

Improved Probabilistic Context-free Grammars for Passwords Using Word Extraction

Haibo Cheng¹, Wenting Li¹, Ping Wang¹, Kaitai Liang²

¹Peking University, ²Delft University of Technology

ICASSP 2021



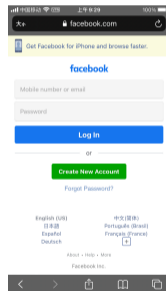
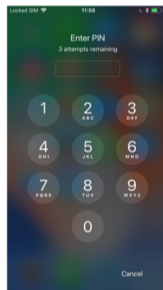
PEKING
UNIVERSITY



Background

Password

- 1 Dominant authentication method.
- 2 Including meaning segments.



Probabilistic context-free grammars (PCFGs)

- 1 Model password distributions.
- 2 Used for password strength meters and password guessing attacks.

Background

The core challenge to model passwords via PCFG models: *how to segment passwords*

① Existing segmentation methods:

① Simple segmentation based on char types (PCFG_W [1], PCFG_M [2]).

① “password123” → “password/123”.

② “1qa2ws3ed” → “1/qa/2/ws/3/ed”.

This is **inaccurate**.

② Improved segmentation with external dictionaries (e.g., PCFG_C [3]).

① “password” is identified as an English word;


② “1qa2ws3ed” is identified as a keyboard pattern.

But external dictionaries cannot **fully** and **accurately** cover the individual segments in passwords, because passwords are different from other types of texts.

② Inaccurate segmentation leads to misestimation of password probability.

Example: “jordan23” consists of Michael Jordan’s name and his jersey number. Current PCFG models divide it to two independent segments and underestimate its probability.

Our work

- ① A word extraction method for passwords, extracting individual segments (called words) from passwords.
- 
- ② A new password model—WordPCFG, achieving better performance on guessing attacks.

Our word extraction method for passwords

Extraction based on **cohesion** and **freedom**, inspired by a method for Chinese words [4].

- 1 Cohesion is the evaluation of a string's internal association.

$$\text{Coh}(s) = \min_{s_1 || s_2 = s} \text{PMI}(s_1; s_2), \text{ where } \text{PMI}(s_1; s_2) = \log \frac{p(s_1 || s_2)}{p(s_1) \cdot p(s_2)}.$$

- 2 Freedom is the evaluation of a string's independence from its context.

$$\text{Fdm}(s) = \min_{x \in \{r, l\}} \text{Fdm}_x(s), \text{ where } \text{Fdm}_l(s) = - \sum_{c \in \Sigma} \text{Pr}(c || s) \cdot \log \text{Pr}(c || s),$$

$$\text{Fdm}_r(s) = - \sum_{c \in \Sigma} \text{Pr}(s || c) \cdot \log \text{Pr}(s || c).$$

Our word extraction method for passwords

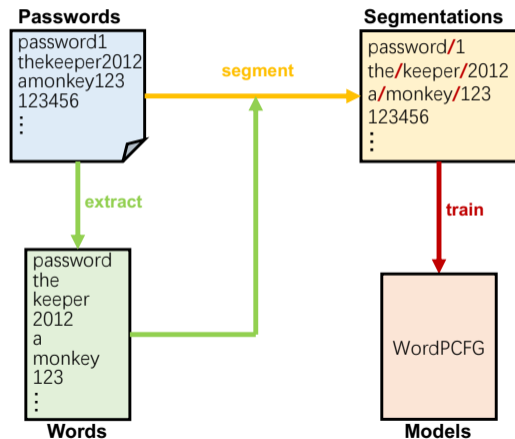
We extract a substring s in passwords as a word if $\text{Coh}(s) \geq T_c$ and $\text{Fdm}(s) \geq T_f$, where T_c and T_f are empirically set to 0.01 and 1.0, respectively.

Table 1: Extracted words from passwords via our method

Type	Examples
Keyboard pattern	qwertasdf 1q2w3e zxcvbn 1qaz 123456
English word	superstar skateboard lucky dragoon
Chinese pinyin	woaini woshi mima baobei haha
Name	steven wangming
Phrase	iloveu teamo byebye mylife howareyou
Hybrid	kobe24 jordan23 welcome2 4ever

Our WordPCFG

- 1 Extract words from passwords.
- 2 Segment passwords using the dictionary of words.
- 3 Train the probabilities of segments and templates.



The accuracy of WordPCFG

Leveraging WordPCFG for guessing attacks

- ① Attack: Crack passwords in descending order of probabilities.
- ② Performance:
 - ① WordPCFG achieves a significant improvement, when the guessing number climbs to 10^{10} .
 - ② WordPCFG can crack 83.04%–95.47% passwords, achieving a 12.96%–71.84% improvement.

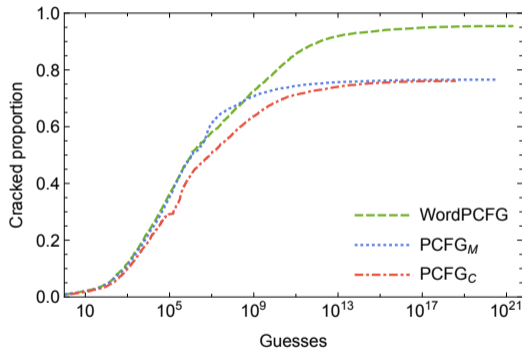


Figure 1: Rockyou

Thank you

References

- [1] Matt Weir et al. “Password Cracking Using Probabilistic Context-Free Grammars”. In: *Proc. IEEE S&P 2009*, pp. 391–405.
- [2] Jerry Ma et al. “A Study of Probabilistic Password Models”. In: *Proc. IEEE S&P 2014*, pp. 689–704.
- [3] Rahul Chatterjee et al. “Cracking-resistant password vaults using natural language encoders”. In: *Proc. IEEE S&P 2015*, pp. 481–498.
- [4] Shan He and Jie Zhu. “Bootstrap method for Chinese new words extraction”. In: *Proc. IEEE ICASSP 2001*. Vol. 1, pp. 581–584.