Analysis of Password Protected Documents Using Statistical Approaches on High Performance Computing

Chapter · January 2022

DOI: 10.1007/978-981-16-8550-7_51

6 authors, including:

Ajeet Singh
University of Hyderabad
17 PUBLICATIONS **4** CITATIONS
SEE PROFILE

Vikas Tiwari
Malaviya National Institute of Technology Jaipur
13 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Tentu Appala Naidu
University of Hyderabad
38 PUBLICATIONS **132** CITATIONS
SEE PROFILE

Srujan kotagiri Raju
CMR Technical Campus
119 PUBLICATIONS **165** CITATIONS
SEE PROFILE

Analysis of Password Protected Documents Using Statistical Approaches on High Performance Computing

Ajeet Singh, Vikas Tiwari, Allu Swamy Naidu, Appala Naidu Tentu, K. Surjan Raju, and Ashutosh Saxena

Abstract Password-based validation frameworks are as yet the most regularly utilized components for ensuring the data regardless of being helpless against dictionary reference-based attacks. Password breaking is the way towards speculating or recuperating a secret key from put away areas or from an information transmission framework. Best state-of-the-art password analysing methods like HashCat, John the Ripper and rainbow crack empower clients to check billions of passwords each second against the secret key hashes. This paper discusses various techniques including traditional, probabilistic and statistical methods for cracking the password protected files. Further, experimental evaluation, rationale and performance analysis on some sample password protected files are presented in this paper. The findings in this paper will also help understanding of both password-composition policies and metrics for quantifying password security.

Keywords Privacy · Password cracking · Dictionary attack · Brute-force attack · Personally identifiable information · Learned patterns

1 Introduction

Despite significant progress in attackers' abilities to crack passwords, text-based passwords remain the most used validation approach in computer-based systems. There are a wide range of approaches to validate clients of a framework; for example, a client can introduce a physical article like a key card, demonstrate character utilizing an individual trademark like a fingerprint or use something that solitary the client

A. Singh (✉) · V. Tiwari · A. S. Naidu · A. N. Tentu · A. Saxena
C.R. Rao Advanced Institute of Mathematics Statistics and Computer Science, University of Hyderabad Campus, Prof. CR Rao Road, Hyderabad 500046, Telangana, India
e-mail: ajeetcs@uohyd.ac.in

K. S. Raju · A. Saxena
CMR Technical Campus, Hyderabad 501401, India
e-mail: hod.cse@cmrtc.ac.in

V. V. S. S. S. Chakravarthy et al. (eds.), *Advances in Micro-Electronics, Embedded Systems and IoT*, Lecture Notes in Electrical Engineering 838,
https://doi.org/10.1007/978-981-16-8550-7_51

knows. Passwords [1, 2] are one of the methods designed to provide authentication [3–5]. As opposed to different methodologies recorded, an essential advantage of utilizing confirmation through a password [6] is that if your password becomes compromised it very well may be effortlessly changed. In this paper, we discuss about general view of password cracking, methodologies for password cracking [7] at the point when an attacker can sign in to the framework by giving a client name and password pair, methods [8, 9] when an attacker approaches how passwords are put away on the framework. Figure 1 represents some scenarios attempts with which password cracking can occur. Figure 2 gives an overview of password hash salting flow.

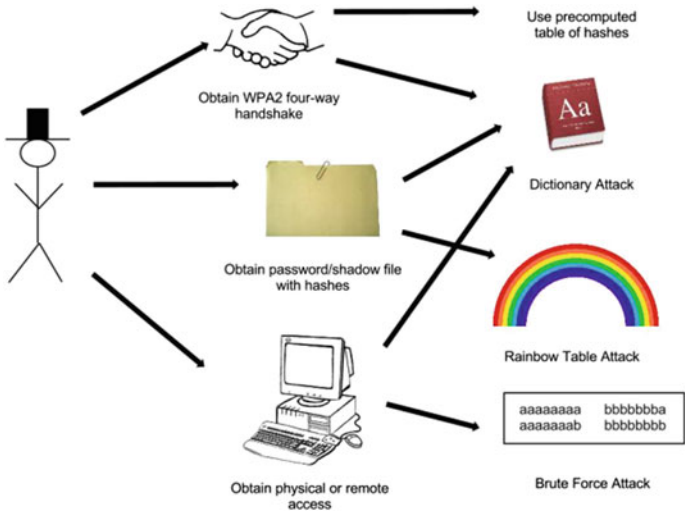

Fig. 1 Various techniques for password cracking

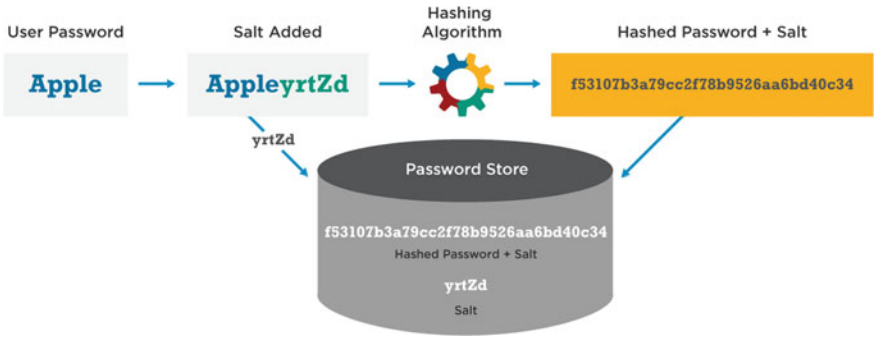

Fig. 2 Password hash salting flow

1.1 Contribution Highlights

Our contribution in this paper is summarized as follows:

- Various techniques including traditional, probabilistic and statistical methods for cracking the password protected files are discussed.
- Experimental evaluation, rationale, computational features and performance analysis on some sample password protected files are presented in this paper.
- Dictionary wordlist creation is performed based on statistical patterns observation by state-of-the-art tools such as *Flex*, *pydictor* and *crunch*.
- Integrated our generated dictionaries with Elcomsoft software tool [10]. Further, tested the functionality for password protected MS-Word input files up to 6-characters length.

1.2 Background and Related Work

This section covers various methods and frameworks developed for password guessing over past years. Kedem et al. [11] discussed the strategy of brute force attack on the Unix passwords in the perspective of SIMD architecture computer. Hitaj et al. [12] given a deep learning methodology for password guessing. Chou [13] given a framework and strategy for password breaking dependent on learned designs from unveiled passwords. Dell et al. [14] and Drmuth et al. [15] proposed a faster platform of password guessing exploiting an ordered Markov enumerator. Juels et al. [16] and Ma et al. [17] given the most occurring passwords and an investigation of probabilistic secret word models. Yampolskiy et al. [18] given an strategy for the dissecting user password selection behaviour for the reduction of certain password space. Weir et al. [19] discussed the attack procedure for password cracking exploiting probabilistic context-free grammars. Kelley et al. [20] discussed techniques for measuring passwords strength measure by resembling password cracking procedures. Bonneau et al. [21] discussed strong passwords choosing strategies, furthermore the development of flawed confirmation over the years.

Wang [22], in his thesis, discussed the key issues in password security. Ma et al. [23] performed the study of probabilistic password models. Narayanan et al. [24] surveyed fast dictionary-oriented attacks on passwords exploiting time-space tradeoff technique. Weir et al. [25] presented framework for password cracking exploiting probabilistic context-free grammars. Veras et al. [26] discussed semantic examples of passwords and their corresponding security impact. Melicher et al. [27] proposed a fast and efficient procedure for modelling password guessability using neural networks. Aggarwal et al. [28] did the survey of different modern technologies in password cracking techniques. Tirado et al. [29] presented another dispersed brute force secret phrase breaking method. Hitaj et al. [30] presented a deep learning-oriented methodology for password guessing. Ji et al. [31] given a massive-scale empirical concentrate on the crackability, relationship and security of passwords. Li et al. [32]

given a huge scope observational investigation of Chinese Web passwords. Yampolskiy [33], in their work, analysed client secret phrase determination conduct for decrease of password space. Gong-Shen et al. [34] performed the password weakness evaluation and recuperation dependent on rules mined from enormous data.

Das et al. [35] identified a couple of basic strategies clients frequently utilize to change an essential password between destinations which can be utilized by an aggressor to make password speculating endlessly simpler. Li et al. [36] given an investigation of individual data in human-picked passwords and their security suggestions. Merhav et al. [37] performed attacks. Lu et al. [38] given an estimation investigation of validation rate-restricting components of present day sites. Pal et al. [39] presented a password similarity model using neural networks. Guri et al. [40] analysed the individual data spillage during secret word recuperation of Internet providers. Bailey et al. [41] given the insights on password re-use and versatile strength for monetary records. Emin Islam [42] presented an strategy to crack more password hashes with specific patterns. Stobert et al. [43] talk about the general password life cycle and client conduct in overseeing passwords. Kelley et al. [44] performed measuring password strength by mimicking secret key breaking calculations. Shay et al. [45] discussed the effect of direction and criticism on password creation conduct. Wang et al. [46] given a framework to analyse the passwords of specific Chinese Web end-users.

1.3 Organization of the Paper

Section 2 covers a brief summary on password storage. Various methods of password cracking and other findings are discussed in Sect. 3. Section 4 covers the theoretical password cracking model using Elcomsoft attack tool. Experimental evaluation, computational features, results discussion and performance analysis are presented in Sect. 5, and conclusions are given in Sect. 6.

2 Password Storage

This section provides a short discussion about how the passwords [47, 48] are stored in the system. Putting away client names and comparing passwords in plaintext is not any more an adequate arrangement. Endeavouring to shroud passwords put away as plaintext (e.g. putting the secret phrase record somewhere down in a tangled registry chain of importance) would add up to security through obscurity which would likewise be inadmissible. The Unix arrangement of record the executives, on the other hand, is superior: one of the agreements. Beginning adaptations of Multics (the forerunner to Unix) put away the secret phrase record in clear content, yet just perceptible with superuser consents. This arrangement likewise attacked when a

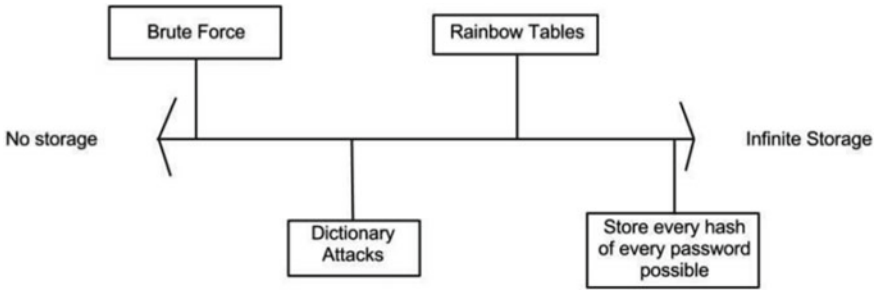

Fig. 3 Variation with respect to storage

bug exchanged some impermanent documents around and the secret key record (in plaintext) was printed for each client upon login [49].

Unix, rather, stores the hashed estimation of passwords in the secret key record rather than the genuine passwords [50, 51]. At that point when a client inputs their secret key, the framework can essentially take the hash of the info and contrast it with the put away hash esteem [52, 53]. Variety regarding capacity is appeared as Fig. 3.

- In a large portion of the Unix-based record frameworks, the secret key document is situated at */and so on/password*. Each line in this record contains data around one record on the framework. The record itself is lucid by all clients, however, is just writable with superuser benefits.
- The secret key document for Windows, known as the security accounts manager (SAM) record, contains seven colon delimited fields: the client name, client number, scrambled secret phrase, hashed secret word, hashed secret phrase under an alternate calculation, complete name of client and lastly home registry. Rather than the Unix secret key record, the Windows SAM document isn't meaningful once the working framework has booted.
- Many sites and online administrations expect clients to sign in with a common secret word plot. This requires the capacity of secret word data. Be that as it may, online administrations normally store passwords for their framework in a non-normalized way, and these frameworks are not generally planned by engineers with foundations in protection or security.

3 Various Methodologies and Findings

This section discusses various methodologies for cracking the password protected files along with some interesting findings.

Table 1 Size of rainbow table and extent of key space covered

Char set	Plaintext length	Key space	Table size (GB)
ascii-32-95	1–7	2^{46}	52
ascii-32-95	1–8	$2^{52.5}$	460
Mix-alpha-numeric	1–8	$2^{47.65}$	127
Mix-alpha-numeric	1–9	$2^{53.6}$	690
Lower-alpha-numeric	1–9	$2^{46.5}$	65
Lower-alpha-numeric	1–10	$2^{51.8}$	316

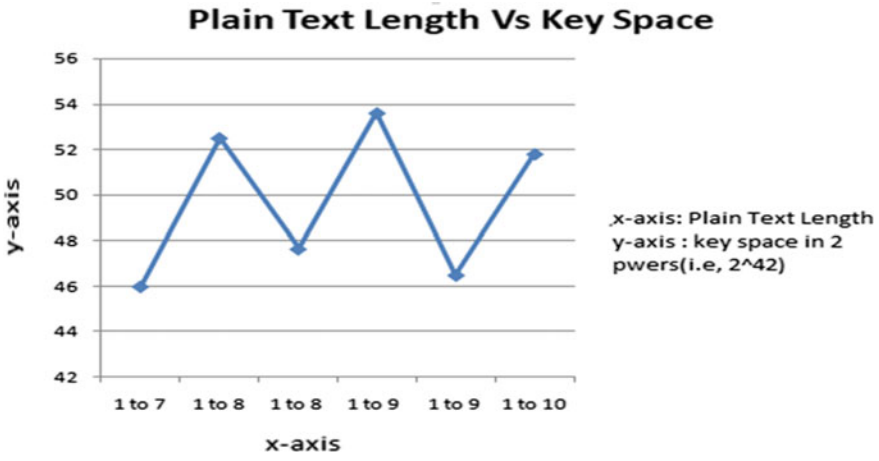

Fig. 4 Variation of plaintext length versus key space covered

3.1 Rainbow Tools for Password Recovery

Here, the strategy is straighforward, i.e. by precomputing some part to the issue (regularly either by explaining subproblems or by discovering normal arrangements), the time cost of taking care of the issue all in all is diminished, while the space necessities are considerably not as much as what might be expected to completely precompute the solution.

3.1.1 RainbowCrack

RainbowCrack [54] is a computer program that was developed by Zhu Shuanglei. It cracks hashes exploiting rainbow tables and recovers the plaintext. Table 1 and graph (shown in Fig. 4) represent the size of rainbow table and extent of key space covered for different character types having different lengths.

Table 2 Sizes of rainbow tables for different Hash algorithms

Char set	Plaintext length	Key space	Table size		
			NTLM (GB)	SHA-1 and MySQLSHA1 (GB)	MD5 (GB)
All-space	1–7	$2^{42.7}$	–	–	–
Alpha-space	1–9	$2^{42.86}$	35	–	23
Lower-alpha-numeric-space	1–9	2^{47}	–	108	108
Mix-alpha-numeric	1–9	$2^{53.7}$	1000	504	1000
Mix-alpha-numeric-space	1–7	$2^{41.9}$	17	–	17

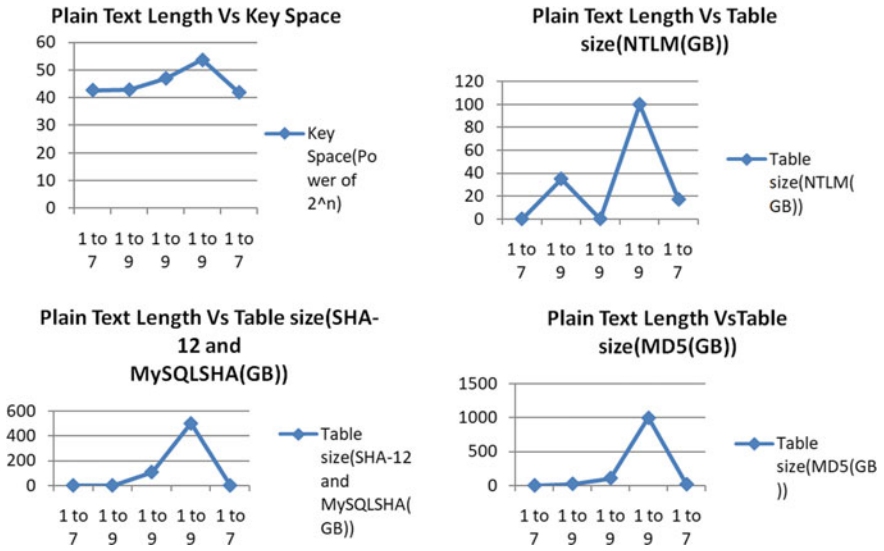

Fig. 5 Comparison between the plaintext lengths and key space, NTLM, MD5, SHA-1 and MySQLSHA1

3.1.2 Rainbow Tables Generation

Rainbow tables generation is a free software application currently available in English, and it was last updated on 2010 and downloadable. Table 2 gives the sizes of rainbow tables for different hash algorithms.

The graph (shown in Fig. 5) gives comparison between the plain text lengths and key space, NTLM, MD5, SHA-1 and MySQLSHA1.

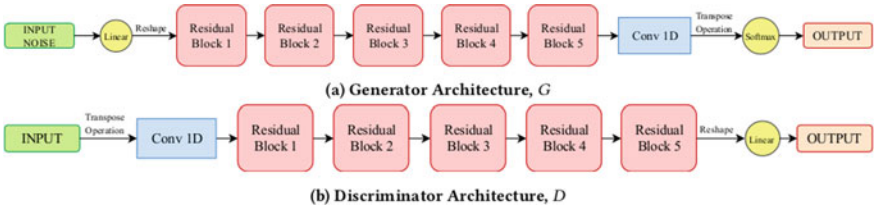

Fig. 6 PassGAN's architecture

Rcracki_mt can be used to perform a rainbow table attack on password hashes. *GUIRainbowCrack* is extended version of RainbowCrack in GUI. GUI RainbowCrack is a new way to crack password using pre-computed password hash table (PCPH). It substantially improves the speed of standard password cracking.

3.2 PassGAN: Generative Adversarial Networks Architecture Based Password Guessing

PassGAN [12] replaces human-generated password rules with theory-grounded machine learning algorithms. PassGAN architecture view is shown in Fig. 6.

3.2.1 Architecture Overview

Figure 6 represents PassGAN's architecture. There are two components named as generator and discriminator.

It utilizes—*matplotlib—2.1.1, numpy—1.13.3, Tensorflow—1.4.1, Tensorflow-gpu—1.4.1*.

4 Password Cracking Using Elcomsoft Attack Tool

The rationale and various attacks provided by Elcomsoft password recovery tool [10] are summarized as follows.

4.1 Rationale

- It supports all versions of Microsoft Office 2.0 to Microsoft Office 2019 and PDF form as password protected input file.

- Dictionary, such as, exploitation of rockyou.txt [55] and brute force attacks with user-defined masks and templates. User can add his own created dictionary of any size in the tool then execute the attack.
- Supports multiple language dictionaries.
- Supports GPU acceleration with state-of-the-art NVIDIA cards.
- Hardware acceleration reduces password recovery time by a factor of ≈50.
- Supports 64 CPUs and up to 8 GPUs.

5 Experimental Evaluation and Performance Analysis

This section presents experimental set-up, simulation environment, obtained results in various test case scenarios and performance analysis in terms of CPU and GPU utilization.

5.1 Set-Up and Simulation Environment

Our high performance computing workstation set-up and simulation environment consist of following software and hardware specifications: Operating system as Windows 10, Intel Core i7-8750H processor, RAM size 32 Gb, NVIDIA Quadro P600 graphics card having 384 NVIDIA CUDA Cores and Python 3.7 installed.

5.2 Obtained Results

The input in the experiments is in the form of password protected word documents. The obtained results in various test case scenarios are given in Table 3. The time to recover password is judged based on attack scenarios, i.e. password character length, chars combination, search space and attack type. Various experimental test cases of 4, 5, 6 characters are taken into consideration.

Performance analysis is done based on CPU and GPU utilization (representation is given as Figs. 7 and 8).

6 Conclusive Discussion

This paper discusses various techniques including traditional, probabilistic and statistical methods for cracking the password protected files. Experimental evaluation, rationale, computational features and performance analysis on some sample password protected files are also presented in this paper.

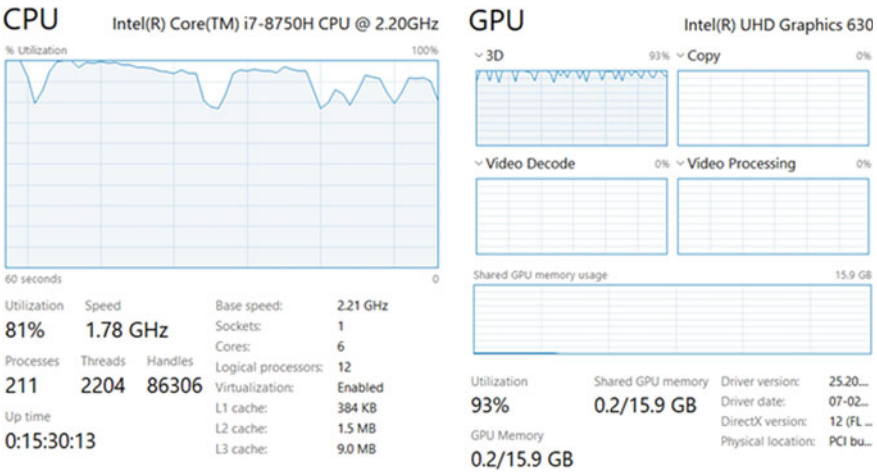

Fig. 7 CPU and GPU utilization

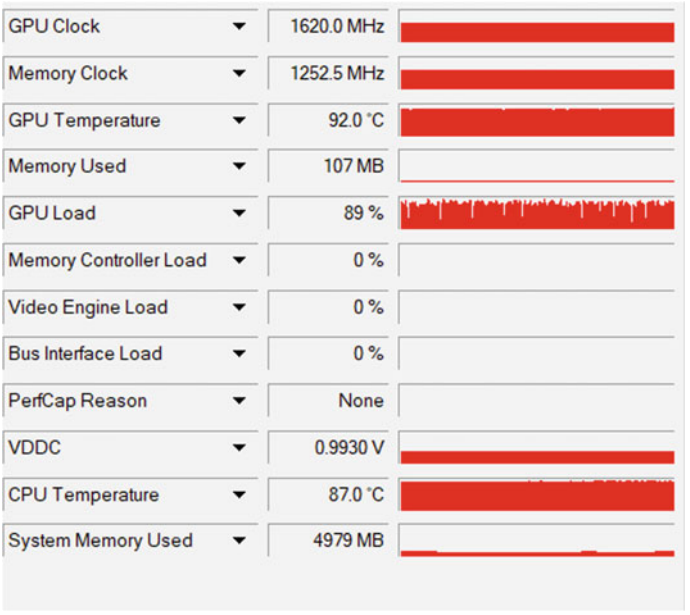

Fig. 8 Sensors view of NVIDIA Quadro P600 graphics card

Table 3 Tabular representation of the obtained results in various test scenarios

Char len	Combination	Search space	Attack type	Time to recover password
4	$< a...z >< 0...9 >$	36^4	Brute force attack	≈65 s
5	$< a...z >< 0...9 >$	36^5	Dictionary attack	≈110 s
6	$< a...z >< A...Z >< 0...9 >$	62^6	Dictionary attack	≈154 s
6	$< a...z >< A...Z >< 0...9 >$	62^6	Dictionary attack	≈1500 s
6	$< a...z >< A...Z >< 0...9 >$	62^6	Dictionary attack	≈1140 s
6	$< a...z >< A...Z >< 0...9 >$	62^6	Dictionary attack	≈240 s
6	$< a...z >< A...Z >< 0...9 >$	62^6	Dictionary attack	≈618 s
6	$< a...z >< A...Z >< 0...9 >$	62^6	Dictionary attack	≈1862 s

References

1. Narayanan M, Shmatikov V (2005) Fast dictionary attacks on passwords using timespace tradeoff. In: Proceedings of the 12th ACM conference on computer and communications security, pp 364–372
2. John the Ripper password cracker. Openwall Project. http://www.openwall.com/john
3. Ur B, Noma F, Bees J, Segreti SM, Shay R, Bauer L, Christin N, Cranor LF (2015) I added'!'at the end to make it secure: Observing password creation in the lab. In: Eleventh symposium on usable privacy and security (SOUPS 2015), pp 123–140
4. Karapanos N, Marforio C, Soriente C, Capkun S (2015) Sound-proof: usable two-factor authentication based on ambient sound. In: 24th USENIX security symposium (USENIX security 15), pp 483–498
5. Wang D, Wang P (2016) Two birds with one stone: two-factor authentication with security beyond conventional bound. IEEE Trans Depend Secure Comput 15(4):708–722
6. Wang D, Cheng H, Wang P, Huang X, Jian G (2017) Zipf's law in passwords. IEEE Trans Inf Forens Secur 12(11):2776–2791
7. Morris R, Thompson K (1979) Password security: a case history. Commun ACM 22(11):594–597
8. Hellman ME (1980) A cryptanalytic time-memory trade-off. IEEE Trans Inf Theory 26(4):401–406
9. Troy Hunt. http://www.troyhunt.com/2011/06/brief-sony-password-analysis.html
10. https://www.elcomsoft.com/aopr.html
11. Kedem G, Ishihara Y (1999) Brute force attack on UNIX passwords with SIMD computer. In: Proceedings of the 8th conference on USENIX security symposium, vol 8, Berkeley, CA, USA, 1999. USENIX Association, p 8
12. Hitaj B, Gasti P, Ateniese G, Perez-Cruz F (2019) PassGAN: a deep learning approach for password guessing. In: International conference on applied cryptography and network security. Springer, Cham
13. Chou HC, Lee HC, Yu HJ, Lai FP, Huang KH, Hsueh CW (2013) Password cracking based on learned patterns from disclosed passwords. Int J Innov Comput Inf Control 9(2)

14. DellAmico M, Michiardi P, Roudier Y (2010) Password strength: an empirical analysis. In: Proceedings of the IEEE INFOCOM. IEEE, pp 1–9
15. Drmuth M, Angelstorf F, Castelluccia C, Perito D, Abdelberi C (2015) OMEN: faster password guessing using an ordered Markov enumerator. In: ESSoS. Springer, pp 119–132
16. Juels A, Rivest RL (2013) Honeywords: making password-cracking detectable. In: Proceedings of the 2013 ACM SIGSAC conference on computer & communications security. ACM, pp 145–160
17. Ma J, Yang W, Luo M, Li N (2014) A study of probabilistic password models. In: IEEE symposium on security and privacy (SP). IEEE, pp 689–704
18. Yampolskiy RV (2006) Analyzing user password selection behavior for reduction of password space. In: Proceedings of the IEEE international Carnahan conferences on security technology, pp 109–115
19. Weir M, Aggarwal S, de Medeiros B, Glodek B (2009) Password cracking using probabilistic context-free grammars. In: Proceedings of the 30th IEEE symposium on security and privacy, pp 391–405
20. Kelley PG, Komanduri S, Mazurek ML, Shay R, Vidas T, Bauer L, Christin N, Cranor LF, Lopez J (2012) Guess again (and again and again): measuring password strength by Simulating Password-Cracking Algorithms. In: 2012 IEEE symposium on security and privacy, San Francisco, CA, pp 523–537
21. Bonneau J, Herley C, Van Oorschot PC, Stajano F (2015) Passwords and the evolution of imperfect authentication. ACM 58(7):78–87
22. Wang D (2017) Research on key issues in password security. PhD Dissertation, Peking University. http://wangdingg.weebly.com/uploads/2/0/3/6/20366987/phd-thesis0103.pdf
23. Ma J, Yang W, Luo M, Li N (2014) A study of probabilistic password models. In: 2014 IEEE symposium on security and privacy. IEEE, pp 689–704
24. Narayanan A, Shmatikov V (2005) Fast dictionary attacks on passwords using time-space tradeoff. In: Proceedings of the 12th ACM conference on computer and communications security, pp 364–372
25. Weir M, Aggarwal S, De Medeiros B, Glodek B (2009) Password cracking using probabilistic context-free grammars. In: 2009 30th IEEE symposium on security and privacy. IEEE, pp 391–405
26. Veras R, Collins C, Thorpe J (2014) On semantic patterns of passwords and their security impact. In: NDSS
27. Melicher W, Ur B, Segreti SM, Komanduri S, Bauer L, Christin N, Cranor LF (2016) Fast, lean, and accurate: modeling password guessability using neural networks. In: 25th USENIX security symposium, pp 175–191
28. Aggarwal S, Houshmand S, Weir M (2018) New technologies in password cracking techniques. In: Cyber security: power and technology, pp 179–198
29. Tirado E, Turpin B, Beltz C, Roshon P, Judge R, Gagneja K (2018) A new distributed brute-force password cracking technique. In: International conference on future network systems and security. Springer, pp 117–127
30. Hitaj B, Gasti P, Ateniese G, Perez-Cruz F (2019) Passgan: a deep learning approach for password guessing. In: International conference on applied cryptography and network security. Springer, pp 217–237
31. Ji S, Yang S, Hu X, Han W, Li Z, Beyah R (2015) Zero-sum password cracking game: a large-scale empirical study on the crackability, correlation, and security of passwords. IEEE Trans Depend Secure Comput 14(5):550–564
32. Li Z, Han W, Xu W (2014) A large-scale empirical analysis of Chinese web passwords. In: 23rd USENIX security symposium, pp 559–574
33. Yampolskiy RV (2006) Analyzing user password selection behavior for reduction of password space. In: Proceedings 40th annual 2006 international Carnahan conference on security technology. IEEE, pp 109–115
34. Gong-Shen MKL, Wei-Dong Q, Jian-Hua L (2016) Password vulnerability assessment and recovery based on rules mined from large-scale real data. Chin J Comput 39(3):454–467

35. Das A, Bonneau J, Caesar M, Borisov N, Wang X (2014) The tangled web of password reuse. In: NDSS symposium 2014, p 7
36. Li Y, Wang H, Sun K (2016) A study of personal information in human-chosen passwords and its security implications. In: IEEE INFOCOM 2016—the 35th annual IEEE international conference on computer communications. IEEE, pp 1–9
37. Merhav N, Cohen A (2020) Universal randomized guessing with application to asynchronous decentralized brute force attacks. IEEE Trans Inf Theory 66(1):114–129
38. Lu B, Zhang X, Ling Z, Zhang Y, Lin Z (2018) A measurement study of authentication rate-limiting mechanisms of modern websites. In: Proceedings of the 34th annual computer security applications conference, pp 89–100
39. Pal B, Daniel T, Chatterjee R, Ristenpart T (2019) Beyond credential stuffing: password similarity models using neural networks. In: 2019 IEEE symposium on security and privacy (SP). IEEE, pp 417–434
40. Guri M, Shemer E, Shirtz D, Elovici Y (2016) Personal information leakage during password recovery of internet services. In: 2016 European intelligence and security informatics conference (EISIC). IEEE, pp 136–139
41. Bailey DV, Dürmuth M, Paar C (2014) Statistics on password re-use and adaptive strength for financial accounts. In: International conference on security and cryptography for networks. Springer, pp 218–235
42. Emin Islam T (2015) Cracking more password hashes with patterns. IEEE Trans Inf Forens Secur 10(8):1656–1665
43. Stobert E, Biddle R (2014) The password life cycle: user behaviour in managing passwords. In: 10th symposium on usable privacy and security (SOUPS 2014), pp 243–255
44. Kelley PG, Komanduri S, Mazurek ML, Shay R, Vidas T, Bauer L, Christin N, Cranor LF, Lopez J (2012) Guess again (and again and again): measuring password strength by simulating password-cracking algorithms. In: 2012 IEEE symposium on security and privacy. IEEE, pp 523–537
45. Shay R, Bauer L, Christin N, Cranor LF, Forget A, Komanduri S, Mazurek ML, Melicher W, Segreti SM, Ur B (2015) A spoonful of sugar? The impact of guidance and feedback on password-creation behavior. In: Proceedings of the 33rd annual ACM conference on human factors in computing systems, pp 2903–2912
46. Wang D, Wang P, He D, Tian Y (2016) Birthday, name and bifacial-security: understanding passwords of Chinese web users. In: 28th USENIX security symposium (USENIX security 19), pp 1537–1555
47. Wang D, Zhang Z, Wang P, Yan J, Huang X (2016) Targeted online password guessing: an underestimated threat. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pp 1242–1254
48. Wang KC, Reiter MK (2019) How to end password reuse on the web. In: Proceedings of ACM CCS
49. Grassi PA, Fenton JL, Newton EM, Perlner RA, Regenscheid AR, Burr WE, Richer JP, Lefkovitz NB, Danker JM, Choong Y-Y et al (2017) NIST special publication 800-63b: Digital identity guidelines. Enrollment and identity proofing requirements. https://pages.nist.gov/800-63-3/sp800-63b.html
50. Jaggard AD, Syverson P (2018) Oft target. In: Proceedings of the PET
51. Adams A, Sasse MA (1999) Users are not the enemy. Commun ACM 42(12):40–46
52. Bonneau J (2012) The science of guessing: analyzing an anonymized corpus of 70 million passwords. In 2012 IEEE symposium on security and privacy. IEEE, pp 538–552
53. Mazurek ML, Komanduri S, Vidas T, Bauer L, Christin N, Cranor LF, Kelley PG, Shay R, Ur B (2013) Measuring password guessability for an entire university. In: Proceedings of the 2013 ACM SIGSAC conference on computer & communications security, pp 173–186
54. https://web.archive.org/web/20080705140750 http://www.antsight.com/zsl
55. https://www.kaggle.com/wjburns/common-password-list-rockyoutxt