口令泄露检测技术

汇报人：汪 定

北京大学 信息科学技术学院
软件工程国家工程研究中心
高可信软件技术教育部重点实验室

2019年1月19日 北京

wangdingg@pku.edu.cn (✉)
*wangdingg.weebly.com (*主页*)*

口令，“密码”，password

基于口令的身份认证无处不在

口令认证的不可替代性

	成本低廉	可用性	可再生性
口　令	是	中	是
硬件token	否	低	是
生物特征	否	高	否

共识：在可预见未来，口令仍将是最主要的认证方式。

“拖库”事件频繁发生

- “拖库” —— 网站口令文件库泄露（被窃取）
- 近年来，数以百计【已经确认的】
  - 2018上半年，**166起**【 GemaltoBreach Level Index 】
  - 2017年，**221起**【Verison 2018 Data Breach Report】
  - 2016年，**1093起**【IRTC Identity Breach Report】
  - 201603-201703，**3785起**【Thomas et al., CCS 2017】
  - 2011-2015，国内96起【http://www.liu16.com/post/476.html】
- 哪些大网站被拖库了？

  Yahoo, Dropbox, LinkedIn, Adobe, 小米, CSDN, 天涯….

口令文件泄露后……


```
85  i believe i can fly
86  password
87  The Quick Brown Fox Jumped Over The Lazy Dog
88  吸烟有害健康
89  transportability
90  sanfranciscobitcoins
91  idk my bff jill
92  nigga please
93  123123123
94  esto es una prueba
95  anthem
96  1231231232
97  praha
98  it was the best of times it was the worst of times
99  1!2@3#4$5%6^7&8*9(0)
100 1234
101 correct horse battery
102 Thorsten
103 dutsu
104 doandroidsdreamofelectricsheep
105 $$$
106 1TOY`
107 Password1
108 koj
109 1231231239
110 portki
111 Pennypacker2
112 zelda
113 correct horse battery staple3
114 We are what we repeatedly do. Excellence, then, is not an act, but a habit.
115 password12341234
116 keineahnung23
117 0123456789
118 pass123
119 zerohedge
120 中国上海
121 YaLiC
122 thimbleful
123 my babe
```

比“拖库”更可怕的是。。。。

泄露的网站	泄露的口令数量	泄露时间	发现时间	历时
Myspace	360,213,049	2008 年	2016.07	8 年
Fling	40,757,760	2011 年	201605	5 年
LinkedIn	1.17 亿	2012.06	2016.05	4 年
Dropbox	68,680,741	2012.06	2016.08	4 年
VK.com	100,544,934	2012 年底~2013 年初	2016.06	4 年
Yahoo	30 亿	2013.08	2017.10	4 年
Yahoo	10 亿	2013.08	2016.09	3 年
Yahoo	5 亿	2014.08	2016.12	2 年
Weebly	43,430,316	2016.02	2016.10	8 个月
Deloitte	500 万	2016.10	2017.03	5 个月
Last.fm	43,570,999	2012.03	2012.06	3 个月
Under Armour	1.5 亿	2018.02	2018.03	1 个月

一个现实的问题

如何降低用户口令文件泄露带来的危害？

潜在的解决办法

策略①

使用机器相关 函数

基本思想：

窃取无用

【ErsatzPasswords，ACSAC'15 最佳论文】

策略②

门限密码学

基本思想：

秘密分割，

窃取部分无用

【Threshold PAKE/PASS，CRYPTO'12, CCS'15】

策略③

放置假口令

基本思想：

欺骗迷惑，及时检测

【Honeywords, Honeyindex，CCS'13, NDSS'18】

使用机器相关 函数

- 最典型要数 ErsatzPasswords方案
  - 在服务器端使用一个机器相关的函数 $\mathrm{HDF}(\cdot)$ 保存：

$$\beta_i =$$

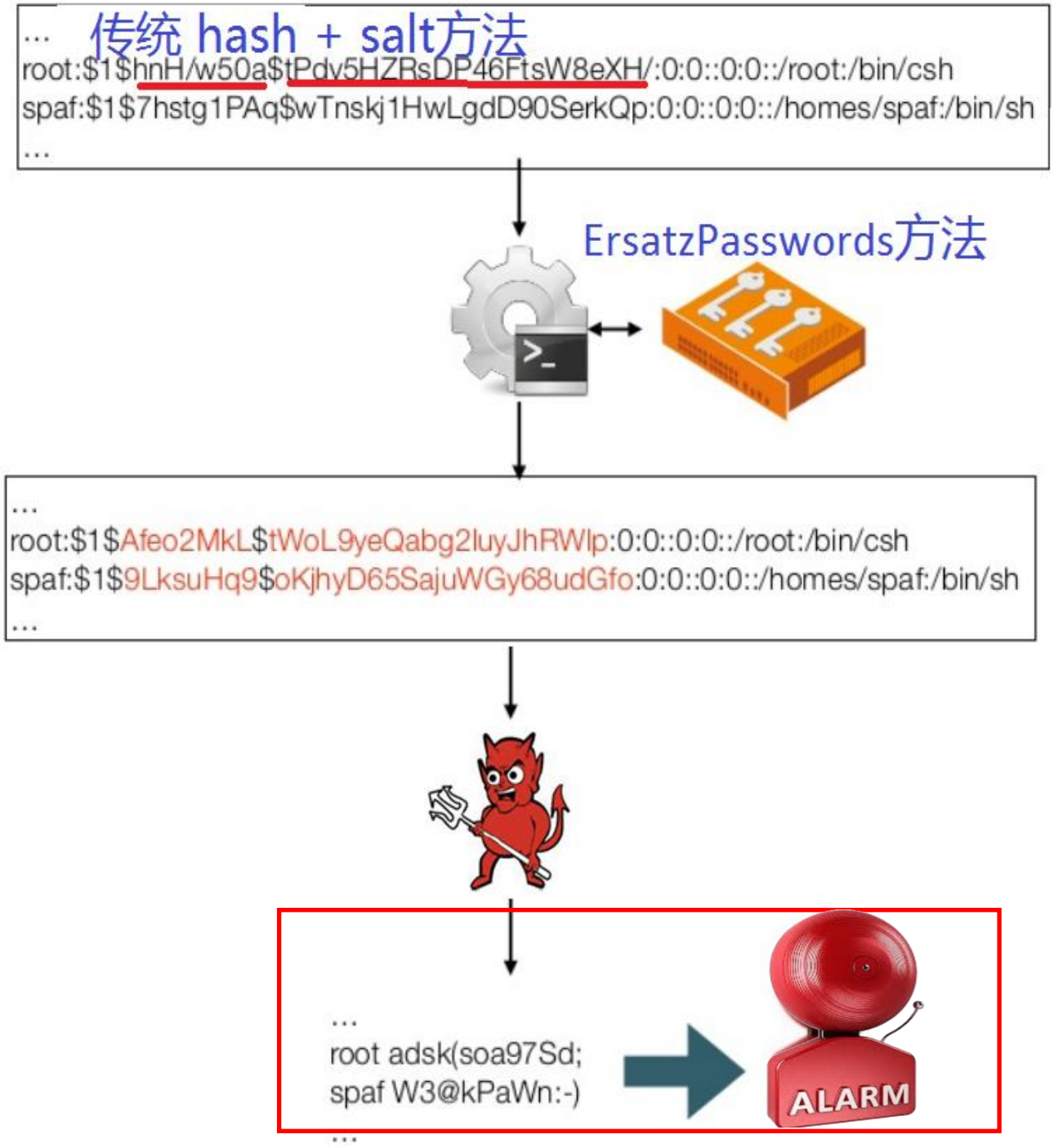
显然，PW_i^* 是真口令经过$\mathrm{HDF}(\cdot)$运算之后的字符串，是假口令。

- 当攻击

验证

如果相等，意味着：

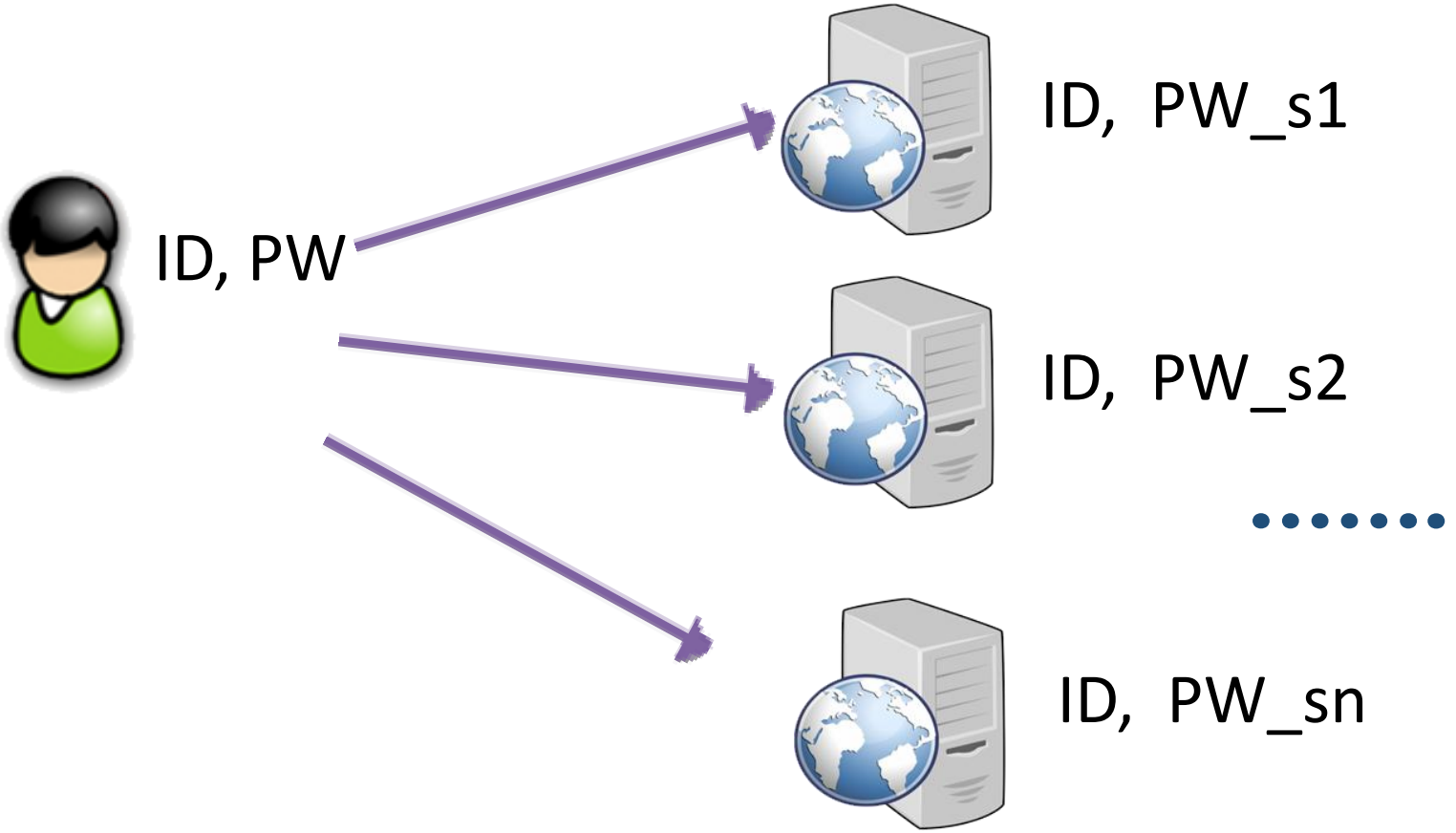
$$PW_i^* = \mathrm{HDF}(PW_i)$$



门限密码学

- 当口令PW分割成n份，放置于 n个服务器，
  当且仅当，k>t个服务器被攻陷， PW 才泄露

PW = (PW_s1,
PW_s2,
…,
PW_sn)

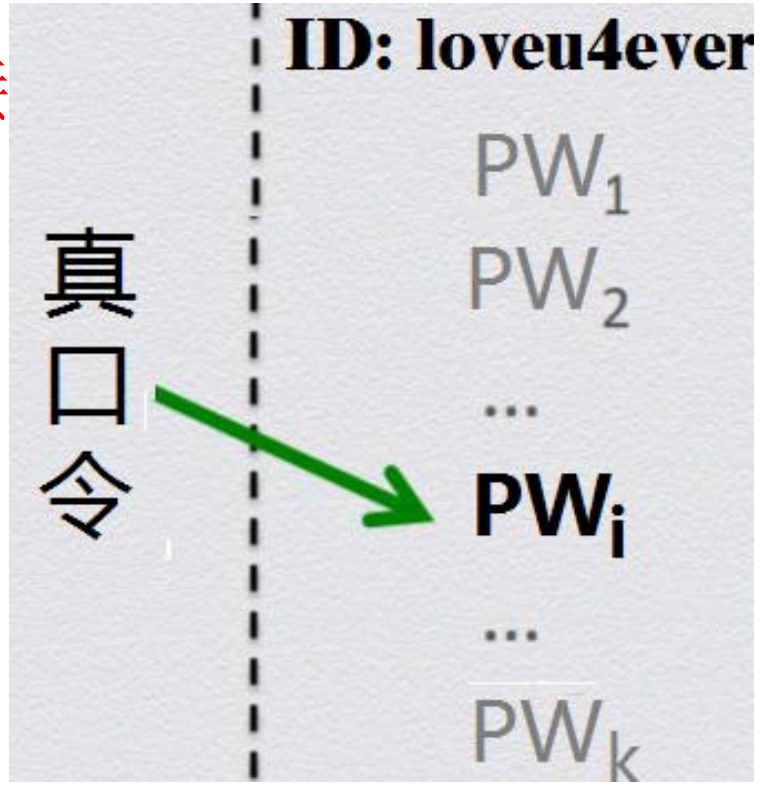


放置假口令

- 在口令文件中，置入一批假帐户，可用来检测口令文件泄露事件。

  如果仅放置假帐户，真帐户的口令仍然会直接

不能防御在线猜测攻击。



在每个帐户中，置入 *k-1* 个 假口令

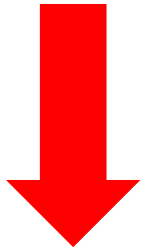
三种解决策略的对比

- 安全性 vs. 可用性的平衡
  - 前两种方案安全性好，但可用性方面至少存在一个严重缺陷
  - 放置假口令这种方案表现最好

潜在策略	客户端影响	服务器端影响	可扩展性	抗离线猜测	抗在线猜测	泄露检测
使用机器相关函数	无 (✓)	大 (✗)	✗	✓	✓	●
分布式密码学	大 (✗)	大 (✗)	✓	✓	✓	●
放置假口令	无 (✓)	小 (✓)	✓	✓	✓	✓

Honeywords：假口令

- 在口令文件中，置入一批假帐户，可用来检测口令文件泄露事件
  - 如果仅放置假帐户，真帐户的口令仍然会直接暴露，不能防御在线猜测攻击



在每个帐户中，置入 k-1 个假口令

要解决的核心问题：**如何生成假口令，做到以假乱真？**

Honeywords 的初步印象

- 方法1：修改真口令的特定位置（如最后2位） tiger03 →tiger82
- 方法2：**模拟词法** tiger03→L_5D_2 →trust52, ilove71,….

ID：loveu4ever@foo.com ； PW： tiger81 ********* ；
Name： Wang Lei; Birthday： 19810117

方法1：

tiger03	tiger82	tiger59	tiger15	tiger87
tiger17	tiger32	tiger81	tiger70	tiger60

方法2：

trust52	ilove22	tiger03	ilove54	tiger81
llove71	yuiop91	tiger66	yuiop10	trust49

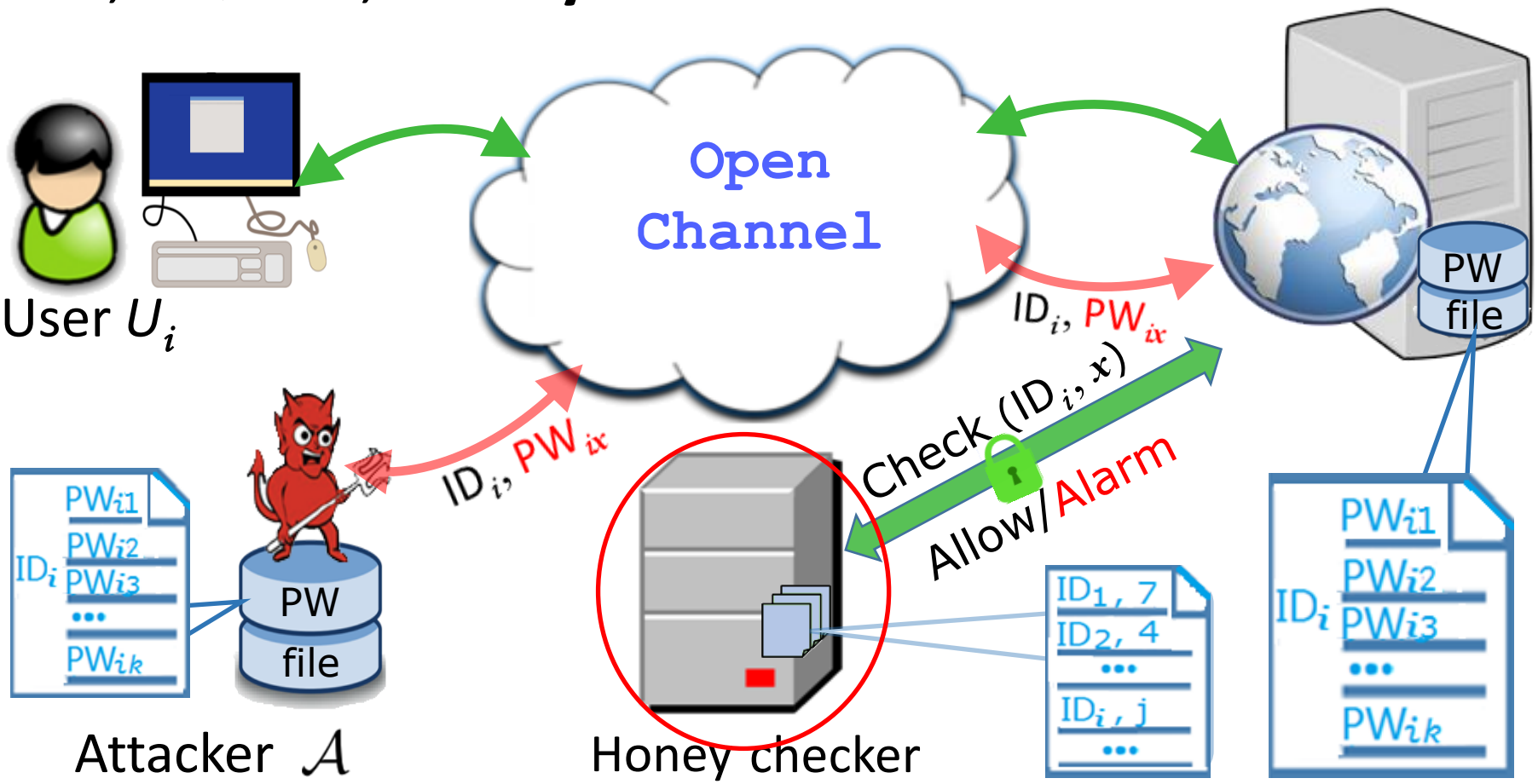
系统模型

- Client-Server 架构
- 三个参与方: 用户, 服务器, **Honey Checker**

报警条件:

1）单个用户假口令尝试次数超过 $\mathcal{T}_1$ (e.g., 3) ；

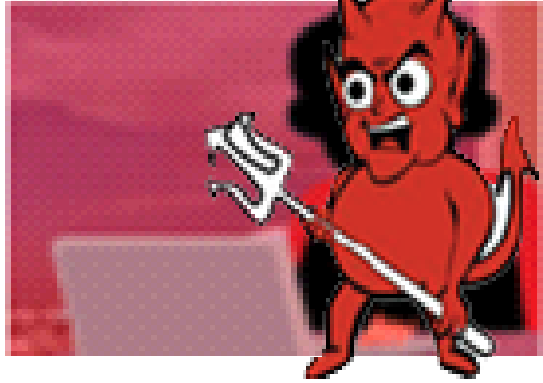
2）系统整体假口令尝试次数超过 $\mathcal{T}_2$ (e.g., 10^4) 。



攻击者能力假设

- 假定所有公开信息都可被攻
  如已泄露的口令集，网站口令
- 假定攻击者已经获得口令的
  意味着，攻击者获得用户的ID
- 根据攻击者是否拥有个人信
  - 漫步攻击：不知道用户个人
  - 定向攻击：知道用户个人信

攻击者目标



- **目标1：**以最少的猜测次数 x_1，找出给定用户 ID_i 的真口令；

  显然 x_1 应满足：$x_1 < \mathcal{T}_1 = 3$

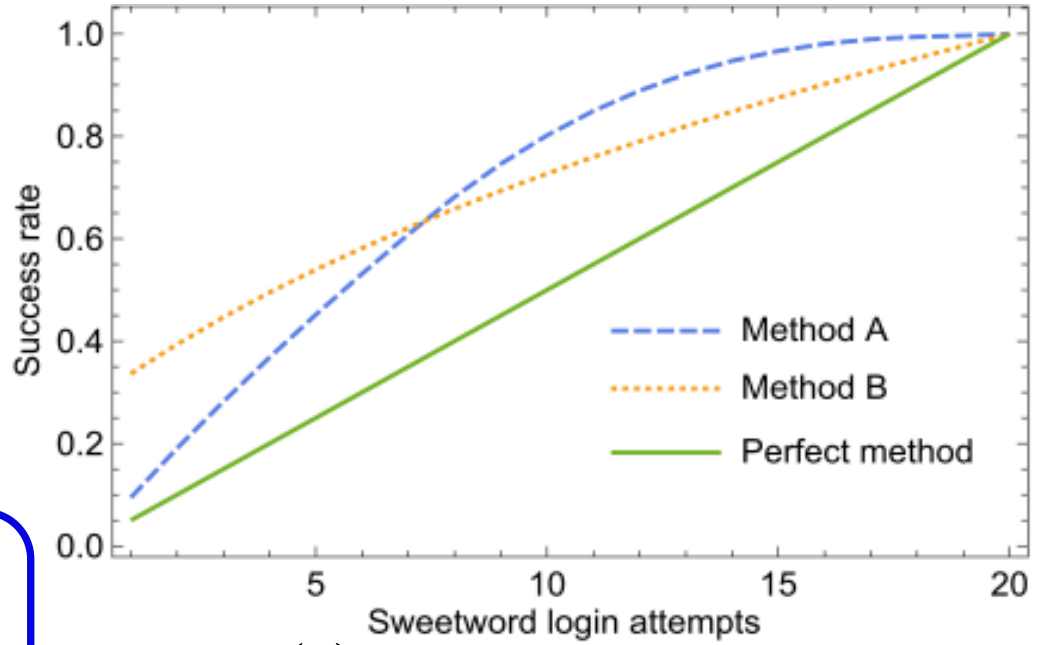
- **目标2：**总体允许 x_2 次猜测的情况下，找出尽量多的真口令

  显然 x_2 应满足：$x_2 < \mathcal{T}_2 = 10^4$

Honeywords方法评价指标

- 平滑（Flatness）图
  - 图中点 (x, y) 表示，对给定用户猜测x次，猜中真口令的概率。

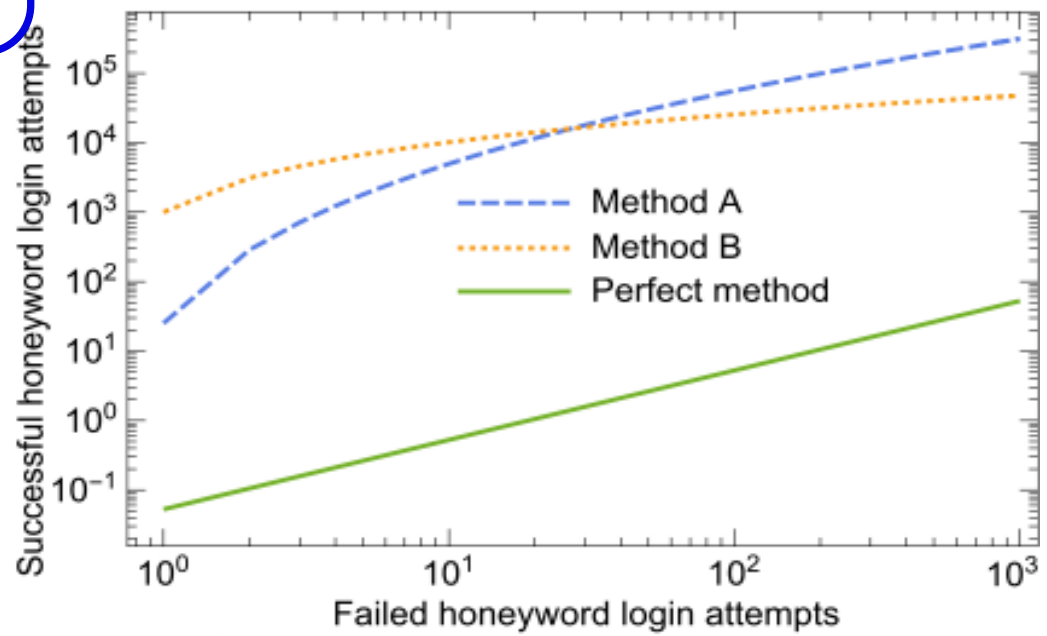
衡量honeyword生成方法的平均安全强度。



(a) Flatness graph.

- 成功次数（Success-number）图
  - 图中点 (x, y) 表示，对全体用户允许最多x次失败猜测，可猜中的用户数量。

衡量honeyword生成方法的worse-case安全强度。



(b) Success-number graph.

Juels-Rivest 的四个honeywords生成方法【CCS 2013】

- 修改口令尾部（Tweak-tail)：修改尾部后2位字符，字符类型保持不变
  比如，口令abcd12，可生成 abcd38, abcd91
- 模拟词法（Modeling syntax）：保持口令结构不变，替换掉对应的组件
  比如，口令password@1，结构为$L_8S_1D_1$，可生成abcdefg*7, iloveyou#3
- 混合方法（Hybrid)：先使用modeling syntax方法，再使用修改尾部方法
  比如，口令password@1，先生成iloveyou#3，再生成password!2, iloveyou%6
- 简单模型 （Sample model）：通过一系列启发式步骤，逐字符生成 honeywords

采用的大规模真实数据集

- 11个口令集
  - 3个含PII的口令集
  - 近一亿的真实口令
- 3辅助PII数据集

TABLE I. BASIC INFO ABOUT OUR 10 PASSWORD DATASETS†

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.

TABLE II. BASIC INFO ABOUT OUR PII DATASETS†

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	69,324	Email, User name, Name, Birthday

设计 Honeywords 攻击方法的挑战

- 传统口令猜测攻击要解决的问题

  如何以尽可能少的猜测次数，更快地猜测出用户的口令，即**按口令的使用概率，从低到高对口令进行排序**。

- Honeyword猜测攻击要解决的问题
  - 如何对某个用户的 k 个口令，**按口令是真口令的概率，从低到高进行排序**？
  - 如何对所有n个用户的n*k个口令，按口令是真口令的概率，从低到高进行排序？

设计**口令概率模型**，如PCFG, Markov

设计新的**假口令概率模型**。

针对单个用户k个口令的两种攻击策略

- Top-PW：直接按照口令的频率进行排序（即传统口令猜测方法）
- Norm Top-PW：先对每个用户的k个口令进行平滑处理，再频率归一化，最后再排序
  - 平滑的原因：由于测试集中的口令不一定在训练集中，如果没有平滑，这些口令都被认为是真口令的概率为0，归一化后，会造成该用户其他口令的概率被高估。
  - 平滑方法：加1平滑。计算口令的频率时，对计数进行加1，即(count(pw)+1)/(count_all_pw+1)，然后再进行归一化。

针对全体用户的攻击策略

- **整体思路：从每个用户的排序，扩展到所有用户的攻击排序**

  1）对每个用户的k个口令，计算概率，然后将每个用户概率最大的口令放入优先队列crackQueue；

  2）从crackQueue取出概率最大的口令；

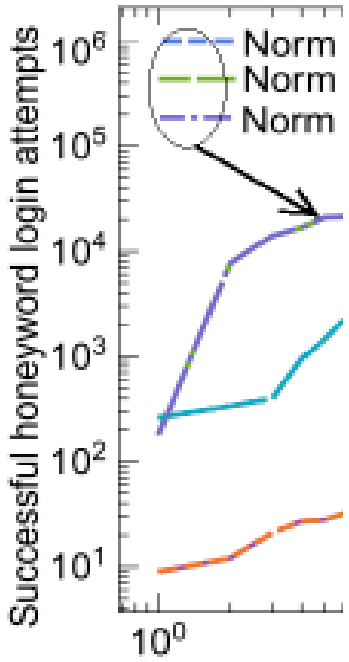
  3）进行在线验证，验证成功则转到2，不成功则转到4；

  4）计算上次攻击的用户，攻击次数是否达到上限。如果达到上限则转到2，否则对该用户剩余口令进行归一化（Top-PW方法不归一），将最大的口令放入队列crackQueue，然后转到 2。

针对Juels-Rivest方法的实证攻击实验

Succ-number 指标

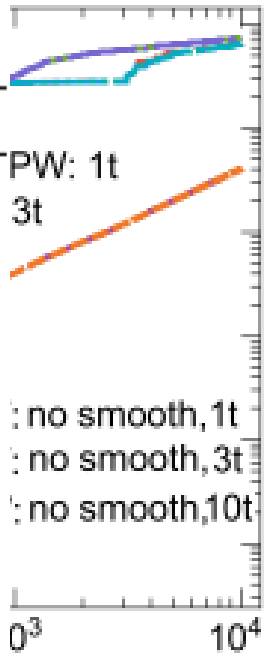
Norm top-PW (smooth)攻击策略在允许1万次错误时，能成功猜中11.03% ~ 16.63%的用户。



(a) Attack

TABLE V. SUCCESS-NUMBER INFORMATION (%)

	Tweak-tail	Model-syntax	Hybrid	Simple model
Tianya	14.41%	13.04%	14.90%	**5.81%**
Dodonew	10.10%	9.06%	10.46%	**8.75%**
CSDN	18.78%	**15.75%**	18.39%	16.32%
12306	9.32%	**7.88%**	9.17%	9.51%
Rockyou	21.63%	7.35%	14.01%	**2.41%**
000webhost	9.56%	14.33%	16.86%	**4.56%**
ClixSense	16.87%	**5.27%**	9.52%	6.08%
Yahoo	24.25%	**7.61%**	13.81%	16.84%
Rootkit	20.39%	**12.72%**	17.82%	19.57%
QNB	20.99%	20.85%	20.97%	**20.48%**
Average	16.63%	11.39%	14.59%	**11.03%**

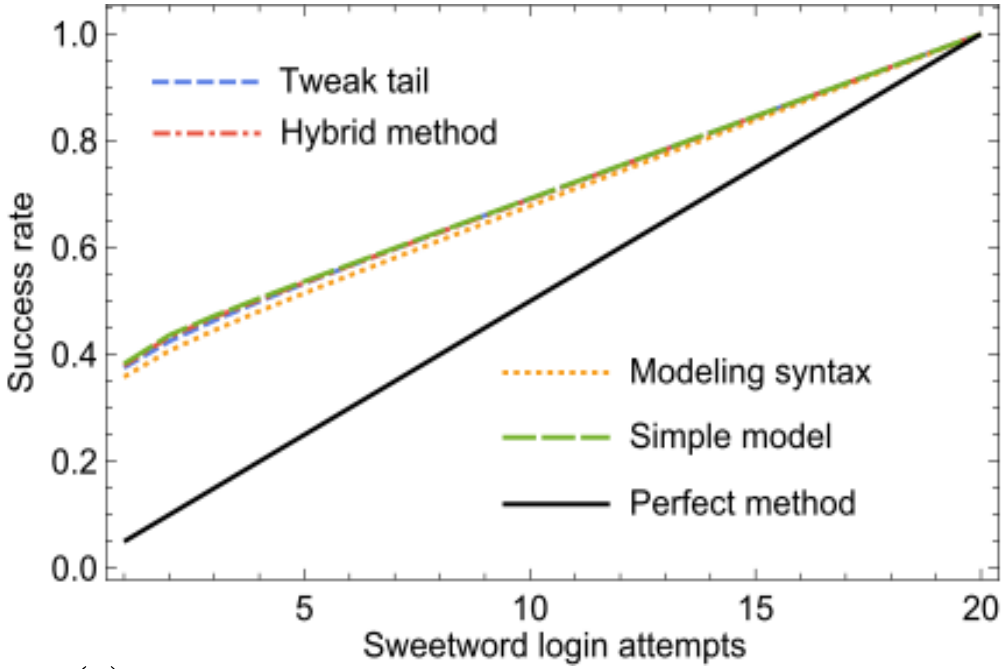


ethod.

针对Juels-Rivest方法的实证攻击实验

Flatness指标

k=20时，攻击者一次猜测成功率都在0.29以上。理想情况下，攻击者猜一次的猜中的概率应为0.05。



(e) The flatness graph of each method (k=20).

TABLE VI. ϵ-FLAT INFA[illegible] FOR EACH HONEYWORD METHOD.

	Tweak			le model
Tianya	**0.43**			4463
Dodonew	0.37			3828
CSDN	0.36			3978
12306	0.130			0.1327
Rockyou	0.5498	**0.4831**	0.5334	0.5035
000webhost	0.3550	0.3587	0.3594	**0.3541**
ClixSense	0.3055	**0.2221**	0.2758	0.2943
Yahoo	0.2785	**0.2080**	0.2527	0.2661
Rootkit	0.2293	**0.1636**	0.2052	0.2210
QNB	0.2348	0.2342	0.2355	**0.2313**
Average	0.3262	**0.2929**	0.3200	0.3230

如果攻击者知道用户个人相关信息，成功率 > 0.56 !

Juels-Rivest 四种方法的对比

- 优劣排序

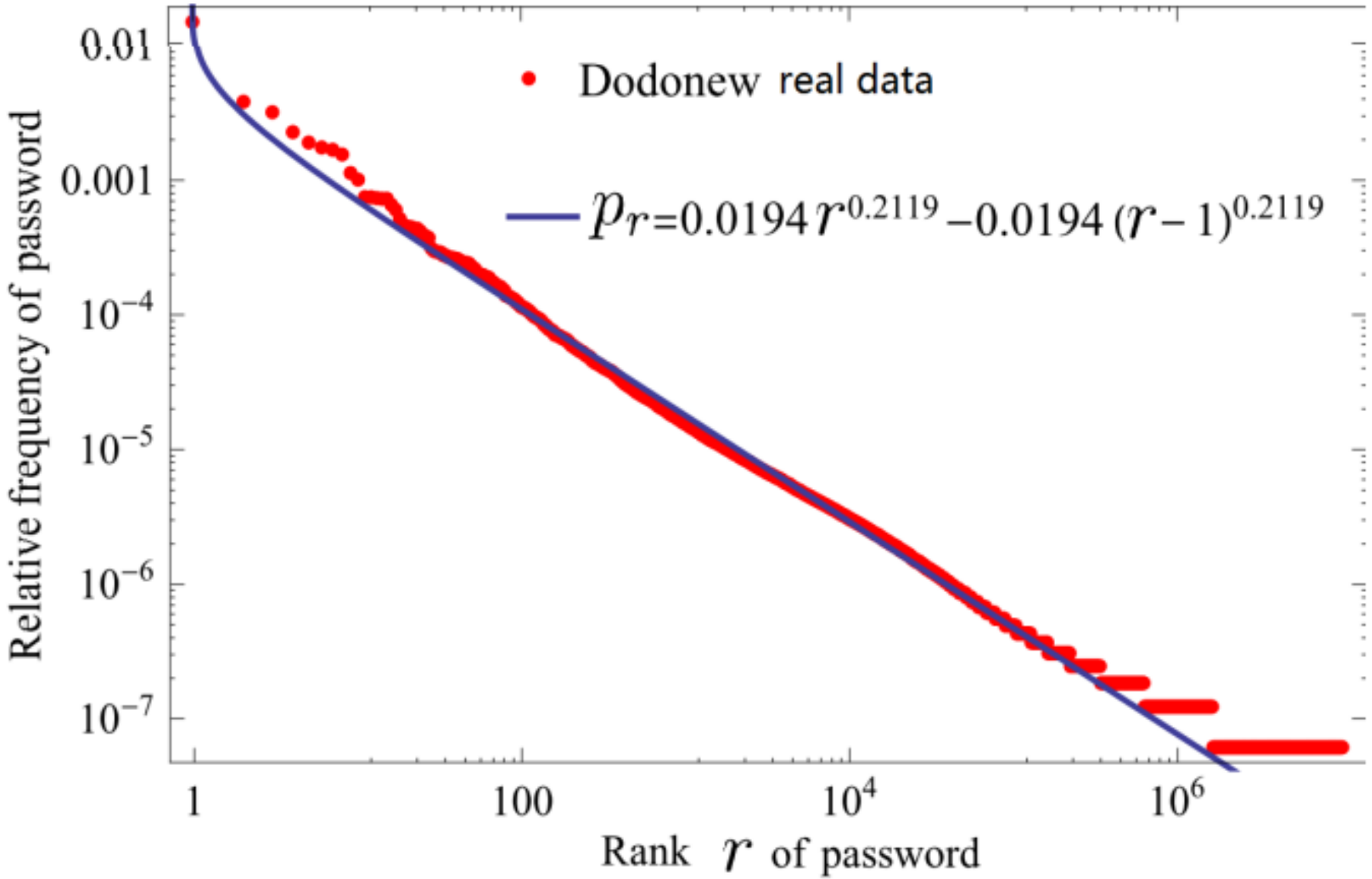
  Model-sy

- 都远未达到

- 四种方法本

$$\Pr(PW = sw_j | SW$$

要想
所有



攻击方法的理论分析(1)

定理1

设用户的k个口令为$(sw_1, sw_2, \ldots, sw_k)$，用户使用pw作为口令的概率为$\Pr_{PW}(pw)$，honeyword生成方法由真口令pw生成hw的概率为 $\Pr_{HW}(hw|sw)$。则已知用户的k个口令为$(sw_1, sw_2, \ldots, sw_k)$，真口令为$sw_j$的概率 $\Pr(PW = sw_j | SW = (sw_1, sw_2, \ldots, sw_k))$为

$$\frac{\Pr_{PW}(sw_j)\prod_{l\neq j}\Pr_{HW}(sw_l|sw_j)}{\sum_{t=1}^{k}\Pr_{PW}(sw_t)\prod_{l\neq t}\Pr_{HW}(sw_l|sw_t)}$$

攻击方法的理论分析(2)

Honeywords生成方法的几个性质

设T(pw)为口令能够生成的honeyword空间(包括pw自身)，则一个好的honeyword生成算法应该有性质1

性质1 $sw' \in T(sw) \Rightarrow T(sw') = T(sw)$

这一性质保证了k个口令中，每个口令都能生成其他口令，否则该口令必为假口令。这一性质将整个口令空间划分为一系列等价类，能互相生成口令归为一类。以下都假设有性质1.

性质2 $\forall sw_3 \in T(sw_1) = T(sw_2), \Pr_{HW}(sw_3|sw_1) = \Pr_{HW}(sw_3|sw_2)$

这一性质是说，honeyword生成的概率只与该等价类有关，与具体的口令无关。所以不妨将 $\Pr_{HW}(sw'|sw)$ 简记为 $\Pr_{HW}(sw'|T(sw))$，从而 $\Pr(PW = sw_j|SW = (sw_1, sw_2, \ldots, sw_k))$ 可化简为 $\dfrac{\frac{\Pr_{PW}(w_j)}{\Pr_{HW}(w_j|T(w_1))}}{\sum_{t=1}^{k} \frac{\Pr_{PW}(w_t)}{\Pr_{HW}(w_t|T(w_1))}}$

攻击方法的理论分析(3)

性质3

$$\forall sw_3, sw_2 \in T(sw_1), \Pr_{HW}(sw_3|sw_1) = \Pr_{HW}(sw_2|sw_1)$$

这一性质是说，生成honeyword的概率分布为平均分布。Juels-Rivest的前三种方法都满足该性质，第四种方法近似满足该性质。在该性质下，概率 $\Pr(PW = sw_j|SW = (sw_1, sw_2, \ldots, sw_k))$ 可以化简为

$$\frac{\Pr_{PW}(sw_j)}{\sum_{t=1}^{k} \Pr_{PW}(sw_t)}$$

该式即对应前述的Norm Top-PW攻击者。

直觉上讲，如果用户A的k个口令中有2个高频口令，用户B只有1个高频口令，显然B的高频口令更有可能是真口令，所以需要归一化。

攻击方法的理论分析(4)

性质4

$$\forall sw_1, sw_2, sw_3, \Pr_{HW}(sw_3|sw_1) = \Pr_{HW}(sw_3|sw_2)$$

这一性质是说，honeyword生成的概率与真口令无关。这一性质的直接推论是T(pw)对任意pw都是整个口令空间。在该性质下，概率 $\Pr(PW = sw_j|SW = (sw_1, sw_2, \ldots, sw_k))$ 可以化简为

$$\frac{\frac{\Pr_{PW}(sw_j)}{\Pr_{HW}(sw_j)}}{\sum_{t=1}^{k} \frac{\Pr_{PW}(sw_t)}{\Pr_{HW}(sw_t)}}$$

由上述公式可知，当且仅当 $\Pr_{PW}(sw_j) = \Pr_{HW}(sw$ 法最优，即 $\Pr(PW = sw_j|SW = (sw_1, sw_2, \ldots, sw_k))$ 为1/k。

知己知彼，百战不殆。
如果我们知道了攻击者如何最优攻击，就可设计相应最优防御办法。

我们提出的honeyword生成方法

基本设计原则

honeywords的概率应该尽可能与它是真口令的概率一致，即

$$\Pr_{PW}(sw_j) = \Pr_{HW}(sw_j)$$

从而：

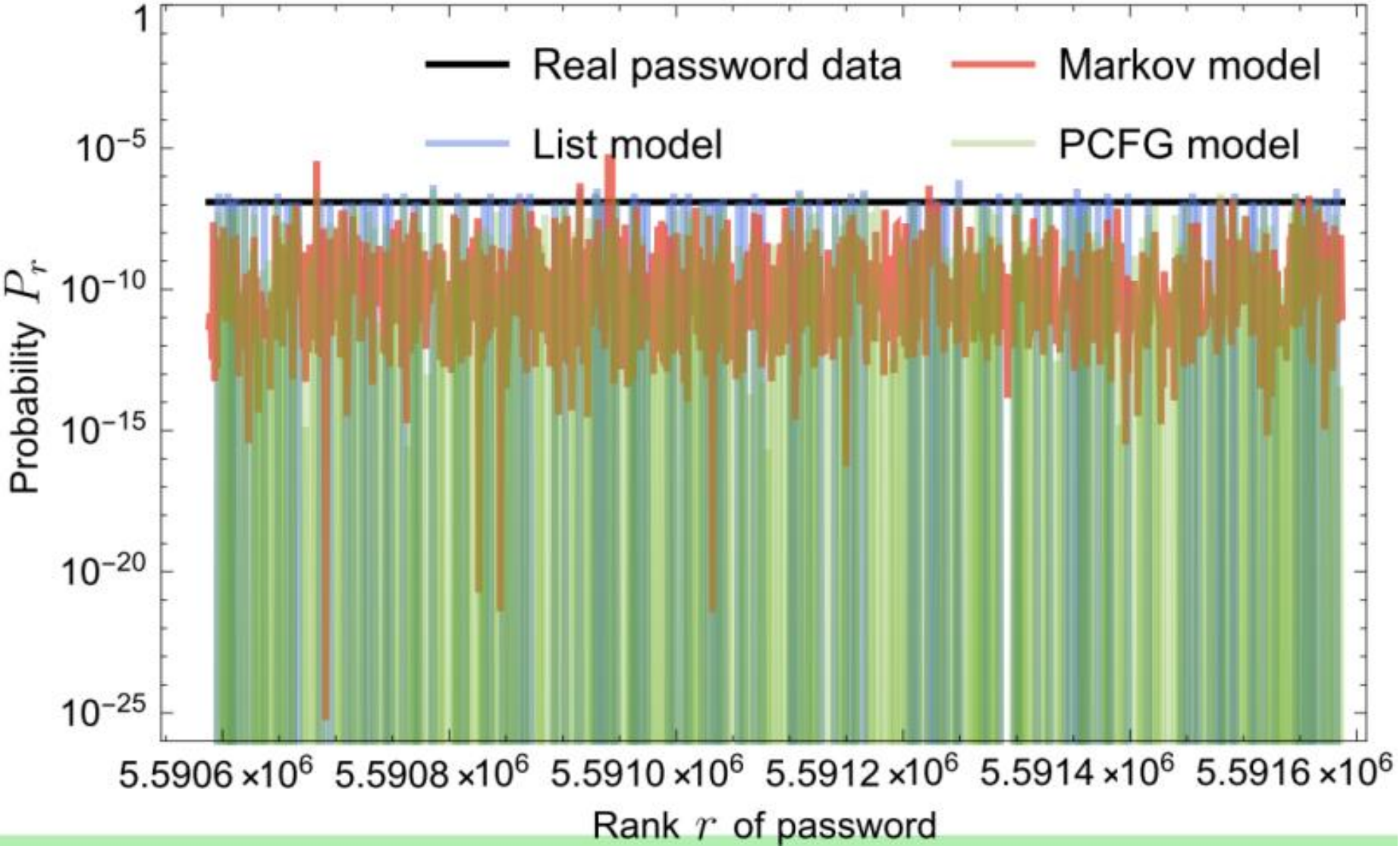
$$\frac{\frac{\Pr_{PW}(sw_j)}{\Pr_{HW}(sw_j)}}{\sum_{t=1}^{k} \frac{\Pr_{PW}(sw_t)}{\Pr_{HW}(sw_t)}} = \frac{1}{k}$$

综合使用口令概率模型生成honeywords

- List
- PCFG
- Markov

三个口令概率模型的对比

- List 、 PCFG 、 Markov



三个概率模型各有缺陷和优点

- List
  - 对高频口令的概率估计准确。
  - 无法生成不在测试集中的口令。
- PCFG
  - 对某些口令，如1qa2ws3ed，严重低估其概率。
  - 无法生成某些口令（训练集中，没有对应的结构或组件）。
- Markov
  - 由于采用平滑技术，能生成整个口令空间中的口令。
  - 对某些过长的口令（超出Markov的阶），如110120130，严重低估其概率。

被低估的典型口令

TABLE V. THE VALUE OF $\frac{\Pr_{\mathrm{PW}}(\cdot)}{\Pr_{\mathrm{HW}}(\cdot)}$ GIVEN BY EACH PASSWORD MODEL FOR TYPICAL PASSWORDS (DODONEW-TR VS. DODONEW-TS).

Typical password	Probability $\Pr_{\mathrm{PW}}(\cdot)$	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

使用混合的方法来生成honeyword

- 三个模型混合的方法：$\frac{1}{3}\text{List}+\frac{1}{3}\text{Markov}+\frac{1}{3}\text{PCFG}$

  步骤1：从List、PCFG、Markov中随机选择一个模型

  步骤2：根据该模型生成一个honeyword

- 效果
  - 除非一个口令被三种模型都低估了概率，才会被低估。
  - 虽然被低估口令的比例下降了，**仍然存在一些**。
  - 如何更准确的预测训练集中未出现口令的概率，依赖于更好的概率模型。

解决低估口令的问题

- **发现1：** List已经解决了流行口令的低估问题，剩余的被低估的口令都是低频非流行口令
- **发现2：** 被低估概率的低频口令通常修改尾部后，还是低频口令，并且同样被低估。
- **对策**

  对于被低估的口令，采用修改尾部的办法生成honeywords

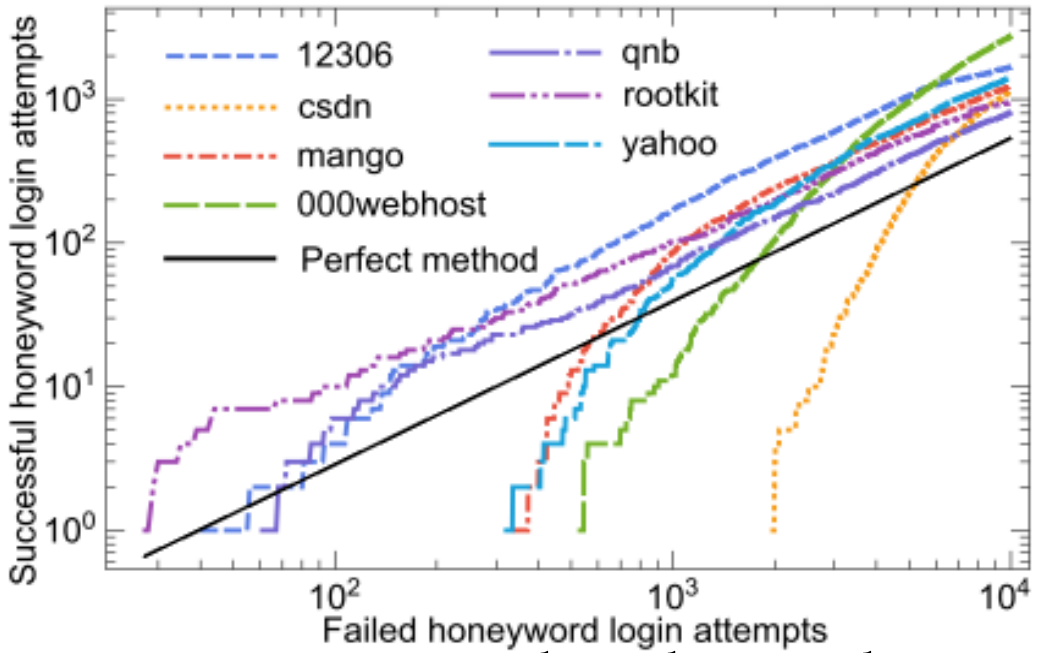
攻击结果

- Flatness指标

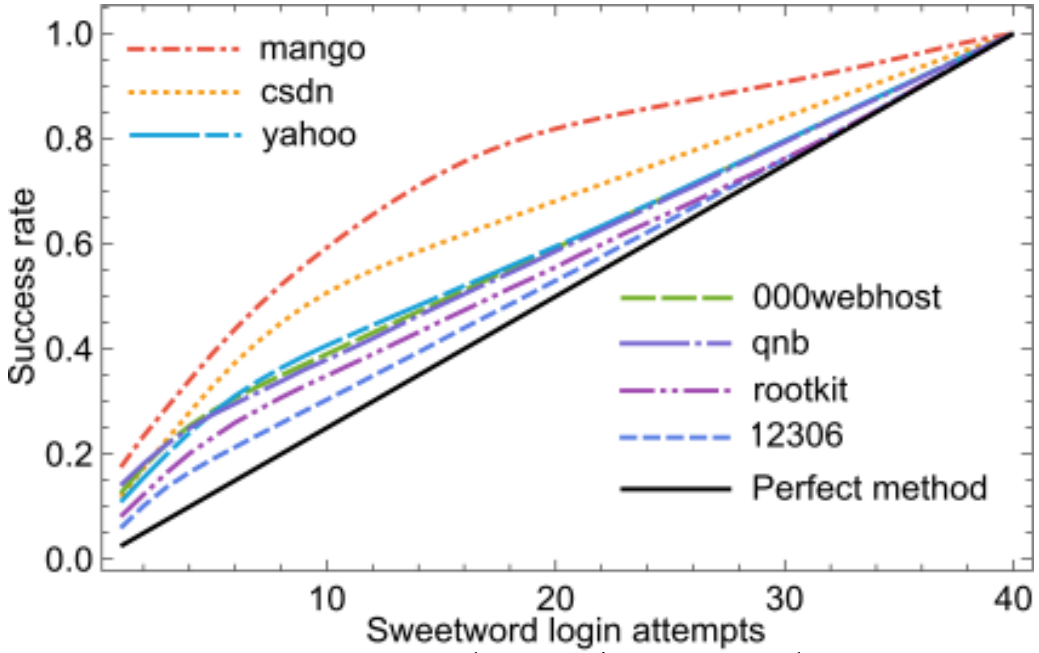
  攻击者猜一次，能猜中概率约7%。

- Succ-number指标

  攻击者被允许失败1万次，成功次数不超过3000次，成功率低于7.2%。



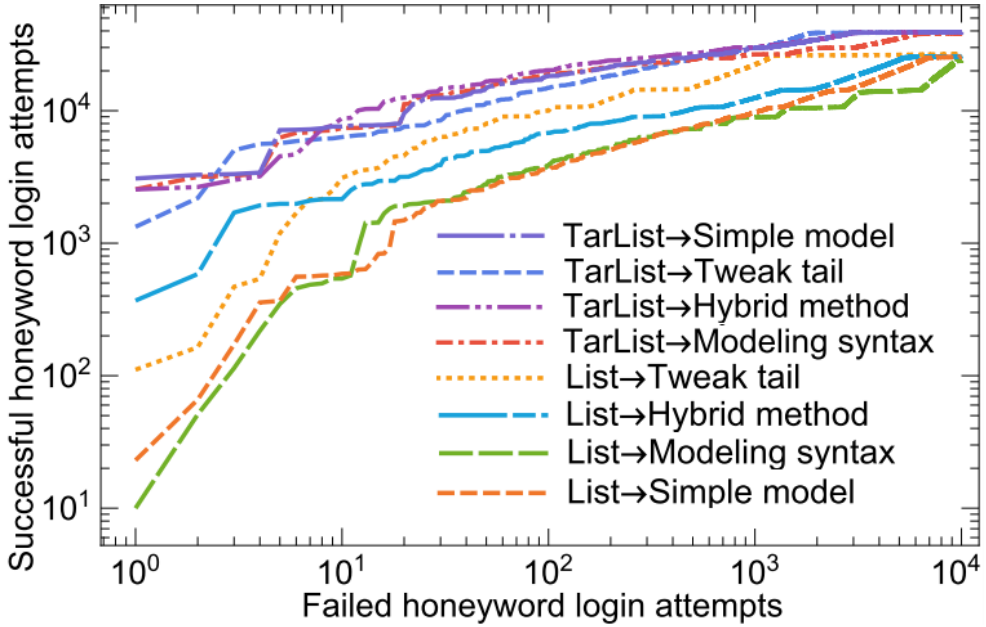
(b) Success-number graph of $\frac{1}{3}$List+$\frac{1}{3}$Markov+$\frac{1}{3}$PCFG.



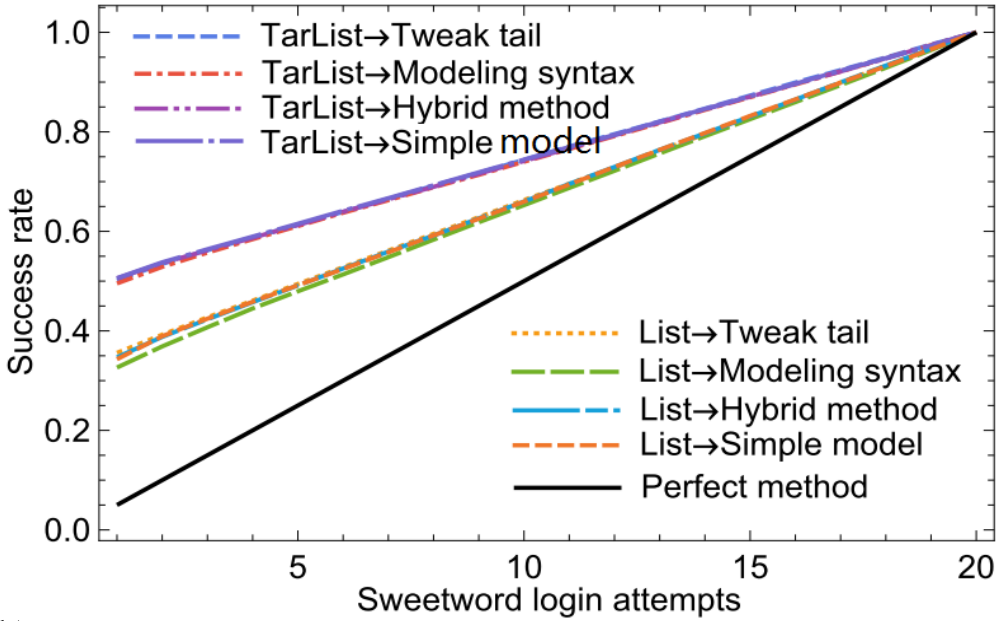
(e) Flatness graph of $\frac{1}{3}$List+$\frac{1}{3}$Markov+$\frac{1}{3}$PCFG.

基于个人信息的定向攻击

- CCS 2016我们提出一种利用用户个人信息猜测用户口令的方法——**定向口令猜测TarGuess;**
- 基于**TarGuess**，可以利用用户的个人信息更准确的 估计用户口令的概率。因此，在honeyword中，该模型可以更 准确的猜测用户的真口令。



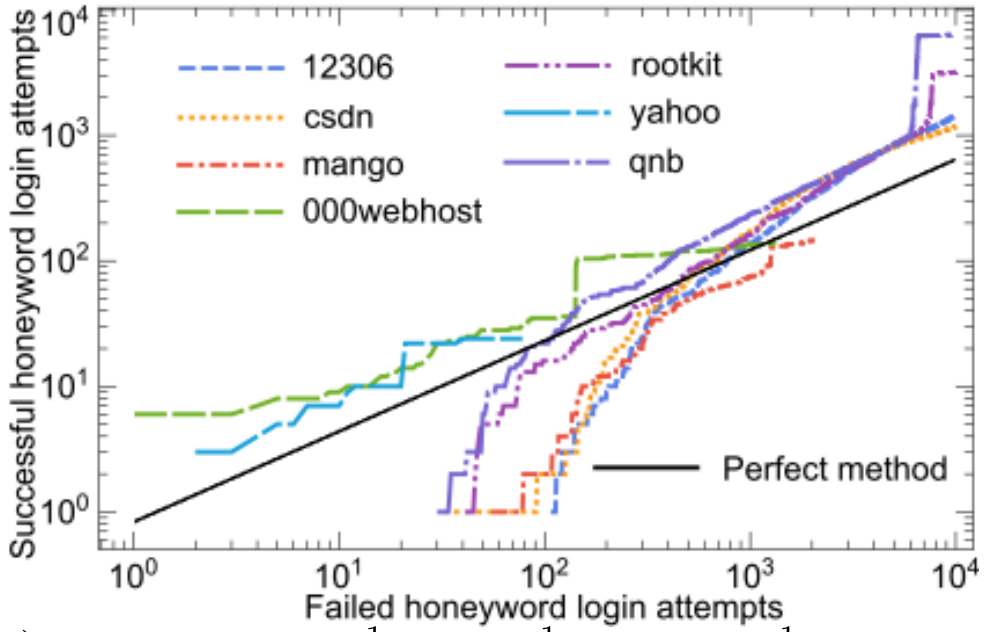
(a) Success-number graph: trained on PII-Dodonew-tr, tested on PII-Dodonew-ts.



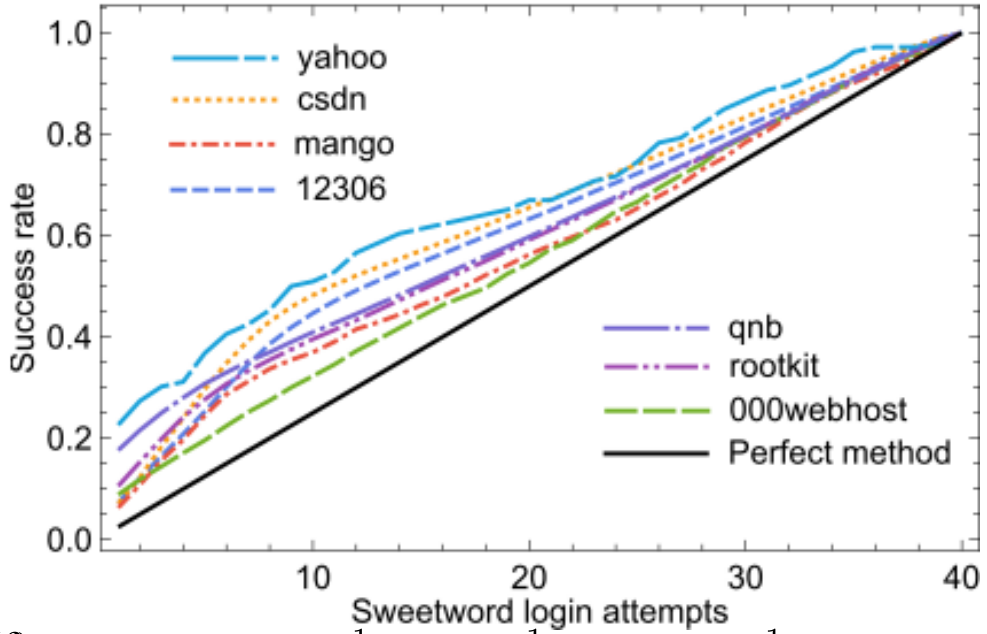
(b) Flatness graph: trained on PII-Dodonew-tr, tested on PII-Dodonew-ts.

抵御定向攻击的方法

- 使用对应的定向口令概率模型生成honeywords。
- 攻击结果：
  - Flatness图。攻击者猜测1次，猜中的概率低于12.1%。
  - Success-number图。攻击者猜测1万次，成功次数不超过6300次，成功率低于8.7%。



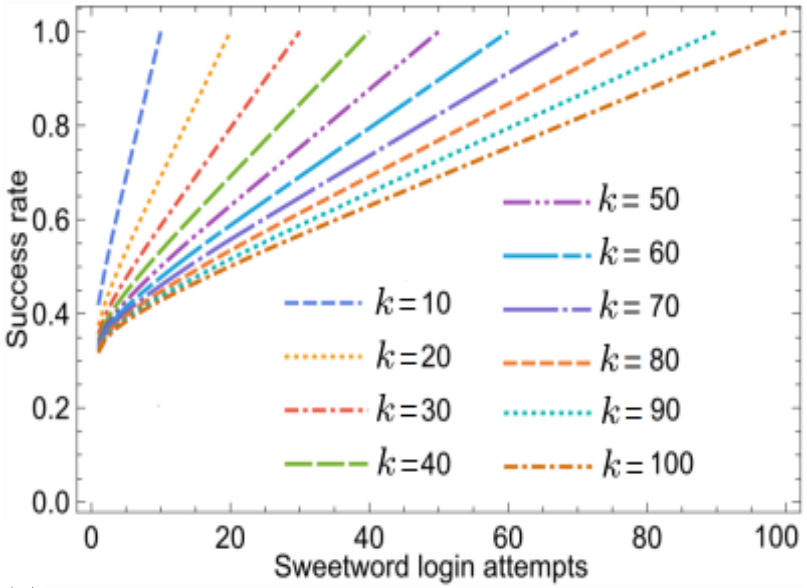
(c) Success graph of $\frac{1}{3}$TarList+$\frac{1}{3}$TarMarkov+$\frac{1}{3}$TarPCFG.



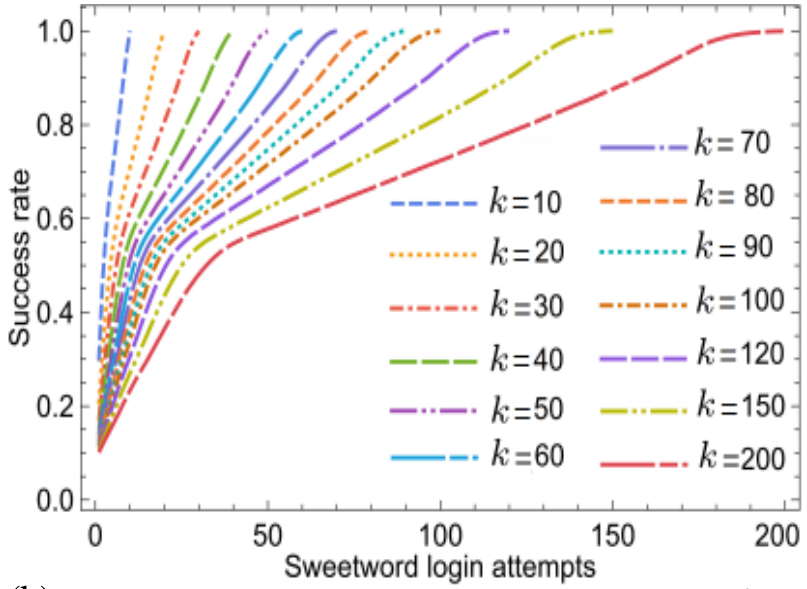
(f) Flatness graph of $\frac{1}{3}$TarList+$\frac{1}{3}$TarMarkov+$\frac{1}{3}$TarPCFG.

参数 k 对安全性的影响

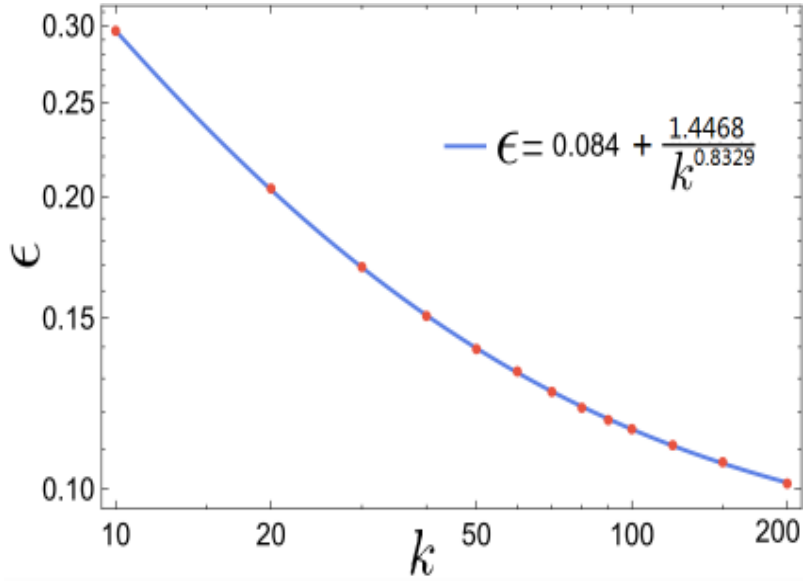
- List、PCFG、Markov三个模型混合。
  - 平滑系数 ϵ 最小为0.084 > 0.05。K=200时，ϵ=0.1。
  - k=40时，ϵ为0.13，到达可接受的安全级别。



(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.

Fig. 9. How the flatness curve varies with k, trained on Dodonew-tr and tested on Dodonew-ts. Here we use tweaking-tail (under $\mathcal{A}_1$) and our method $\frac{1}{3}$List+$\frac{1}{3}$Markov+$\frac{1}{3}$PCFG (under $\mathcal{A}_3$) as examples. The sub-fig(c) explores how ϵ (=$y_{x=1}$ in sub-fig(b)) evolves with k.

有待解决的问题

- 混合式Honeywords生成方法中，如何进行各模型的最优组合？ 也就是，如何选取 α， β，γ？

$$\alpha\,\mathrm{List}+\beta\,\mathrm{Markov}+\ \gamma\,\mathrm{PCFG}$$

其中，$\alpha+\ \beta+\gamma=1$.

- 如何设计更好的口令概率模型？

致 谢

wangdingg@pku.edu.cn (✉)
Tel: +86 18511345776

致 谢

wangdingg@pku.edu.cn (✉)
Tel: +86 18511345776