调查密码选择的分布

文章 2011 年 4 月

资料来源:arXiv

2 名作者，包括:

大卫·马龙
梅努斯爱尔兰国立大学
165 份出版物 3，523 次引用

查看个人资料

调查密码选择的分布

大卫·马龙，凯文·马厄
·汉密尔顿研究所，
NUI·梅努

斯。2011年4

月20日

摘要

在本文中，我们将研究选择密码的分布。齐夫定律通常出现在精选词汇列表中。使用来自四个不同在线来源的密码列表，我们将调查Zipf定律是否是描述密码被选择的频率的一个好的候选。我们查看了许多用于测量密码分布安全性的标准统计数据，并查看了使用Zipf定律对数据建模是否会产生对这些统计数据的良好估计。然后，我们使用猜测作为衡量标准，查看来自每个来源的密码分布的相似性。这表明这些发行版提供了破解密码的有效工具。最后，我们将展示如何通过偶尔要求用户选择不同的密码来形成正在使用的密码的分布。

1 介绍

在本文中，我们研究了密码频率分布是否可以用Zipf定律来建模。Zipf定律是一种概率分布，其中事件的频率与其在频率表中的排名成反比。这里最常见事件的等级是1，第二常见事件的等级是2，依此类推。在观察自然语言中单词的使用频率时，我们观察到了齐夫定律。在我们的例子中，事件是用户使用特定的密码。为了研究这个问题，我们使用了来自hotmail.com 、 flirtlife.de 、 computer-bits.ie和rockyou.com的用户和密码列表。在每一个案例中，用户名和密码列表都是在发生安全事故后公开的。每个列表有1800到3200万用户。

有许多实质性的区别

密码的选择和自然语言中单词的使用。特别是，密码通常被选择为难以猜测，并且有大量关于如何选择密码的建议文献(例如[10])。然而，不根据建议选择密码的原因有很多:建议很费力，需要很长时间才能产生符合要求的密码，产生的密码很可能是难以记忆的字母和数字的散列。甚至有人认为，一般来说，选择这些密码中的一个的成本远远大于丢失它试图保护的信息的成本[6]。

因此，普通用户可能会选择他们容易记住的密码，例如他们的名字、他们居住的城市、最喜欢的球队等等，而不是从非字典单词列表中统一选择，这导致某些密码比其他密码使用得更频繁。由于Zipf定律已经在许多经验数据集中被观察到，我们想调查它是否为我们看到的密码提供了一个合理的模型。据我们所知，这是第一篇研究Zipf定律是否适用于密码选择的论文。

看到Zipf定律或任何其他偏向于密码数量的分布，对安全性有影响。如果可以确定密码的正确分布，就可以降低猜测密码的成本。人们很自然地期望，比如说，一个网站的用户人口统计可以用来确定攻击目标；与带有. fr域名的网站相比，带有. ie域名的网站更有可能使用爱尔兰主题密码。

Zipf定律告诉我们，某种事物出现的次数与它在频率表中的排名成反比，y∞r s，其中s是一个接近1的参数。通过绘制数据集并拟合s值，我们

将会看到，虽然 Zipf 分布不能完全描述我们的数据，但它提供了一个合理的模型，特别是密码选择的长尾理论。算法设计者可以使用密码选择的重尾分布来更有效地处理密码，如[13]中所述。

为了确定 Zipf 分布等模型是否能提供有用的预测，我们使用了猜测[11]和香农熵等指标。我们为以下两者计算这些指标

拟合模型和实际数据，并比较结果。我们发现实际的度量在 Zipf 分布预测的两个因子之内，并且 Zipf 模型通常比简单的统一模型提供更好的预测。

另一个重要的问题是，一组密码选择总体上能告诉我们多少关于密码选择的信息。因此，我们使用猜测作为衡量标准来比较不同数据集的相似性。我们将展示，通过使用一个列表中的常用密码，可以在猜测另一个列表中的密码时提高速度。

最后，我们介绍一种使用户使用的密码更加统一的技术。当用户设置或重置密码时，这项技术可能会要求他们选择不同的密码。通过将这种技术建立在 Metropolis-Hastings 算法的基础上，我们可以将其设计为在使用中产生更均匀的密码分布。

2 数据集概述

我们收集了以前被黑客攻击过的网站的密码，这些密码后来被公开泄露。由于这些集合是通过不同的方法收集的，如键记录、网络嗅探或数据库转储，因此这些列表可能只包含随机的、可能有偏见的用户样本。我们的榜单来自 2009 年 hotmail.com 榜单、2006 年 flirtlife.de 榜单、2009 年 computerbits.ie 榜单和 2009 年 rockyou.com 榜单。

一些列表还为少数用户提供了多个密码。在这种情况下，我们通过将用户的密码作为该用户看到的最后一个条目来清理集合，这可能对应于用户最初键入错误的密码，然后键入正确的密码，或者在密码是

位置	用户数量	#通过	#通过 用户数量
hotmail	7300	6670	0.91
调情生活	98930	43936	0.44
计算机比特	1795	1656	0.92
摇滚你	32603043	14344386	0.44

表 1:每个站点的用户数量和不同密码的数量。已更改，最新的密码。我们也

忽略任何具有空白密码的用户。清理完数据后，我们制作了一个表格，按照用户使用频率递减的顺序排列密码。表 1 显示了每组数据的用户数量和不同密码的数量。从表中可以明显看出，对于较小的列表，唯一密码相对较多。

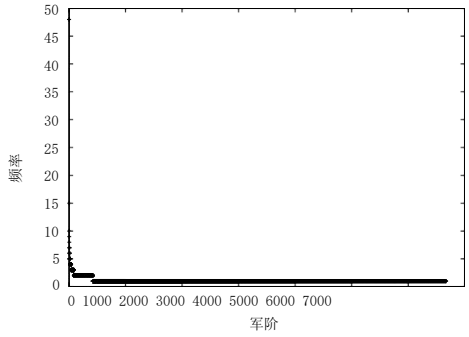
表 2 总结了每个列表中的前 10 个密码。我们看到 123456、password 之类的密码很常见。最常见的密码，“123456”在 hotmail 数据中占总密码的 0.7%，在 flirtlife 列表中占 3.3%，在 rockyou 列表中占 2.0%；“密码”占 computerbits 列表总密码的 1.2%。这表明密码分布偏向于一些常见的密码。

从每个 ccTLD 列表的前 10 个密码来看，每个列表中用户的人口统计非常清楚。请注意，热门 mail.com 数据被认为是通过针对拉丁美洲社区的网络钓鱼收集的。flirtlife 列表显示了用户说德语和土耳其语的明显迹象，computerbits 列表包含爱尔兰感兴趣的地名。如果我们看看 computerbits 和 rockyou 的数据，我们会发现网站的名字出现在每个列表的前十名。这种选择密码的方法似乎也可以用在其他网站上。

通过获取明显的密码并了解网站所针对的用户的人口统计数据，人们可以建立一个全面的字典，该字典可以涵盖网站上最常用的密码。这意味着，如果用户担心自己的账户被黑，他们应该少用常用密码。甚至一些简单的事情，如将一些字符改为大写或在“leet speak”中书写单词，都有可能将密码移出最大的-

军阶	hotmail	用户数量	调情生活	用户数量	计算机比特	用户数量	摇滚你	用户数量
一	123456	48	123456	1432	密码	20	123456	290729
2	123456789	15	菲肯	407	计算机比特	10	12345	79076
3	111111	10	12345	365	123456	七	123456789	76789
四	12345678	9	喂	348	都柏林	6	密码	59462
5	泰奎罗	8	123456789	258	莱特梅因	5	我爱你	49952
6	000000	七	沙茨	230	标准英语打字键盘的	四	公主	33291
七	亚历杭德罗	七	12345678	223	爱尔兰	四	1234567	21725
8	塞巴斯蒂安	6	丹尼尔	185	1234567	3	摇滚你	20901
9	埃斯特里拉丝毛绉	6	1234	175	利物浦	3	12345678	20553
10	1234567	6	阿斯金	171	明斯特	3	abc123	16648

表 2:每个列表的前 10 个密码



(a) hotmail

图 3.1:线性范围内等级与频率的关系图

常用列表。这些结果也证实了系统管理员根据经验观察到的一些事情，即当使用 crack 检查密码哈希时，包含本地化的字典通常会增加恢复的密码的数量。

3 密码的分发

在这一节中，我们将看看密码在我们的列表中是如何分布的，以及这些分布与 Zipf 模型的匹配程度。我们将对我们看到的第 I 个最流行的密码的频率 fi 感兴趣。如果密码出现的次数相等，我们就随机打破平局。

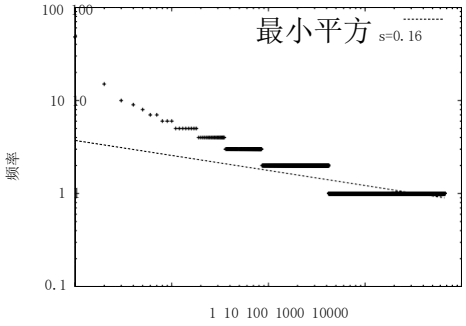
图 3.1 显示了 hotmail 列表中数据的等级与频率的线性关系。有少数密码出现的频率很高，很多密码出现的频率为 1 或 2，导致图形难以阅读。相反，我们画出了频率

在图 3.2 中，sus 按对数标度排列。这些图表当然显示了重尾行为的证据，频率下降比指数下降慢得多。Zipf 分布在双对数图上表现为一条直线，其中参数 s 是斜率的负值。

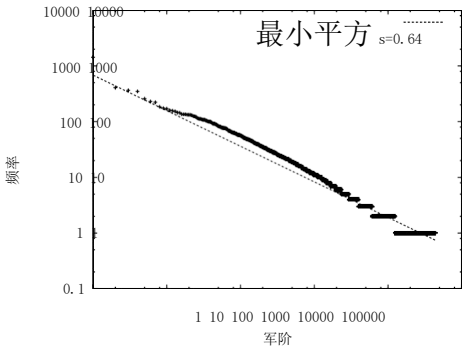
如果我们用最小二乘法拟合该数据，如图 3.2 所示，我们会得到一个太浅的斜率，因为大部分点的频率为 1 或 2，使斜率偏向 0。为了说明这一点，我们遵循[1]中的方法，对数据进行对数分类，如图 3.3 所示。这里，我们将 2n 和 2n+11 之间的所有等级的频率相加。我们看到，这给了我们一个更好地拟合我们的数据的斜率，并且这条线看起来是一个相对较好的拟合。我们使用这个分级斜率作为基础，用 Zipf 分布对我们的数据进行建模。

查看数据的另一种方法是查看 k 个用户使用的密码数量 nk。我们在图 3.4 中绘制了对数标度。如[1]中所述，如果数据是 Zipf 分布，我们预计该图也是一条斜率为(1 + 1/s)的直线。正如我们所看到的，在拟合直线之前，我们也需要收集这些数据，结果如图 3.5 所示。再次，我们看到一条线是一个相对较好的匹配，最大的差异出现在计算机位，最小的列表。由此产生的斜率总结在表 3 中。

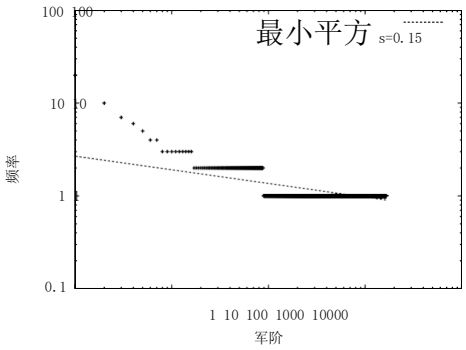
我们还可以为截尾 Zipf 分布建立最大似然估计(MLE)，它将与 r 成比例的概率分配给秩 r = 1 的密码。。。N 的 MLE 只是具有非零频率的密码的数量，s 的 MLE 可以使用标准技术构建，如[3]中所述。这具有提供两种估计的优点



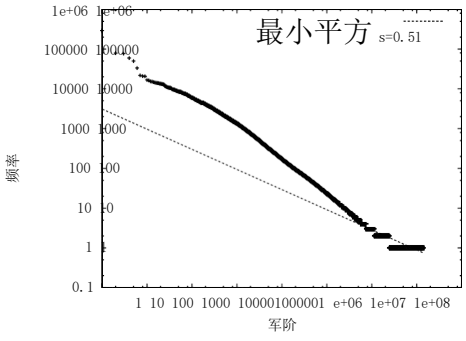
(a) hotmail



(b) 调情生活

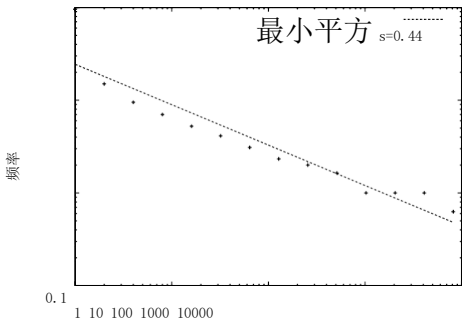


(c) 计算机比特

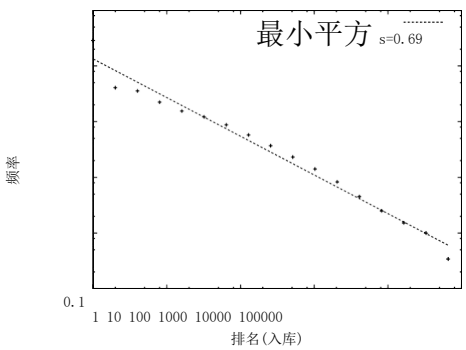


(d) 摇滚你

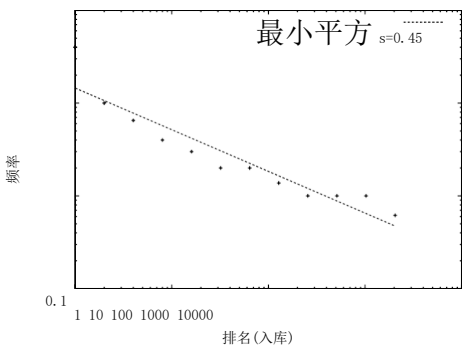
图 3.2:双对数标度的图等级与频率的关系图



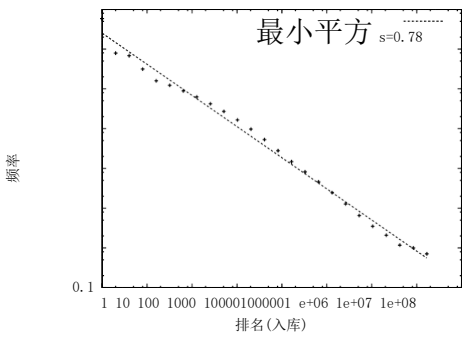
(a) hotmail



(b) 调情生活

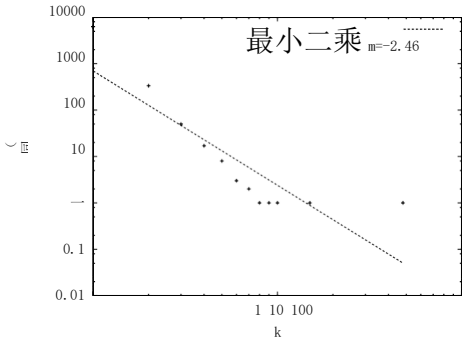


(c) 计算机比特

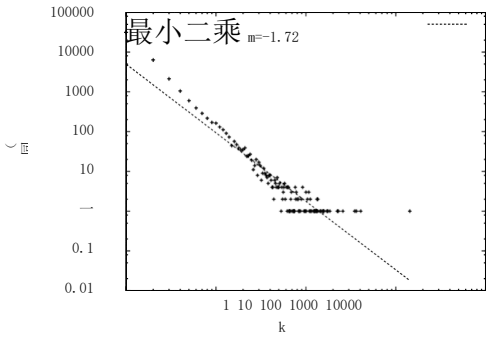


(d) 摇滚你

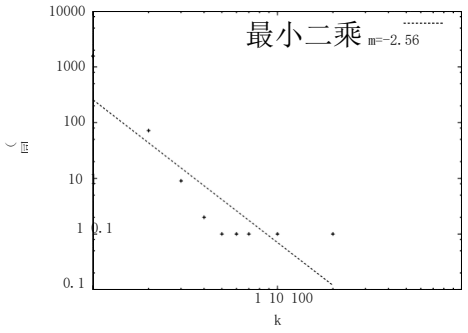
图 3.3:图 3.2 的指数面元图



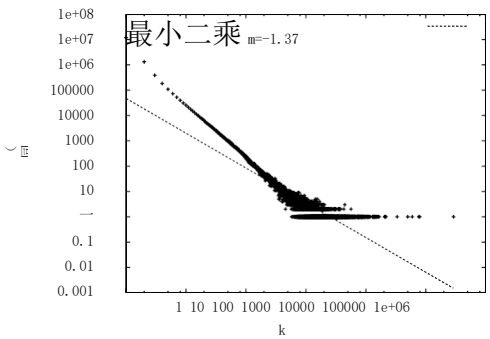
(a) hotmail



(b) 调情生活

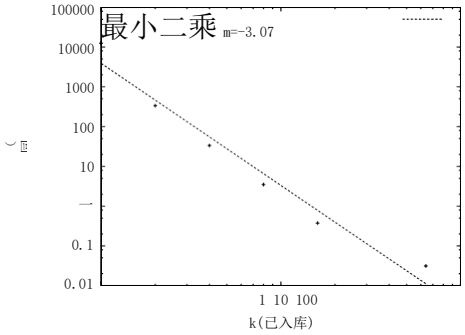


(c) 计算机比特

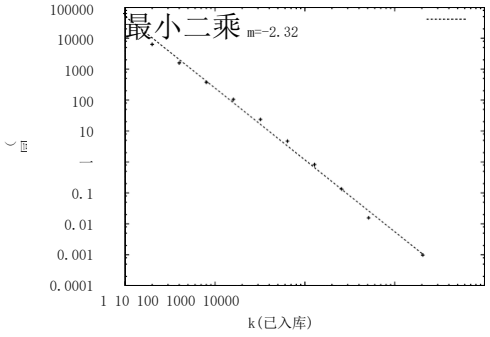


(d) 摇滚你

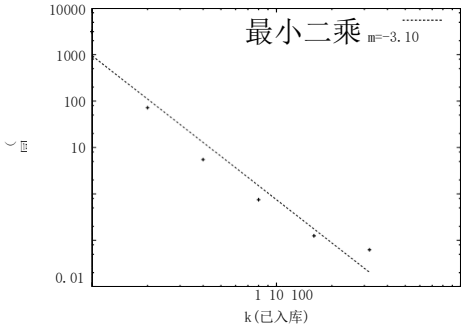
图 3.4:k 对 nk 的双对数坐标图。



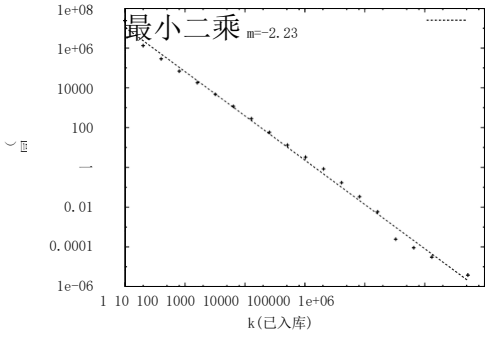
(a) hotmail



(b) 调情生活



(c) 计算机比特



(四)摇滚你

图 3.5:k 与 nk 的关系图，以对数-对数标度指数分级。

不太可能从齐夫定律中精确推导出来。

	hotmail	调情生活	c 位	摇滚你
s 生的	0.16	0.64	0.15	0.51
s 扔掉	0.44	0.69	0.45	0.78
—m 生的	2.46	1.72	2.56	1.37
—m 扔掉	3.07	2.32	3.10	2.23
1+1/秒 扔掉	3.27	2.45	3.22	2.28

表 Zipf 分布的参数，通过最小二乘法拟合频率和 nk 对 k 图进行估计。

	hotmail	调情生活	c 位	摇滚你
s MLE	0.246	0.695	0.23	0.7878
标准错误	0.009	0.001	0.02	*< 0.0001*

表 Zipf 分布的参数，用最大似然估计。

s 和 p 值 1 中标准误差的配对。结果如表 4 所示。

我们看到 MLE 为 flirtlife 和 rockyou 数据提供的 s 估计值与最小二乘估计值非常接近。较小数据集的最大似然估计值介于分箱值之间(大约
0.45)和原始值(大约 0.15)。我们
请注意，p 值表明 hotmail、computerbits 和 rockyou 数据不太可能

实际上被 Zipf 分发了。然而，对于 hotmail 和 computerbits 数据，Zipf 定律和数据之间的最大差异是前几个密码，这表明数据的尾部可以通过具有更高 p 值的 Zipf。

总之，我们已经看到，通过将密码词频数据绘制在双对数图上，它具有重尾特征。最小二乘法和最大似然估计表明，如果 Zipf 分布，s 参数很小。然而，p 值表明数据是

1 P 值是通过使用带有估计参数的 Zipf 生成样本，应用相同的排序和估计过程来计算的。然后，我们计算出超过实际数据的 K-S 统计的安德森-达林修正的分数。

4 密码统计

在这一节中，我们将研究一些与密码相关的统计数据，这些数据可以从密码的选择分布中获得。我们将研究这些统计数据是从列表中直接计算出来的，还是使用两个简单的列表模型计算出来的。对于真实列表，我们计算我们的统计数据，假设等级 I 的密码出现的概率是 fi/N，其中 fi 是我们观察到该密码的频率，N 是观察到的密码总数。

第一个模型假设所有看到的密码的密码选择是一致的，即如果密码的数量是 N，那么密码被选择的概率是 1/N。第二个模型假设密码选择的分布是 Zipf 分布，即等级为 I 的密码被使用的概率是 Pi = Ki s，其中 s 是第 3 节中的参数，K 是归一化常数。

现在让我们考虑感兴趣的统计数字。这些统计数据中的一些非常强调分布的尾部，并且对模型和数据之间相对较小的差距非常敏感。第一个统计是猜测，这是正确猜测密码[11]所需的平均猜测次数，此时已知密码的排序列表，但不知道确切的密码。猜测是
由...给出

$$G = \sum_{i=1}^{N} iPi,$$

其中 Pi 是等级 I 的密码的概率。

根据给定的分布，猜测密码的另一个策略是尝试普通密码，但是当分布的某个分数 α 被覆盖时就放弃。与此相关的平均猜测次数被称为 α 猜测，Gα [11]。其值由下式给出

$$G\alpha = \sum_{i=1}^{r\alpha} iPi,$$

其中 rα 是当
成功的累积概率至少为 α 。我们将使用 α = 0.85，这样我们就覆盖了大部分分布，但避免了尾部。

香农熵，

$$h = -\sum Pi \log_2 Pi,$$

是与随机变量相关的不确定性位数的常用度量。虽然香农熵已被用作密码和密钥分发安全性的衡量标准，但它与猜测密码的难易程度没有直接关系[9，7]。伊恩熵是香农熵的推广，它是

由...给出

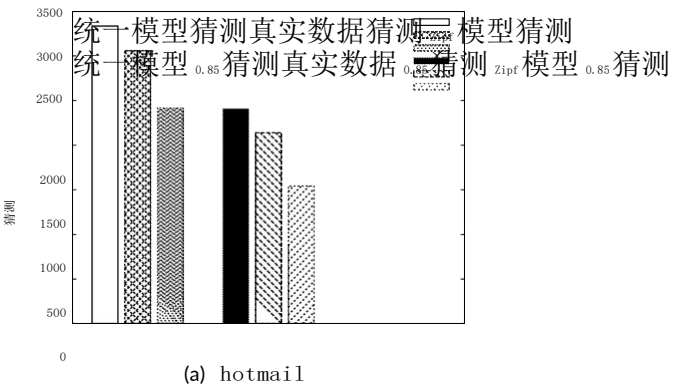
$$R = \log_2\left(\sum \sqrt{Pi}\right)^2。$$

它与密码的可猜测性密切相关[2，8]。最小熵也被用作密码/密钥安全性的保守度量[5]。

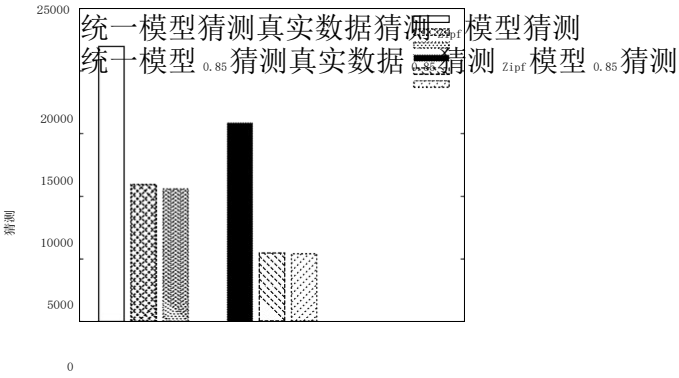
图 4.1 比较了统一模型、真实数据和 Zipf 模型的猜测统计。左边的三个条显示猜测值，右边的三个条显示 0.85 的猜测值。不出所料，对统一模型的猜测估计过高地估计了所需的猜测次数。hotmail.com 和 computerbits.ie 列表中的密码总数中只有相对较小的一部分是共享的。这似乎反映在猜测的预测中，均匀分布为 hotmail 和 computerbits 提供了相对较好的预测，而 Zipf 模型低估了这一点。对于 flirtlife 和 rockyou 来说，共享密码在总密码中所占的比例更大，猜测的比例远低于统一猜测，但 Zipf 模型提供了更好的预测，尽管它仍然低估了。

图 4.2 显示了实际数据和模型的熵值。香农熵显示在左边，最小熵显示在中间，伊恩熵显示在右边。同样如预期的那样，统一模型倾向于高估熵。然而，对于伊恩熵，模型和数据给出的结果似乎很接近。Zipf 模型似乎在所有情况下都提供了相对较好的近似值。

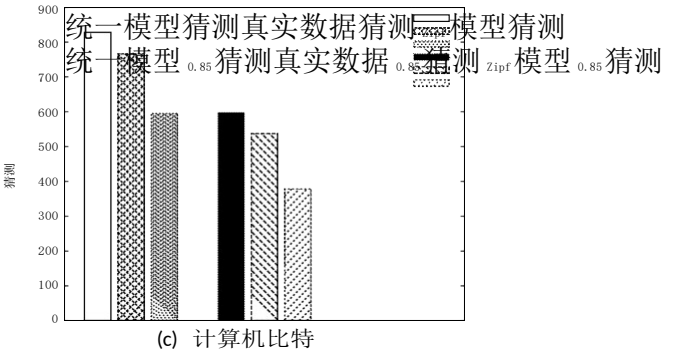
5 分布之间的关系



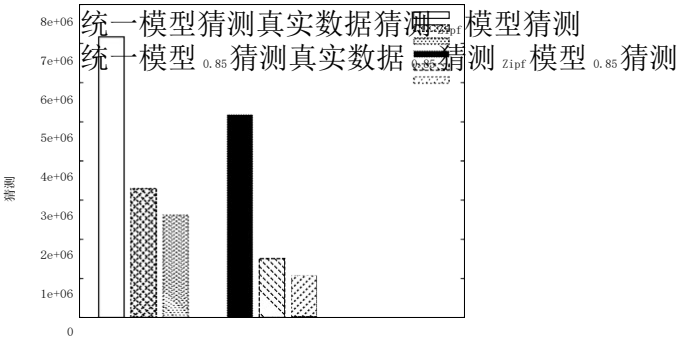
(a) hotmail



(b) 调情生活



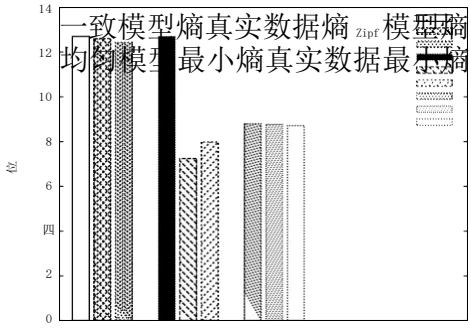
(c) 计算机比特



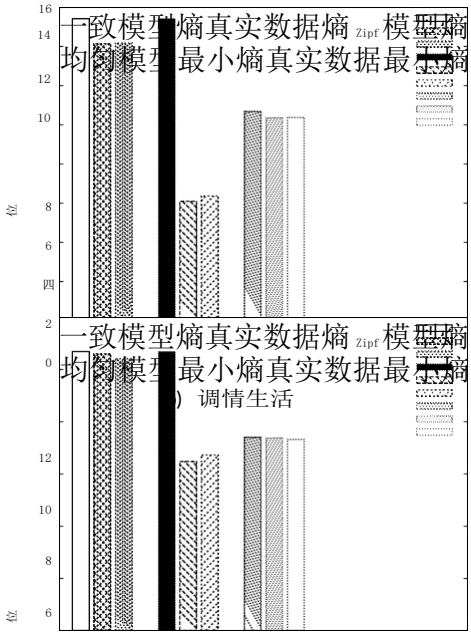
(d) 摇滚你

我们已经看到虽然我们列表中的流行密码有一些共同点(例如，

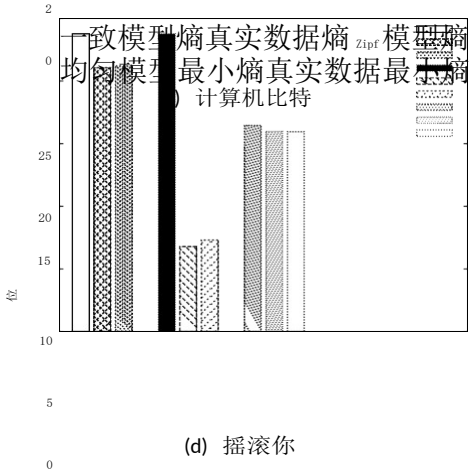
图 4.1:统一模型、真实数据和 Zipf 模型的猜测统计。



(a) hotmail



(b) 调情生活



(c) 计算机比特

(d) 摇滚你

统一模型的熵值，

图 4.2:
真实数据和 Zipf 模型。

密码 23456)，它们还显示网站或服务的特定功能。在第 3 节中，我们还看到所有列表都有一些相对频繁使用的密码，后面跟着一长串不常用的密码。在这一部分，我们将量化这些列表中的密码之间有多少重叠。考虑从我们的列表中随机选择一个用户来猜测密码的问题。如果我们按照列表中最受欢迎到最不受欢迎的顺序猜测密码，那么在 t 次猜测之后，我们将会猜到

使用的密码

用户。如果我们每次尝试猜测一个密码，按照这个顺序猜测可以尽快恢复用户的密码，从这个意义上来说，这是最佳的。图 5.1 显示了每个数据集的实线 C(t)。右边的轴表示 C(t)/N，我们可以将其解释为在 t 次猜测中成功猜测的概率，或者密码被猜到的用户的比例。例如，在使用 hotmail 数据进行 100 次猜测后，我们已经恢复了大约 400 个用户的密码，这表示针对特定用户的成功概率为 5%。因为我们猜测在最佳顺序中，其他顺序必须恢复更少的用户，并且具有更低的成功概率。

如果我们不知道猜测密码的最佳顺序，我们可能会以另一个参考数据集的最佳顺序来猜测密码。假设我们在参考数据集中有一个秩为 I 的密码，它在被猜测的数据集中秩为 σ(i)。如果我们按照参考数据集给出的顺序猜测，在 t 次猜测之后，我们将会猜到

用户，其中如果密码 I 不在我们猜测的列表中，我们假设 fσ(i)为零。如果两个列表中密码的流行度排序相同，那么这个函数将是 C(t σ) = C(t)，否则 C(t σ) ≤ C(t)。

图 5.1 显示了 C(t σ)，当使用其他列表作为参考时。考虑 1000 次试猜后的情况。密码匹配的用户数量

这 1000 个猜测中的一个，C(1000 σ)，可以是

根据用作参考的列表，看起来相差几乎一个数量级。因此，为了快速猜出密码，我们需要一个好的参考列表。

请注意，对于任何列表，一旦我们进行了10-100次以上的猜测，较大的参考列表会比较小的参考列表获得更多的成功。这表明，除了最流行的密码之外，可能还有一种更通用的密码
更明显的密码的一般顺序

从更大的数据集中输入。rockyou列表非常有效，排名前100万的rock- you密码覆盖了全球近40%的用户
其他列表。

我们可以将此直接应用于密码破解问题。2010年12月，Gawker.com的密码数据库被泄露。该数据库不包含明文密码，而是包含使用众所周知的DES和Blowfish密码散列方案的密码散列。我们可以使用列表中的单词作为离线破解的猜测-
对Gawker hashes的攻击。

Gawker数据集包含748，090个用户的潜在有效(即13个字符)DES哈希。哈希使用3844种不同的盐。一个简单的

perl脚本可以以

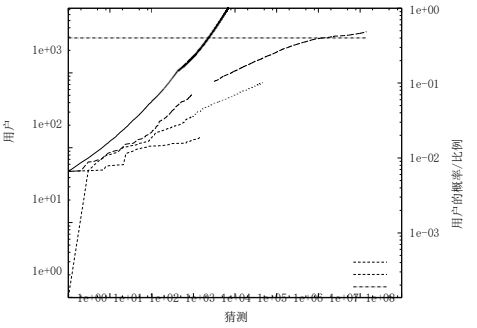
在现代CPU上，每个内核每小时大约进行80，000次试验。由于DES哈希将密码转换为8个字符，我们截断长密码并重新聚合之前的列表。图5.2显示了t次尝试后密码被破解的用户数量。同样，大名单提供了最快的密码恢复，并恢复了40%的用户

在不到一百万次的试验中。甚至我们更小的
列表做得很好，在大约1000次尝试(不到一分钟的CPU时间)中恢复了超过10，000个用户的密码。

为了比较，在图5.2中我们展示了
按词汇顺序使用字典作为猜测列表的结果。该词典基于Mac OS X上/usr/share/dict/的内容，被截断为8个字符，并使用Unix sort命令进行排序。这导致付出的努力的回报大大低于使用分级密码列表。

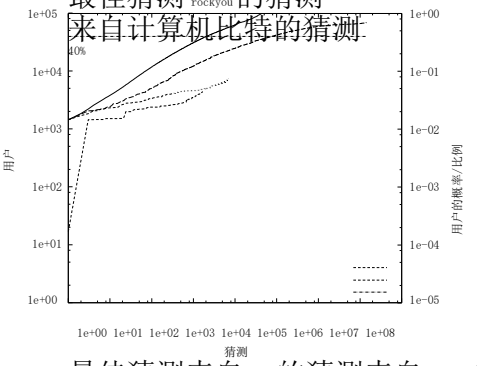
到目前为止，我们已经测量了有效的-

通过计算在t次猜测后其密码将被正确猜测的不同用户的数量来猜测密码的准确性。此指标的替代方法是查看t次猜测后正确猜测的不同密码的数量。在这种情况下，我们



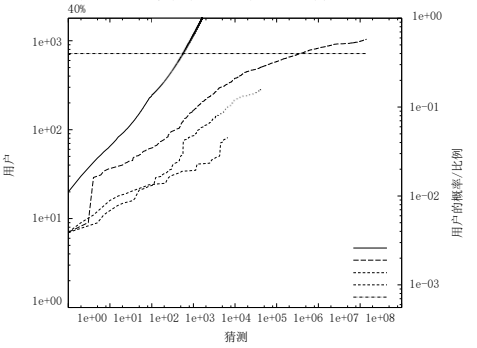
(a) hotmail

最佳猜测 rockyou 的猜测
来自计算机比特的猜测



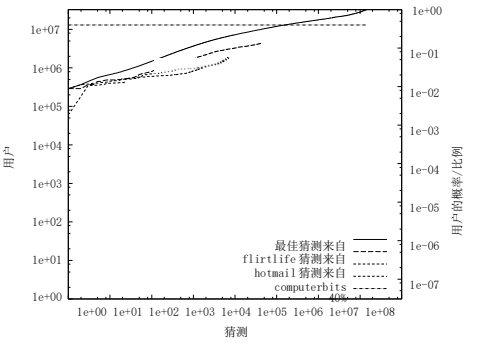
(b) 调情生活

最佳猜测来自 的猜测来自 hotmail 的猜测
来自计算机比特的猜测



(c) 计算机比特

最佳猜测来自 rockyou 猜测来自 flirtlife 猜测来自 hotmail



(d) 摇滚你

图5.1:通过使用一个分布的密码排序来猜测另一个分布，从而恢复密码的用户数量。

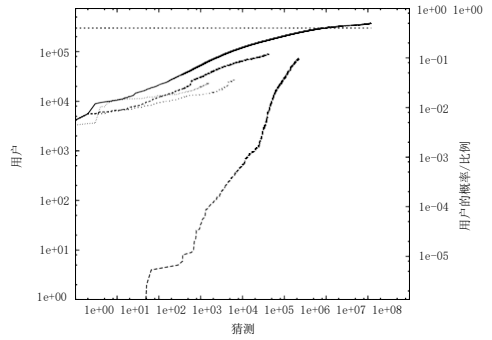


图 5.2:通过使用 Gawker.com 密码哈希上不同分布的密码排序，密码被破解的用户数量。

每次猜测后恢复零个或一个密码。

图 5.3 显示了我们的主要列表的结果。我们可以恢复通过的最佳速度-

单词是每次猜测 1 个，所以我们绘制了最佳线 y = x。我们看到，大约 500，000 - 5，000，000 次猜测，我们可以获得大约 40%的密码，除非是在猜测 rock- you 密码时，因为其他列表的猜测次数不足以达到 40%。不管
当从 rockyou 数据集中猜测密码时，其他列表的曲线接近最佳线，表明每次猜测都有很好的回报。

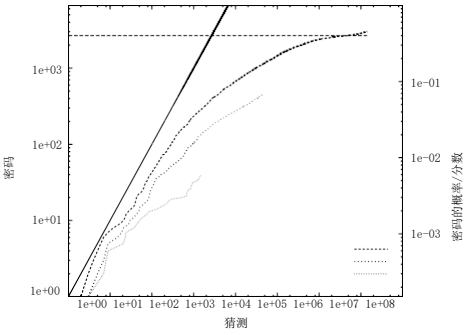
图 5.4 显示了根据通过率猜测 Gawker 密码的结果
单词恢复了。因为不是所有的散列都被

破解和哈希加盐，我们不知道不同密码的总数。然而，我们可以把这个数字的上限定为-
苏明认为所有未破解的密码都是唯一的。

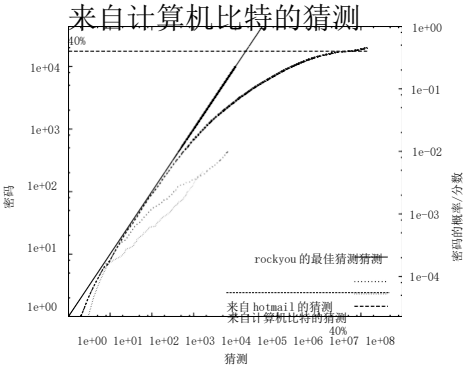
首先，我们注意到使用其他密码列表来猜测仍然比使用字典提供更好的回报。事实上，基于 rockyou 数据集，对于 10 到 10，000 次猜测，曲线相对接近最佳猜测线，表明成功率几乎为 100%。在使用了这个列表中的 1400 万个密码后，我们已经破解了接近

40%的密码。字典曲线
单词与最佳单词距离很远，这表明在 Gawker 的数据集中，只有不到 10%的词典单词被用作密码。

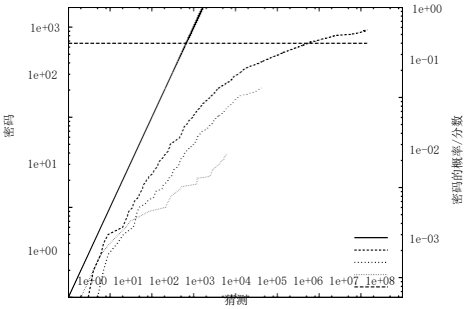
在[4]中，作者考虑了各种技术



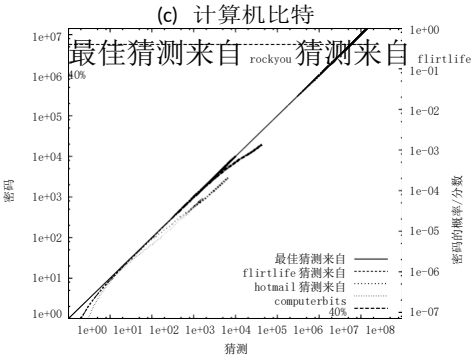
(a) hotmail



(b) 调情生活



(c) 计算机比特



(d) 摇滚你

图 5.3:通过使用一个分布中的密码排序来猜测另一个分布而恢复的密码数量。

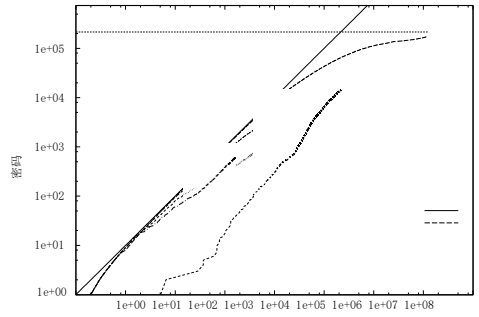


图 5.4:使用每个列表的排序来破解 rockyou 的猜测散列。

来自调情生活的猜测来自 hotmail 的猜测
来自计算机科学的猜测字典中的猜测
利用一列表的排序来破解 rockyou 的猜测
40% 的上限

用于生成猜测/破解的候选密码。这些技术包括字典攻击、损坏的字典和马尔可夫生成器，它们可以在样本密码上进行训练。他们使用三组数据中恢复的密码部分来评估这些技术。他们表明为了恢复大部分密码，比如 40%，需要猜测的次数超过 1 亿次-

lion，除非密码生成技术是在类似的数据集上训练的。

我们的结果表明使用大量的作为猜测来源的密码似乎提供了比这些技术更好的回报，能够在不到 1000 万次的猜测中恢复 40%的密码。当然，一旦耗尽，列表就不会提供更多的候选猜测，而 mangling 和 Markov 技术理论上可以产生无界的数据集。

6 使密码选择更加统一

我们看到，人们对密码的选择相当不统一，导致一些密码出现的频率很高。上一节演示了这样做的一个后果:相对较少的单词有很高的概率匹配用户的密码。相对常见的做法是禁止字典中的单词或常用密码(例如 Twitter 禁止的密码列表[14])，以努力使用户远离更常用的密码。

事实上，如果密码选择是统一的(超过一大组密码)一些基于普通密码存在的攻击变得无效。基于此，在[13]中提出了一种方案，当密码变得相对太流行时，该方案防止用户选择特定的密码，以减少密码分布的不均匀性。

然而，存在众所周知的方案，其使用来自随机发生器的输出并操纵它来实现不同的分布。例如，Metropolis-Hastings 方案允许我们生成期望的分布 P(.)通过概率性地接受/拒绝马尔可夫链 Q(., .).它具有一个有用的特征，即只要已知与密度成比例的函数，就不需要知道期望分布的密度。

基本的 Metropolis-Hastings 方案如下:

1. 设 t → 0，选 xt。
2. 用分布 Q(x′ ，xt)生成 x′ 。
3. 有可能

，t ，

部 $1, \frac{P(x)Q(x', xt)}{P(x)Q(x', xt)}$

转到步骤 4(接受)，否则返回步骤 2(拒绝)。

4. t → t + 1，XT→x′ 。

其中序列 xt 的项是输出。通常，初始输出被丢弃，以洗去 x0 的初始选择，并允许序列向其稳定行为移动。这有时被称为老化。

我们可以应用这样的方案来产生更统一的密码选择。我们期望的分布 P(.)对于用户愿意使用的所有密码都是统一的，因此 P(x′)/P(x)= 1。接下来，我们假设如果用户被要求选择一个密码，他们选择它独立于先前的可能性为 Q(.).因为我们不知道 Q(。)，我们将通过跟踪频率 F(.)用户用它来建议通行字。

当用户选择一个密码时，这暗示了下面的方案。

1. 从之前看到的所有密码中统一选择一个密码 x。

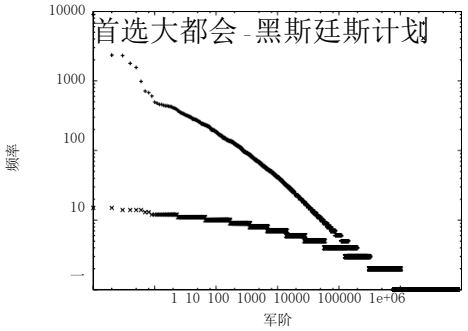


图 6.1:从 rockyou 数据集生成的用户排名与频率的关系图，有自由选择和 Metropolis-Hastings 方案。

2. 要求用户输入新密码 x’。
3. 在[0，F(x’)]范围内生成一个均匀实数 u，然后递增 F(x’)。如果 u ≤ F(x),转到步骤 4(接受)，否则返回步骤 2(拒绝)。
4. 接受使用 x ’作为密码。

该方案旨在使用 Metropolis- Hastings 方案，通过用户的选择生成统一的分布。有几件事需要注意。首先，通过在所有看到的密码中统一选择 x，我们的目标是避免需要一个磨合阶段，因为我们从我们想要的分布中选择开始。第二，实际上从来没有使用过密码 x，只有它的频率 F (x),所以如果方案没有看到以前的密码，我们可以用 0 来表示 F (x)。最后，方案学习 F(。)在线，所以它看到的密码选择越多，我们就越希望它能更好地做出统一的选择。

我们实现了这个方案，并通过从 rockyou 数据集中选择密码进行了测试，其中密码被选中的概率与其在数据集中的出现频率成比例。我们使用 Metropolis-Hastings 方案为 1，000，000 个用户生成了密码，作为比较，用户可以自由选择任何密码。结果如图 6.1 所示。我们看到，在 Metropolis-Hastings 方案中，分布更加均匀，最常见密码的频率从 1000 多次减少到 10 多次，减少了两个数量级以上。这种方案的一个问题是它可能会拒绝用户的密码选择并且使用户感到沮丧。然而，在这些试验中，用户平均被要求
1.28 个密码，方差为 0.61。

我们注意到这个方案与[13]中的方案有一些相似之处。这两种方案都必须存储密码的频率信息。为了避免以纯文本形式存储密码，可以存储散列密码的频率。然而，我们的方案存储的是用户选择密码的频率信息，而不是频率信息

正在使用的密码。如果频率表被攻击者窃取，这有一些好处，因为即使散列可以被破解，由于 Metropolis-Hastings 方案，频繁选择并不常用。

这两种方案都可以用 count-min 草图实现，而不是使用简单的频率表。这是一种存储频率估计值的有效数据结构。如[13]中详细描述的，将信息存储在草图中是有好处的，特别是当草图被攻击者窃取时。这是因为它使用多个输出空间较小的哈希函数，如果攻击者试图识别高频密码，会导致误报。

这些方案的一个区别是 Metropolis-Hastings 算法旨在使整个分布更加均匀，而不是限制最流行的密码的频率。正如我们在第 5 节中看到的，中等排名的密码，比如排名 10-1，000，对于提高猜测用户密码的成功率非常重要；这些猜测将成功概率从百分之几增加到百分之几。通过降低这些密码的频率，我们降低了使用高等级和中等等级密码的攻击的有效性。

我们还注意到，虽然该方案在线学习密码频率分布，但是它也可以使用已知的密码频率列表来初始化。虽然我们选择了统一分布的目标，但是该方案也可以与禁止密码的列表相结合(通过将期望的频率 P (x)设置为零)或者对一些密码实施软禁止(通过减少这些密码的 P (x))。

7 讨论

我们已经看到，虽然 Zipf 定律并不完全匹配，但它似乎对选择密码的频率提供了一个尚可接受的描述。参数 s 的估计值远小于 1。虽然这可能被解释为表明强重尾行为，但另一种解释是，随着 s → 0，分布变得单一，这实际上是密码所希望的。这些观察可能对算法设计者有用，用于确定数据结构的大小，或者甚至利用用户选择的相对重尾特性。

我们还看到，拟合分布提供了香农熵、猜测和评估密码分布时感兴趣的其他统计数据的相对较好的近似值。使用统一的模型，其中所有通行字都是同等可能的，为具有较小 s 的数据集提供了合理的近似，但是提供了对最小熵的较差估计。

我们已经看到，在最流行的密码中，选择密码的用户群的人口统计学特征是显而易见的，甚至网站的名称也可能是密码。一些网站，例如 Twitter，已经注意到了这一点，并实施了禁止密码列表[14]，其中包括许多更常见的密码，包括网站的名称。这也给了这样一个建议，即管理员在使用密码破解软件检查其站点上使用的密码的安全性时，应该包括包含本地使用的术语的自定义词典。

Zipf 分布衰减相对较慢，因此我们预计会有大量相对常用的密码。我们调查了这些密码从一个列表到另一个列表是否变化很大。我们看到的就是这种情况，虽然不是最佳的，但是较大的列表提供了关于在其他列表中通行字的排序的良好指导。我们已经证明，这可以在猜测或破解中等猜测次数的密码时提供显著的加速，特别是在简单的字典攻击上，但也在[4]中描述的一系列猜测生成技术上。

从一系列网站收集泄漏密码的攻击者有了破解密码的有用起点。如果暴露了散列密码，攻击者尝试 2000 万个密码的时间是相对的即使在单个 CPU 上也非常小。我们注意到，这增加了在网站之间重复使用密码是一种风险的建议的额外分量，即使攻击者没有办法识别哪些用户对是网站所共有的。这是因为，如果只有一个站点以未散列的格式存储密码，并且该密码被泄露，那么在密码被散列的系统上，这有助于随后对该密码的破解。

禁止更常用的密码可能会导致使用中的密码分布更加均匀。有趣的是，我们发现大多数英语词典中的单词并不一定是常见的密码:在超过 220，000 个词典单词中，只有不到 15，000 个在 Gawker 数据集中作为通行单词出现。我们提出了一个基于 Metropolis-Hastings 算法的方案，旨在生成更统一的密码选择，而不必事先知道一系列常用密码。其基本实现相对简单，可以很容易地集成到 PAM 模块中[12]。

8 结论

我们已经看到，对于用户选择密码的频率，Zipf 分布是一个相对较好的匹配。我们还看到，在我们研究的列表中发现的密码具有相对相似的排序。因此，当猜测或破解另一个列表中的密码时，一个列表中的密码是很好的候选。最后，我们提出了一个可以引导用户更均匀地分配密码的方案。

鸣谢:作者要感谢 Ken Duffy 发人深省的评论和 Dermot Frost 提供的备用 CPU 周期。

参考

[1] 拉达·阿·亚当。Zipf，幂律和帕累托-排名教程。Xe- rox 帕洛阿尔托研究中心，http://www.hpl.hp.com/research/idl/papers/ranking/r 2000.

[2] 埃达尔·阿里坎。关于猜测的一个不等式及其在顺序译码中的应用。

IEEE信息论汇刊，第42期，1996年1月。

[3]A.克劳塞特，C.R .沙立兹，和M.E.J .纽曼。经验数据中的幂律分布。暹罗评论，51(4):661－703，2009年。

[4]马特奥·德拉米科、皮埃特罗·米希尔迪和伊夫·鲁迪尔。密码强度:实证分析。INFOCOM，第1－47页，2010年。

[5]唐纳德·e·伊斯特莱克、杰弗里·I·席勒和史蒂夫·克罗克。RFC 4086:安全性的随机性要求。第1－47页，2005年。

[6]C.赫利。再见，不要感谢外部性:用户对安全建议的理性拒绝。《2009年新安全范例研讨会论文集》，第133－144页。美国计算机学会，2009年。

[7]D.马龙和w .沙利文。猜测不能代替熵。在信息技术和电信会议上。Citeseer，2005年。

[8]大卫·马龙和韦恩·g·沙利文。猜测和熵。IEEE信息论汇刊，50(3):525－526，2004。

[9]詹姆斯·l·梅西。猜测和熵。1994年IEEE国际信息理论研讨会会议录，第204页，1994。

[10]明迪·麦克道尔，杰森·拉斐尔和肖恩·赫尔南。网络安全提示st04-002。http://www.us-cert.gov/cas/tips/ST04-002.html，2009.

[11]约翰·奥·普利姆。密码学中功和熵的差异。密码学理论图书馆:记录，98-24页，1998。

[12]维平·萨马尔。使用可插入身份验证模块(PAM)的统一登录。《第三届美国计算机学会计算机与通信安全会议论文集》(CCS'96)，第1-10页，1996年。

[13]斯图尔特·谢克特、科马克·赫利和迈克尔·米岑马赫。流行就是一切:保护密码免受统计猜测攻击的新方法。在…里

HotSec'10第五届USENIX安全热点会议论文集，第1－6页，2010年。

[14]twitter.com。twitter注册页面的源代码。view-source:https://Twitter .com/sign up(搜索twttr。禁用密码)，2010年。 -