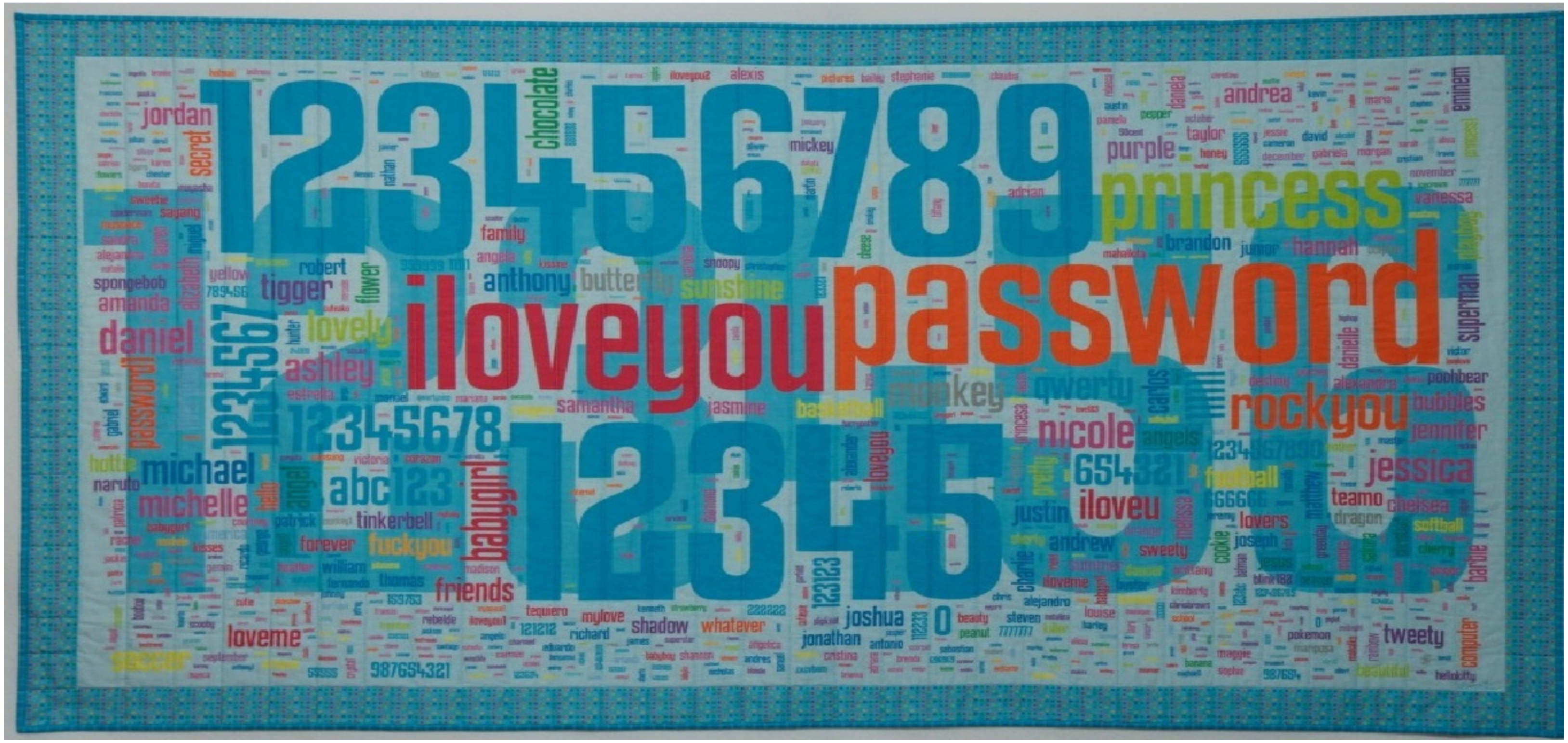
A Security Analysis of Honeywords

Ding Wang, Haibo Cheng, Ping Wang, Jeff Yan, Xinyi Huang

Password



123456789
princess
password
iloveyou
12345
12345678
1234567
rockyou
nicole
daniel
michael
michelle
ashley
abc123
jessica
tigger
lovely
anthony
sunshine
monkey
qwerty
babygirl
friends
tinkerbell
iloveu
654321
jordan
chocolate
purple
andrea
amanda
football
justin
andrew
joshua
shadow
whatever
loveme
tweety

Password-based authentication is still ubiquitous

Millions of passwords were leaked

- Thousands of data breaches were confirmed
  - 2016，**3141**【 Verizon 2016 Data Breach Report 】
  - 2016，**1093**【IRTC Identity Breach Report】
  - 201603-201703，**3785**【Thomas et al., CCS 2017】
  - 2011-2015，96 in China【http://www.liu16.com/post/476.html】
- Some popular websites didn't survive

  Yahoo, Dropbox, LinkedIn, Adobe, Xiaomi, CSDN, Tianya….

Password cracking

- The plaintext of most passwords can be recovered in a short time.
- Password distribution follows **Zipf law** [1]. Most users' passwords are in a small set of popular passwords.
- Websites should inform the users as soon as possible after a data breach occurs.

[1] Ding Wang et al. Zipf's Law in Passwords (2017 TIFS)

Websites did not realize the data breach

Websites	Account	Leak time	Notice time	Time interval
Myspace	360,213,049	2008	2016.07	8 years
Fling	40,757,760	2011	2016.05	5 years
LinkedIn	117 million	2012.06	2016.05	4 years
Dropbox	68,680,741	2012.06	2016.08	4 years
VK.com	100,544,934	2012	2016.06	4 years
Yahoo	3 billion	2013.08	2017.10	4 years
Yahoo	1 billion	2013.08	2016.09	3 years
Yahoo	0.5 billion	2014.08	2016.12	2 years
Weebly	43,430,316	2016.02	2016.10	8 months
Last.fm	43,570,999	2012.03	2012.06	3 months
Deloitte	5 million	2016.10	2017.03	5 months

How to make the data leakage detectable?

- **Traditional storage method**
  One sever (password file): (ID, pw)
- **Honeyword scheme** proposed by Juels and Rivest (CCS'13)
  Two severs:
  - **Password file: (ID, (sw_1, sw_2, …, sw_k))**
    one real password and **k-1 decoy passwords** (honeywords)
  - **Honeychecker: (ID, i)**
    the position of real password

Honeyword system

- One parameter
  - **k**: the number of sweetwords (one real password and k-1 honeywords). E.g., k=20.
- Two thresholds
  - $\boldsymbol{\mathcal{T}\downarrow 1}$: A user will be alarmed, when the honeyword login times of this user reaches $\mathcal{T}\downarrow 1$. E.g., $\mathcal{T}\downarrow 1 = 1$.
  - $\boldsymbol{\mathcal{T}\downarrow 2}$: The website will be alarmed, when the total honeyword login times of all user on the website reaches $\mathcal{T}\downarrow 2$. E.g., $\mathcal{T}\downarrow 2 = 10^4$.

How to generate honeywords

- Four Juels-Rivest methods
  - Tweak tail.
    Replace the tail characters with the same type characters. E.g., abcd12→abck40 (d→k, 1→4, 2→0).
  - Modeling syntax.
    Replace the segments with same type segments. E.g., abcd12→efgh40 (abcd→efgh, 12→40)
  - Hybrid.
    Hybrid of tweak tail and modeling syntax.
  - Simple model.
    A heuristic method that generates passwords character-by-character.

Our contribution

Focus on the honeyword generation method:

- Propose an efficient distinguish attack.
- Propose two security metrics based on attack.
- Evaluate the four Juels-Rivest methods on real datasets.
- Evaluate the password probability model method.

Efficient distinguish attackers

The order of attack:

- For a given user and his k sweetwords (sw_1, sw_2, …, sw_k).
- For n users on the website and their n×k sweetwords.

A straightforward idea:

- Top-PW: The decreasing order of probability $Pr(sw_i)$.

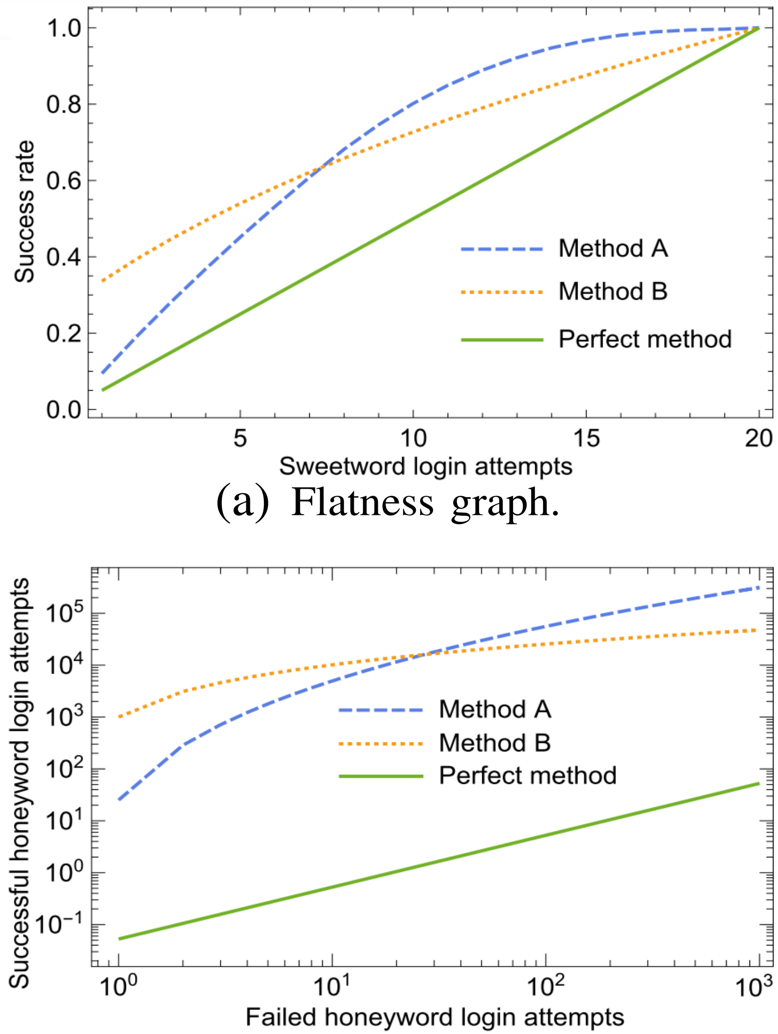
Efficient distinguish attackers

A more efficient method:

- Norm top-PW: The decreasing order of normalized probability $Pr(sw_i)/\Sigma_t Pr(sw_t)$.
  - For a given user, the order is the same as Top-PW.
  - For all users, the order is adaptive:
    1. Compute $Pr(sw_i)/\Sigma_t Pr(sw_t)$ for every sweetword.
    2. Crack the user with the maximum sweetword.
    3. If succeed, exclude the user and go back to Step 2. If fail, normalize the remaining sweetwords of the user and go back to Step 2.

Two security metrics

- **Flatness graph**
  The point (x,y) means a given user can be successfully cracked with y probability when logged in x times.
- **Success-number graph**
  The point (x,y) means y users on the website can be successfully cracked when logged in x times with honeywords.



(a) Flatness graph.

(b) Success-number graph.

Real password datasets

- 10 datasets
  - 104.36 million passwords
  - 9 different web services

TABLE I. BASIC INFO ABOUT OUR 10 PASSWORD DATASETS†

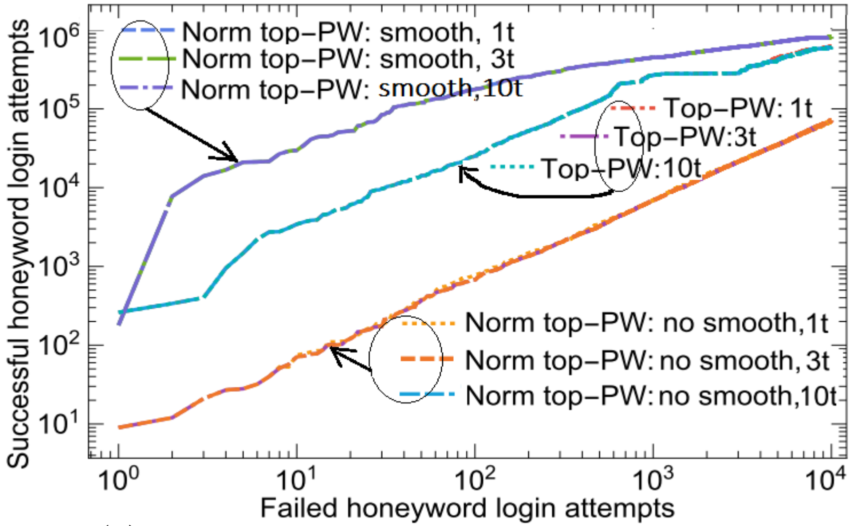
Dataset	Web service	Language	When leaked	Total PWs	With PII
Tianya	Social forum	Chinese	Dec., 2011	30,901,241	
Dodonew	E-commerce	Chinese	Dec., 2011	16,258,891	
CSDN	Programmer	Chinese	Dec., 2011	6,428,277	
Rockyou	Social forum	English	Dec., 2009	32,581,870	
000webhost	Web hosting	English	Oct., 2015	15,251,073	
Yahoo	Web portal	English	July, 2012	442,834	
12306	Train ticketing	Chinese	Dec., 2014	129,303	✓
ClixSense	Paid task platform	English	Sep., 2016	2,222,045	✓
Rootkit	Hacker forum	English	Feb., 2011	69,418	✓
QNB*	E-bank	English	April, 2016	79,580	✓

†PW stands for password, PII for personally identifiable information.
*QNB passwords are from e-Bank and used as high-value targets.

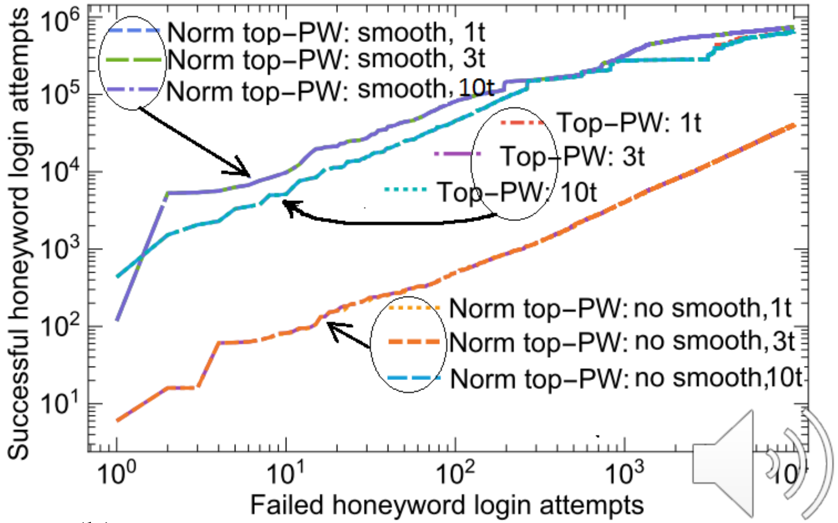
Evaluate the four Juels-Rivest methods

Success-number graph

- Norm top-PW(smooth): At least **615,664 (8.75%)** users are sucessfully cracked when the honeyword login times reaches $\mathbf{10^4}$ (on dodonew-ts).
- Expected value: **526** (10^4/19)



(a) Attacks on the tweaking-tail method.

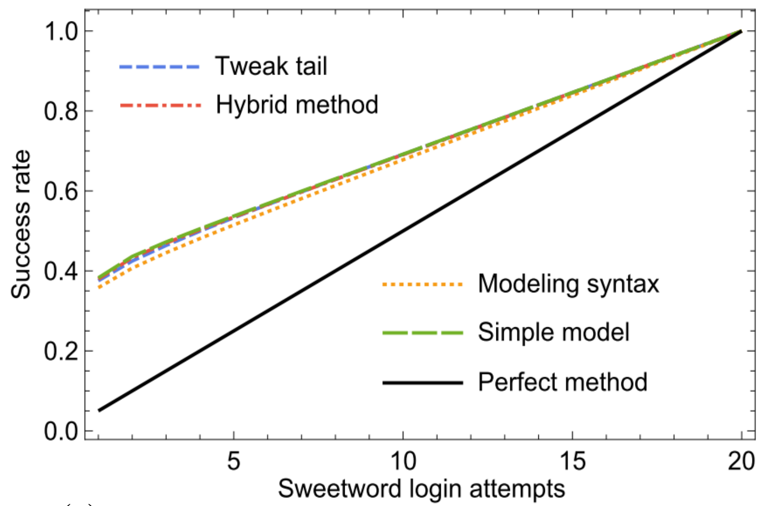


(b) Attacks on the modelling-syntax method.

Evaluate the four Juels-Rivest methods

Flatness graph

- Norm top-PW(smooth): At least **35%** users can be successfully cracked at the **first try** (on dodonew-ts).
- Expected value: **5%** (1/20)



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

Evaluate the four Juels-Rivest methods

- Same result on other datasets.
- The four methods fail to provide the expected security.
  - Success-number graph: on average at least **11%** users can be successfully cracked when the honeyword login times reaches 10^4.
  - Flatness graph: on average at least **29%** users can be successfully cracked at the first try.

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%**

TABLE VI. ϵ-FLAT INFO ABOUT EACH HONEYWORD METHOD.

	Tweak-tail	Model-syntax	Hybrid	Simple model
Tianya	**0.4368**	0.4400	0.4580	0.4463
Dodonew	0.3755	**0.3582**	0.3796	0.3828
CSDN	0.3664	**0.3437**	0.3716	0.3978
12306	0.1309	**0.1177**	0.1287	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	**0.231**[illegible]
Average	0.3262	**0.2929**	0.3200	0.3230

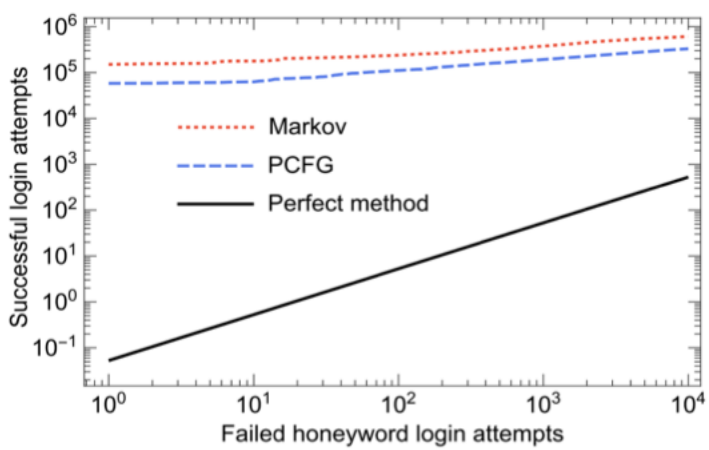
The inherent defect of the four Juels-Rivest methods

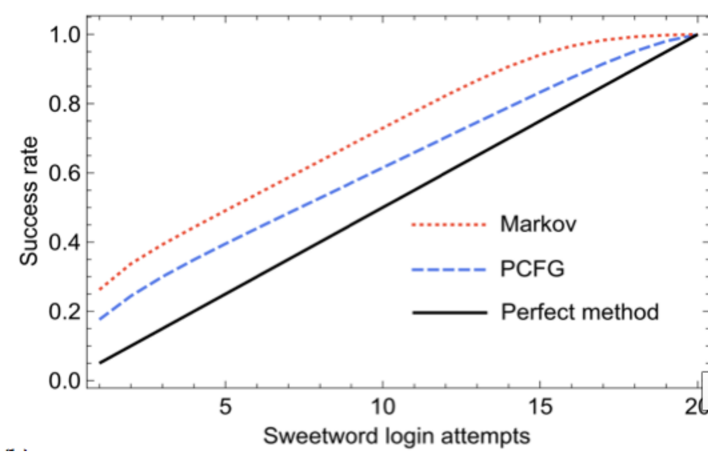
- The honeyword distribution is uniform distribution.
- The password distribution follows the Zipf law.
- The honeyword distribution should be the same as the password distribution.

Password probability model generating method

Password probability model generating method

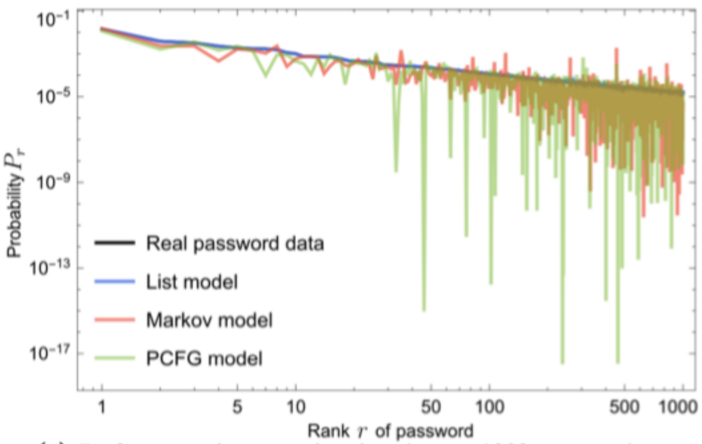
- Two state-of-the-art probability models:
  - PCFG-based model.
  - Markov-based model.
- Better on the flatness graph but still **vulnerable** on the success-number graph.



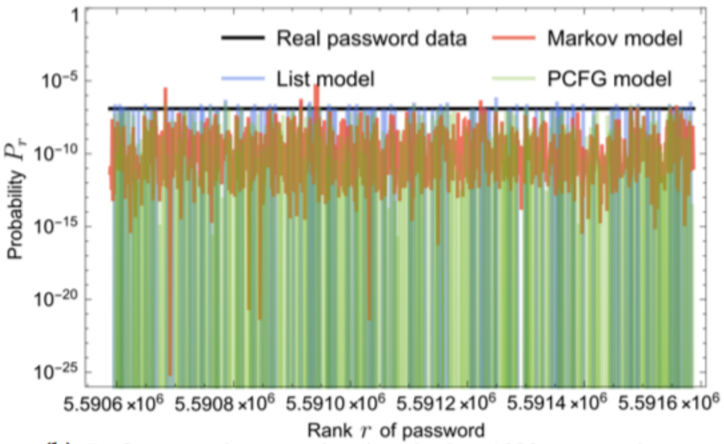


Password probability model generating method

- Every model is not good enough.
- The probability of a large number of passwords is **underestimated**.



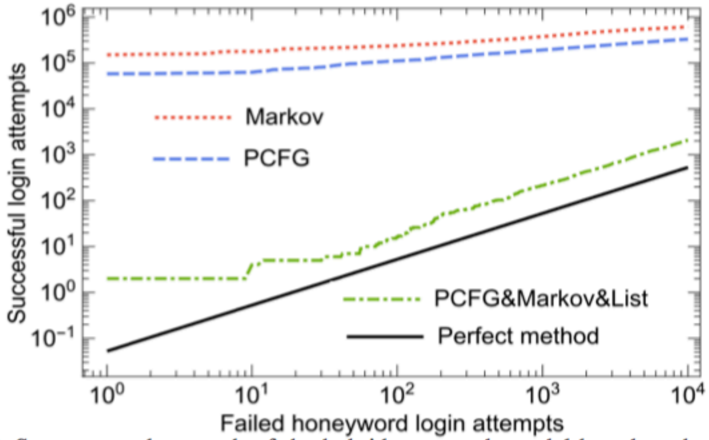
(a) Performance in approximating the top 1000 passwords.



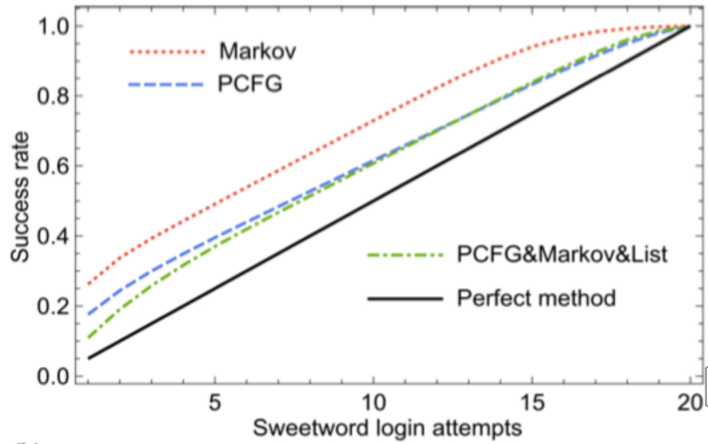
(b) Performance in approximating the last 1000 passwords.

Password probability model generating method

- A possible solution: hybrid model of password models. E.g., List&Markov&PCFG.
  $Pr_{List\&Markov\&PCFG}(pw)=1/3Pr_{List}(pw)+1/3Pr_{Markov}(pw)+1/3Pr_{PCFG}(pw)$
- Hybrid model is the best on both metrics.
  - Flatness graph: 11% (expected value 5%)
  - Success-number graph: 1113 (expected value 526)



(a) Success-number graph of the hybrid password- model based method.



(b) Flatness graph of the hybrid password- model based method.

Conclusion

Honeyword-generation method:

- The four methods proposed by Juels and Rivest have inherent defect.
- Password probability model method:
  - Single model is vulnerable.
  - Hybrid model is the best on success-number graph and flatness graph.

THANK YOU