|K_U)$和会话密钥$sk_U=\mathcal{H}_3(ID_i\|ID_S\|\ Y_1\|C_2\|k\|K_U)$。
2. **Rule S3$^{(4)}$** – 计算$C_4^*=\mathcal{H}_2(ID_i\|ID_S\|Y_1\|C_2\ \|k\|K_S)$，并验证$C_4^*$是否等于接收到的$C_4$，若相等，则服务器在$\Lambda_{\mathcal{A}}$列表中查询$(0, *, P_i)$，在$\Lambda_{\Psi}$列表中查询$((C_1, M_i, CAK_i, CID_i), (C_2,\ C_3), C_4)$。如果记录不存在，则终止游戏。

由于之前的会话未出现C_1和C_2，并且它们中至少有一个是诚实实体产生的，它们中至少有一个是随机的。此外，我们在$\mathbf{G_2}$中已排除Hash输出碰撞的可能性, 故除非k由$\mathcal{A}$猜测而来, 否则$\mathbf{G_4}$和$\mathbf{G_3}$是不可区分的, 可得

$$\Big|\Pr[\mathrm{Succ}_4]-\Pr[\mathrm{Succ}_3]\Big| \leq \frac{q_{send}}{2^{l_0}} \tag{8.5}$$

Game $\mathbf{G_5}$: 本游戏中，如果攻击者计算出正确的安全参数k，且成功仿冒用户或服务器，则终止游戏。我们通过修改参与者处理查询的方式来达到此目的，规则如下：

- **Rule U2$^{(5)}$** – 在$\Lambda_{\mathcal{A}}$列表中查询记录$(0, *\|ID_i\|*, k)$，若存在，则终止游戏；否则，计算$K_U=(C_2)^u \bmod p$， $C_3^*=\mathcal{H}_1(ID_i\|ID_S\|Y_1\ \|C_2\|k\|K_U)$，并验证$C_3^*$ 是否等于接收到的C_3。若不相等，则终止协议执行，否则继续。

- **Rule S3$^{(5)}$** – 计算$C_4^* = \mathcal{H}_2(ID_i \| ID_S \| Y_1 \| \ C_2 \| k \| K_S)$，并验证$C_4^*$是否等于接收到的$C_4$。若相等，则服务器在$\Lambda_{\mathcal{A}}$列表中查询$(0, *, P_i)$，在$\Lambda_{\Psi}$列表中查询$((C_1, M_i, CAK_i, CID_i), (C_2, C_3), C_4)$。如果记录存在，则终止游戏。

在事件**AskPara$_5$**不发生的情况下，**G$_5$**和**G$_4$**是不可区分的，可得：

$$\left|\Pr[\text{Succ}_5] - \Pr[\text{Succ}_4]\right| \leq \Pr[\text{AskPara}_5] \tag{8.6}$$

为得到$\Pr[\text{AskPara}_5]$的上限，在随后的游戏中都需假设$\mathcal{A}$正确计算出安全参数k。

Game G$_6$: 本游戏中，如果攻击者计算出正确的C_3或C_4 (即查询对应的$\mathcal{H}_1$或$\mathcal{H}_2$)，且成功仿冒用户或服务器，则终止游戏。我们通过修改参与者处理查询的方式来达到此目的，规则如下：

- **Rule U3$^{(6)}$** – 验证$(1, ID_i \| ID_S \| Y_1 \| C_2 \| k \| *, \ C_3) \in \Lambda_{\mathcal{A}}$是否成立。若不成立，则终止游戏；否则，用户计算$C_4$和$sk_U$。
- **Rule S3$^{(6)}$** – 计算$C_4^* = \mathcal{H}_2(ID_i \| ID_S \| Y_1 \| \ C_2 \| k \| K_S)$，并验证$C_4^*$是否等于接收到的$C_4$。若相等，则服务器在$\Lambda_{\mathcal{A}}$列表中查询记录$(0, *, P_i)$或$(2, ID_i \| ID_S \| Y_1 \| C_2 \| k \| *, C_4)$，在$\Lambda_{\Psi}$列表中查询$((C_1, CAK_i, CID_i), (C_2, C_3), C_4)$。如果记录存在，则终止游戏。

在事件**AskAuth$_6$**不发生的情况下，**G$_6$**和**G$_5$**是不可区分的，可得：

$$\left|\Pr[\text{Succ}_6] - \Pr[\text{Succ}_5]\right| \leq \Pr[\text{AskAuth}_6] \tag{8.7}$$

$$\left|\Pr[\text{AskPara}_6] - \Pr[\text{AskPara}_5]\right| \leq \Pr[\text{AskAuth}_6] \tag{8.8}$$

Game G$_7$: 本游戏中，我们使用私有的 Hash 函数$\mathcal{H}_i'(i = 1, 2, 3)$来替换$\mathcal{H}_i$：

$$C_3 = \mathcal{H}_1'(ID_i \| ID_S \| C_1 \| C_2);$$
$$C_4 = \mathcal{H}_2'(ID_i \| ID_S \| C_1 \| C_2);$$
$$sk_U = sk_S = \mathcal{H}_3'(ID_i \| ID_S \| C_1 \| C_2)$$

这样C_3, C_4, sk_U, sk_S与k, K_U和K_S完全独立。在事件**AskH$_7$**不发生的情况下，**G$_7$**和**G$_6$**是不可区分的，故有：

$$\left|\Pr[\text{Succ}_7] - \Pr[\text{Succ}_6]\right| \leq \Pr[\text{AskH}_7] \tag{8.9}$$

$$\left|\Pr[\text{AskPara}_7] - \Pr[\text{AskPara}_6]\right| \leq \Pr[\text{AskH}_7] \tag{8.10}$$

$$\left|\Pr[\text{AskAuth}_7] - \Pr[\text{AskAuth}_6]\right| \leq \Pr[\text{AskH}_7] \tag{8.11}$$

引理 2 在游戏**G$_7$**中，事件**Succ$_7$**和**AskPara$_7$**出现的概率上界如下：

$$\left|\Pr[\text{Succ}_7]\right| = \frac{1}{2} \quad \left|\Pr[\text{AskPara}_7]\right| \leq C' \cdot q_{send}^{s'} + \frac{q_{send}}{2^{l_0}} \tag{8.12}$$

证明. 在游戏$\mathbf{G_7}$中，会话密钥是通过$\mathcal{A}$未知的私有 Hash 函数计算而来，因此$\Pr[\text{Succ}_7] = \frac{1}{2}$。

令$R(U)$表示用户实例接收到的消息(C_2, C_3)的集合，令$R(S)$表示服务器实例接收到的消息C_4的集合。虽然$\mathbf{G_4}$中已经去除了$\mathcal{A}$幸运地猜测出安全参数k的情况，但$\mathcal{A}$可以通过Corrupt$(I = U^i,\ 1)$-查询或Corrupt$(I = U^i,\ 2)$-查询的辅助来计算出核心安全参数k。以上两种不同情况下的概率分别表示为$\Pr[\text{AskPara}_7\text{WithCorr}_1]$ 和$\Pr[\text{AskPara}_7\text{WithCorr}_2]$。如第二章所述，用户口令服从 Zipf 分布而不是传统假设下的“均匀分布”(见文献[21, 22, 278, 472])，依据 Zipf 分布刻画的攻击者优势比均匀分布假设下刻画的攻击者优势精确$2 \sim 4$个量级。我们在$\mathbf{G_2}$中已排除了协议消息碰撞的可能性，从信息论的角度，可得

$$\begin{aligned}\left|\Pr[\text{AskPara}_7\text{WithCorr}_1]\right| = &\Pr_{pw}[\exists C_3 \in R(U), (1, ID_i \| ID_S \| Y_1 \| C_2 \| k \| *, C_3) \in \Lambda_{\mathcal{A}}] \\ &+ \Pr_{pw}[\exists C_4 \in R(S), (0, *, P_i) \in \Lambda_{\mathcal{A}}] \leq C' \cdot q_{send}^{s'}\end{aligned} \tag{8.13}$$

$$\left|\Pr[\text{AskPara}_7\text{WithCorr}_2]\right| \leq \frac{q_{send}}{2^{l_0}} \tag{8.14}$$

Game $\mathbf{G_8}$: 本游戏中，任给一个随机CDH实例(A, B)，我们利用 Diffie-Hellman 问题的自归约特性[474]来模拟运行。需要注意的是，由于不需要借助K_U和K_S的值来计算认证元或会话密钥，因此，我们无需获知θ 和φ 的值：

1. **Rule U1$^{(8)}$** – 取随机数$\alpha \in Z_p^*$，计算$C_1 = A^\alpha \bmod p$，$Y_1 = y^\alpha \bmod p$，$k = \mathcal{H}_0(x \| ID_i \| T_{reg}) = N_i \oplus \mathcal{H}_0(b \| PW_i)$，$CID_i = ID_i \oplus \mathcal{H}_0(C_1 \| Y_1)$，$CAK_i = (a_i \| k) \oplus \mathcal{H}_0(Y_1 \| C_1)$和$M_i = \mathcal{H}_0(Y_1 \| k \| CID_i)$。同时，在$\Lambda_A$列表中增加$(C_1, \alpha)$。
2. **Rule S2$^{(8)}$** – 取随机数$\beta \in Z_p^*$，计算$C_2 = B^\beta$和$C_3 = \mathcal{H}_1'(ID_i \| ID_S \| C_1 \| C_2)$。同时，在$\Lambda_B$列表中增加$(C_2, \beta)$。

$$\Pr[\text{AskH}_7] = \Pr[\text{AskH}_8] \tag{8.15}$$

AskH$_8$意味着攻击者$\mathcal{A}$已对$(ID_i \| ID_S \| \ Y_1 \| C_2 \| * \| CDH(C_1, C_2))$查询了$\mathcal{H}_i (i = 1, 2, 3)$。通过随机在$\Lambda_{\mathcal{A}}$列表中选取，可以有$\frac{1}{q_h}$的概率得到三元组$(C_1, C_2, \text{CDH}(C_1, C_2))$对应的Diffie-Hellman秘密参数。具体来说，在列表Λ_A和Λ_B 中很容易找到α和β的值，因此有$C_1 = A^\alpha$，$C_2 = B^\beta$:

$$\text{CDH}(C_1, C_2) = \text{CDH}(A^\alpha, B^\beta) = \text{CDH}(A, B)^{\alpha\beta}$$

可得:

$$\left|\Pr[\text{AskH}_8]\right| \leq q_h \text{Adv}_{\mathcal{P}}^{\text{CDH}}(t') \tag{8.16}$$

其中$t' \leq t + (q_{send} + q_{exe} + 1) \cdot \tau_e$.

综上所述，首先，通过等式(1)-(5)可得：

$$\left|\Pr[\text{Succ}_4] - \Pr[\text{Succ}_0]\right| \leq \frac{(q_{send}+q_{exe})^2}{2p} + \frac{q_h^2}{2^{l+1}} + \frac{q_{send}}{2^l}.$$

其次，通过等式(6)-(9)可得：

$$\left|\Pr[\text{Succ}_7] - \Pr[\text{Succ}_4]\right| \leq \Pr[\text{AskPara}_5] + \Pr[\text{AskAuth}_6] + \Pr[\text{AskH}_7].$$

根据定义有

$$\Pr[\text{AskAuth}_6] \leq \Pr[\text{AskH}_6].$$

最后，通过等式8.9∼8.16可得：

$$\begin{aligned}\text{Adv}_{\mathcal{P}}^{\text{ake}}(\mathcal{A}) &= 2\Pr[\text{Succ}_7] - 1 + 2(\Pr[\text{Succ}_0] - \Pr[\text{Succ}_7]) \\ &\leq C' \cdot q_{send}^{s'} + 12 q_h \text{Adv}_{\mathcal{P}}^{\text{CDH}}(t') + \frac{q_h^2 + 6q_{send}}{2^l} + \frac{(q_{send}+q_{exe})^2}{p},\end{aligned}$$

我们采用表2.6中Tianya的CDF-Zipf模型，其中，$C' = 0.062239$和$s' = 0.155478$；$n_0 = 2^8$；$t' \leq t + (q_{send} + q_{exe} + 1) \cdot \tau_e$和$l = min\{l_i\}, i = 0,1,2,3$. □

定理 8 $\mathbb{G}$，$\mathcal{D}$，n_0和$\mathcal{P}$与定理7中描述的一致。$\mathcal{A}$ 表示时间t内破坏双向认证的攻击者，且$Send$查询次数小于$q_{send}(\leq m_0 = 10)$，$Execution$查询小于q_{exe}，随机预言机查询小于q_h，因此有

$$\begin{aligned}Adv_{\mathcal{P},\mathcal{D}}^{\text{auth}}(\mathcal{A}) \leq C' \cdot q_{send}^{s'} + \frac{q_h^2 + 8q_{send}}{2^{l+1}} + \frac{(q_{send}+q_{exe})^2}{2p} \\ + 5q_h Adv_{\mathcal{P}}^{\text{CDH}}(t + (q_{send} + q_{exe} + 1) \cdot \tau_e),\end{aligned}$$

其中，τ_e 表示$\mathbb{G}$和$l = min\{l_i\}, i = 0,1,2,3$中模幂运算时间。

证明. 证明过程与定理 7 类似。

首先，我们再定义一个事件：

– $\mathbf{Auth_n}$：$\mathcal{A}$在游戏$\mathbf{G_n}, n = 0,1,\ldots,7$中成功猜测出认证元$C_3$ 或C_4。

因此，我们定义

$$\text{Adv}_{\mathcal{P}}^{\text{auth}}(\mathcal{A}) = \Pr[\text{Auth}_0]$$

接着，采用与上一节完全相同的一系列仿真游戏，扩展式8.1∼8.7可得：

$$\begin{aligned}
&\left|\Pr[\text{Auth}_1]-\Pr[\text{Auth}_0]\right| = 0\\
&\left|\Pr[\text{Auth}_2]-\Pr[\text{Auth}_1]\right| \leq \frac{(q_{send}+q_{exe})^2}{2p}+\frac{q_h^2}{2^{l+1}}\\
&\left|\Pr[\text{Auth}_3]-\Pr[\text{Auth}_2]\right| \leq \frac{q_{send}}{2^l}\\
&\left|\Pr[\text{Auth}_4]-\Pr[\text{Auth}_3]\right| \leq \frac{q_{send}}{2^l}\\
&\left|\Pr[\text{Auth}_5]-\Pr[\text{Auth}_4]\right| \leq \Pr[\text{AskPara}_5]\\
&\left|\Pr[\text{Auth}_6]-\Pr[\text{Auth}_5]\right| \leq \Pr[\text{AskAuth}_6] \leq 2\Pr[\text{AskH}_7]\\
&\Pr[\text{Auth}_7] = \Pr[\text{Auth}_6] = 0
\end{aligned}$$

因此有

$$\begin{aligned}
\text{Adv}_{\mathcal{P}}^{\text{auth}}(\mathcal{A}) \leq C' \cdot q_{send}^{s'} + 5q_h \text{Adv}_{\mathcal{P}}^{\text{CDH}}(t+(q_{send}+q_{exe}\\
+1)\cdot\tau_e) + \frac{q_h^2+6q_{send}}{2^{l+1}} + \frac{(q_{send}+q_{exe})^2}{2p}. \ \square
\end{aligned}$$

8.4 性能评估

本节从效率和节7.7中12项评价指标的实现情况这两个方面，对比我们的协议与相关协议[150, 221, 383, 384, 386, 398, 412, 422]。效率评估结果如表8.4.2所示，评价指标的满足情况见表8.4.3。

不失一般性，假设安全参数n_0为32比特，用户名ID_i、口令PW_i、随机数、时间戳、安全单向Hash 函数的输出值都为128位，y和g为1024位。令T_H，T_E，T_S 和 T_C 分别表示Hash 操作、模幂运算、对称加密、Chebysev多项式运算的时间复杂度，其它像异或操作等轻量级操作在此省略。

表 8.4.1 典型密码原语的运算时间

Experimental platform	Exp. T_E ($\|p\|$=1024)	Chebysev T_C ($\|p\|$=1024)	Symm. T_S (AES-128)	Hash T_H (SHA-1)
Philips HiPerSmart 36 MHz	140.0 ms	439.5 ms	4.972 ms	12.261 ms
Intel(R) T5870 2.00 GHz	10.257 ms	32.200 ms	2.012 µs	2.580 µs
Intel(R) i7-4790 3.60GHz	2.871 ms	10.257 ms	0.086 µs	0.598 µs

为更直观的展现我们协议的计算开销，表8.4.1列出了相关密码原语操作在不同计算平台上的运行时间。文献[380]中给出了相关操作在Philips HiPerSmart$^{\text{TM}}$智能卡上运行时间，该种智能卡中配备32 位RISC mips处理器，提供最大36 MHz时钟速度和2 KB指令缓存，256 KB 闪存和16 KB RAM。我们采用此智能卡模拟用户设备。此外，我们使用普通笔记本电脑模拟服务器端计算开销。注意，我们的实验和文献[380]一样，都采用了标准密码库MIRACL[339]。

表 8.4.2 相关认证协议的性能比较

	Protocol rounds[a]	Computation overhead		Communication cost		Storage cost
		User side	Server side	User side	Server side	
Xu et al. (2009)[150]	2	$3T_E$+$5T_H$≈ 481.305 ms	$3T_E$+$4T_H$≈ 8.615 ms	1408 bits	1408 bits	3200 bits
Wang et al. (2012)[412]	3	$3T_E$+$7T_H$≈ 505.827 ms	$3T_E$+$5T_H$≈ 8.616 ms	1408 bits	1152 bits	3456 bits
Wu et al. (2012)[398]	3	$4T_E$+$5T_H$≈ 621.305 ms	$3T_E$+$5T_H$≈ 8.616 ms	2304 bits	1152 bits	3456 bits
Li et al. (2013)[422]	2	$4T_E$+$4T_H$≈ 609.044 ms	$3T_E$+$3T_H$≈ 8.615 ms	1408 bits	1408 bits	4096 bits
Byun (2015)[221]	2	$5T_E$+T_S+$5T_H$≈766.277 ms	$5T_E$+T_S+$3T_H$≈ 14.357 ms	2176 bits	1152 bits	2176 bits
Jiang et al. (2015)[384]	2	$4T_E$+$4T_H$≈ 609.044 ms	$2T_E$+$4T_H$≈ 5.744 ms	2304 bits	1152 bits	3328 bits
Truong et al. (2015)[383]	3	T_C+$7T_H$≈ 525.327 ms	$3T_C$+$7T_H$≈ 30.775 ms	640 bits	1152 bits	384 bits
Islam et al. (2016)[386]	2	$3T_E$+$3T_H$≈ 456.783 ms	$2T_E$+$3T_H$≈ 5.744 ms	1408 bits	1408 bits	2176 bits
Our scheme	3	$3T_E$+$9T_H$≈530.349 ms	$3T_E$+$7T_H$≈ 8.617 ms	1536 bits	1152 bits	3616 bits

[a] All the schemes with 2 rounds are either prone to clock synchronization issue or reply attack.

表 8.4.3 采用节7.7中评价标准对相关认证协议进行评估

	The proposed twelve evaluation criteria in Sec. 7.7											
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Xu et al. (2009)[150]	✓	×	×	×[a]	×	×	✓	×	×	✓	×	✓
Wang et al. (2012)[412]	✓	✓	✓	×[b]	✓	×	✓	✓	✓	✓	✓	✓
Wu et al. (2012)[398]	✓	✓	✓	×	✓	✓	✓	✓	✓	×	✓	✓
Li et al. (2013)[422]	✓	✓	×	×[b]	✓	×	✓	×	✓	✓	×	✓
Byun (2015)[221]	×	×	✓	✓	✓	✓	✓	✓	×	✓	✓	✓
Jiang et al. (2015)[384]	✓	✓	×	✓	✓	×	✓	×	✓	✓	×	×
Truong et al. (2015)[383]	✓	✓	×	×[b]	✓	×	✓	✓	✓	✓	×	✓
Islam et al. (2016)[386]	✓	✓	✓	×[b]	✓	✓	✓	✓	✓	✓	✓	✓
Our scheme	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

[a] This scheme is vulnerable to user impersonation attack as shown in [422];
[b] All the schemes in [383, 386, 412, 422] are subject to the security-usability conflict—an offline password guessing attack will succeed once the smart card security is breached.

如表8.4.2和8.4.3所示，我们的协议实现了所有12个标准，同时保持了合理的效率。由于“安全性 vs. 可用性”这一内在冲突，其它仅采用传统密码学工具(即未引入“Honeywords+fuzzy-verifier”这一系统安全技术)的协议至少有一个评价指标未实现。

8.5　本章小结

本章研究了第七章中遗留的两个重要问题：(1)如何利用大规模真实口令数据集来评估节7.3.2所提“模糊验证因子”方法的有效性？(2)是否可以设计一个可实现节7.7所提12个指标的协议？为解决这两个问题，本章利用了前文中的口令Zipf分布理论和匿名性的复杂度理论等口令安全理论，特别是将第四章中的Honeywords 技术引入双因子安全协议设计中。通过结合“honeywords”和“fuzzy-verifier”，本文提出的协议可及时检测攻击者通过利用受害者智能卡的参数发起的在线口令猜测攻击，有效降低攻击者的猜测优势，较好地解决了文献[397]中遗留的“安全性 vs. 可用性”问题。值得一提的是，我们采用 11 个大规模口令集，包含1.026亿真实用户口令，覆盖多种流行服务和多样化的用户群体，来评估“honeywords+fuzzy-verifier”技术的有效性。

第九章　结束语和展望

9.1　论文研究成果

身份认证是保障信息系统安全的第一道防线，在很多情况下甚至是唯一的一道防线。在众多身份认证方法中，基于口令的认证由于简单易用、成本低廉、口令容易更新等特点，获得了广泛的应用。虽然口令安全研究有几十年的历史，近年来也涌现出了一批成果。但口令研究整体尚处于起步阶段，学术界虽然针对一些现实口令安全问题提出了解决方案，但往往缺乏可行性和前瞻性，指导性不强，口令学术界没有能够引领口令工业界。本文认为产生这一现状的重要原因是，长期以来口令安全研究被视为实验科学，忽略现实问题背后的深层次理论问题，多局限于实验实证或对现象的简单统计，极少尝试采用数学理论模型或形式化方法，缺乏从理论层面进行规律性、原则性和系统性的探索。

鉴于此，本文以用户脆弱行为为切入点，采用数据驱动的方法，基于可证明安全理论、统计学习理论和自然语言处理技术等多学科交叉知识，研究了 6 个亟待解决的关键理论问题，取得了如下七项研究成果：

1. 发现人类生成的口令服从 Zipf 分布，结束了长期假设口令服从均匀分布的历史。学者们曾怀疑“口令服从均匀分布”这一假设的现实性，但又立即陷入一个新的难题：如果口令不服从均匀分布，那到底服从什么分布？因此，数以百计的口令文献情愿或不情愿地采用了这一假设(最典型的是成百上千的PAKE协议)。我们的PDF-Zipf模型和CDF-Zipf口令分布模型使摆脱这一不现实假设成为可能。CDF-Zipf模型非常精确：在14 个大规模真实口令集下，其理论分布与实际分布的最大累积分布误差(即KS统计量D)仅为0.006170∼0.045874(均值0.018457)。这一成果将对整个口令研究产生基础性影响：本文显示口令的 Zipf 分布理论在三个相关领域(密码协议、生成策略和强度评价)具有重要应用，目前该成果也已经被国内外学者应用于另外三个领域(口令加密算法[223]、口令Hash 函数[192] 和口令集生成算法[92])，并进入普度大学等欧美高校课程。
2. 提出了一个定向在线口令猜测理论攻击框架，为科学评估攻击者的定向猜测优势奠定了理论基础。该框架包含7个理论攻击模型以刻画7种不同能力的现实攻击者。大规模真实数据测试结果显示，在允许猜测100次的情况下，如果仅知道目标用户的一些常见个人信息，成功率可达20%；如果知道用户在其它网站曾经泄露的一个口令，成功率可达73%。这一结果大大超出了此前人们的预期，被200多个国际媒体报道，引起了国际社会的广

泛关注。本研究确凿地显示了定向在线口令猜测威胁的严重性，使这一威胁得到正视：美国国家身份认证标准NIST SP800-63-3根据这一研究修订了在线猜测相关部分，并征询了我们的防御措施[①]。

3. 提出了一个 Honeywords 攻击和设计理论体系，为设计最优的 Honeywords生成方法奠定了理论基础。当前口令文件泄漏事件频发生，并且没有很好的检测手段，而滞后发现会产生巨大的负面影响。Honeywords 技术是实现主动、及时的泄露检测的有效手段，本文为如何攻击、设计和评估honeywords生成方法提供了坚实的理论基础。
4. 设计了一个基于口令重用行为的口令评价算法，实现了对人类口令生成行为的精确刻画。几乎每一个主流的网站都会执行一个口令强度评价算法，但目前的算法都假设用户是从无到有构造口令，未能有效刻画用户的口令生成行为。本文基于对494个用户的调研，发现大多数用户在注册新帐户时会重用现有口令，因此设计了我们的fuzzPSM。大规模实验显示，fuzzPSM优于当前主流的口令评价算法，尤其适于评测弱口令。
5. 提出了一个口令认证协议中实现匿名性的基本原则，为设计匿名口令认证协议奠定理论基础。由于智能卡、移动设备和传感器节点是资源受限设备，协议设计者们不断尝试仅基于对称密码技术来设计匿名口令认证协议。我们基于Halevi–Krawczyk (1999)和Impagliazzo–Rudich (1989)这两个经典结果，成功证明这从根本上是行不通的：在广泛采用的非抗窜扰智能卡假设下，公钥密码技术是口令认证协议实现匿名性的基本组件。
6. 提出了一个新的基于口令双因子认证协议评价指标体系，为全面准确地评估双因子认证协议提供准绳。我们指出现有评价标准都存在严重缺陷(如冗余和歧义)，不适于应用；此外，现有研究往往仅关注评价标准而忽略了与之相辅相成的安全模型，在定义具体评价指标时，未明确安全模型，使指标的含义不明确。为此，我们明确地定义了一个当前最为严格(但现实)的攻击者模型，并在综合现有各评价标准优点的基础上，提出一个含有12项指标的新标准，这二者共同构成一个新的协议评价指标体系。最后，通过评估 67 个典型双因子协议，显示了该评价指标体系的有效性。
7. 设计了一个 ROM 下可证明安全的基于口令双因子认证协议，证实了实现所有 12 项评价指标的可能性。由于“安全性 vs. 可用性”这一内在冲突，现有仅采用传统密码学工具的协议均顾此失彼，至少有一个评价指标无法实现。我们将“honeywords”与“fuzzy-verifier”相结合，成功解决了前述内在冲突，实现了一个在当前最严格的攻击者模型下，可满足所有12项指标的协议。最后，在ROM模型下给出了协议的形式化安全证明。

[①] http://wangdingg.weebly.com/uploads/2/0/3/6/20366987/nist800-63-3revised.png

9.2　未来工作展望

虽然口令存在众多的安全性和可用性问题，但由于口令使用简单、成本低廉、容易修改，克服了基于物理硬件的认证技术和生物特征认证技术的缺陷，学术界逐渐认识到：在可预见的未来，口令仍无可替代。结合本文的工作，我们认为下面几个问题还值得深入研究和探索：

- 口令Zipf分布形成的动力学机制。人类生成口令的行为是一种非常特殊的行为，受用户自身因素的影响，如语言、文化习惯、姓别、学历等，也受外界因素的制约，如系统的服务类型和口令生成策略等。虽然单个用户看似独立地生成口令，但在自身和外界多种综合因素影响下，人类生成的口令奇妙地服从一个简单优美的规律：Zipf 分布！口令的这些影响因素与其它服从Zipf 分布的现象(如人类语言、收入分布和战争频率)的影响因素差异巨大。如何建模这些影响因素，探寻口令Zipf分布的形成机制是一项十分具有挑战性和重要科学价值的工作。
- 特定群体、应用环境下用户口令行为的理解。现有研究在理解不同语言、地域用户的口令方面取得了一系列重要发现，但在理解特定群体、特定应用环境下的用户口令方面鲜有公开成果。比如，男性口令和女性口令在漫步或定向猜测攻击下谁更强？黑客的口令与普通用户的口令有何不同？移动设备上的口令与传统PC上口令有何差异？
- 基于隐语义 LDA模型的漫步口令猜测模型。口令猜测模型是建模人类口令行为的重要工具，是口令安全研究的核心课题之一。基于NLP 的漫步猜测攻击模型效果不佳的一个重要原因是，NLP 方法在语义标注和抽象化的粒度方面难以调控，而隐语义LDA 模型正好可以弥补这一不足。如何将LDA 模型引入口令猜测模型，以实现细粒度的语义标注和抽象是一个有前景的研究方向。
- 基于个人可标识信息的口令强度评测算法。当前口令强度评测研究成果主要集中在，如何基于漫步猜测攻击模型，设计更逼近理想漫步猜测攻击者的口令强度评测算法。随着用户越来越多的个人相关信息被公开，在其它网站使用的口令被泄露，设计基于定向口令猜测模型的强度评测算法，对用户口令进行准确评测的需求十分迫切。本文提出了基于口令重用行为的口令强度评测算法fuzzyPSM，但未考虑用户个人可标识信息(PII)，如何将PII信息引入我们的fuzzyPSM是一个具有重要现实意义的研究方向。

参考文献

[1] Robert Morris, Ken Thompson. Password security: A case history[J]. Commun ACM, 1979, **22**(11):594–597

[2] Lawrence O'Gorman. Comparing passwords, tokens, and biometrics for user authentication[J]. Proc of the IEEE, 2003, **91**(12):2021–2040

[3] Security on the IBM Mainframe[EB/OL], Dec. 2014. `http://www.redbooks.ibm.com/redbooks/pdfs/sg247803.pdf`

[4] Mark Keith, Benjamin Shao, Paul John Steinbart. The usability of passphrases for authentication: An empirical field study[J]. Int J Human-Comput Studies, 2007, **65**(1):17–28

[5] Joseph Bonneau. The science of guessing: Analyzing an anonymized corpus of 70 million passwords[C]. Proc. IEEE S&P 2012, pp. 538–552

[6] Daniel V Klein. Foiling the cracker: A survey of, and improvements to, password security[C]. Proc. USENIX SEC 1990, pp. 5–14

[7] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, XiaoFeng Wang. The tangled web of password reuse[C]. Proc. NDSS 2014, pp. 1–15

[8] Blake Ives, Kenneth R Walsh, Helmut Schneider. The domino effect of password reuse[J]. Commun ACM, 2004, **47**(4):75–78

[9] Gates predicts death of the password[EB/OL], Feb. 2004. `http://www.cnet.com/news/gates-predicts-death-of-the-password/`

[10] Luke St Clair, Lisa Johansen, William Enck, Matthew Pirretti, Patrick Traynor, Patrick McDaniel, Trent Jaeger. Password exhaustion: Predicting the end of password usefulness[C]. Proc. ICISS 2006, pp. 37–55

[11] Yinqian Zhang, Fabian Monrose, Michael Reiter. The security of modern password expiration:an algorithmic framework and empirical analysis[C]. Proc. ACM CCS 2010, pp. 176–186

[12] Robert Biddle, Sonia Chiasson, Paul C Van Oorschot. Graphical passwords: Learning from the first twelve years[J]. ACM Comput Surv, 2012, **44**(4):1–41

[13] Anil K Jain, Arun Ross, Sharath Pankanti. Biometrics: a tool for information security[J]. IEEE Trans Inf Foren Secur, 2006, **1**(2):125–143

[14] Nan Zheng, Aaron Paloski, Haining Wang. An efficient user verification

system via mouse movements[C]. Proc. ACM CCS 2011, pp. 139–150

[15] Xinyi Huang, Yang Xiang, Ashley Chonka, Jianying Zhou, Robert H Deng. A generic framework for three-factor authentication: Preserving security and privacy in distributed systems[J]. IEEE Trans Parallel Distrib Syst, 2011, **22**(8):1390–1397

[16] Joseph Bonneau, Cormac Herley, Paul C Van Oorschot, Frank Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes[C]. Proc. IEEE S&P 2012, pp. 553–567

[17] Jonathan T Weinberg. Biometric identity[J]. Commun ACM, 2015, **59**(1):30–32

[18] Cormac Herley, Paul Van Oorschot. A research agenda acknowledging the persistence of passwords[J]. IEEE Secur & Priv, 2012, **10**(1):28–36

[19] Markus Dürmuth, David Freeman, Battista Biggio. Who are you? A statistical approach to measuring user authenticity[C]. Proc. NDSS 2016, pp. 1–15

[20] Thomas Wu. The SRP authentication and key exchange system[EB/OL], 2000. `http://tools.ietf.org/html/rfc2945`

[21] Jonathan Katz, Rafail Ostrovsky, Moti Yung. Efficient and secure authenticated key exchange using weak passwords[J]. J ACM, 2009, **57**(1):1–41

[22] Michel Abdalla, Fabrice Benhamouda, Philip MacKenzie. Security of the J-PAKE Password-Authenticated Key Exchange Protocol[C]. Proc. IEEE S&P 2015, pp. 571–587

[23] William Cheswick. Rethinking passwords[J]. Commun ACM, 2013, **56**(2):40–44

[24] Abe Singer, Warren Anderson, Rik Farrow. Rethinking password policies[J]. USENIX Login, 2013, **38**(4):14–18

[25] Joseph Bonneau, Cormac Herley, Paul van Oorschot, Frank Stajano. Passwords and the evolution of imperfect authentication[J]. Commun ACM, 2015, **58**(7):78–87

[26] 2016 Data Breach Investigations Report[EB/OL], April 2016. `http://www.verizonenterprise.com/resources/reports/rp_dbir-2016-executive-summary_xg_en.pdf`

[27] Tips for Producing Strong Passwords[EB/OL], Oct. 2016. `http://news.erau.edu/headlines/tips-for-producing-strong-passwords/`

[28] Amid Widespread Data Breaches in China[EB/OL], Dec. 2011.

Http://www.techinasia.com/alipay-hack/

[29] Patrick Heim. Resetting passwords to keep your files safe[EB/OL], Aug. 2016. Https://blogs.dropbox.com/dropbox/2016/08/resetting-passwords-to-keep-your-files-safe/

[30] Dado Ruvic. Hackers stole data from more than 1 billion user accounts - Yahoo[EB/OL], Dec. 2016. https://www.rt.com/usa/370323-yahoo-hack-billion-users/

[31] Shine Wong. Xiaomi Forum 8 million users' private data leaked[EB/OL], May, 2014. http://www.gizmochina.com/2014/05/13/xiaomi-forum-8-million-users-private-data-leaked/

[32] Dan Goodin. Twitch resets user passwords following breach[EB/OL], March 2015. http://arstechnica.com/security/2015/03/twitch-resets-user-passwords-following-breach/

[33] Shape Security Inc. 2017 Credential Spill Report[EB/OL], Jan. 2017. http://info.shapesecurity.com/rs/935-ZAM-778/images/Shape-2017-Credential-Spill-Report.pdf

[34] Jerry Ma, Weining Yang, Min Luo, Ninghui Li. A Study of Probabilistic Password Models[C]. Proc. IEEE S&P 2014, pp. 538–552

[35] Jeremi Gosney. Password cracking HPC[C]. Proc. Password 2012, pp. http://passwords12.at.ifi.uio.no/Jeremi_Gosney_Password_Cracking_HPC_Passwords12.pdf

[36] Mechanicus Adeptus. Hashdumps and Passwords[EB/OL], May 2014. http://www.adeptus-mechanicus.com/codex/hashpass/hashpass.php

[37] William Burr, Donna Dodson, William Polk. NIST SP800-63-2: Electronic Authentication Guideline[R]. Tech. rep., NIST, Reston, VA, Aug. 2013

[38] Paul A. Grassi, James L. Fenton. NIST SP800-63B: Digital Authentication Guideline[R]. Tech. rep., NIST, Reston, VA, 2017. https://pages.nist.gov/800-63-3/sp800-63b.html

[39] David Raven. Apple iCloud password hack could be 'responsible for security breach'[EB/OL], Sep. 2014. http://www.mirror.co.uk/3am/celebrity-news/jennifer-lawrence-leaked-nude-photos-4145139

[40] Zeljka Zorz. GitHub accounts compromised in wake of reused password attack[EB/OL], June 2016. https://www.helpnetsecurity.com/2016/06/17/github-reused-password-attack/?utm_source=dlvr.it&utm_medium=twitter

[41] Peter Allen. Obama's Twitter password revealed after French hacker arrested for breaking into U.S. president's account[EB/OL], March 2010. Http://www.dailymail.co.uk/news/article-1260488/Barack-Obamas-Twitter-password-revealed-French-hacker-arrested.html

[42] Emil Protraltnski. Mark Zuckerberg' s Twitter and Pinterest accounts hacked, LinkedIn password dump likely to blame[EB/OL], June 2016. http://bit.ly/2ncZwq8

[43] Joseph Steinberg. Katy Perry's Twitter Account–With the Most Followers Anywhere–Was Hacked[EB/OL], May 2016. http://www.inc.com/joseph-steinberg/katy-perrys-twitter-account-was-just-hacked-heres-how-to-protect-yours.html

[44] Cheyenne Macdonald, Abigall Beall. Passwords used in the biggest ever cyberattack revealed - and '12345' and 'password' were top[EB/OL], Oct. 2016. http://www.dailymail.co.uk/sciencetech/article-3825740/Passwords-used-biggest-DDoS-attack-revealed-12345-password-top.html

[45] Ted Samson. Amazon EC2 enables brute-force attacks on the cheap[EB/OL], Jan. 2011. http://www.infoworld.com/t/data-security/amazon-ec2-enables-brute-force-attacks-the-cheap-148447

[46] 中国网信网. "十三五"规划纲要：实施网络强国战略加快建设数字中国[EB/OL], March 2016. http://www.cac.gov.cn/2016-03/18/c_1118372649.htm

[47] 今日中国. 中国全面开启网络实名制[EB/OL], June 2015. http://www.chinatoday.com.cn/chinese/society/zx/201506/t20150617_800034265.html

[48] Maurice Vincent Wilkes. Time-sharing computer systems[M]. American Elsevier, 1968

[49] Simson Garfinkel, Heather Richter Lipford. Usable security: History, themes, and challenges[J]. Synthesis Lectures on Inform Secur Priv Trust, 2014, **5**(2):1–124

[50] Yimin Song, Chao Yang, Guofei Gu. Who is peeping at your passwords at Starbucks? To catch an evil twin access point[C]. Proc. IEEE/IFIP DSN 2010, pp. 323–332

[51] Fengwei Zhang, Kevin Leach, Haining Wang, Angelos Stavrou. TrustLogin: Securing password-login on commodity operating systems[C]. Proc.

ASIACCS 2015, pp. 333–344

[52] Leslie Lamport. Password authentication with insecure communication[J]. Commun ACM, 1981, **24**(11):770–772

[53] C. C. Chang, T. C. Wu. Remote password authentication with smart cards[J]. IEE Comput Digital Tech, 1991, **138**(3):165–168

[54] Steven M Bellovin, Michael Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks[C]. Proc. IEEE S&P 1992, pp. 72–84

[55] Neil Haller. The S/KEY one-time password system[EB/OL], Feb. 1995. Https://tools.ietf.org/html/rfc1760

[56] S. Halevi, H. Krawczyk. Public-key cryptography and password protocols[J]. ACM Trans Inform Syst Secur, 1999, **2**(3):230–268

[57] Michel Abdalla, Pierre Alain Fouque, David Pointcheval. Password-based authenticated key exchange in the three-party setting[C]. Proc. PKC 2005, pp. 65–84

[58] Jonathan Katz, Philip MacKenzie, Gelareh Taban, Virgil Gligor. Two-server password-only authenticated key exchange[J]. J Comput Syst Sci, 2012, **78**(2):651–669

[59] Mario Di Raimondo, Rosario Gennaro. Provably secure threshold password-authenticated key exchange[C]. Proc. EUROCRYPT 2003. Springer, 2003, pp., 507–523

[60] Jiangshan Yu, Guilin Wang, Yi Mu, Wei Gao. An efficient generic framework for three-factor authentication with provably secure instantiation[J]. IEEE Trans Inform Foren Secur, 2014, **9**(12):2302–2313

[61] George B Purdy. A high security log-in procedure[J]. Commun ACM, 1974, **17**(8):442–445

[62] Jerome H Saltzer, Michael D Schroeder. The protection of information in computer systems[J]. Proc IEEE, 1975, **63**(9):1278–1308

[63] Chris Davies, Ravi Ganesan. Bapasswd: A new proactive password checker[C]. 16th National Computer Security Conference. 1993, pp., 1–15

[64] Thomas Wu. A Real-World Analysis of Kerberos Password Security.[C]. Proc. NDSS 1999, pp. 13–22

[65] Jeff Yan, Alan F Blackwell, Ross J Anderson, Alasdair Grant. Password memorability and security: Empirical results.[J]. IEEE Secur & Priv, 2004, **2**(5):25–31

[66] Arvind Narayanan, Vitaly Shmatikov. Fast dictionary attacks on passwords using time-space tradeoff[C]. Proc. ACM CCS 2005, pp. 364–372

[67] Dinei Florêncio, Cormac Herley, Baris Coskun. Do strong web passwords accomplish anything?[C]. Proc. HotSec 2007, pp. 1–6

[68] Matt Weir, Sudhir Aggarwal, Breno de Medeiros, Bill Glodek. Password cracking using probabilistic context-free grammars[C]. Proc. IEEE S&P 2009, pp. 391–405

[69] Rafael Veras, Christopher Collins, Julie Thorpe. On the semantic patterns of passwords and their security impact[C]. Proc. NDSS 2014, pp. 1–16

[70] Shiva Houshmand, Sudhir Aggarwal. Building better passwords using probabilistic techniques[C]. Proc. ACSAC 2012, pp. 109–118

[71] Claude Castelluccia, Markus Dürmuth, Daniele Perito. Adaptive password-strength meters from Markov models[C]. Proc. NDSS 2012, pp. 1–15

[72] A Post-Password World?[EB/OL], April 2016. `https://www.dyadicsec.com/post-password_world/`

[73] Ding Wang, Ping Wang. The Emperor's New Password Creation Policies. Proc. ESORICS 2015, pp. 456–477

[74] Anne Adams, Martina Angela Sasse. Users are not the enemy[J]. Commun ACM, 1999, **42**(12):40–46

[75] Dinei Florencio, Cormac Herley. A large-scale study of web password habits[C]. Proc. WWW 2007, pp. 657–666

[76] Ding Wang, Hai Bo Cheng, Ping Wang. Understanding passwords of Chinese users: A Data-Driven Approach[J]. IEEE Trans Depend Secur Comput, 2017. In press, `http://bit.ly/2maZLCd`

[77] Steven Furnell. An assessment of website password practices[J]. Comput Secur, 2007, **26**(7):445–451

[78] Adam Beautement, M Angela Sasse, Mike Wonham. The compliance budget: managing security behaviour in organisations[C]. Proc. ACM NSPW 2009, pp. 47–58

[79] Rishab Nithyanand, Rob Johnson. The password allocation problem: strategies for reusing passwords effectively[C]. Proc. ACM WPES 2013, pp. 255–260

[80] Elizabeth Stobert, Robert Biddle. The password life cycle: user behaviour in managing passwords[C]. Proc. SOUPS 2014, pp. 243–255

[81] Joseph Bonneau, Sören Preibusch. The password thicket: Technical and

market failures in human authentication on the Web.[C]. Proc. WEIS 2010, pp. 1–48

[82] Dinei Florêncio, Cormac Herley, Paul C Van Oorschot. Password portfolios and the finite-effort user: Sustainably managing large numbers of accounts[C]. Proc. USENIX SEC 2014, pp. 575–590

[83] 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[J]. 计算机学报, 2016, **39**(3):454–467

[84] Zhigong Li, Weili Han, Wenyuan Xu. A Large-Scale Empirical Analysis on Chinese Web Passwords[C]. Proc. USENIX SEC 2014, pp. 559–574

[85] Daniel V Bailey, Markus Dürmuth, Christof Paar. Statistics on Password Re-use and Adaptive Strength for Financial Accounts. Proc. SCN 2014, pp. 218–235

[86] Michelle L Mazurek, Saranga Komanduri, Timothy Vidas, Lorrie Faith Cranor, Patrick Gage Kelley, Richard Shay, Blase Ur. Measuring password guessability for an entire university[C]. Proc. ACM CCS 2013, pp. 173–186

[87] Patrick Gage Kelley, Saranga Komanduri, Michelle L Mazurek, et al. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms[C]. Proc. IEEE S&P 2012, pp. 523–537

[88] Blase Ur, Fumiko Noma, Jonathan Bees, et al. “I added ‘!’ at the end to make it secure”: Observing password creation in the lab[C]. Proc. SOUPS 2015, pp. 123–140

[89] Richard Shay, Lujo Bauer, Nicolas Christin, et al. A spoonful of sugar?: The impact of guidance and feedback on password-creation behavior[C]. Proc. ACM CHI 2015, pp. 2903–2912

[90] SM Taiabul Haque, Matthew Wright, Shannon Scielzo. Hierarchy of users? web passwords: Perceptions, practices and susceptibilities[J]. Int J Hum-Comput Stud, 2014, **72**(12):860–874

[91] Gang Han, Yu Yu, Xiangxue Li, Kefei Chen, Hui Li. Characterizing the semantics of passwords: The role of Pinyin for Chinese Netizens[J]. Comput Stand Inter, 2017, **54**:20–28

[92] 韩伟力, 袁琅, 李思斯, 王晓阳. 一种基于样本的模拟口令集生成算法[J]. 计算机学报, 2017, **40**(5):1151–1157

[93] Matt Weir, Sudhir Aggarwal, Michael Collins, Henry Stern. Testing metrics for password creation policies by attacking large sets of revealed passwords[C]. Proc. ACM CCS 2010, pp. 162–175

[94] Shouling Ji, Shukun Yang, Xin Hu, Weili Han, Zhigong Li, Raheem Beyah. Zero-Sum Password Cracking Game: A Large-Scale Empirical Study on the Crackability, Correlation, and Security of Passwords[J]. IEEE Trans Depend Secur Comput, 2015. Doi: 10.1109/TDSC.2015.2481884

[95] Mindi McDowell, J Rafail, S Hernan. Choosing and Protecting Passwords[EB/OL], Oct. 2016. https://www.us-cert.gov/ncas/tips/ST04-002

[96] David Jacoby. False Perceptions of IT Security: Passwords[EB/OL], Dec. 2014. https://blog.kaspersky.com/false-perception-of-it-security-passwords/7036/

[97] Jeremiah Blocki, Manuel Blum, Anupam Datta. Naturally rehearsing passwords[C]. Proc. ASIACRYPT 2013, pp. 361–380

[98] Centre for the Protection of National Infrastructure. Password guidance: executive summary[EB/OL], Sept. 2015. https://www.gov.uk/government/publications/password-policy-simplifyig-your-approach/password-policy-executive-summary

[99] Brett Stone Gross, Marco Cova, Lorenzo Cavallaro, et al. Your botnet is my botnet: analysis of a botnet takeover[C]. Proc. ACM CCS 2009, pp. 635–647

[100] Steve Wise. Survey Finds Vast Majority Of Americans Memorize Or Write Passwords On Paper[EB/OL], July 2015. https://www.passwordboss.com/news/survey-finds-vast-majority-of-americans-memorize-or-write-passwords-on-paper/

[101] David Jaeger, Chris Pelchen, Hendrik Graupner, Feng Cheng, Christoph Meinel. Analysis of Publicly Leaked Credentials and the Long Story of Password (Re-)use[C]. Proc. Password 2016, pp. 1–19

[102] Roman V Yampolskiy. Analyzing user password selection behavior for reduction of password space[C]. Proc. 40th Annual IEEE Int. Carnahan Conf. Secur. Techn. 2006, pp., 109–115

[103] Yue Li, Haining Wang, Kun Sun. A Study of Personal Information in Human-chosen Passwords and Its Security Implications[C]. Proc. INFOCOM 2016, pp. 1–9

[104] Matteo Dell'Amico, Pietro Michiardi, Yves Roudier. Password strength: An empirical analysis[C]. Proc. INFOCOM 2010, pp. 1–9

[105] Matteo Dell'Amico, Maurizio Filippone. Monte Carlo strength evaluation: Fast and reliable password checking[C]. Proc. CCS 2015, pp. 158–169

[106] Eugene H Spafford. Opus: Preventing weak password choices[J]. Comput Secur, 1992, **11**(3):273–278

[107] Cynthia Kuo, Sasha Romanosky, Lorrie Faith Cranor. Human selection of mnemonic phrase-based passwords[C]. Proc. SOUPS 2006, pp. 67–78

[108] Bruce Schneier. Real-World Passwords[EB/OL], Dec. 2006. `https://www.schneier.com/blog/archives/2006/12/realworld_passw.html`

[109] Matt Bishop, Daniel V Klein. Improving system security via proactive password checking[J]. Comput & Secur, 1995, **14**(3):233–249

[110] Blase Ur, Patrick Gage Kelley, Saranga Komanduri, et al. How does your password measure up? The effect of strength meters on password creation[C]. Proc. USENIX SEC 2012, pp. 65–80

[111] Serge Egelman, Andreas Sotirakopoulos, Konstantin Beznosov, Cormac Herley. Does my password go up to eleven?: the impact of password meters on password selection[C]. Proc. ACM CHI 2013, pp. 2379–2388

[112] Stanford University IT. SUNet ID Passwords[EB/OL], Feb. 2017. `https://uit.stanford.edu/service/accounts/passwords`

[113] Dinei Florêncio, Cormac Herley. Where do security policies come from?[C]. Proc. SOUPS 2010, pp. 1–14

[114] Steven Furnell. Assessing password guidance and enforcement on leading websites[J]. Comput Fraud Secur, 2011, **2011**(12):10–18

[115] Roger A. Grimes. Password size does matter[EB/OL], July 2006. `http://www.infoworld.com/article/2655121/security/password-size-does-matter.html`

[116] Richard Shay, Saranga Komanduri, Adam Durity, et al. Designing Password Policies for Strength and Usability[J]. ACM Trans Inf Syst Secur, 2016, **18**(4):1–34

[117] Dinei Florêncio, Cormac Herley, Paul C van Oorschot. Pushing on string: the "don't care" region of password strength[J]. Commun ACM, 2016, **59**(11):66–74

[118] Eugene Panferov. A Canonical Password Strength Measure[EB/OL], 2015. `https://arxiv.org/pdf/1505.05090.pdf`

[119] Blase Ur, Jonathan Bees, Sean Segreti, et al. Do users' perceptions of password security match reality?[C]. Proc. CHI 2016, pp. 3748–3760

[120] Xavier Carnavalet, Mohammad Mannan. From very weak to very strong: Analyzing password-strength meters[C]. Proc. NDSS 2014, pp. 1–16

[121] Xavier Carnavalet, Mohammad Mannan. A Large-Scale Evaluation of High-Impact Password Strength Meters[J]. ACM Trans Inform Syst Secur, 2015, **18**(1):1–32

[122] Dan Wheeler. zxcvbn: Low-Budget Password Strength Estimation[C]. Proc. USENIX SEC 2016, pp. 157–173

[123] Dominik Reichl. Details on the quality/strength estimations in KeePass[EB/OL], July 2015. http://keepass.info/help/kb/pw_quality_est.html

[124] Password strength meter of Tecent[EB/OL], Feb. 2017. http://zc.qq.com/chs/v2/

[125] David Malone, Kevin Maher. Investigating the distribution of password choices[C]. Proc. WWW 2012, pp. 301–310

[126] The Avalanche Technology Group. All Data Breach Sources[EB/OL], Dec. 2015. https://breachalarm.com/all-sources

[127] Nicole Perlroth. Yahoo Says Hackers Stole Data on 500 Million Users in 2014[EB/OL], Sep. 2016. http://www.nytimes.com/2016/09/23/technology/yahoo-hackers.html?_r=0

[128] Michael Heller. Yet another Yahoo breach compromises more than 1 billion accounts[EB/OL], Dec. 2016. http://searchsecurity.techtarget.com/news/450404755/Yet-another-Yahoo-breach-compromises-more-than-one-billion-accounts

[129] George Kingsley Zipf. Human behavior and the principle of least effort.[M]. Addison-Wesley Press, 1949

[130] Mihir Bellare, Viet Tung Hoang. Adaptive witness encryption and asymmetric password-based cryptography. Proc. PKC 2015, pp. 308–331

[131] Kristian Gjosteen, Oystein Thuen. Password-Based Signatures. EuroPKI 2011, 2012, vol. 7163 of LNCS, pp. 17–33

[132] Xun Yi, Feng Hao, Liqun Chen, Joseph K Liu. Practical Threshold Password-Authenticated Secret Sharing Protocol. Proc. ESORICS 2015, pp. 347–365

[133] Franziskus Kiefer, Mark Manulis. Zero-Knowledge Password Policy Checks and Verifier-Based PAKE. Proc. ESORICS 2014, vol. 8713 of LNCS, pp. 295–312

[134] Michel Abdalla, Fabrice Benhamouda, David Pointcheval. Public-Key Encryption Indistinguishable Under Plaintext-Checkable Attacks. Proc.

PKC 2015, pp. 332–352

[135] Stuart Schechter, Cormac Herley, Michael Mitzenmacher. Popularity is everything: A new approach to protecting passwords from statistical-guessing attacks[C]. Proc. HotSec 2010, pp. 1–8

[136] Stanislaw Jarecki, Aggelos Kiayias, Hugo Krawczyk, Jiayu Xu. Highly-Efficient and Composable Password-Protected Secret Sharing[C]. Proc. IEEE EuroS&P 2016, pp. 1–16

[137] Claude Elwood Shannon. A mathematical theory of communication[J]. ACM SIGMOBILE Mobile Comput Commun Rev, 2001, **5**(1):3–55

[138] Alfréd Rényi, et al. On measures of entropy and information[C]. Proc. 4th Berkeley Symp. Mathematical Statistics and Probability. 1961, pp., 547–561

[139] James L Massey. Guessing and entropy[C]. Proc. IEEE ISIT 1994, pp. 204–204

[140] S Boztas. Entropies, guessing, and cryptography[EB/OL], 1999

[141] John O Pliam. On the incomparability of entropy and marginal guesswork in brute-force attacks[C]. Proc. Indocrypt 2000, pp. 67–79

[142] Rahul Chatterjee, Joseph Bonneau, Ari Juels, Thomas Ristenpart. Cracking-resistant password vaults using natural language encoders[C]. Proc. IEEE S&P 2015, pp. 481–498

[143] Jun Ho Huh, Seongyeol Oh, Hyoungshick Kim, Konstantin Beznosov, Apurva Mohan, S Raj Rajagopalan. Surpass: System-initiated User-replaceable Passwords[C]. Proc. ACM CCS 2015, pp. 170–181

[144] Sebastian Uellenbeck, Markus Dürmuth, Christopher Wolf, Thorsten Holz. Quantifying the security of graphical passwords: The case of android unlock patterns[C]. Proc. ACM CCS 2013, pp. 161–172

[145] Bin B Zhu, Jeff Yan, Dongchen Wei, Maowei Yang. Security Analyses of Click-based Graphical Passwords via Image Point Memorability[C]. Proc. ACM CCS 2014, pp. 1217–1231

[146] Youngbae Song, Geumhwan Cho, Seongyeol Oh, Hyoungshick Kim, Jun Ho Huh. On the Effectiveness of Pattern Lock Strength Meters: Measuring the Strength of Real World Pattern Locks[C]. Proc. ACM CHI 2015, pp. 2343–2352

[147] Simon Blake Wilson, Don Johnson, Alfred Menezes. Key agreement protocols and their security analysis. Crytography and Coding 1997, Springer, 1997, vol. 1355 of LNCS, pp. 30–45

[148] Hugo Krawczyk. HMQV: A High-Performance Secure Diffie-Hellman Protocol. Proc. CRYPTO 2005, pp. 546–566

[149] Mihir Bellare, Phillip Rogaway. Entity Authentication and Key Distribution[C]. Proc. CRYPTO 1993, pp. 232–249

[150] Jing Xu, Wentao Zhu, Dengguo Feng. An improved smart card based password authentication scheme with provable security[J]. Comput Stand Inter, 2009, **31**(4):723–728

[151] Yongge Wang. Password Protected Smart Card and Memory Stick Authentication Against Off-line Dictionary Attacks. Proc. SEC 2012, Springer Boston, 2012, vol. 376, pp. 489–500

[152] Rongxing Lu, Zhenfu Cao. Simple three-party key exchange protocol[J]. Comput Secur, 2007, **26**(1):94–97

[153] Jin Wook Byun, Ik Rae Jeong, Dong Hoon Lee, Chang-Seop Park. Password-authenticated key exchange between clients with different passwords[C]. Proc ICICS 2002, pp. 134–146

[154] Lomas Gong, Mark A Lomas, Roger M Needham, Jerome H Saltzer. Protecting poorly chosen secrets from guessing attacks[J]. IEEE J Selected Areas in Commun, 1993, **11**(5):648–656

[155] Yanjiang Yang, Jianying Zhou, Jun Wen Wong, Feng Bao. Towards practical anonymous password authentication[C]. Proc. ACSAC 2008, pp. 59–68

[156] Zhenfeng Zhang, Kang Yang, Xuexian Hu, Yuchen Wang. Practical Anonymous Password Authentication and TLS with Anonymous Client Authentication[C]. Proc. ACM CCS 2016, pp. 1179–1191

[157] Debiao He, Sherali Zeadally, Neeraj Kumar, Wei Wu. Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures[J]. IEEE Trans Inform Foren Secur, 2016, **11**(9):2052–2064

[158] Xun Yi, Feng Hao, Elisa Bertino. ID-Based Two-Server Password-Authenticated Key Exchange. Proc. ESORICS 2014, pp. 257–276

[159] Michel Abdalla, Olivier Chevassut, Pierre-Alain Fouque, David Pointcheval. A simple threshold authenticated key exchange from short secrets[C]. Proc. AISACRYPT 2005, pp. 566–584

[160] Hung Min Sun. An efficient remote use authentication scheme using smart cards[J]. IEEE Trans Consum Electr, 2000, **46**(4):958–961

[161] Shyi Tsong Wu, Bin Chang Chieu. A user friendly remote authentication

scheme with smart cards[J]. Comput Secur, 2003, **22**(6):547–550

[162] Amit K Awasthi, Sunder Lal. A remote user authentication scheme using smart cards with forward secrecy[J]. IEEE Trans Comsum Electr, 2003, **49**(4):1246–1248

[163] Paul Kocher, Joshua Jaffe, Benjamin Jun. Differential power analysis[C]. Proc. CRYPTO 1999, pp. 789–789

[164] Werner Schindler. A timing attack against RSA with the chinese remainder theorem[C]. Proc. CHES 2000, pp. 109–124

[165] Karsten Nohl, David Evans, Starbug Starbug, Henryk Plötz. Reverse-engineering a cryptographic RFID tag[C]. Proc. USENIX SEC 2008, pp. 185–193

[166] Wei Chi Ku, Shuai Min Chen. Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards[J]. IEEE Trans Comsum Electr, 2004, **50**(1):204–207

[167] Eun Jun Yoon, Eun Kyung Ryu, Kee Young Yoo. Further improvement of an efficient password based remote user authentication scheme using smart cards[J]. IEEE Trans Comsum Electr, 2004, **50**(2):612–614

[168] Xiaomin Wang, Wenfang Zhang, Jiashu Zhang, Muhammad Khurram Khan. Cryptanalysis and improvement on two efficient remote user authentication scheme using smart cards[J]. Comput Stand Inter, 2007, **29**(5):507–512

[169] Han Cheng Hsiang, Wei Kuan Shih. Weaknesses and improvements of the Yoon-Ryu-Yoo remote user authentication scheme using smart cards[J]. Comput Commun, 2009, **32**(4):649–652

[170] R. Madhusudhan, RC Mittal. Dynamic ID-based remote user password authentication schemes using smart cards: A review[J]. J Netw Comput Appl, 2012, **35**(4):1235–1248

[171] Bin Liu, Yurong Jiang, Fei Sha, Ramesh Govindan. Cloud-enabled privacy-preserving collaborative learning for mobile sensing[C]. Proc. Sensys 2012, pp. 57–70

[172] Feng Zhu, Sandra Carpenter, Ajinkya Kulkarni. Understanding identity exposure in pervasive computing environments[J]. Perva Mobile Comput, 2012, **8**(5):777–794

[173] Han Cheng Hsiang, Wei Kuan Shih. Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment[J]. Comput Stand Inter, 2009, **31**(6):1118–1123

[174] Muhammed Khurram Khan, Soo-Kyun Kim, Khaled Alghathbar. Cryptanalysis and security enhancement of a more efficient & secure dynamic ID-based remote user authentication scheme'[J]. Comput Commun, 2011, **34**(3):305–309

[175] Daojing He, Maode Ma, Yan Zhang, Chun Chen, Jiajun Bu. A strong user authentication scheme with smart cards for wireless communications[J]. Comput Commun, 2011, **34**(3):367–374

[176] Xiong Li, Yong Xiong, Jian Ma, Wendong Wang. An enhanced and security dynamic identity based authentication protocol for multi-server architecture using smart cards[J]. J of Netw Comput Appl, 2012, **35**(2):763–769

[177] Kaiping Xue, Peilin Hong, Changsha Ma. A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture[J]. J Comput Syst Sci, 2014, **80**(1):195–206

[178] Jenq Shiou Leu, Wen Bin Hsieh. Efficient and secure dynamic ID-based remote user authentication scheme for distributed systems using smart cards[J]. IET Inf Secur, 2014, **8**(2):104–113

[179] Ruhul Amin, Neeraj Kumar, GP Biswas, R Iqbal, Victor Chang. A light weight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment[J]. Future Gener Comput Syst, 2016. Doi: 10.1016/j.future.2016.12.028

[180] Prosanta Gope, Tzonelih Hwang. An efficient mutual authentication and key agreement scheme preserving strong anonymity of the mobile user in global mobility networks[J]. J Netw Comput Appl, 2016, **62**:1–8

[181] Ruhul Amin, SK Hafizul Islam, GP Biswas, Muhammad Khurram Khan, Lu Leng, Neeraj Kumar. Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks[J]. Comput Netw, 2016, **101**:42–62

[182] Tanmoy Maitra, SK Islam, Ruhul Amin, et al. An enhanced multi-server authentication protocol using password and smart-card: cryptanalysis and design[J]. Secur Comm Netw, 2016, **9**(17):4615–4638

[183] Debiao He, Ding Wang. Robust biometrics-based authentication scheme for multiserver environment[J]. IEEE Syst J, 2015, **9**(3):816–823

[184] Chun I Fan, Yi Hui Lin. Provably secure remote truly three-factor authentication scheme with privacy protection on biometrics[J]. IEEE Trans

Inform Foren Secur, 2009, **4**(4):933–945

[185] Yanjiang Yang, Haibing Lu, Joseph K Liu, Jian Weng, Youcheng Zhang, Jianying Zhou. Credential Wrapping: From Anonymous Password Authentication to Anonymous Biometric Authentication[C]. Proc. ACM ASIACCS 2016, pp. 141–151

[186] Vanga Odelu, Ashok Kumar Das, Adrijit Goswami. A Secure Biometrics-Based Multi-Server Authentication Protocol using Smart Cards[J]. IEEE Trans Inform Foren Secur, 2015, **10**(9):1953–1966

[187] Saranga Komanduri, Richard Shay, Patrick Gage Kelley, Michelle L Mazurek, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Serge Egelman. Of passwords and people: measuring the effect of password-composition policies[C]. Proc. of CHI 2011, pp. 2595–2604

[188] Jim Blythe, Ross Koppel, Sean W Smith. Circumvention of Security: Good Users Do Bad Things[J]. IEEE Secur & Priv, 2013, **11**(5):80–83

[189] Richard Shay, Saranga Komanduri, Adam L Durity, Phillip Seyoung Huh, Michelle L Mazurek, Sean M Segreti, Blase Ur, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor. Can long passwords be secure and usable?[C]. Proc. ACM CHI 2014, pp. 2927–2936

[190] Dinei Florêncio, Cormac Herley, P van Oorschot. An Administrator's Guide to Internet Password Research[C]. Proc. USENIX LISA 2014, pp. 44–61

[191] Ari Juels, Ronald L Rivest. Honeywords: Making password-cracking detectable[C]. Proc. ACM CCS 2013, pp. 145–160

[192] Jeremiah Blocki, Anupam Datta. CASH: A Cost Asymmetric Secure Hash Algorithm for Optimal Password Protection[C]. Proc. IEEE CSF 2016, pp. 1–10. `http://arxiv.org/pdf/1509.00239v1.pdf`

[193] Alan S Brown, Elisabeth Bracken, Sandy Zoccoli. Generating and remembering passwords[J]. Applied Cogn Psych, 2004, **18**(6):641–651

[194] Ron Bowes. Password dictionaries[EB/OL], Oct. 2011. `https://wiki.skullsecurity.org/Passwords`

[195] S. Designer. John the Ripper password cracker[EB/OL], Feb. 1996. `http://www.openwall.com/john/`

[196] Weili Han, Zhigong Li, Lang Yuan, Wenyuan Xu. Regional Patterns and Vulnerability Analysis of Chinese Web Passwords[J]. IEEE Trans Inform Foren Secur, 2016, **11**(2):258–272

[197] Blase Ur, Sean M Segreti, Lujo Bauer, et al. Measuring real-world accuracies

and biases in modeling password guessability[C]. Proc. USENIX SEC 2015, pp. 463–481

[198] Campbell Allan. 32 million Rockyou passwords stolen[EB/OL], Dec. 2009. Available at http://www.hardwareheaven.com/news.php?newsid=526

[199] Dan Goodin. Personal data is exposed as a result of a five-month-old hack on 000Webhost[EB/OL], Oct. 2015. http://t.cn/R4tKrEU

[200] Lucian Constantin. Security Gurus 0wned by Black Hats[EB/OL], July 2009. Http://news.softpedia.com/news/Security-Gurus-0wned-by-Black-Hats-117934.shtml

[201] Jason Mick. Russian Hackers Compile List of 10M+ Stolen Gmail, Yandex, Mailru[EB/OL], Sep. 2014. http://t.cn/R4tmJE3

[202] Bruce Schneier. Password data of Flirtlife.de compromises[EB/OL], May 2006. https://www.schneier.com/blog/archives/2006/05/common_password.html

[203] Casey Johnston. Why your password can't have symbols[EB/OL], April 2013. http://t.cn/zTNoEEs

[204] Miranda Lee Pao. An empirical examination of Lotka's law[J]. J Amer Soc Inform Sci, 1986, **37**(1):26–33

[205] Aaron Clauset, Cosma Rohilla Shalizi, Mark EJ Newman. Power-law distributions in empirical data[J]. SIAM Review, 2009, **51**(4):661–703

[206] Mark EJ Newman. Power laws, Pareto distributions and Zipf's law[J]. Contemp Phys, 2005, **46**(5):323–351

[207] T Maillart, D Sornette, S Spaeth, G Von Krogh. Empirical tests of Zipf's law mechanism in open source Linux distribution[J]. Phys Rev Lett, 2008, **101**(21):218701

[208] Réka Albert, Hawoong Jeong, Albert-László Barabási. Error and attack tolerance of complex networks[J]. Nature, 2000, **406**(6794):378–382

[209] Ding Wang, Qianchen Gu, Xinyi Huang, Ping Wang. Understanding Human-Chosen PINs: Characteristics, Distribution and Security[C]. Proc. ACM ASIACCS 2017, pp. 372–385

[210] David L Russell. Optimization theory[M], vol. 1. WA Benjamin Advanced Book Program, 1970

[211] Susan Nolan. Study Guide for Essentials of Statistics for the Behavioral Sciences[M]. Worth Publishers, 2013

[212] Richard M Royall. The effect of sample size on the meaning of significance

tests[J]. The Amer Statist, 1986, **40**(4):313–315

[213] Liqun Chen, Hoon Wei Lim, Guomin Yang. Cross-domain password-based authenticated key exchange revisited[J]. ACM Trans Inform Syst Secur, 2014, **16**(4):1–37

[214] Xinyi Huang, Yang Xiang, Elisa Bertino, Jianying Zhou, Li Xu. Robust Multi-Factor Authentication for Fragile Communications[J]. IEEE Trans Depend Secur Comput, 2014, **11**(6):568–581

[215] Stanislaw Jarecki, Hugo Krawczyk, Maliheh Shirvanian, Nitesh Saxena. Device-Enhanced Password Protocols with Optimal Online-Offline Protection[C]. Proc. ASIACCS 2016, pp. 177–188

[216] Ali Bagherzandi, Stanislaw Jarecki, Nitesh Saxena, Yanbin Lu. Password-protected secret sharing[C]. Proc. ACM CCS 2011, pp. 433–444

[217] Jonathan Katz, Vinod Vaikuntanathan. Round-optimal password-based authenticated key exchange[J]. J Crypt, 2013, **26**(4):714–743

[218] 魏福山, 张刚, 马建峰, 马传贵. 标准模型下隐私保护的多因素密钥交换协议[J]. 软件学报, 2016, **27**(6):1511–1522

[219] Mihir Bellare. Practice-oriented provable-security[C]. Proc. ISC 1997, pp. 221–231

[220] Fabrice Benhamouda, Olivier Blazy, Céline Chevalier, David Pointcheval, Damien Vergnaud. New Techniques for SPHFs and Efficient One-Round PAKE Protocols. Proc. CRYPTO 2013, pp. 449–475

[221] Jin Wook Byun. Privacy preserving smartcard-based authentication system with provable security[J]. Securi Commun Netw, 2015, **8**(17):3028–3044

[222] 1st NSA Annual Best Scientific Cybersecurity Paper Competition[EB/OL], July 2013. `http://cps-vo.org/group/sos/papercompetition2012`

[223] Zhicong Huang, Erman Ayday, Jacques Fellay, Jean Hubaux, Ari Juels. GenoGuard: Protecting genomic data against brute-force attacks[C]. Proc. IEEE S&P 2015, pp. 447–462

[224] Masayuki Fukumitsu, Shingo Hasegawa, Jun Iwazaki, et al. A Proposal of a Password Manager Satisfying Security and Usability by Using the Secret Sharing and a Personal Server[C]. Proc. IEEE AINA 2016, pp. 661–668

[225] All Data Breach Sources[EB/OL], Nov. 2016. `https://breachalarm.com/all-sources`

[226] Now it's easy to see if leaked passwords work on other sites[EB/OL], July 2016. `http://bit.ly/29AJANh`

[227] Sarah Achappell. Turkey: personal data of 50 million citizens leaked online[EB/OL], April 2016. http://bit.ly/1TPA4j4

[228] Nearly 80 percent of Internet users suffer identity leaks[EB/OL], July 2015. http://bit.ly/2b9TEdn

[229] Thu Pham. Four Years Later, Anthem Breached Again: Hackers Stole Credentials[EB/OL], Feb. 2015. http://duo.sc/2ene0Pr

[230] Markus Jakobsson, Mayank Dhiman. The Benefits of Understanding Passwords[C]. Proc. HotSec 2012, pp. 1–6

[231] Abdelberi Chaabane, Gergely Acs, Mohamed Kaafar. You are what you like! information leakage through users' interests[C]. Proc. NDSS 2012, pp. 1–15

[232] Erika McCallister, Tim Grance, Karen Scarfone. NIST SP800-122: Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)[R]. Tech. rep., NIST, Reston, VA, 2010

[233] Senate Bill No. 1386: Personal information[EB/OL], Sep. 2002. http://bit.ly/1WJIIpK

[234] Ding Wang, Debiao He, Haibo Cheng, Ping Wang. fuzzyPSM: A New Password Strength Meter Using Fuzzy Probabilistic Context-Free Grammars[C]. Proc. IEEE/IFIP DSN 2016, pp. 595–606. http://bit.ly/2ahJ8CO

[235] Gigya Inc. Survey Guide: Businesses Should Begin Preparing for the Death of the Password[EB/OL], March 2016. http://info.gigya.com/rs/672-YBF-078/images/original-201603_gigya_wp_businesses_preparing_death_password-web4.pdf

[236] Jeremiah Onaolapo, Enrico Mariconti, Gianluca Stringhini. What Happens After You Are Pwnd: Understanding The Use Of Leaked Account Credentials In The Wild[C]. Proc. ACM IMC 2016, pp. 65–79

[237] C. Custer. China now has 656m mobile web users, and 710m total internet users[EB/OL], Aug. 2016. https://www.techinasia.com/china-656m-mobile-web-users-710m-total-internet-users

[238] William Melicher, Blase Ur, Sean Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, Lorrie Cranor. Fast, lean and accurate: Modeling password guessability using neural networks[C]. Proc. USENIX SEC 2016, pp. 1–17

[239] Rahul Chatterjee, Anish Athalye, Devdatta Akhawe, Ari Juels, Thomas Ristenpart. Password typos and How to Correct Them Securely[C]. Proc. IEEE S&P 2016, pp. 799–818

[240] Charles Weir. Cracking the MySpace List–First Impressions[EB/OL], July 2016. Http://reusablesec.blogspot.com/2016/07/cracking-myspace-list-first-impressions.html

[241] Mohit Kumar. Hey, Music Lovers! Last.Fm Hack Leaks 43 Million Account Passwords[EB/OL], Sep. 2016. http://thehackernews.com/2016/09/lastfm-hacked.html

[242] Steve Ragan. Weebly data breach affects 43 million customers[EB/OL], Oct. 2016. http://bit.ly/2kP4EA2

[243] Markus Dürmuth, Thorsten Kranz. On Password Guessing with GPUs and FPGAs[C]. Proc. Password 2014, pp. 19–38

[244] Jeffrey Goldberg. Bcrypt is great, but is password cracking infeasible?[EB/OL], Mar. 2015. Https://blog.agilebits.com/2015/03/30/bcrypt-is-great-but-is-password-cracking-infeasible/

[245] Mohammed H Almeshekah, Christopher N Gutierrez, Mikhail J Atallah, Eugene H Spafford. ErsatzPasswords: Ending Password Cracking and Detecting Password Leakage[C]. Proc. ACSAC 2015, pp. 311–320

[246] Jan Camenisch, Robert R Enderlein, Gregory Neven. Two-Server Password-Authenticated Secret Sharing UC-Secure Against Transient Corruptions. Proc. PKC 2015, Springer, 2015, pp. 283–307

[247] Jan Camenisch, Anja Lehmann, Gregory Neven. Optimal Distributed Password Verification[C]. Proc. ACM CCS 2015, pp. 182–194

[248] Imran Erguler. Achieving Flatness: Selecting the Honeywords from Existing User Passwords[J]. IEEE Trans Depend Secur Comput, 2016, **13**(2):284–295

[249] Nilesh Chakraborty, Samrat Mondal. Few notes towards making honeyword system more secure and usable[C]. Proc. ACM SIN 2015, pp. 237–245

[250] Nilesh Chakraborty, Samrat Mondal. On designing a modified-UI based honeyword generation approach for overcoming the existing limitations[J]. Comput & Secur, 2017, **66**:155–168

[251] Jeff Goldman. Chinese Hackers Publish 20 Million Hotel Reservations[EB/OL], Dec. 2013. http://bit.ly/2aVKyBw

[252] F. Adowsett. What has been leaked: impacts of the big data breaches[EB/OL], April 2016. Http://bit.ly/2lTOypH

[253] Swati Khandelwal. Sony Pictures Scarier Hack – Hackers Leak Scripts, Celebrity Phone Numbers and Aliases[EB/OL], Dec. 2014. http://thehackernews.com/2014/12/sony-pictures-scarier-hack-h

ackers-leak.html

[254] Lisa Vaas. Change your password! Yahoo confirms data breach of 500 million accounts[EB/OL], Sep. 2016. http://bit.ly/2dE43zR

[255] Mandy Zuo. Widespread online leaks of personal information in China prompt new data collection laws[EB/OL], May 2016. http://bit.ly/2dA09Zh

[256] William Melicher, Blase Ur, Sean M Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor. Fast, lean and accurate: Modeling password guessability using neural networks[C]. Proc. Usenix SEC 2016, pp. 175–191

[257] Varun Haran. Qatar National Bank Suffers Massive Breach[EB/OL], April 2016. Http://www.bankinfosecurity.com/qatar-national-bank-suffers-massive-breach-a-9068

[258] Hristo Bojinov, Elie Bursztein, Xavier Boyen, Dan Boneh. Kamouflage: Loss-resistant password management[C]. Proc. ESORICS 2010, pp. 286–302

[259] How Many Times Has Your Personal Information Been Exposed to Hackers?[EB/OL], Sep. 2016. http://nyti.ms/1SdFv0s

[260] David Bisson. Ubuntu Forums Hack Exposed 2M Users' Information[EB/OL], July 2016. http://www.tripwire.com/state-of-security/latest-security-news/ubuntu-forums-hack-potentially-compromised-2m-users-information/

[261] Eduard Kovacs. PhpBB Asking Users to Change Passwords Following Hack[EB/OL], Dec. 2014. Http://www.securityweek.com/phpbb-asking-users-change-passwords-following-hack

[262] Maximilian Golla, Benedict Beuscher, Markus Dürmuth. On the Security of Cracking-Resistant Password Vaults[C]. Proc. CCS 2016, pp. 1230–1241

[263] Robert W Proctor, Mei Ching Lien, Kim Phuong L Vu, E Eugene Schultz, Gavriel Salvendy. Improving computer security for authentication of users: Influence of proactive password restrictions[J]. Behav Res Meth Instr & Comput, 2002, **34**(2):163–169

[264] Richard Shay, Saranga Komanduri, Patrick Kelley, et al. Encountering stronger password requirements: user attitudes and behaviors[C]. Proc. SOUPS 2010, pp. 1–20

[265] Patrick Gage Kelley, Saranga Komanduri, Michelle L Mazurek, Richard

Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Julio Lopez. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms[C]. Proc. IEEE S&P 2012, pp. 523–537

[266] Benjamin Strahs, Chuan Yue, Haining Wang. Secure Passwords Through Enhanced Hashing[C]. Proc. LISA 2009, pp. 93–106

[267] Mansour Alsaleh, Mohammad Mannan, P Van Oorschot. Revisiting defenses against large-scale online password guessing attacks[J]. IEEE Trans Depend Secur Comput, 2012, **9**(1):128–141

[268] SéRgio SC Silva, Rodrigo MP Silva, Raquel CG Pinto, Ronaldo M Salles. Botnets: A survey[J]. Comput Netw, 2013, **57**(2):378–403

[269] Dan Wheeler. zxcvbn: realistic password strength estimation[EB/OL], April 2012. Https://blogs.dropbox.com/tech/2012/04/zxcvbn-realistic-password-strength-estimation/

[270] Robert Graham. PHPbb Password Analysis[EB/OL], June 2009. `http://www.darkreading.com/risk/phpbb-password-analysis/d/d-id/1130335?`

[271] Edward E Cureton. The average spearman rank criterion correlation when ties are present[J]. Psychometrika, 1958, **23**(3):271–272

[272] Leta Adler. A modification of Kendall's tau for the case of arbitrary ties in both rankings[J]. J Amer Statist Assoc, 1957, **52**(277):33–35

[273] Omprakash Gnawali, Ki Young Jang, Jeongyeup Paek, Marcos Vieira, Ramesh Govindan, Ben Greenstein, August Joki, Deborah Estrin, Eddie Kohler. The tenet architecture for tiered sensor networks[C]. Proc. ACM SenSys 2006, pp. 153–166

[274] Jing Shi, Rui Zhang, Yanchao Zhang. Secure range queries in tiered sensor networks[C]. Proc. INFOCOM 2009, pp. 945–953

[275] Dejun Yang, Satyajayant Misra, Xi Fang, Guoliang Xue, Junshan Zhang. Two-tiered constrained relay node placement in wireless sensor networks: computational complexity and efficient approximations[J]. IEEE Trans Mobile Comput, 2012, **11**(8):1399–1411

[276] Wensheng Zhang, Hui Song, Sencun Zhu, Guohong Cao. Least privilege and privilege deprivation: towards tolerating mobile sink compromises in wireless sensor networks[C]. Proc. MobiHoc 2005, pp. 378–389

[277] Daojing He, Jiajun Bu, Sencun Zhu, Sammy Chan, Chun Chen. Distributed access control with privacy support in wireless sensor networks[J]. IEEE

Trans Wireless Commun, 2011, **10**(10):3472–3481

[278] Guomin Yang, Duncan S. Wong, Huaxiong Wang, Xiaotie Deng. Two-factor mutual authentication based on smart cards and passwords[J]. J Comput Syst Sci, 2008, **74**(7):1160–1172

[279] Ding Wang, Ping Wang. Offline Dictionary Attack on Password Authentication Schemes using Smart Cards. Proc. ISC 2013, pp. 221–237

[280] Steven J Murdoch, Saar Drimer, Ross Anderson, Mike Bond. Chip and PIN is Broken[C]. Proc. IEEE S&P 2010, pp. 433–446

[281] Nikos Mavrogiannopoulos, Andreas Pashalidis, Bart Preneel. Security implications in Kerberos by the introduction of smart cards[C]. Proc. ACM ASIACCS 2012, pp. 59–63

[282] Manik Lal Das. Two-factor user authentication in wireless sensor networks[J]. IEEE Trans Wireless Commun, 2009, **8**(3):1086–1090

[283] Office of the Science Advisor. Taking the Pulse of the Planet: EPA' s Remote Sensing Information Gateway[EB/OL]. `http://www.epa.gov/geoss`

[284] The National Oceanographic Partnership Program (NOPP)[EB/OL]. `http://www.nopp.org/`

[285] Kumar Mangipudi, Rajendra Katti. A secure identification and key agreement protocol with user anonymity (SIKA)[J]. Comput Secur, 2006, **25**(6):420–425

[286] Jinyuan Sun, Chi Zhang, Yanchao Zhang, Yuguang Fang. SAT: A security architecture achieving anonymity and traceability in wireless mesh networks[J]. IEEE Trans Depend Secur Comput, 2011, **8**(2):295–307

[287] Chengzhe Lai, Hui Li, Xiaohui Liang, Rongxin Lu, Kuan Zhang, Xuemin Shen. CPAL: A Conditional Privacy-Preserving Authentication with Access Linkability for Roaming Service[J]. IEEE Internet of Things J, 2014, **1**(1):2327–4662

[288] Dominic Hughes, Vitaly Shmatikov. Information hiding, anonymity and privacy: a modular approach[J]. J Comput Secur, 2004, **12**(1):3–36

[289] Xiangxue Li, Weidong Qiu, Dong Zheng, Kefei Chen, Jianhua Li. Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards[J]. IEEE Trans Ind Elec, 2010, **57**(2):793–800

[290] Ding Wang, Chunguang Ma. Cryptanalysis of a remote user authentication scheme for mobile client-server environment based on ECC[J]. Inform

Fusion, 2013, **14**(4):498–503

[291] Daojing He, Yi Gao, Sammy Chan, Chun Chen, Jiajun Bu. An enhanced two-factor user authentication scheme in wireless sensor networks[J]. Ad Hoc & Sensor Wireless Netw, 2010, **10**(4):361–371

[292] Muhammad Khurram Khan, Khaled Alghathbar. Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'[J]. Sensors, 2010, **10**(3):2450–2459

[293] Pardeep Kumar, Amlan Jyoti Choudhury, Mangal Sain, Sang Gon Lee, Hoon Jae Lee. RUASN: A Robust User Authentication Framework for Wireless Sensor Networks[J]. Sensors, 2011, **11**(5):5020–5046

[294] Jiang Qi, MA Zhuo, Ma Jianfeng, Li Guangsong. Security Enhancement of Robust User Authentication Framework for Wireless Sensor Networks[J]. China Commun, 2012, **9**(10):103–111

[295] Kaiping Xue, Changsha Ma, Peilin Hong, Rong Ding. A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks[J]. J Netw Comput Appl, 2013, **36**(1):316–323

[296] Chengzhe Lai, Hui Li, Rongxing Lu, Xuemin Sherman Shen. SE-AKA: A secure and efficient group authentication and key agreement protocol for LTE networks[J]. Comput Netw, 2013, **57**(17):3492–3510

[297] DaeHun Nyang, Mun Kyu Lee. Improvement of Das's Two-Factor Authentication Protocol in Wireless Sensor Networks[EB/OL], 2009. `http://eprint.iacr.org/2009/631.pdf`

[298] Tien Ho Chen, Wei Kuan Shih. A robust mutual authentication protocol for wireless sensor networks[J]. ETRI J, 2010, **32**(5):704–712

[299] Hsiu Lien Yeh, Tien Ho Chen, Pin Chuan Liu, Tai Hoo Kim, Hsin Wen Wei. A secured authentication protocol for wireless sensor networks using elliptic curves cryptography[J]. Sensors, 2011, **11**(5):4767–4779

[300] Dazhi Sun, Jianxin Li, Zhiyong Feng, Zhenfu Cao, Guangquan Xu. On the security and improvement of a two-factor user authentication scheme in wireless sensor networks[J]. Pers Ubiquit Comput, 2013, **17**(5):895–905

[301] Ashok Kumar Das, Pranay Sharma, Santanu Chatterjee, Jamuna Kanta Sing. A dynamic password-based user authentication scheme for hierarchical wireless sensor networks[J]. J Netw Comput Appl, 2012, **35**(52):1646–1656

[302] Jian Jun Yuan. An enhanced two-factor user authentication in wireless sensor networks[J]. Telecommun Syst, 2014, **55**(1):105–113

[303] Binod Vaidya, Dimitrios Makrakis, Hussein Mouftah. Two-factor mutual authentication with key agreement in wireless sensor networks[J]. Secur Commun Netw, 2016, **9**(2):171–183

[304] Rong Fan, Dao-jing He, Xue zeng Pan, Ling di Ping. An efficient and DoS-resistant user authentication scheme for two-tiered wireless sensor networks[J]. J Zhejinag Univ-Sci C, 2011, **12**(7):550–560

[305] M Turkanovic, M Holbl. An Improved Dynamic Password-based User Authentication Scheme for Hierarchical Wireless Sensor Networks[J]. Electron & Electr Eng, 2013, **19**(6):109–116

[306] Chun Ta Li, Chi Yao Weng, Cheng Chi Lee, Chin Wen Lee. Towards Secure and Dynamic Password Based User Authentication Scheme in Hierarchical Wireless Sensor Networks[J]. Int J Secur & Its Appl, 2013, **7**(3):249–258

[307] Madeline González Muñiz, Peeter Laud. On the (im) possibility of perennial message recognition protocols without public-key cryptography[C]. Proc. ACM SAC 2011, pp. 1510–1515

[308] DanielR. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? Proc. EUROCRYPT 1998, vol. 1403 of LNCS, pp. 334–345

[309] Michael Backes, Birgit Pfitzmann. Limits of the Cryptographic Realization of Dolev-Yao-Style XOR. Proc. ESORICS 2005, vol. 3679 of LNCS, pp. 178–196

[310] Ahto Buldas, Aivo Jürgenson. Does Secure Time-Stamping Imply Collision-Free Hash Functions? Proc. ProvSec 2007, vol. 4784 of LNCS, pp. 138–150

[311] Russell Impagliazzo, Steven Rudich. Limits on the provable consequences of one-way permutations[C]. Proc. STOC 1989, pp. 44–61

[312] DongGook Park, Colin Boyd, Sang-Jae Moon. Forward Secrecy and Its Application to Future Mobile Communications Security. Proc. PKC 2000, vol. 1751 of LNCS, pp. 433–445

[313] Minh Huyen Nguyen. The Relationship Between Password-Authenticated Key Exchange and Other Cryptographic Primitives. Proc. TCC 2006, vol. 3378 of LNCS, pp. 457–475

[314] Peng Zeng, Zhenfu Cao, Kim-kwang Choo, Shengbao Wang. On the anonymity of some authentication schemes for wireless communications[J]. IEEE Commun Lett, 2009, **13**(3):170–171

[315] Jianming Zhu, Jianfeng Ma. A new authentication scheme with anonymi-

ty for wireless environments[J]. IEEE Trans Consum Electron, 2004, **50**(1):231–235

[316] Cheng chi Lee, Min shiang Hwang, I En Liao. Security enhancement on a new authentication scheme with anonymity for wireless environments[J]. IEEE Trans Ind Elec, 2006, **53**(5):1683–1687

[317] Chia Chun Wu, Wei Bin Lee, Woei Jiunn Tsaur. A secure authentication scheme with anonymity for wireless communications[J]. IEEE Commun Lett, 2008, **12**(10):722–723

[318] Ruhul Amin, GP Biswas. A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks[J]. Ad Hoc Netw, 2016, **36**:58–80

[319] Prosanta Gope, Tzonelih Hwang. A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks[J]. IEEE Trans Ind Electro, 2016, **63**(11):7124–7132

[320] Jaewook Jung, Jiye Kim, Younsung Choi, Dongho Won. An Anonymous User Authentication and Key Agreement Scheme Based on a Symmetric Cryptosystem in Wireless Sensor Networks[J]. Sensors, 2016, **16**(8):1–30

[321] Mohammad Sabzinejad Farash, Muhamed Turkanović, Saru Kumari, Marko Hölbl. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment[J]. Ad Hoc Netw, 2016, **36**:152–176

[322] Chun Ta Li, Chi Yao Weng, Cheng Chi Lee. An Advanced Temporal Credential-Based Security Scheme with Mutual Authentication and Key Agreement for WSNs[J]. Sensors, 2013, **13**(8):9589–9603

[323] Pardeep Kumar, Sang Gon Lee, Hoon Jae Lee. E-SAP: Efficient-Strong Authentication Protocol for Healthcare Applications Using Wireless Medical Sensor Networks[J]. Sensors, 2012, **12**(2):1625–1647

[324] Thomas S Messerges, Ezzat Dabbish, Robert H Sloan. Examining smart-card security under the threat of power analysis attacks[J]. IEEE Trans Comput, 2002, **51**(5):541–552

[325] Konstantinos Markantonakis, Michael Tunstall, Gerhard Hancke, Ioannis Askoxylakis, Keith Mayes. Attacking smart card systems: Theory and practice[J]. Inform Secur Tech Rep, 2009, **14**(2):46–56

[326] Tae Hyun Kim, ChangKyun Kim, IlHwan Park. Side channel analysis

attacks using AM demodulation on commercial smart cards with SEED[J]. J Syst Soft, 2012, **85**(12):2899 – 2908

[327] Julien Lancia. Java Card Combined Attacks with Localization-Agnostic Fault Injection. Proc. CARDIS 2013, pp. 31–45

[328] Mohammad Sabzinejad Farash, Shehzad Ashraf Chaudhry, Mohammad Heydari, Sajad Sadough, S Mohammad, Saru Kumari, Muhammad Khurram Khan. A lightweight anonymous authentication scheme for consumer roaming in ubiquitous networks with provable security[J]. Int J Communi Syst, 2017, **30**(4):1–14

[329] Huixian Li, Yafang Yang, Liaojun Pang. An efficient authentication protocol with user anonymity for mobile networks[C]. Proc. IEEE WCNC 2013, pp. 1842–1847

[330] Muhamed Turkanović, Boštjan Brumen, Marko Hölbl. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion[J]. Ad Hoc Networks, 2014, **20**:96–112

[331] Chin Chen Chang, Ting Fang Cheng, Hsiao Ling Wu. An authentication and key agreement protocol for satellite communications[J]. Int J Commun Syst, 2014, **27**(10):1994—2006

[332] Soobok Shin, Hongjin Yeh, Kangseok Kim. An efficient secure authentication scheme with user anonymity for roaming user in ubiquitous networks[J]. Peer-to-Peer Netw Appl, 2015, **8**(4):674–683

[333] Feng Bao, Robert Deng. Privacy Protection for Transactions of Digital Goods. Proc. ICICS 2001, pp. 202–213

[334] Manik Lal Das, Ashutosh Saxena, Ved P. Gulati. A dynamic ID-based remote user authentication scheme[J]. IEEE Trans Comsum Electr, 2004, **50**(2):629–631

[335] Yi Pin Liao, Shuenn Shyang Wang. A secure dynamic ID based remote user authentication scheme for multi-server environment[J]. Comput Stand Inter, 2009, **31**(1):24–29

[336] Sheng Zhong, Fan Wu. A collusion-resistant routing scheme for noncooperative wireless ad hoc networks[J]. IEEE/ACM Trans Netw, 2010, **18**(2):582–595

[337] Qingjun Xiao, Kai Bu, Zhijun Wang, Bin Xiao. Robust localization against outliers in wireless sensor networks[J]. ACM Trans Sensor Netw, 2013,

9(2):24

[338] Danny Dolev, Andrew Yao. On the security of public key protocols[J]. IEEE Trans Inform Theory, 1983, **29**(2):198–208

[339] Shamus Software Ltd. Miracl library[EB/OL], May 2013. `http://www.shamus.ie/index.php?page=home`

[340] Joseph Bonneau, Mike Just, Greg Matthews. What' s in a Name? Proc. FC 2010, pp. 98–113

[341] Hung Min Sun, Wei Chih Ting, King Hang Wang. On the security of Chien's ultralightweight RFID authentication protocol[J]. IEEE Trans Depend Secur Comput, 2011, **8**(2):315–317

[342] Huaqun Wang, Yuqing Zhang. On the Security of a Ticket-Based Anonymity System with Traceability Property in Wireless Mesh Networks[J]. IEEE Trans Depend Secur Comput, 2012, **9**(3):443–446

[343] Gui Lin Wang, Jiang Shan Yu, Qi Xie. Security Analysis of a Single Sign-On Mechanism for Distributed Computer Networks[J]. IEEE Trans Ind Inform, 2013, **9**(1):294–302

[344] Muhammad Khurram Khan, Debiao He. A new dynamic identity-based authentication protocol for multi-server environment using elliptic curve cryptography[J]. Secur Commun Netw, 2012, **5**(11):1260–1266

[345] Ding Wang, Qianchen Gu, Haibo Cheng, Ping Wang. The Request for Better Measurement: A Comparative Evaluation of Two-Factor Authentication Schemes[C]. Proc. ACM ASIACCS 2016, pp. 475–486

[346] Ding Wang, Ping Wang. Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks[J]. Ad Hoc Netw, 2014, **20**:1–15

[347] Chun-Guang Ma, Ding Wang, Sen-Dong Zhao. Security flaws in two improved remote user authentication schemes using smart cards[J]. Int J Commun Syst, 2014, **27**(10):2215–2227

[348] Ding Wang, Chun guang Ma, De li Gu, Zhen shan Cui. Cryptanalysis of Two Dynamic ID-Based Remote User Authentication Schemes for Multi-server Architecture. Proc. NSS 2012, pp. 462–475

[349] Pardeep Kumar, Andrei Gurtov, Mika Ylianttila, Sang Gon Lee, HoonJae Lee. A Strong Authentication Scheme with User Privacy for Wireless Sensor Networks.[J]. ETRI J, 2013, **35**(5):889–899

[350] Fan Wu, Lili Xu. Security analysis and Improvement of a Privacy

Authentication Scheme for Telecare Medical Information Systems[J]. J Med Syst, 2013, **37**(11):1–9

[351] Muhammad Khurram Khan, Saru Kumari. Cryptanalysis and Improvement of "An Efficient and Secure Dynamic ID-based Authentication Scheme for Telecare Medical Information Systems" [J]. Secur Comm Netw, 2014, **7**(2):399–408

[352] Fengtong Wen, Xuelei Li. An improved dynamic ID-based remote user authentication with key agreement scheme[J]. Comput & Electr Eng, 2012, **38**(2):381–387

[353] Ming Chin Chuang, Jeng Farn Lee, Meng Chang Chen. SPAM: A Secure Password Authentication Mechanism for Seamless Handover in Proxy Mobile IPv6 Networks[J]. IEEE Syst J, 2013, **7**(1):102–113

[354] Qi Xie, Duncan S Wong, Guilin Wang, Xiao Tan, Kefei Chen, Liming Fang. Provably Secure Dynamic ID-Based Anonymous Two-Factor Authenticated Key Exchange Protocol With Extended Security Model[J]. IEEE Trans Inform Foren Secur, 2017, **12**(6):1382–1392

[355] Alavalapati Goutham Reddy, Eun Jun Yoon, Ashok Kumar Das, Vanga Odelu, Kee Young Yoo. Design of Mutually Authenticated Key Agreement Protocol Resistant to Impersonation Attacks for Multi-Server Environment[J]. IEEE Access, 2017, (5):3622–3639

[356] Chandra Sekhar Vorugunti, Bharavi Mishra, Ruhul Amin, et al. Improving Security of Lightweight Authentication Technique for Heterogeneous Wireless Sensor Networks[J]. Wirel Personal Commun:1–26. Doi:10.1007/s11277-017-3988-7

[357] Ding Wang, Ping Wang, Jing Liu. Improved Privacy-Preserving Authentication Scheme for Roaming Service in Mobile Networks[C]. Proc. WCNC 2014, pp. 3178–3183

[358] Fengtong Wen, Willy Susilo, Guomin Yang. A Secure and Effective Anonymous User Authentication Scheme for Roaming Service in Global Mobility Networks[J]. Wireless Pers Commun, 2013, **73**(3):993–1004

[359] Chun Chen, Daojing He, Sammy Chan, Jiajun Bu, Rong Fan. Lightweight and provably secure user authentication with anonymity for the global mobility network[J]. Int J Commun Syst, 2011, **24**(3):347–362

[360] Miyoung Kang, Hyun Sook Rhee, Jin Young Choi. Improved user authentication scheme with user anonymity for wireless communications[J]. IEICE

Trans Fund Electron Commun Comput Sci, 2011, **94**(2):860–864

[361] Xiong Li, Jian Ma, Wendong Wang, Junsong Zhang. A novel smart card and dynamic ID based remote user authentication scheme for multi-server environments[J]. Math Comput Model, 2013, **58**(2):85–95

[362] Jiye Kim, Donghoon Lee, Woongryul Jeon, Youngsook Lee, Dongho Won. Security Analysis and Improvements of Two-Factor Mutual Authentication with Key Agreement in Wireless Sensor Networks[J]. Sensors, 2014, **14**(4):6443–6462

[363] Saru Kumari, Muhammad Khurram Khan, Xiong Li. An improved remote user authentication scheme with key agreement[J]. Comput & Electr Eng, 2014, **40**(6):97–112

[364] Xavier Leroy. Smart card security from a programming language and static analysis perspective[EB/OL], 2013. `http://pauillac.inria.fr/~xleroy/talks/language-security-etaps03.pdf`

[365] Dazhi Sun, Jinpeng Huai, Jizhou Sun, Jianxin Li, Zhiyong Feng. Improvements of Juang et al.'s Password-Authenticated Key Agreement Scheme Using Smart Cards[J]. IEEE Trans Ind Elec, 2009, **56**(6):2284–2291

[366] Tao Zhou, Jing Xu. Provable secure authentication protocol with anonymity for roaming service in global mobility networks[J]. Comput Netw, 2011, **55**(1):205–213

[367] Daojing He, Chun Chen, Jiajun Bu, Sammy Chan, Yan Zhang. Security and efficiency in roaming services for wireless networks: challenges, approaches, and prospects[J]. IEEE Commun Mag, 2013, **51**(2):142–150

[368] Daojing He, Chun Chen, Sammy Chan, Jiajun Bu. Secure and efficient handover authentication based on bilinear pairing functions[J]. IEEE Trans Wireless Commun, 2012, **11**(1):48–53

[369] David Chaum, Eugène Heyst. Group Signatures. Proc. EUROCRYPT 1991, pp. 257–265

[370] Jian Ren, Lein Harn. An Efficient Threshold Anonymous Authentication Scheme for Privacy-Preserving Communications[J]. IEEE Trans Wireless Commun, 2013, **12**(3):1018–1025

[371] Jing Xu, Wen Tao Zhu. A Generic Framework for Anonymous Authentication in Mobile Networks[J]. J Comput Sci Tech, 2013, **28**(4):732–742

[372] Ding Wang, Nan Wang, Ping Wang, Sihan Qing. Preserving privacy for free: efficient and provably secure two-factor authentication scheme with

user anonymity[J]. Inform Sci, 2015, **321**:162–178

[373] Kyungho Son, Dong Guk Han, Dongho Won. A Privacy-Protecting Authentication Scheme for Roaming Services with Smart Cards[J]. IEICE Trans Commun, 2012, **95**(5):1819–1821

[374] Eiichiro Fujisaki, Tatsuaki Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. Proc. CRYPTO 1999, pp. 537–554

[375] Mihir Bellare, David Pointcheval, Phillip Rogaway. Authenticated Key Exchange Secure against Dictionary Attacks. Proc. EUROCRYPT 2000, pp. 139–155

[376] Dan Boneh. Simplified OAEP for the RSA and Rabin functions. Joe Kilian, (Editor) Proc. CRYPTO 2001, Springer-Verlag, 2001, vol. 2139 of LNCS, pp. 275–291

[377] Ran Canetti, Oded Goldreich, Shai Halevi. The random oracle methodology, revisited[J]. J ACM, 2004, **51**(4):557–594

[378] Shih Ying Chang, Yue Hsun Lin, Hung Min Sun, Mu En Wu. Practical RSA signature scheme based on periodical rekeying for wireless sensor networks[J]. ACM Trans Sen Netw, 2012, **8**(2):13

[379] Arvinderpal S Wander, Nils Gura, Hans Eberle, Vipul Gupta, Sheueling Chang Shantz. Energy analysis of public-key cryptography for wireless sensor networks[C]. Proc. PerCom 2005. IEEE, 2005, pp., 324–328

[380] Michael Scott, Neil Costigan, Wesam Abdulwahab. Implementing cryptographic pairings on smartcards[C]. Proc. CHES 2006, pp. 134–147

[381] Piotr Szczechowiak, Anton Kargl, Michael Scott, Martin Collier. On the application of pairing based cryptography to wireless sensor networks[C]. Proc. ACM WiSec 2009, pp. 1–12

[382] Kuo Hui Yeh, Chunhua Su, N. W. Lo. Two robust remote user authentication protocols using smart cards[J]. J Syst Soft, 2010, **83**(12):2556–2565

[383] Toan Thinh Truong, Minh Triet Tran, Anh Duc Duong, Isao Echizen. Chaotic Chebyshev Polynomials Based Remote User Authentication Scheme in Client-Server Environment[C]. Proc. IFIP SEC 2015, pp. 479–494

[384] Qi Jiang, Jianfeng Ma, Guangsong Li, Xinghua Li. Improvement of robust smart-card-based password authentication scheme[J]. Int J Commun Syst, 2015, **28**(2):383–393

[385] Yanrong Lu, Lixiang Li, Haipeng Peng, Yixian Yang. Robust anonymous two-factor authenticated key agreement schemem for mobile client-server

environment[J]. Secur Commun Netw, 2016, **9**(11):1331–1339

[386] SK Islam. Design and analysis of an improved smartcard-based remote user password authentication scheme[J]. Int J Commun Syst, 2016, **29**(11):1708–1719

[387] Mohammad Masdari, Safiyyeh Ahmadzadeh. A survey and taxonomy of the authentication schemes in Telecare Medicine Information Systems[J]. J Netw Comput Appl, 2017, **87**:1–19

[388] I En Liao, Cheng Chi Lee, Min Shiang Hwang. A password authentication scheme over insecure networks[J]. J Comput Syst Sci, 2006, **72**(4):727–740

[389] Chwei Shyong Tsai, Cheng Chi Lee, Min Shiang Hwang. Password authentication schemes: current status and key issues[J]. Int J Netw Securi, 2006, **3**(2):101–115

[390] Ren Chiun Wang, Wen Sheng Juang, Chin Laung Lei. Robust authentication and key agreement scheme preserving the privacy of secret key[J]. Comput Commun, 2011, **34**(3):274–280

[391] Joseph Bonneau, Cormac Herley, Paul C Oorschot, Frank Stajano. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes[C]. Proc. IEEE S&P 2012, pp. 553–567

[392] Cormac Herley, Paul Van. A research agenda acknowledging the persistence of passwords[J]. IEEE Secur Priv, 2012, **10**(1):28–36

[393] Thomas Wu. The Secure Remote Password Protocol[C]. Proc. NDSS 1998, pp. 1–15

[394] Thomas Fox Brewster. 15 Million Passwords Appear To Have Leaked From 000webhost[EB/OL], Oct. 2015. `http://www.forbes.com/sites/thomasbrewster/2015/10/28/000webhost-database-leak/`

[395] Daphna Weinshall. Cognitive authentication schemes safe against spyware[C]. Proc. IEEE S&P 2006, pp. 295–230

[396] Qiang Yan, Jin Han, Yingjiu Li, Robert H Deng. On limitations of designing leakage-resilient password systems: Attacks, principles and usability[C]. Proc. NDSS 2012, pp. 1–16

[397] Xinyi Huang, Xiaofeng Chen, Jin Li, Yang Xiang, Li Xu. Further observations on smart-card-based password-authenticated key agreement in distributed systems[J]. IEEE Trans Para Distrib Syst, 2014, **25**(7):1767–1775

[398] Shuhua Wu, Yuefei Zhu, Qiong Pu. Robust smart-cards-based user

authentication scheme with user anonymity[J]. Secur Commun Netw, 2012, **5**(2):236–248

[399] Chun-I Fan, Yung-Cheng Chan, Zhi-Kai Zhang. Robust remote authentication scheme with smart cards[J]. Comput Secur, 2005, **24**(8):619–628

[400] Jia Lun Tsai, Nai Wei Lo, Tzong Chen Wu. Novel Anonymous Authentication Scheme Using Smart Cards[J]. IEEE Trans Ind Inform, 2013, **9**(4):2004–2013

[401] Shehzad Ashraf Chaudhry, Mohammad Sabzinejad Farash, Husnain Naqvi, Saru Kumari, Muhammad Khurram Khan. An enhanced privacy preserving remote user authentication scheme with provable security[J]. Securi Commun Netw, 2015. Doi:10.1002/sec.1299

[402] Saru Kumari, Muhammad Khurram Khan. More secure smart card-based remote user password authentication scheme with user anonymity[J]. Secur Comm Netw, 2014, **7**(11):2039–2053

[403] Fahad T Bin Muhaya. Cryptanalysis and security enhancement of Zhu's authentication scheme for Telecare medicine information system[J]. Secur Commun Netw, 2015, **8**(2):149–158

[404] Chenyu Wang, Guoai Xu. Cryptanalysis of Three Password-based Remote User Authentication Schemes with Non-tamper Resistant smart card[J]. Secur Commun Netw, 2017. Doi: 10.1002/sec.817

[405] Yanyan Wang, Jiayong Liu, Fengxia Xiao, Jing Dan. A more efficient and secure dynamic ID-based remote user authentication scheme[J]. Comput Commun, 2009, **32**(4):583–585

[406] Saru Kumari, Muhammad Khurram Khan. Cryptanalysis and improvement of 'a robust smart-card-based remote user password authentication scheme'[J]. Int J Commun Syst, 2014, **27**(12):3939–3955

[407] Kyung Kug Kim, Myung Hwan Kim. An Enhanced Anonymous Authentication and Key Exchange Scheme Using Smartcard. Proc. ICISC 2012, vol. 7839 of LNCS, pp. 487–494,

[408] Junrong Liu, Yu Yu, Francois-Xavier Standaert. Small Tweaks do Not Help: Differential Power Analysis of MILENAGE Implementations in 3G/4G USIM Cards. Proc. ESORICS 2015, pp. 468–480

[409] Bae Ling Chen, Wen Kuo. Robust smart-card-based remote user password authentication scheme[J]. Int J Commun Syst, 2014, **27**(2):377–389

[410] Saru Kumari, Muhammad Khurram Khan. Cryptanalysis and improve-

ment of 'a robust smart-card-based remote user password authentication scheme'[J]. Int J Commun Syst, 2014, **27**(12):3939–3955

[411] Feng Hao. On Robust Key Agreement Based on Public Key Authentication. Proc. FC 2010, vol. 6052 of LNCS, pp. 383–390

[412] Ding Wang, Chunguang Ma, Ping Wu. Secure Password-Based Remote User Authentication Scheme with Non-tamper Resistant Smart Cards. Proc. DBSec 2012, Springer, 2012, vol. 7371 of LNCS, pp. 114–121

[413] Chun Ta Li, Cheng Chi Lee. A Robust Remote User Authentication Scheme Using Smart Card[J]. Info Tech Control, 2011, **40**(3):236–245

[414] Qi Xie. Dynamic ID-Based Password Authentication Protocol with Strong Security against Smart Card Lost Attacks. Proc. ICWCA 2012, pp. 412–418

[415] Jangirala Srinivas, Sourav Mukhopadhyay, Dheerendra Mishra. An Efficient Password Authentication Scheme for Smart Card.[J]. Ad Hoc Netw, 2017, **54**:147–169

[416] Emile Morse, Mary Theofanos, Y Choong, Celeste Paul, Aiping Zhang, Hannah Wald. NIST-IR-7867 Usability of PIV Smartcards for Logical Access[R]. Tech. rep., National Institute of Standards and Technology, McLean, VA, 2012. Doi:`http://dx.doi.org/10.6028/NIST.IR.7867`

[417] Alessandro Barenghi, Luca Breveglieri, Israel Koren, David Naccache. Fault Injection Attacks on Cryptographic Devices: Theory, Practice, and Countermeasures[J]. Proc of the IEEE, 2012, **100**(11):3056–3076

[418] Wen-Shenq Juang, Sian-Teng Chen, Horng-Twu Liaw. Robust and efficient password-authenticated key agreement using smart cards[J]. IEEE Trans Ind Elec, 2008, **55**(6):2551–2556

[419] Certicom Research. Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curve Domain Parameters, Version 2.0[EB/OL], Jan. 2010. Available at `http://www.secg.org/download/aid-784/sec2-v2.pdf`

[420] Junghyun Nam, Seungjoo Kim, Dongho Won. Security analysis of a nonce-based user authentication scheme using smart cards[J]. IEICE Trans Fund Electron Comm Comput Sci, 2007, **90**(1):299–302

[421] Chun Ta Li. A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card[J]. IET Inform Secur, 2013, **7**(1):3–10

[422] Xiong Li, Jianwei Niu, Muhammad Khurram Khan, Junguo Liao. An enhanced smart card based remote user password authentication scheme[J].

J Netw Comput Appl, 2013, **36**(5):1365–1371

[423] Ding Wang, Ping Wang, Chunguang Ma, Zhong Chen. Robust Smart Card based Password Authentication Scheme against Smart Card Security Breach[EB/OL], 2012. http://eprint.iacr.org/2012/439.pdf

[424] Dan's Data. Thinking Putty defeats Fingerprint Scanners![EB/OL], Sep. 2013. Available at http://www.puttyworld.com/thinputdeffi.html

[425] SK Hafizul Islam, G.P. Biswas. Design of improved password authentication and update scheme based on elliptic curve cryptography[J]. Math Comput Model, 2013, **57**(11-12):2703–2717

[426] Feng Bao. Security Can Only Be Measured by Attacks[EB/OL], 2008. http://wenku.baidu.com/view/ff14a31ea300a6c30c229f7a.html

[427] Dorothy E Denning, Giovanni Maria Sacco. Timestamps in key distribution protocols[J]. Commun ACM, 1981, **24**(8):533–536

[428] Shafi Goldwasser, Silvio Micali. Probabilistic encryption[J]. J Comput Syst Sci, 1984, **28**(2):270–299

[429] Ran Canetti, Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. Proc. EUROCRYPT 2001, vol. 2045 of LNCS, pp. 453–474

[430] Emmanuel Bresson, Olivier Chevassut, David Pointcheval. Security proofs for an efficient password-based key exchange[C]. Proc. ACM CCS 2003, pp. 41–50

[431] David Pointcheval, Sébastien Zimmer. Multi-factor authenticated key exchange[C]. Proc. ACNS 2008, pp. 277–295

[432] Santanu Chatterjee, Sandip Roy, Ashok Kumar Das, et al. Secure Biometric-Based Authentication Scheme using Chebyshev Chaotic Map for Multi-Server Environment[J]. IEEE Trans Depend Secure Comput, 2016. Doi: 10.1109/TDSC.2016.2616876

[433] Alfred Menezes. Another Look at Provable Security. Proc. EUROCRYPT 2012, vol. 7237 of LNCS, pp. 8–8. http://www.cs.bris.ac.uk/eurocrypt2012/Program/Weds/Menezes.pdf

[434] Neal Koblitz, Alfred J Menezes. Another look at "provable security"[J]. J Cryptology, 2007, **20**(1):3–37

[435] Tie Yan Li, Gui Lin Wang. Analyzing a Family of Key Protection Schemes against Modification Attacks[J]. IEEE Trans Depend Secur Comput, 2011, **8**(5):770–776

[436] Michael Scott. Cryptanalysis of a recent two factor authentication scheme[EB/OL], 2012. http://eprint.iacr.org/2012/527.pdf

[437] HemantaK. Maji, Manoj Prabhakaran, Mike Rosulek. Attribute-based signatures. Proc. CT-RSA 2011, pp. 376–392

[438] Ya Fen Chang, Wei Liang Tai, Hung Chin Chang. Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update[J]. Int J Commun Syst, 2014, **27**(11):3430–3440

[439] Wen Her Yang, Shiuh Pyng Shieh. Password authentication schemes with smart cards[J]. Comput Secur, 1999, **18**(8):727–733

[440] Dheerendra Mishra, Ankita Chaturvedi, Sourav Mukhopadhyay. Design of a lightweight two-factor authentication scheme with smart card revocation[J]. J Inform Secur Appl, 2015, **23**:44–53

[441] Vanga Odelu, Ashok Kumar Das, Adrijit Goswami. An Effective and Robust Secure Remote User Authenticated Key Agreement Scheme Using Smart Cards in Wireless Communication Systems[J]. Wirel Pers Commun, 2015, **84**(4):2571–2598

[442] Lili Xu, Fan Wu. An improved and provable remote user authentication scheme based on elliptic curve cryptosystem with user anonymity[J]. Secur Commun Netw, 2015, **8**(2):245–260

[443] Prosanta Gope, Tzonelih Hwang. Lightweight and Energy-Efficient Mutual Authentication and Key Agreement Scheme With User Anonymity for Secure Communication in Global Mobility Networks[J]. IEEE Syst J, 2015. Doi:10.1109/JSYST.2015.2416396

[444] Qi Xie, Na Dong, Duncan S Wong, Bin Hu. Cryptanalysis and security enhancement of a robust two-factor authentication and key agreement protocol[J]. Int J Commun Syst, 2016, **29**(3):478–487

[445] Benchaa Djellali, Kheira Belarbi, Abdallah Chouarfia, Pascal Lorenz. User authentication scheme preserving anonymity for ubiquitous devices[J]. Secur Commun Netw, 2015, **8**(17):3131–3141

[446] Dheerendra Mishra, Ashok Kumar Das, Ankita Chaturvedi, Sourav Mukhopadhyay. A secure password-based authentication and key agreement scheme using smart cards[J]. J Inform Secur Appl, 2015, **23**:28–43

[447] Fan Wu, Lili Xu, Saru Kumari, Xiong Li, Abdulhameed Alelaiwi. A new authenticated key agreement scheme based on smart cards providing user anonymity with formal proof[J]. Secur Commun Netw, 2015, **8**(18):3847–

3863

[448] Da Zhi Sun, Zhen Fu Cao. On the Privacy of Khan et al.'s Dynamic ID-Based Remote Authentication Scheme with User Anonymity[J]. Cryptologia, 2013, **37**(4):345–355

[449] Tian Fu Lee, Chuan Ming Liu. A Secure Smart-Card Based Authentication and Key Agreement Scheme for Telecare Medicine Information Systems[J]. J Med Syst, 2013, **37**(3)

[450] Xiaowei Li, Yuqing Zhang. A simple and robust anonymous two-factor authenticated key exchange protocol[J]. Secur Commun Netw, 2013, **6**(6):711–722

[451] Rajaram Ramasamy, Amutha Muniyandi. An Efficient Password Authentication Scheme for Smart Card.[J]. Int J Netw Secur, 2012, **14**(3):180–186

[452] Zhian Zhu. An efficient authentication scheme for telecare medicine information systems[J]. J Med Syst, 2012, **36**(6):3833–3838

[453] Ding Wang, Chunguang Ma. Cryptanalysis and security enhancement of a remote user authentication scheme using smart cards[J]. J China Univ Posts Telecommun, 2012, **19**(5):104–114

[454] Wen bin Hsieh, Jenq shiou Leu. Exploiting hash functions to intensify the remote user authentication scheme[J]. Comput Secur, 2012, **31**(6):791–798

[455] Debiao He, Jianhua Chen, Jin Hu. Improvement on a smart card based password authentication scheme[J]. J Internet Tech, 2012, **13**(3):38–42

[456] Tien Ho Chen, Han Cheng Hsiang, Wei Kuan Shih. Security enhancement on an improvement on two remote user authentication schemes using smart cards[J]. Future Gener Comput Syst, 2011, **27**(4):377–380

[457] Jung-Yoon Kim, Hyoung-Kee Choi, John A. Copeland. Further Improved Remote User Authentication Scheme[J]. IEICE Trans Fund Electron Comm Comput Sci, 2011, **94**(6):1426–1433

[458] Amit K Awasthi, Keerti Srivastava, RC Mittal. An improved timestamp-based remote user authentication scheme[J]. Comput & Electr Eng, 2011, **37**(6):869–874

[459] Sandeep K. Sood. Secure Dynamic Identity-Based Authentication Scheme Using Smart Cards[J]. Inform Secur J: A Global Persp, 2011, **20**(2):67–77

[460] Jia Lun Tsai, Tzong Chen Wu, Kuo Yu Tsai. New dynamic ID authentication scheme using smart cards[J]. Int J Commun Syst, 2010, **23**(12):1449–1462

[461] Ronggong Song. Advanced smart card based password authentication

protocol[J]. Comput Stand Inter, 2010, **32**(5):321–325

[462] Sandeep K. Sood, Anil K. Sarje, Kuldip Singh. An improvement of Xu et al.'s authentication scheme using smart cards[C]. Proc. ACM Compute 2010, pp. 1–5

[463] Wen Bing Horng, Cheng Ping Lee, Jian Wen Peng. A secure remote authentication scheme preserving user anonymity with non-tamper resistant smart cards[J]. WSEAS Trans Inform Sci Appl, 2010, **7**(5):619–628

[464] Sang-Kyun Kim, Min Gyo Chung. More secure remote user authentication scheme[J]. Comput Commun, 2009, **32**(6):1018–1021

[465] Hao Chung, Wei Ku, Maw Tsaur. Weaknesses and improvement of Wang et al.'s remote user password authentication scheme for resource-limited environments[J]. Comput Stand Inter, 2009, **31**(4):863–868

[466] Rajaram Ramasamy, Amutha Prabakar Muniyandi. New remote mutual authentication scheme using smart cards[J]. Trans Data Priv, 2009, **2**:141–152

[467] Tzung Chen, Wei Lee. A new method for using hash functions to solve remote user authentication[J]. Comput & Electr Eng, 2008, **34**(1):53–62

[468] Ren Chiun Wang, Wen Shenq Juang, Chin Laung Lei. A Simple and Efficient Key Exchange Scheme Against the Smart Card Loss Problem. Proc. IEEE/IFIP EUC 2007, pp. 728–744

[469] Sung Woon Lee, Hyun Sung Kim, Kee Young Yoo. Improvement of Chien et al.'s remote user authentication scheme using smart cards[J]. Comput Stand Inter, 2005, **27**(2):181–183

[470] Rongxing Lu, Zhenfu Cao. Efficient remote user authentication scheme using smart card[J]. Comput Netw, 2005, **49**(4):535–540

[471] Min Shiang Hwang, Li Hua Li. A new remote user authentication scheme using smart cards[J]. IEEE Trans Consum Electr, 2000, **46**(1):28–30

[472] Maliheh Shirvanian, Stanislaw Jarecki, Nitesh Saxena, Naveen Nathan. Two-Factor Authentication Resilient to Server Compromise Using Mix-Bandwidth Devices[C]. Proc. NDSS 2014, pp. 1–16

[473] Hugh Wimberly, Lorie Liebrock. Using fingerprint authentication to reduce security: An empirical study[C]. Proc. IEEE S&P 2011, pp. 32–46

[474] Nelly Fazio, Rosario Gennaro, Irippuge Milinda Perera, William E Skeith III. Hard-core predicates for a Diffie-Hellman problem over finite fields. Proc. CRYPTO 2013, pp. 148–165

博士期间取得的学术成果

已发表的成果：

[1] Ding Wang, Zijian Zhang, Ping Wang, Jeff Yan, Xinyi Huang. Targeted Online Password Guessing: An Underestimated Threat. *Proceedings of the 23nd ACM Conference on Computer and Communications Security (ACM CCS 2016)*, pp. 1242–1254.

[2] Ding Wang, Haibo Cheng, Ping Wang, Xinyi Huang, Gaopeng Jian. Zipf's Law in Passwords. *IEEE Transactions on Information Forensics and Security*, 2017, 12(11): 2776-2791.

[3] Ding Wang, Debiao He, Ping Wang, Chao-Hsien Chu. Anonymous Two-Factor Authentication in Distributed Systems: Certain Goals Are Beyond Attainment. *IEEE Transactions on Dependable and Secure Computing*, 2015, 12(4): 428-442.

[4] Ding Wang, Ping Wang. Two Birds with One Stone: Two-Factor Authentication with Security Beyond Conventional Bound. *IEEE Transactions on Dependable and Secure Computing*, 2016, pp. 1-22, Doi: 10.1109/TDSC.2016.2605087

[5] Ding Wang, Ping Wang. On the Anonymity of Two-Factor Authentication Schemes for Wireless Sensor Networks: Attacks, Principle and Solutions. *Computer Networks*, 2014(73): 41–57.

[6] Ding Wang, Ping Wang. On the Implications of Zipf's Law in Passwords. *Proceedings of the 21th European Symposium on Research in Computer Security (ESORICS 2016)*, Springer, LNCS 9878, pp. 111-131.

[7] Ding Wang, Debiao He, Haibo Cheng, Ping Wang. fuzzyPSM: A New Password Strength Meter Using Fuzzy Probabilistic Context-Free Grammars. *Proceedings of the 46th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2016)*, pp. 595-606.

[8] Ding Wang, Ping Wang. The Emperor's New Password Creation Policies. *Proceedings of the 20th European Symposium on Research in Computer Security (ESORICS 2015)*, Springer, LNCS 9237, pp. 456-477.

[9] Ding Wang, Qianchen Gu, Xinyi Huang, Ping Wang. Understanding Human-Chosen PINs: Characteristics, Distribution and Security.

Proceedings of the 12th ACM ASIA Conference on Computer and Communications Security (ACM ASIACCS 2017), pp. 372-385.

[10] Ding Wang, Qianchen Gu, Haibo Cheng, Ping Wang. The Request for Better Measurement: A Comparative Evaluation of Two-Factor Authentication Schemes. *Proceedings of the 11th ACM Asia Conference on Computer and Communications Security (ACM AISACCS 2016)*, pp. 475-486.

[11] Ding Wang, Nan Wang, Ping Wang, Sihan Qing. Preserving Privacy for Free: Efficient and Provably Secure Two-Factor Authentication Scheme with User Anonymity. *Information Sciences*, 2015(321): 162-178.

[12] Ding Wang, Ping Wang. Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks. *Ad Hoc Networks*, 2014 (20):1-15.

[13] Ding Wang, Haibo Cheng, Debiao He, Ping Wang. On the Challenges in Designing Identity-based Privacy-Preserving Authentication Schemes for Mobile Devices. *IEEE Systems Journal*, 2016, pp. 1-10, Doi: 10.1109/JSYST. 2016.2585681

[14] Ding Wang, Ping Wang. On the Usability of Two-Factor Authentication. *Proceedings of the 10th International Conference on Security and Privacy in Communication Networks (SecureComm 2014)*, Springer, pp. 141–150.

[15] Ding Wang, Ping Wang. Offline Dictionary Attack on Password Authentication Schemes using Smart Cards. *Proceedings of the 16th Information Security Conference (ISC 2013)*, Springer, LNCS 7807, pp. 221-237.

[16] Ding Wang, Ping Wang, Jing Liu. Improved Privacy-Preserving Authentication Scheme for Roaming Service in Mobile Networks. *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC 2014)*, IEEE Communications Society, pp. 3178-3183.

[17] 汪定，李文婷，王平. 对三个多服务器环境下匿名身份认证协议的安全性分析. *软件学报*, 2017, Doi: 10.13328/j.cnki.jos.005361

[18] 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.

[19] 汪定, 王平, 雷鸣. 基于RSA的网关口令认证协议的分析与改进. *电子学报*, 2015, 43(1): 176-184.

[20] 汪定, 王平, 李增鹏, 马春光. 可证明安全的基于RSA的远程用户口令认证协议. *系统工程理论与实践*, 2015, 35(1): 191-204.

[21] Chengyu Wang, Ding Wang, Guoai Xu, Yanhui Guo. A lightweight

password-based authentication protocol using smart card. *International Journal of Communication Systems*, 2017, Doi: 10.1002/dac.3336

[22] Debiao He, Ding Wang. Robust biometric-based user authentication scheme multi-server environment. *IEEE Systems Journal*, 2015, 9(3): 816-823.

[23] Debiao He, Ding Wang, Qi Xie, Kefei Chen. Anonymous Handover Authentication Protocol for Mobile Wireless Networks with Conditional Privacy Preservation. *Science China: Information Sciences*, 2017, 60(5): 1-17.

已投稿的成果：

[24] Ding Wang, Haibo Cheng, Ping Wang. Understanding Passwords of Chinese Users: A Data-Driven Approach. *IEEE Transactions on Dependable and Secure Computing*, 2017. (Major revision, ID: TDSCSI-2016-11-0476.R1)

[25] Ding Wang, Haibo Cheng, Ping Wang, Jeff Yan, Xinyi Huang. A Security Analysis of Honeywords. *ACM Conference on Computer and Communications Security (ACM CCS)*, 2017. (Under review, Manuscript ID: #851)

[26] Ding Wang, Haibo Cheng, Qiancheng Gu, Ping Wang, Debiao He. How Best to Attack and Generate Honeywords. *ACM Conference on Computer and Communications Security (ACM CCS)*, 2017. (Under review, ID: #861)

上述学术成果与博士论文各章节间关系

表 A 总结了上述 26 项成果与本学位论文 8 个章节间的关系，更详细描述可见节1.3。此外，本学位论文 8 个章节内部之间的关系，可见节1.3中的图 1.13。

表 A 博士期间 26 项学术成果与博士论文各章节间的关系

章节	研究内容	相关的学术成果
第一章	口令安全研究现状	成果[8]: §1.2.5 口令强度评价方法现状 成果[18]: §1.2 口令安全研究现状
第二章	口令分布函数	成果[2]: §2.4.1 Zipf模型 成果[6]: §2.6 Zipf 分布的启示 成果[9]: §2.4.2 PIN码服从Zipf分布
第三章	口令定向猜测	成果[1]：第§三章全章
第四章	口令泄露检测	成果[25]: §4.3 对Juels-Revist方案的攻击 成果[26]: §4.4∼4.6 新Honeywords方案设计和分析
第五章	口令强度评测	成果[7]：§5.4 新方法fuzzyPSM 成果[24]：§5.3 用户习惯调研
第六章	协议匿名性本质计算复杂度	成果[5]：§6.4 公钥技术原则 成果[12,17]：§6.4.2 原则普适性实证
第七章	协议评价指标体系	成果[4]：§7.6 现有评价标准分析 成果[3]：§7.7 新评价标准的提出 成果[10,13,15]：§7.2.2 攻击者模型和§7.6 现有评价标准分析
第八章	可证明安全协议	成果[4]：第§八章全章 成果[14]：§7.2.3 Fuzzy-verifier有效性分析 成果[11,19∼23]：§8.3 可证明安全方法的应用

参与的科研项目

[1] 国家自然科学基金面上项目：云计算环境下的匿名多因子认证协议研究。项目编号：61472016。时间：2015.01-2018.12。核心骨干。

[2] 国家重点研发计划项目：网络空间数字虚拟资产保护基础科学问题研究。项目编号：2016YFB0800603。时间：2016.07-2019.06。核心骨干。

获得与学术相关荣誉

* ACM China Doctoral Dissertation Award（全国2人，2018年）
* ACM SIGSAC China Doctoral Dissertation Award（全国2人，2018年）
* 教育部 自然科学奖一等奖（排名第二，全国53项，2017年）
* 中国计算机学会 优秀博士学位论文（全国10人，2017年）
* 北京大学 优秀博士学位论文（信息科学类排名第1，2017年）
* 北京大学 “博雅”博士后计划（全校首批24人，2017年）
* 北京大学 研究生“学术精英”（全校20人，2017年）
* 中国大学生 “年度人物”入围奖（全国200人，2017年）
* Elsevier Computers & Security杰出审稿人奖（2017年）
* 北京大学信息科学技术学院 “学术十杰”（全院10人，2016年）
* 北京大学 “学生年度人物”（全校10人，2016年）
* 信息安全国家重点实验室 博士生奖学金（全国10人，2015年）
* Elsevier JISA 杰出审稿人奖（2015年）
* 北京大学 “创新奖”（学术类，2015年）

致 谢

转眼倏忽间又至毕业季。回首在北京大学攻读博士学位的四年时间，往事历历在目仿佛就在昨天。记得申请读博时，暗暗给自己定的目标是博士期间争取发表一篇顶级刊物论文，没有想到在博一还没有结束时就完成了这一心愿。博士期间，还有很多类似的没有想到，如成果引起美国标准修改，进入普度大学授课内容，获得“2016·北京大学学生年度人物”。所有这些没想到，都是导师悉心培养的结果，是团队成员支持的结果，是亲朋好友们关心的结果。博士四年，虽然经历了无数科研岁月的艰辛与挫折，但体会更多的是播洒汗水后收获果实的欢乐和欣慰。一段攀登，一次涅槃，倍感幸运。

首先感谢我的恩师王平老师，是王老师带我走上了口令安全的研究之路，找到了研究的乐趣和自信。在我攻读博士期间，导师前瞻的思维、渊博的学识、敏锐的洞察力、自由的学术思想、谦逊低调的为人使我受益匪浅，王老师将永远是我辈之学习楷模。王老师为我们营造了宽松自由的学习环境，大力支持我去探索感兴趣的研究课题。在读博四年里，导师倾注了大量心血指导我进行科学研究，帮助我建立自己的研究兴趣小组，使我得以克服无数艰难险阻，并在关键时刻帮我把握方向。王老师不但教会了我如何做学术，也指引着我如何走人生。如果说能取得一点小小的成果，都是王老师悉心栽培的结果。在此，谨向导师王平教授表示最衷心的感谢。

在北大的四年时间里，亲眼见证了iCSL的成长和壮大，身为这样一个宽松而不失奋进，自由而不失严谨的集体而感到自豪。感谢马萌师兄，刘京师兄，万志涛师兄，徐景明师兄，刘岭师兄，耿雄飞师兄，刘岭师兄，能够沿着你们走过的足迹继续前行十分荣幸；感谢程海波师弟，顾乾辰师弟，沈耀晟师弟，张子健师弟，郑志雄师弟，李斌师弟，许繁华师弟，雷鸣师弟，舒龙师弟和李文婷师妹，赵奕明师妹，王晨宇师妹，郭奕旻师妹，张蕾师妹，赵圣楠师妹，能和你们在一个战壕里，感受团队作战的力量，倍感珍惜；感谢刘伟杰师弟，林杨欣师弟，你们思维敏捷，充满活力，你们蓬勃的朝气也时常感染着我。

感谢多年来一直关心和支持我的本科导师和硕士导师，虽毕业多年但始终保持着联系与合作。感谢南开信息安全系的贾春福老师，难忘本科时您逐字逐句帮我改第一篇论文的情形，难忘您对我们“踏实做人，扎实做事”的谆谆教诲。感谢信息工程大学的郭渊博老师，虽在您的直接指导下只有半年多时间，但您的长者风范，为人处世给我留下了深刻印象，是学生人生道路上学习的楷模。感谢硕士导师马春光老师，在您的关心和培养下，我参加了人生中的第一

次学术会议，发表了人生中的第一篇高被引论文，获得了人生中的第一个“学术十杰”称号，并最终决定继续求学。

衷心感谢导师组中的王立福老师，Chu Chao-Hsien老师，卿斯汉老师，陈钟老师和徐茂智老师在博士求学期间给予的精心指导和大力帮助，每一个培养环节都离不开您们的辛勤汗水。感谢软件所许进老师，张世琨老师，谢冰老师，张路老师，熊英飞老师，黄雨老师润物于无声的关心和指导。特别感谢李杰老师和林金龙老师一直以来温暖的支持和鼓励。

感谢杨义先老师，曹珍富老师，马建峰老师，刘建伟老师，林东岱老师和Hugo Krawczyk等安全界前辈的关心和支持。感谢福建师范大学的黄欣沂老师，武汉大学的何德彪老师，英国兰卡斯特大学Jeff Yan老师，西安电子科技大学的陈晓峰老师，北京航空航天大学的伍前红老师和广州大学的李进老师，与您们的交流使我对研究方向有了更准确的把握和更前沿的视野。感谢西安电子科技大学的姜奇老师，解放军信息工程大学的魏福山老师、胡学先老师，桂林电子科技大学的刘忆宁老师，电子科技大学的李经纬老师，东北大学的史闻博老师，华东师范大学的何道敬老师和北卡罗莱州立大学的王永革教授，普度大学的Jeremiah Blocki 教授，斯坦福大学的Joseph Bonneau 教授，德州大学的林志强教授，萨里大学的李树钧教授，波鸿鲁尔大学的Maximilian Golla 博士在身份认证领域的学术交流和讨论，以及论文写作、投稿方面的经验分享。感谢我们的信息安全兴趣小组，虽然我们分布在不同的实验室，不同的院系，不同的大学，甚至远隔重洋，但地理的距离并没有影响我们体验协同作战的愉快!

感谢局机关的丛局长、覃局长，您们恩师般的关心和培养是我不断奋进的动力，今后不管在哪里，在什么岗位上，我都会牢记您们的指示，立足本职，扎实工作，不给您丢人；感谢单位的薛校长、葛部长、战主任、张主任、薛参谋和张教导员，感谢您们兄长般的关怀和为我营造的良好的学习大后方，为我提供的难得求学机遇，难忘一段多彩的军旅生涯!

最后，深深感谢我的父母，您们默默的支持是我勇于面对一切困难的动力。您们勤劳的一生，是为后代无私奉献的一生，您们平凡而伟大，小时候作文中虽常这样写，但至今天才真正理解它。感谢我的爱人洪培真，有你一路陪伴、关怀和理解，再难也不难，我愿为你而奋斗！感谢所有亲人，你们一直以来的支持和理解是我毕生不断前行的力量源泉!

北京大学学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

论文作者签名：　　日期：　　年　　月　　日

学位论文使用授权说明

（必须装订在提交学校图书馆的印刷本）

本人完全了解北京大学关于收集、保存、使用学位论文的规定，即：

- 按照学校要求提交学位论文的印刷本和电子版本；
- 学校有权保存学位论文的印刷本和电子版，并提供目录检索与阅览服务，在校园网上提供服务；
- 学校可以采用影印、缩印、数字化或其它复制手段保存论文；
- 因某种特殊原因需要延迟发布学位论文电子版，授权学校在 □ 一年 / □ 两年 / □ 三年以后在校园网上全文发布。

（保密论文在解密后遵守此规定）

论文作者签名：　　导师签名：　　日期：　　年　　月　　日