"© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works."
https://ieeexplore.ieee.org/document/9746203

WORDMARKOV: A NEW PASSWORD PROBABILITY MODEL OF SEMANTICS

Jiahong Xie[¶] *Rong Zhu*[¶] *Haibo Cheng*[¶*] *Ping Wang*[¶*] *Kaitai Liang*[§]

[¶]Peking University
{xjhshare, zhurong, hbcheng, pwang}@pku.edu.cn
[§]Delft University of Technology
kaitai.liang@tudelft.nl

ABSTRACT

To date there are few researches on the semantic information of passwords, which leaves a gap preventing us from fully understanding the passwords characteristic and security. We propose a new password probability model for semantic information based on Markov Chain, called *WordMarkov*, that can capture the semantic essence of password samples. Further, we evaluate our design via password guessing attacks, on six real-world datasets, and we show that WordMarkov obtains 24.29%–67.37% improvement over the state-of-the-art password probability models. We also reveal some interesting password habits from the semantics on "long" passwords. Based on those findings, WordMarkov achieves 75.35%–96.34% attack improvement.

***Index Terms*—** Markov Chain, password probability model, word segmentation, semantic information of password

1. INTRODUCTION

Password probability model is one of the cornerstones for password systems, and academia and industry have drawn their attentions to yield practical and secure designs over a decade, e.g., [1, 2, 3, 4, 5, 6]. The model has been widely investigated under the umbrella of password security, such as *password guessing attack* [1, 2, 7], *password-strength meters* [4, 8], *honeywords* [9], *honeyvaults* [10]. To provide strong password security one should fully capture a throughout and structural understanding on passwords [11]. Some research works may be only limited to superficial investigations on patterns [8, 12]. For example, they mainly focus on password length and the presence of printable characters via such as *PCFG* [1], *Markov* [2] and *LSTM* [4]. However, the "in-depth patterns", in particular the ones associated with the semantic information, should be further explored.

To address the problem, a few studies have been proposed in the literature, e.g., [3, 6]. But the most recent semantic password probability models may suffer from some practical shortcomings. In [3], Veras *et al.* used external corpus to investigate the semantic information. Unlike natural languages, passwords do not have a regular grammatical structure [13]. And this makes one difficult to accurately describe the semantic information of passwords from natural language dictionary. As for [6], Cheng *et al.* extracted password semantic information by a Chinese word extraction approach. Although they did not use external corpus, their PCFG model still cannot provide generalization [2]. It is impossible to calculate or sample structures that are not included in the training set.

1.1. Our Contributions

We propose a new password probability model based on the Markov Chain, called *WordMarkov*. The Markov model (hereafter we call it *Markov*) is one of the mainstream tools used to study password distribution in the context of password security. In the design of *Markov*, a password is treated as a whole, in which we only consider the association between characters. With the help of the word extraction method proposed in [6], we divide the password into independent semantic segments (also called *words*) and regard a password as multiple words connected. Due to the special features of *Markov*, our WordMarkov can identify the semantic information in the passwords more accurately (than the current research works), but also inherit the advantages of the *Markov* to provide superior generalization.

We further find some interesting password habits. Language words or names (e.g., "swan","4ever", "Carlos") are more frequently used in long passwords, than keyboard patterns (e.g., "qwer", "1q2w3e"). Note by long passwords we here mean those with the length > 12 characters. This indicates that using words/names pattern may help one to perform password guessing attacks. But the phenomenon for short passwords is the other way round.

Then we perform an empirical evaluation of the WordMarkov under the password guessing attack on six practical

---

*Corresponding author.

This research is supported by National Natural Science Foundation of China (6207071129) and by National Key R&D Program of China (2020YFB1806400).

"© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works."
https://ieeexplore.ieee.org/document/9746203

datasets, and the experimental results show that the WordMarkov is able to achieve 24.29%–67.37% improvement over the current models. In particular, for long password guessing, its improvement can reach 75.35%–96.34%.

2. PASSWORD PROBABILITY MODEL

Password probability models usually assign a probability value to each string [2]. They may help one to understand what makes users choose strong or weak passwords. And well-design models can further be applied in password strength meters [4], password cracking utilities [1] and honey encryption [14]. Generally speaking, there are two types of models to describe password distribution: one is *char-based* model [2, 4, 12, 14] and the other is *template-based* model [1, 3, 5, 10].

The char-based model is built on an intuitive idea that the probability of a user entering the current character only depends on his/her historical inputs (e.g., previously input characters). The password probability is calculated as the product of the probabilities on all the characters from a given password. Narayanan et al. [12] first used Markov to guess passwords in the char-based model. And later Ma et al. [2] proposed a more comprehensive study. A series of improvements have been proposed to optimize Markov, such as *Length normalization*, *End-symbol normalization* and *Laplace smoothing*. In 2016, Melicher et al. [4] applied deep learning to the password probability model and proposed LSTM of password. The overall framework is still based on Markov Chain. The difference is that when calculating the character probability, the model obtains the probability of the next character by inputting the prefix into the neural network, rather than simply counting the frequency of the string in the training set. The char-based model can provide good generalization, and it is possible to generate any passwords in the password space. However, because char-based model only focuses on characters, it is hard to reveal the semantics from passwords in the model .

The core assumption of template-based model is *users habitually choose several different meaning segments and group them together as a password* [6]. A password's probability is now the probability of its structure multiplied by those of its segments. Weir et al. [1] propose the first PCFG model for passwords, which divides a password into three types (namely letter, digit, special-symbol) of segments and marks the length for each segment. Cheng et al. [6] proposed WordPCFG in 2021, introducing the concept of *word* in PCFG by a Chinese word extraction approach. More specifically, the *words* are independent semantic segments of passwords. They introduced a new type of segments, *word*, to the original PCFG, which significantly improves the accuracy of capturing password distributions. The template-based model assumes that the each segment and template in the password are independent, and it is unable to generate those segments and templates which do not exist in the training set.

To get rid of this shortcoming, we investigate the semantic information in passwords based on Markov Chain, and further propose a semantic password probability model with both generalization and accuracy.

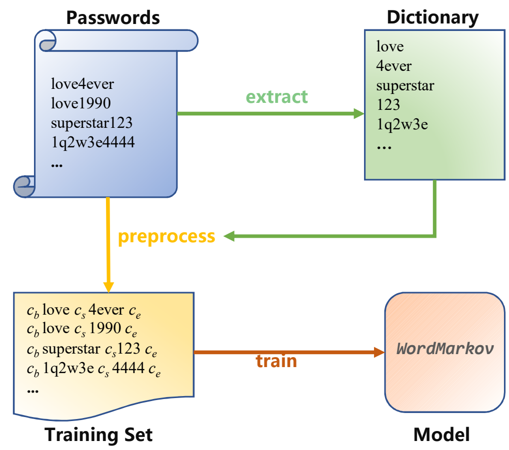


Fig. 1. The training process by WordMarkov

3. OUR APPROACH

3.1. Word Segmentation Method

Passwords do not have a regular grammatical structure [13] and thus, the word extraction method in natural language is not applicable here. We leverage the word extraction method proposed in [6] to extract the word dictionary from passwords. The words of a password are multiple independent semantic segments. We use the default configuration in the research [6], but we set the maximum and minimum length of each word in order to achieve practical performance (after several attempts): $min\_length = 3$, and $max\_length = 8$. In word segmentation, we recognize words in a password by *Positive Maximum Matching*, and the rest parts are also regarded as words even if they are not extracted into words.

3.2. The Design of WordMarkov

We first segment password into multiple independent semantic segments, calculate the probability of each segment, and finally multiply them to obtain the probability of password. Recall that the probability of a user entering the current character only depends on the historical characters, in the Markov Chain [12, 2]. The core assumption of Markov Chain is defined as:

$$\Pr\left(x_i \mid x_{i-1}, \ldots, x_1\right) = \Pr\left(x_i \mid x_{i-1}, \ldots, x_{i-k}\right) \quad (1)$$

For example, 2-order Markov calculates the probability of string $c_1c_2 \ldots c_{n-1}c_n$ as:

$$\Pr(s) = \Pr\left(c_2 \mid c_1\right) \cdot \Pr\left(c_3 \mid c_1c_2\right) \ldots \Pr\left(c_n \mid c_{n-2}c_{n-1}\right) \quad (2)$$

“© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”
https://ieeexplore.ieee.org/document/9746203

We use the *Laplace Smoothing method* to compensate for overfitting and improve the generalization. Then the transition probability of the Markov Chain is defined as follows:

$$\Pr\left(c_{k+1} \mid c_1 c_2 \ldots c_k\right) = \frac{\operatorname{count}\left(c_1 \ldots c_k c_{k+1}\right) + \sigma}{\sum_{c'_{k+1} \in \Omega} \left(\operatorname{count}\left(c_1 \ldots c_k c'_{k+1}\right) + \sigma\right)} \tag{3}$$

where Ω is the set of all characters, and σ is the parameter for *Laplace Smooth*.

However, *Markov* does not consider the semantics of the password [3]. For example, for the password “password123”, it is easy to recognize that the password is composed of two segments. But *Markov* believes that the probability of character “1” is strongly related to its previous three characters “ord”. This clearly contradicts to common sense.

To solve this problem, we introduce a *Split-Symbol* which divides password into independent semantic segments—*words*. The Split-Symbol clears the historical information of the previous word, and the new word can be generated from the Split-Symbol without relying on the historical information. We then have the core idea of WordMarkov:

$$\Pr_{\mathrm{pw}}(password) = \prod_{1}^{n} \Pr_{\mathrm{w}}(word_i) \tag{4}$$

As shown in Fig. 1, in the training phase of WordMarkov, we add a *Begin-Symbol* to the head and a *End-Symbol* to the tail of each password. The *Begin-Symbol*, *End-Symbol* and *Split-Symbol* are regarded as ordinary characters. For readers' convenience, we use the “c_b, c_s, c_e” to represent the *Begin-Symbol*, *Split-Symbol* and *End-Symbol* in the following.

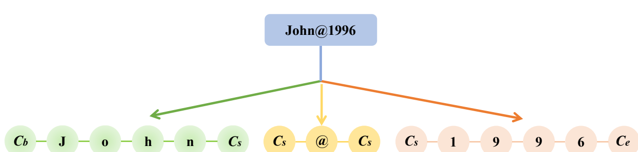


Fig. 2. An example of WordMarkov processing password

In Fig. 2, the password “John@1996” is divided into three words “John”, “@” and “1996”, and the WordMarkov can be regarded as a chain of multiple independent sub-chains. The probability of “John@1996” is calculated by 1-order WordMarkov as follows (for demonstration purposes, the order here is 1):

$$\begin{aligned}
&\Pr_{\mathrm{pw}}(\text{John1996}) = \Pr_{\mathrm{w}}(\text{John}) \cdot \Pr_{\mathrm{w}}(@) \cdot \Pr_{\mathrm{w}}(1996) \\
&\Pr_{\mathrm{w}}(\text{John}) = P(\text{J} \mid c_b)\, P(\text{o} \mid \text{J})\, P(\text{h} \mid \text{o})\, P(\text{n} \mid \text{h})\, P(c_s \mid \text{n}) \\
&\Pr_{\mathrm{w}}(@) = P(@ \mid c_s)\, P(c_s \mid @) \\
&\Pr_{\mathrm{w}}(1996) = P(1 \mid c_s)\, P(9 \mid 1)\, P(9 \mid 9)\, P(6 \mid 9)\, P(c_e \mid 6)
\end{aligned} \tag{5}$$

Table 1. Information of the leaked password datasets

Dataset	Language	Total	>12 chars	Percentage
Rockyou	English	32,581,870	1,143,282	3.50%
000Webhost	English	15,250,725	2,512,525	16.47%
Clixsense	English	2,222,046	150,631	6.77%
Tianya	Chinese	30,901,241	1,272,043	4.11%
Dodonew	Chinese	16,258,891	371,830	2.28%
CSDN	Chinese	6,428,277	467,985	7.28%

Table 2. Word count of the datasets

Dataset	Total	>12 chars	Percentage
Rockyou	7,510,232	754,568	10.04%
000Webhost	5,932,846	1,839,758	31.01%
Clixsense	948,363	126,135	13.30%
Tianya	6,362,217	886,940	13.94%
Dodonew	4,929,769	210,781	4.27%
CSDN	2,122,924	358,838	16.90%

Table 3. Cracking rate under the guesses

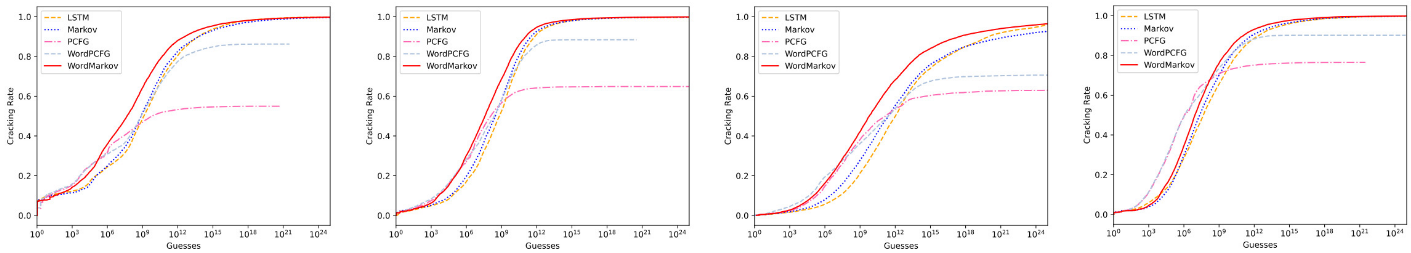
Dataset	Model	Total		>12 chars	
		10^9	10^{12}	10^9	10^{12}
000Webhost	WordMarkov	**42.17%**	**68.22%**	**21.03%**	**34.04%**
	Markov	27.84%	55.36%	2.84%	10.69%
	LSTM	21.29%	49.93%	2.39%	6.55%
	WordPCFG	36.21%	53.78%	3.72%	12.73%
	PCFG	38.19%	53.47%	16.82%	25.94%
CSDN	WordMarkov	**64.31%**	**88.04%**	**35.42%**	**58.38%**
	Markov	51.74%	82.50%	9.15%	31.91%
	LSTM	48.49%	80.47%	8.72%	26.26%
	WordPCFG	51.40%	77.49%	8.44%	26.17%
	PCFG	46.89%	53.27%	18.04%	24.43%
Rockyou	WordMarkov	**76.10%**	**93.15%**	**28.52%**	**47.89%**
	Markov	70.20%	90.68%	4.81%	22.11%
	LSTM	65.58%	88.68%	6.12%	18.82%
	WordPCFG	73.68%	88.33%	11.07%	27.31%
	PCFG	70.69%	75.15%	19.50%	24.00%
Dodonew	WordMarkov	**68.73%**	**94.73%**	**42.92%**	**70.84%**
	Markov	58.16%	92.66%	25.52%	56.49%
	LSTM	52.52%	91.15%	22.96%	51.39%
	WordPCFG	58.15%	85.83%	16.24%	45.06%
	PCFG	56.73%	64.14%	28.45%	32.81%
Clixsense	WordMarkov	**64.90%**	**85.63%**	**31.82%**	**47.06%**
	Markov	51.84%	78.44%	3.86%	14.97%
	LSTM	49.19%	82.33%	6.09%	22.17%
	WordPCFG	45.40%	67.32%	7.98%	23.57%
	PCFG	36.74%	52.27%	16.44%	20.43%
Tianya	WordMarkov	**77.96%**	**93.75%**	**32.26%**	**56.05%**
	Markov	71.73%	90.97%	7.24%	29.42%
	LSTM	63.82%	86.78%	7.02%	20.28%
	WordPCFG	69.71%	82.13%	12.97%	26.55%
	PCFG	69.88%	73.50%	17.25%	23.33%

4. EXPERIMENTS AND RESULTS

4.1. Datasets

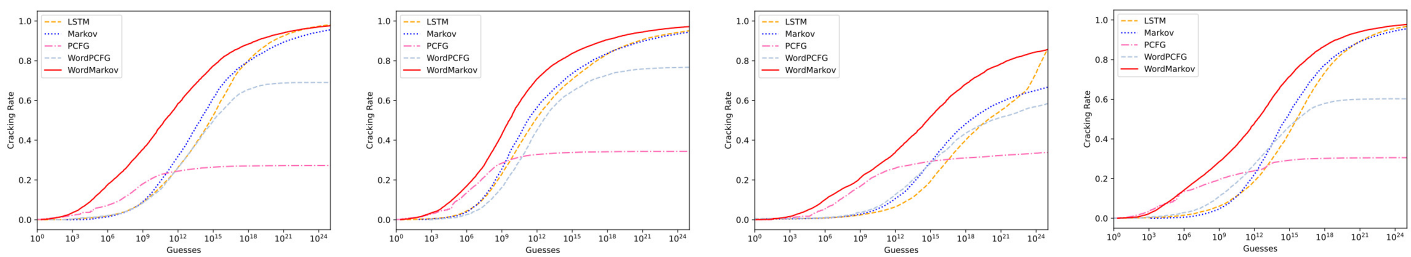
As is shown in Table 1, we collect over 103 million plaintext passwords from the public datasets [5, 15, 16] to simulate password guessing attacks. Our experiments thus can present

"© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works."
https://ieeexplore.ieee.org/document/9746203



(a) CSDN (b) Dodonew (c) 000Webhost (d) Rockyou

Fig. 3. The performance of WordMarkov on password guessing attack



(a) CSDN->12 chars (b) Dodonew->12 chars (c) 000Webhost->12 chars (d) Rockyou->12 chars

Fig. 4. The performance of WordMarkov on long password guessing attack

a scalable and comprehensive view of password habits in the real-world applications. In Table 2, the distribution of words in passwords is not uniform. The proportion of *words* in long passwords is relatively high, which indicates that long passwords contain more semantic information (i.e. more *words*). We state that the datasets have been widely used and studied in the previous works [4, 2, 6, 5, 15, 17]. And our research does not violate any ethical practice and privacy guidance, because we only exploit attacks on the public known datasets, which does not harm user privacy and application security.

4.2. Experiment Setup

We evaluate the efficiency of WordMarkov via offline guessing on the datasets given in Table 1. It is generally accepted that the more precise probability a password model can provide, the more efficient the attack can be [1, 2, 4, 6]. To preclude the impact of non-uniformly training data, we employ a random sampling; and then we adopt a cross-validation approach of k-*fold* (where $k = 4$). Specifically, we randomly split passwords into *four-folds*, and adopt any three of them as the training data and the last *one* for the testing data. In the evaluation, we use the Monte Carlo method [17] to show the results for the large number of guesses.

4.3. Evaluation of WordMarkov

We use the curve of cracking rate vs. guesses to show the performance of the WordMarkov model. As Fig. 3 and Table 3 shown, in the large-scale datasets (e.g., Dodonew, Rockyou), the performances of other models are close, and WordMarkov obtains 8.99%–24.29% improvement over others. For the small-scale sets (e.g., CSDN, 000Webhost), our improvement can reach 31.22%–67.30%, as compared to others. We further perform a series of experiments on long passwords in Fig. 4 and Table 3. The WordMarkov obtains 75.35%–96.34% improvement, outperforming other models in long password guessing.

5. CONCLUSION

We propose *WordMarkov* that is a novel semantic password probability model. We perform the experiments on the real-world datasets and the results show that WordMarkov achieves a significant improvement on password guessing and outperforms the state-of-the-art models' accuracy and generalization. And WordMarkov performs particularly excellent under long password guessing. We state that WordMarkov may be used in other models which are based on Markov Chain. The improvement for the word segmentation approach could be left as an interesting open problem.

"© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works."
https://ieeexplore.ieee.org/document/9746203

6. REFERENCES

[1] Matt Weir, Sudhir Aggarwal, Breno De Medeiros, and Bill Glodek, "Password cracking using probabilistic context-free grammars," in *Proc. IEEE S&P, 2009*, pp. 391–405.

[2] Jerry Ma, Weining Yang, Min Luo, and Ninghui Li, "A study of probabilistic password models," in *Proc. IEEE S&P, 2014*, pp. 689–704.

[3] Rafael Veras, Christopher Collins, and Julie Thorpe, "On semantic patterns of passwords and their security impact.," in *Proc. NDSS, 2014*, pp. 1–16.

[4] William Melicher, Blase Ur, Sean M Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor, "Fast, lean, and accurate: Modeling password guessability using neural networks," in *Proc. USENIX Security, 2016*, pp. 175–191.

[5] Weili Han, Ming Xu, Junjie Zhang, Chuanwang Wang, Kai Zhang, and X Sean Wang, "Transpcfg: transferring the grammars from short passwords to guess long passwords effectively," *TIFS, 2020*, vol. 16, pp. 451–465.

[6] Haibo Cheng, Wenting Li, Ping Wang, and Kaitai Liang, "Improved probabilistic context-free grammars for passwords using word extraction," in *Proc. IEEE ICASSP, 2021*, pp. 2690–2694.

[7] Robert Morris and Ken Thompson, "Password security: A case history," *Commun. ACM, 1979*, vol. 22, no. 11, pp. 594–597.

[8] Claude Castelluccia, Markus Dürmuth, and Daniele Perito, "Adaptive password-strength meters from markov models.," in *Proc. NDSS, 2012*, pp. 1–16.

[9] Ari Juels and Ronald L Rivest, "Honeywords: Making password-cracking detectable," in *Proc. ACM CCS, 2013*, pp. 145–160.

[10] Rahul Chatterjee, Joseph Bonneau, Ari Juels, and Thomas Ristenpart, "Cracking-resistant password vaults using natural language encoders," pp. 481–498.

[11] Markus Jakobsson and Mayank Dhiman, "The benefits of understanding passwords," in *Proc. Mobile Authentication, 2013*, pp. 5–24.

[12] Arvind Narayanan and Vitaly Shmatikov, "Fast dictionary attacks on passwords using time-space tradeoff," in *Proc. ACM CCS, 2005*, pp. 364–372.

[13] Ashwini Rao, Birendra Jha, and Gananand Kini, "Effect of grammar on security of long passwords," in *Proc. ACM CODASPY, 2013*, pp. 317–324.

[14] Maximilian Golla, Benedict Beuscher, and Markus Dürmuth, "On the security of cracking-resistant password vaults," in *Proc. ACM CCS, 2016*, pp. 1230–1241.

[15] Haibo Cheng, Zhixiong Zheng, Wenting Li, Ping Wang, and Chao-Hsien Chu, "Probability model transforming encoders against encoding attacks," in *Proc. USENIX Security, 2019*, pp. 1573–1590.

[16] Dario Pasquini, Ankit Gangwal, Giuseppe Ateniese, Massimo Bernaschi, and Mauro Conti, "Improving password guessing via representation learning," in *Proc. IEEE S&P, 2021*, pp. 1382–1399.

[17] Matteo Dell'Amico and Maurizio Filippone, "Monte carlo strength evaluation: Fast and reliable password checking," in *Proc. ACM CCS, 2015*, pp. 158–169.