INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

PROBABILISTIC CONTEXT-FREE GRAMMAR (PCFG) WISER PASSWORD CRACKING TECHNIQUES

Archana.A[1], Kohila.N[2]

[1] *M.Phil Research Scholar, chana2591@gmail.com*
[2] *Assistant Professor, padmeshraj@gmail.com*
Department of Computer Applications, Vivekanandha College of Arts and Science for Women (Autonomous), Namakkal (TN) - India.

Abstract: - Passwords continue to remain an important authentication technique. The probabilistic context-free grammar-based password cracking system of Weir et *al.* was an important addition to dictionary-based password cracking approaches. In this paper, we show how to substantially improve upon this system by systematically adding keyboard patterns and multiword patterns (two or more words in the alphabetic part of a password) to the context-free grammars used in the probabilistic password cracking. Our results on cracking multiple data sets show that by learning these new classes of patterns, we can achieve up to 22% improvement over the original system. In this paper, we also define metrics to help analyze and improve attack dictionaries. Using our approach to improving the dictionary, we achieve an additional improvement of ~33% by increasing the coverage of a standard attack dictionary. Combining both approaches, we can achieve a 55% improvement over the previous system. Our tests were done over fairly long password guessing sessions (up to 85 billion) and thus show the uniform effectiveness of our techniques for long cracking sessions**.**

Keywords*:* Keyboard patterns, multiwords, password cracking, dictionaries, probabilistic grammars, authentication.

1. Introduction

Passwords today remain a key to authentication security despite many newer proposed techniques such as biometric based techniques and dual factor authentication. The primary reason is the ease of use and the ability of humans to remember reasonable sized passwords. Because of their fairly universal use, it is often necessary for law enforcement to be able to crack passwords and thus it has been important to make progress in cracking techniques. The forensic setting we consider is an offline attack in which law enforcement has the hashes of a set of passwords and needs to make guesses until a sufficient number of the hashes are identified. Additionally, password cracking is often used in corporations to crack the passwords of machines for which the password has been forgotten or for which the password is no longer available because an employee has left.

In this paper we also discuss how to analyze, create and use attack dictionaries more effectively. We develop measures using precision and coverage to compare such dictionaries with respect to reference sets. This work also explores use of secondary attack dictionaries that can be used in the probabilistic password cracking to give more weight to some words over others. Note that attack dictionaries in PPC are used only to replace an alphabet component when generating a password guess. In the literature, some of the studies analyzing dictionaries.

2. Background and Previous Work

A. Previous Work

There have been many studies that have explored how users choose passwords [13]–[15] and recent studies have turned to greater exploration of the strength of passwords [4], [7], [9], [16]. Interestingly enough, however, there are not many studies that explore keyboard shapes and how often users choose keyboard combinations or that explore the strength of such keyboard patterns. For example, the focus of Luca et al. [17] is mainly on how to define such structures. The authors use directional line segments (of varying lengths) they call a stroke and can thus describe a shape on a 10 digit PIN pad.

B. Probabilistic Password Cracking

We substantially improve the probabilistic password cracking system (PPC) of Weir et al. [1] In our work. The authors derive a probabilistic context-free grammar from training on large sets of revealed passwords. This grammar is then used to generate guesses based on the various probabili- ties found in the training data. The authors show how guesses can be generated by PPC in highest probability order which is the optimal off-line attack if no other information were known. There are two major phases in this password cracking approach:

1) ***Training:*** This phase generates the context-free grammar from a training set of disclosed real user passwords. The observed base structures and their frequencies are derived from the training set of passwords. Probability information for digits, special characters and capitalization are also obtained. This information is used to generate the probabilistic context- free grammar. The probability of any string derived from the start symbol is then the product of the probabilities of the productions used in its derivation.
2) ***Generating Guesses:*** The guess generation phase gener- ates the possible password guesses in decreasing probability order using the context-free grammar obtained from the previ- ous step. Multiple dictionaries can be used with probabilities associated to each.

3. Keyboard patterns

A keyboard pattern is generally considered to be a sequence of keystrokes on the keyboard, without regards to the actual characters typed, but that instead creates a physical structure (shape) of the characters that can be remembered.

A. Finding Keyboard Patterns and Ambiguity Issues

Generating guesses in highest probability order using a context-free grammar relies on the grammar being unambiguous. This ensures that there is a well-defined probability for a guess which only depends on a single unique derivation. A grammar is ambiguous if there are two or more different derivation trees for a terminal string. Consider the following grammar:

S → L5D3|L5K3
L5 → alice
D3 → 131|123
K3 → 123| asd

B. Keyboard Patterns and PCFGs

In lines 1 and 2, for passwords asdf and q1q1 respectively, the count of the grammar rule S → K4 would increase by two, and there would not be an increased count for the rules S → L4 and S → LDLD. These counts are ultimately turned into transition probabilities for these rules. Further, if a password such
as john was encountered, then the original base structure L4 would be increased by one but there would of course be no corresponding keyboard structure. Notice that we can maintain the notion of a context-free grammar.

C. Using a Training Dictionary

The above rules would however classify the password component tree as a K4 rather than an L4. It seems that it might be preferable to view such components as English words.

D. Implementing Probability Smoothing

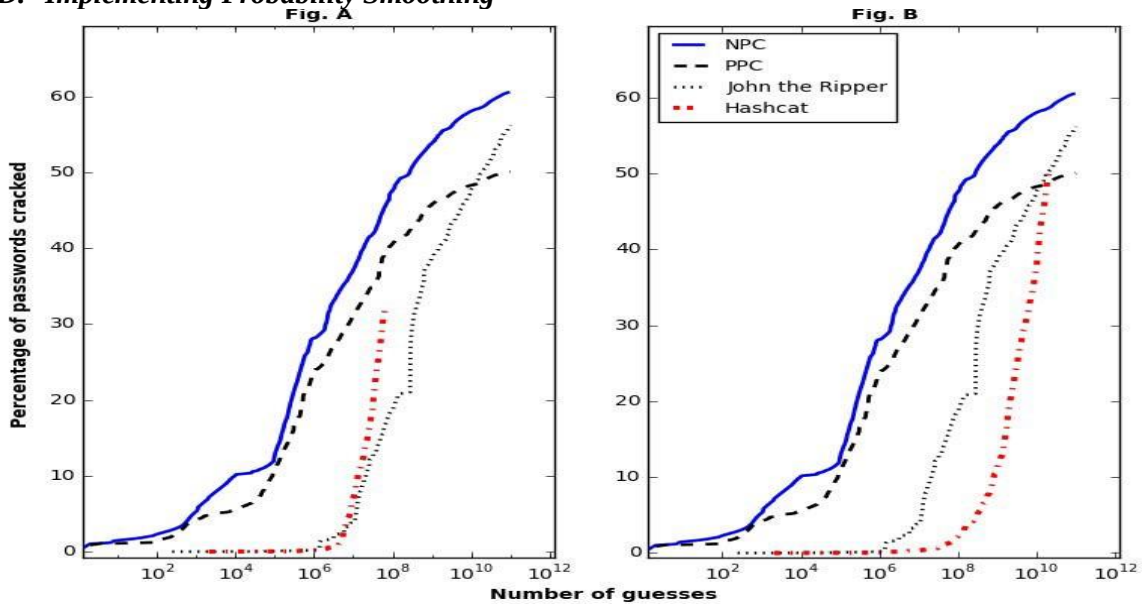


Fig. 1. Comparing password crackers using Combined-set in log scale:
A) Hashcat is used with Best64. **B) Hashcat is used with Deadone.**

NPC is extremely dominant in both figures.

A. Cross Testing

In the series of tests reported on so far, we have been training and testing on sets from similar data. This is where both NPC and PPC have the advantage of training on a set that is similar to the target set. It is often the case that one might not know the target or might not have sufficient training passwords available. So in the next series of tests shown in Fig. 2 we explore this issue.

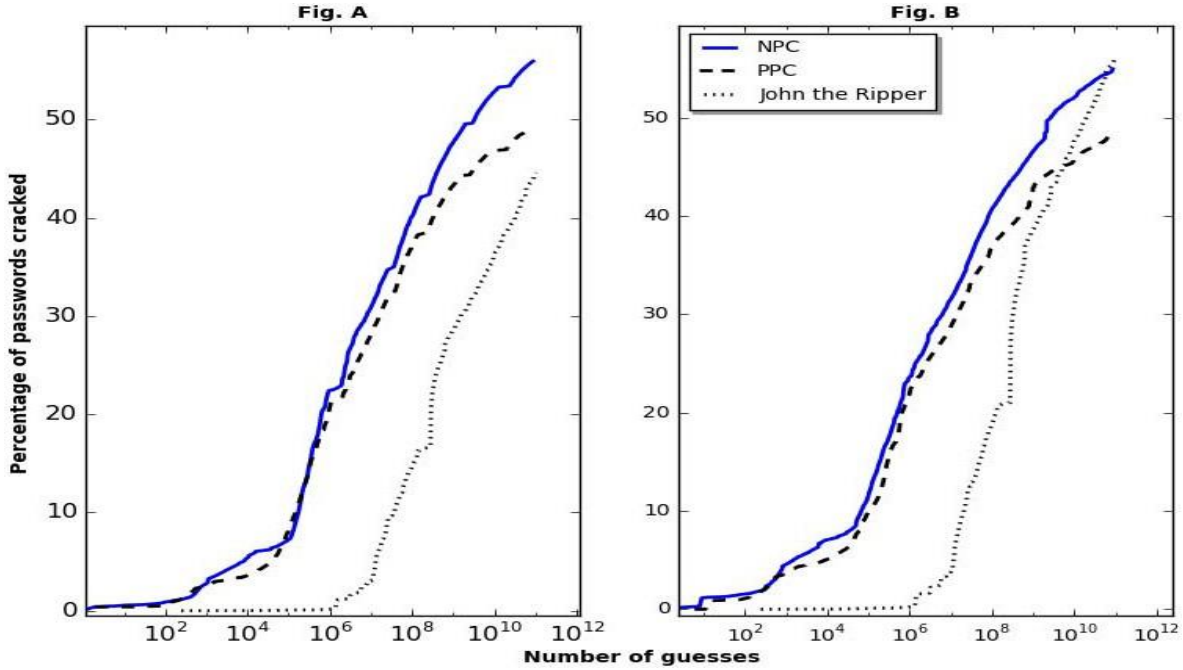


Fig. 2. Comparing password crackers in log scale using:

A) Combined- training and Yahoo-test, **B) Yahoo-training and Combined-test.**

4. Attack Dictionaries

In dictionary based attacks, a list of words called an attack dictionary (or just dictionary) is used along with different mangling rules to create password guesses. Therefore in order to correctly guess a password the attacker needs to not only

Yahoo-test in Fig. 3A and Rockyou-test in Fig. 3B and found that the cracking curves were consistent with the precision and coverage metrics, with better rates of cracking for dictionaries having higher coverage/precision. Note that the Rockyou dictionary has higher coverage since it is calcu- lated with respect to Combined-test, which contains mostly Rockyou passwords.

We next created different dictionaries from dic0294 by systematically altering coverage and precision to see how the cracking changes. In our first series of experiments we used the baseline dic0294 and calculated its metrics with respect to the reference Combined-test.

$$C\ (dic0294, R_{Combined}) = 0.55,$$
$$P\ (dic0294, R_{Combined}) = 0.06$$

We then created two dictionaries as variants of dic0294, increasing the coverage to 0.7 and 0.9 respectively without changing the precision. We call these variants dic0294_c70 and dic0294_c90. The sizes of these variants increased to about 1.56 million and 2.58 million respectively.

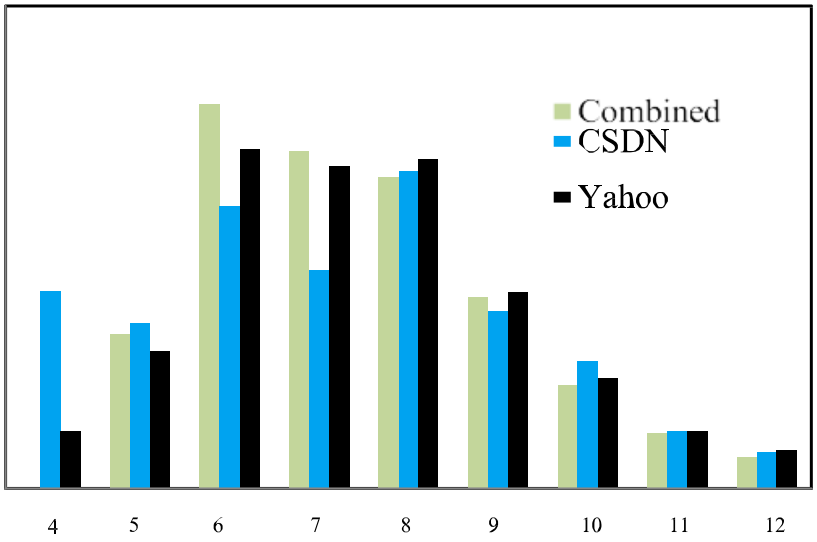
1

$$nn^{-} = nr\,(_{P} - 1) \qquad (1)$$

Since in cracking we would not know the actual target set, we explored the use of the metrics derived from reference Combined-test by testing.

Keyboard and Alpha String Usage

The CSDN passwords have a lower percentage of passwords that partially contain keyboard patterns, but a higher percentage of passwords that entirely consist of keyboard patterns. However, CSDN passwords also have an unusually large number of all digits (3 times as often as Combined-set) which may include keyboard patterns.



Length

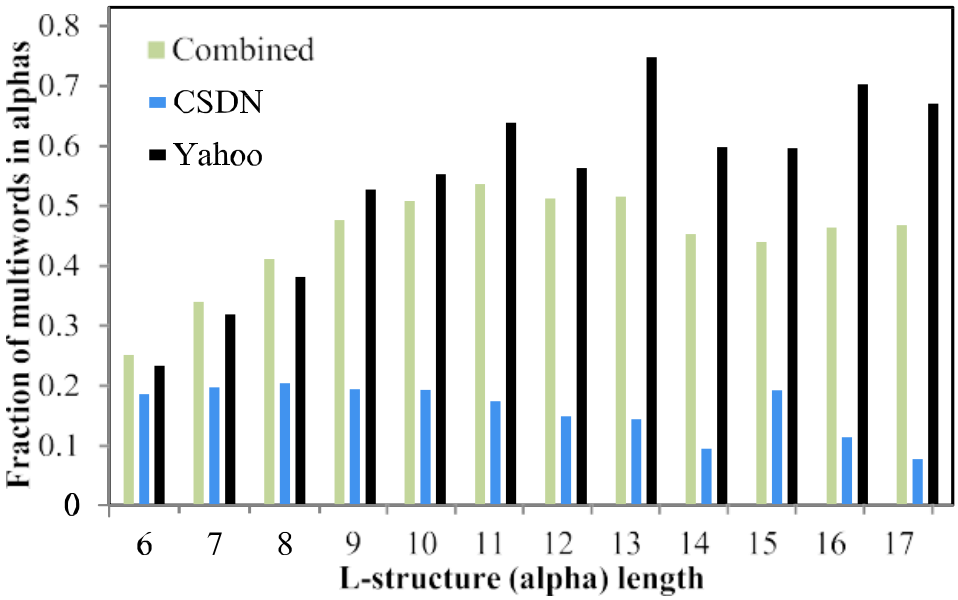


Fig. 4. Fraction that are multiwords for each length L-structure.

Conclusion and Future Work

We have shown how to systematically incorporate keyboard and alpha string patterns into PCFG-based password cracking and how to ensure that the resulting grammars are still essentially unambiguous. A similar approach to other types of patterns could also be undertaken, for example Let speak replacements and removal of vowels in words. We also showed how smoothing for keyboard patterns can be done. For multiword smoothing, the work of other researchers related to models for passphrases and use of linguistic structures could be very useful. We also introduced metrics for attack dictionaries and explored how they can be used for comparing and improving dictionaries. The improvements resulting in NPC are significant and do not result in any loss of efficiency so far on long cracking runs. The techniques we have developed can likely be applied to other cracking systems.

Our results for long cracking sessions of 85 billion guesses show the consistent effectiveness of our system both in the early and later phase of the cracking curve. We believe that the probabilistic password cracking approach is extremely useful, but in one way it is also partially limiting. After some time, the context-free grammar gets "stuck" in a cul-de-sac and will try guesses from patterns seen in the training but which probably should no longer be tried. The way out of this is to either add new classes of patterns or add patterns that include the ability to essentially do brute forcing. We intend to explore this in future work.

REFERENCES

[1] M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek, "Password cracking using probabilistic context-free grammars," in *Proc. 30th IEEE Symp. Secur. Privacy*, May 2009, pp. 391–405.
[2] R. Veras, C. Collins, and J. Thorpe, "On the semantic patterns of passwords and their security impact," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, Feb. 2014.
[3] M. Dell'Amico, P. Michiardi, and Y. Roudier, "Password strength: An empirical analysis," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.
[4] P. G. Kelley *et al.*, "Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2012, pp. 523–537.
[5] Y. Zhang, F. Monrose, and M. K. Reiter, "The security of modern password expiration: An algorithmic framework and empirical analysis," in *Proc. 17th ACM Conf. Comput. Commun. Secur.*, 2010, pp. 176–186.
[6] M. Weir, S. Aggarwal, M. Collins, and H. Stern, "Testing metrics for password creation policies by attacking large sets of revealed passwords," in *Proc. 17th ACM Conf. Comput. Commun. Secur.*, 2010, pp. 162–175.
[7] S. Houshmand and S. Aggarwal, "Building better passwords using probabilistic techniques," in *Proc. 28th Annu. Comput. Secur. Appl. Conf.*, 2012, pp. 109–118.
[8] A. Narayanan and V. Shmatikov, "Fast dictionary attacks on passwords using time-space tradeoff," in *Proc. 12th ACM Conf. Comput. Commun. Secur.*, 2005, pp. 364–372.

[9] C. Castelluccia, M. Dürmuth, and D. Perito, "Adaptive password- strength meters from Markov models," in *Proc. NDSS*, 2012.
[10] J. Ma, W. Yang, M. Luo, and N. Li, "A study of probabilistic pass- word models," in *Proc. 35th IEEE Symp. Secur. Privacy*, May 2014, pp. 689–704.
[11] *John the Ripper Password Cracker*. [Online]. Available: http://www.openwall.com, accessed Apr. 30, 2015.
[12] *Hashcat Advanced Password Recovery*. [Online]. Available: http://www.hashcat.net, accessed Apr. 30, 2015.
[13] S. Riley, "Password security: What users know and what they actually do," *Usability News*, vol. 8, no. 1, pp. 2833–2836, 2006.
[14] B. Stone-Gross *et al.*, "Your botnet is my botnet: Analysis of a botnet takeover," in *Proc. 16th ACM Conf. Comput. Commun. Secur.*, 2009, pp. 635–647.
[15] R. Shay *et al.*, "Encountering stronger password requirements: User attitudes and behaviors," in *Proc. 6th Symp. Usable Privacy Secur.*, 2010, Art. ID 2.
[16] S. Schechter, C. Herley, and M. Mitzenmacher, "Popularity is every- thing: A new approach to protecting passwords from statistical-guessing attacks," in *Proc. 5th USENIX Conf. Hot Topics Secur.*, 2010.
[17] A. De Luca, R. Weiss, and H. Hussmann, "PassShape: Stroke based shape passwords," in *Proc. 19th Austral. Conf. Comput.-Human Interact., Entertaining User Interf.*, 2007, pp. 239–240.
[18] D. Schweitzer, J. Boleng, C. Hughes, and L. Murphy, "Visualizing keyboard pattern passwords," in *Proc. 6th Int. Workshop Vis. Cyber Secur. (VizSec)*, 2009, pp. 69–73.
[19] J. Bonneau, "The science of guessing: Analyzing an anonymized corpus of 70 million passwords," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2012, pp. 538–552.
[20] A. Vance. (Jan. 20, 2010). If your password is 123456, just make it hackme. The New York Times. [Online]. Available: http://www.nytimes.com/2010/01/21/technology/21password.html
[21] *6 Million Users' Privacy Leaked*. [Online]. Available: http://www.china- online-marketing.com/news/anti-virus-news/csdn-tianya-renren-kaixin- hacked-6-million-users-privacy-leaked/, accessed Apr. 30, 2015.
[22] J. Yan, A. Blackwell, R. Anderson, and A. Grant, "Password memo- rability and security: Empirical results," *IEEE Security Privacy*, vol. 2, no. 5, pp. 25–31, Sep./Oct. 2004.
[23] C. Kuo, S. Romanosky, and L. F. Cranor, "Human selection of mnemonic phrase-based passwords," in *Proc. 2nd Symp. Usable Privacy Secur.*, 2006, pp. 67–78.
[24] R. Shay *et al.*, "Can long passwords be secure and usable?" in *Proc. 32nd Annu. ACM Conf. Human Factors Comput. Syst.*, 2014, pp. 2927–2936.
[25] J. Bonneau and E. Shutova, "Linguistic properties of multi-word passphrases," in *Financial Cryptography and Data Security*. Berlin, Germany: Springer-Verlag, 2012, pp. 1–12.
[26] D. V. Klein, "'Foiling the cracker': A survey of, and improvements to, password security," in *Proc. 2nd USENIX Secur. Workshop*, 1990, pp. 5–14.
[27] R. McMillan. (Oct. 27, 2006). *Phishing Attack Targets MySpace Users*. [Online]. Available: http://www.infoworld.com/d/security-central/ phishing-attack-targets-myspace-users-614
[28] T. Warren. (2009). *Thousands of Hotmail Passwords Leaked*. [Online]. Available: http://www.neowin.net/news/main/09/10/05/thousands-of -hotmail-passwords-leaked-online
[29] *Yahoo Credentials*. [Online]. Available: http://news.cnet. com/8301-1009_3-57470786-83/hackers-post-450k-credentials-pilfered- from-yahoo, accessed Apr. 30, 2015.
[30] *The English Open Word List*. [Online]. Available: http://dreamsteep. com/projects/the-english-open-word-list.html, accessed Apr. 30, 2015.
[31] *Most Common Male and Female First Names in the U.S.* [Online]. Available: http://names.mongabay.com/, accessed Apr. 30, 2015.
[32] *Top 40, 000 Words From TV and Movie Scripts*. [Online]. Available: http://en.wiktionary.org/wiki/Wiktionary:Frequency_lists#TV_and_movie _scripts, accessed Apr. 30, 2015.
[33] (2005). *The Dic0294 Wordlist*. [Online]. Available: http://www.outpost9.com/files/WordLists.html, accessed Apr. 30, 2015.
[34] *KoreLogic Rule Set*. [Online]. Available: http://openwall.info/wiki/ john/rules, accessed Apr. 30, 2015.
[35] M. L. Mazurek *et al.*, "Measuring password guessability for an entire university," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 173–186.
[36] D. R. Cox, "Regression models and life-s," in *Breakthroughs in Statistics*. New York, NY, USA: Springer-Verlag, 1992, pp. 527–541.
[37] R. Peto and J. Peto, "Asymptotically efficient rank invariant test proce- dures," *J. Roy. Statist. Soc. A (General)*, vol. 135, no. 2, pp. 185–207, 1972.
[38] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA, USA: Freeman, 1979.