Accelerated and Intelligent Password Cracking with Performance Optimization

Suresh Kumar C., Raja J. and Ragavan R.
Department of Computer Science and Engineering, School of Computing, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai-600062, Tamil Nadu, India

Keywords: Password Authentication, Biometrics, Cyber Security, Optimize Password Cracking, Multi-Factor Authentication, Brute Force, Dictionary Attack, Hybrid Attack.

Abstract: In the increasingly digitised world, security is a huge concern. Among all the security mechanisms, password authentication is the most popular one, however, weak or simple passwords are still susceptible to security attacks. This work introduces a system for distributed password cracking, for helping recover lost or forgotten passwords as well as to assess the weaknesses of current password management methodologies Furthermore, the system leverages cloud computing resources in order to allocate tasks towards workers, in-order to enhance productivity and scalability. The study also examines the efficacy of password strength, hashing algorithm, and encryption method in enhancing password security. The resulting system combines many attack tactics such as brute-force, dictionary, or hybrid approach as well as state-of-the-art optimization strategies. Anticipating passwords with heuristic-based predictions and machine learning model, and running upon GPU-accelerated parallel devices, the system is orders of magnitude faster than the approaches that try with candidate passwords alone. Ultimately, the project seeks to raise cybersecurity awareness by showing how weaknesses exist in password-based authentication and encourage the implementation of stronger, more secure authentication measures, such as multi-factor authentication (MFA) and biometrics.

1 INTRODUCTION

Passwords are the most widespread and least expensive access control mechanism applied to defend sensitive information on the internet, and they can be found on a wide variety of online platforms, for an endless range of applications. There are still weak, common, or reused passwords, and we know those are a gold mine, open to brute force, dictionary, and credential-stuffing attacks. Hackers and other cyber criminals always take advantage of these gaps, which result in breaches, identity theft, and data leaks. 1.2.2 Project Purpose The purpose of the Optimized Password Cracker System Project is to create a faster and efficient password recovery system based on modern computing methodologies. Combining established machine learning algorithms, heuristic based predictions, GPU accelerated computation, and distributed computing, this project aims to improve the reliability of password cracking, and to highlight flaws in current password security methods. This project is written with an ethics-first approach and it is specifically aimed at penetration testing, security auditing and other research into password-based system security, so that security professionals can pick weak passwords from a list and recommend the system's owner to implement strong security measures. In addition, we also analyse the performances of some well-known hashing and encryption methods in order to compare and evaluate them. By pointing out the vulnerabilities in weak password implementation, this initiative encourages the implementation of stronger forms of authentication, including multi-factor authentication (MFA), biometric verification and password managers. Ultimately, the Optimized Password Cracker acts as a cybersecurity educator that urges people and companies to start using better password management practices. “The project improves upon traditional password recovery methods by using a smart alternate approach to crack passwords with the help of the word lists provided. Through GPU acceleration of high speed computations, adaptive learning for evolving strategy, and AI-based recognition of password patterns, this system minimizes computational burden and enhances the

C., S. K., J., R. and R., R.
Accelerated and Intelligent Password Cracking with Performance Optimization.
DOI: 10.5220/0013920100004919

In *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT'25 2025) - Volume 4*, pages 741-749
ISBN: 978-989-758-777-1

success rate of the attack. Ethical security norm is followed to make it responsible use to avoid unauthorized misuse. The work also stresses on the use of strong password policy, secure hashing mechanism (bcrypt, Argon2), multi-factor authentication (MFA) to reduce the security threats.

This research focuses on bridging the gap between traditional brute-force techniques and modern AI-enhanced password recovery methods, demonstrating how performance optimization can drastically improve security assessments while maintaining compliance with ethical hacking practices.

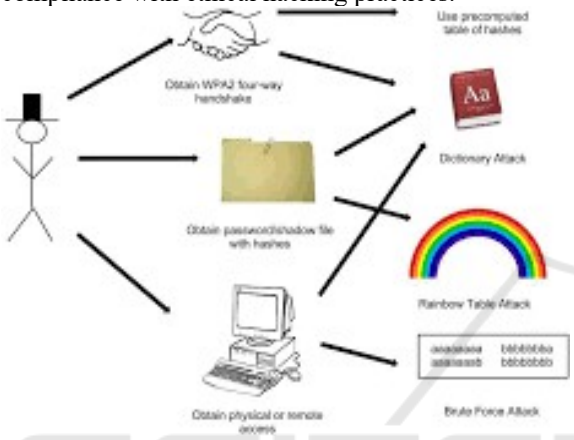

Figure 1: Use case diagram.

Figure 1 depicts the different ways in which attackers, assuming they get hold of authentication-data would try to crack a password. It as an example showing how an attacker may be able to extract credentials (i.e capturing a WPA2 4-way handshake, obtaining a password hash file, physical access, remote access). Techniques for cracking of password Once the authentication data has been obtained, there are various ways to crack the password. Fall: precompute hash tables make it possible for an attacker to find passwords by comparing them to pre-computed hash values. Dictionary attacks perform password guessing on the basis of predefined wordlists of common words and charac-ter patterns. Rainbow table attacks rely upon massive tables of precalculated hashes to speed up cracking. A brute force attack is an attempt to break the password of a user account by systematically trying every possible combination of characters. This demonstrates the risks of short/simple passwords and the key defence strong encryption and security practices bring to bear against unauthorized access.

2 TRADITIONA METHODOLOGY

The conventional process of password cracking involves algorithms to reverse-engineer passwords from encrypted or hashed data in a structured manner using classical algorithmic tools. These techniques are brute force attacks, dictionary attacks and rainbow table attacks. Brute force attacks entail guessing combinations of characters in sequence until the correct password is guessed. This method is exhaustive, however grows in inefficiency as the password size and complexity goes higher. Dictionary attacks rely on lists of preselected passwords and common words or phrases, allowing far fewer trials than the naïve and explicit password cracking methods. Rainbow table attacks use large tables of precomputed hash values that enable attackers to reverse cryptographic hashes rapidly rather than computing them on the fly.

- **Brute Force Attack**: This method involves systematically generating and testing every possible password combination until the correct one is found. It is highly time-consuming, especially for complex passwords.
- **Dictionary Attack:** Instead of testing random combinations, this technique uses a predefined list of commonly used passwords, words, and phrases to expedite the cracking process.
- **Rainbow Table Attack:** This method utilizes large precomputed tables of hashed values, enabling quick reversal of cryptographic hashes without generating them in real time. **Hybrid Attacks** – A combination of dictionary and brute force methods where known words are modified with numbers and special characters to guess more complex passwords.
- **Keylogging & Phishing**: Some traditional approaches involve capturing keystrokes or tricking users into revealing their passwords rather than breaking encryption directly.

These constraints are over come through the aid of deep learning techniques with optimization algorithms.

3 PROPOSED SYSTEM

The novelty IT expert system yet based on an advanced and clever way (with artificial intelligence, machine learning, GPU CUDA, heuristic and other

modern approaches) for cracking in pairing list is suggested. This method, unlike brute-force and dictionary-based attack, dynamically changes the way of the attack with the lessons drawn from the patterns until the system finds the password. With AI-fed models, the system is capable of predictive analysis and can therefore focus on likely combinations rather than just trying them all.

Moreover, GPU parallel process is introduced to highly accelerate computation speed, which can efficiently conduct large-scale password-cracking process. What is more, cloud-based distributed computing further facilitates the scalability of the system, in which multiple computing elements cooperate smoothly to decrease processing time and improve the success rates.

The overall security of the proposed scheme and its security and ethical requirements, restrict the password-cracking behavior to be conducted according to the legitimate working range. Access controls and audit logging are incorporated to track all access to the data, minimizing unauthorized use and achieving compliance with cybersecurity requirements.

The UI has also been designed with the user in mind and have been optimized for ease of use, complete with an interactive dashboard to upload password hashes, select the target attack strategy, and monitor real-time progress. It's designed for use by cybersecurity professionals in ethical hacking, pentesting, and security auditing, and in any scenario in which systems are to be evaluated and/or tested on for changes in the security infrastructure.

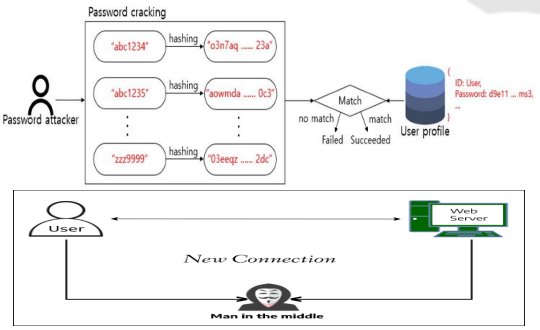

Figure 2: Password cracking process.

Leveraging AI predictions as well as high-performance computing and an adaptive approach, the developed system is faster, more accurate and more resource-sensitive than traditional approaches, while remaining ethically sound.

Figure 2 repesent password-cracking attack, where an attacker systematically guesses passwords by hashing input strings and comparing them to stored password hashes.

The image highlights the brute-force approach, where multiple possible passwords are hashed and compared against the stored values in a database. If a match is found, the attacker successfully retrieves the password. Common password-cracking techniques:

- Brute-force attack: Tries every possible combination. Dictionary attack: Uses a predefined list of common passwords.
- Rainbow table attack: Uses precomputed hash values to quickly reverse-engineer a password.
- Preventive measures: Use strong hashing algorithms (bcrypt, PBKDF2, Argon2), enable multi-factor authentication (MFA), enforce strong password policies, and implement rate-limiting techniques.

4 ACCELERATED AND INTELLIGENT PASSWORD CRACKING WITH PERFORMANCE OPTIMIZATION

Password security continues to be an issue of vital importance even in the new digital era while sophisticated attacks are launched to beat authentication systems. Classic password-cracking methods like brute force and dictionary attacks have advanced to a new level where optimization-based approaches and smart computing models are taken advantage of.

Speedy password cracking is made possible thanks to the help of hardware acceleration, parallelisation and optimalisation algorithms. Cryptographic hash calculations are often accelerated by Graphics Processing Units (GPUs) and Field-Programmable Gate Arrays (FPGAs). Compared with traditional, CPUs, GPUs are well known for their good performance in taking massive parallelisable tasks, thus being suitable for hash-rate computation in password cracking.

Solutions Intelligent password-cracking algorithms combine machine learning and artificial intelligence to better guess passwords. If attacker has access to the password patterns of the user such that the AI based model can generate likely guesses then the time a attacker needs to successfully guess is drastically decreased. Leveraging deep learning models built with massive password leak databases, the attackers bypass conventional security by directly

targeting commonly used passwords, rather than brute-forcing all the possible combinations.

Algorithm and heuristic based performance improvements and adaptive approaches are used for password cracking. Hashing mechanisms like bcrypt, PBKDF2 and Argon2 are designed to mitigate high speed attacks by introducing computational delays and memory-hard functions. Consider rule-based attack, hybrid attack, and the use of precomputed hash tables (rainbow table) as optimizations and you can see that cracking passwords is already much faster.

The increasing use of fast and smart password breaking methods illustrates the necessity of having good security practices. Companies and users need to enforce stronger password policies and 2FA, and also implement strong cryptography to protect sensitive credentials. With the growth of computing power, authentication mechanisms need to adapt to meet new password threats.

Password cracking has changed drastically with increasing computing power and optimized algorithms. Adversaries use high speed hardware, key derivation functions, and performance-enhancing tools to circumvent password security. Nowadays brute-force attacks are complemented with ai-based methods, heuristic examination or data mining which makes the security of passwords a daunting task to organizations as well as to people.

Table 1: Password searches.

	lower case	lower/upper	lower/ upper/ digits	lower/upper/ digits/ symbols
1	26	52	62	95
2	676	2704	3844	9025
4	456,976	7,311,616	14,766,336	81,450,625
8	2.09×10^{11}	5.35×10^{13}	2.18×10^{14}	6.63×10^{15}
16	4.36×10^{22}	2.86×10^{27}	4.77×10^{28}	4.40×10^{31}

Table 1 The table illustrates the total number of possible password combinations based on different character sets and password lengths. It categorizes password strength into four different character set groups: lowercase letters (26 characters), lowercase and uppercase letters (52 characters), lowercase, uppercase, and digits (62 characters), and a full set including lowercase, uppercase, digits, and symbols (95 characters). Each row represents an increasing password length, starting from 1 character up to 16 characters. The table demonstrates how the total number of possible passwords increases exponentially with length. For instance, with just one character, the possible combinations range from 26 (lowercase) to 95 (all characters). However, at 16 characters, the possibilities range from 4.36×10224.36 \times 10^{22}4.36×1022 (lowercase) to 4.40×10314.40 \times 10^{31}4.40×1031 (full character set).
This data highlights the importance of increasing password length and complexity to enhance security. A longer password with a diverse character set significantly improves security by making brute-force attacks infeasible. The exponential growth in possible combinations as length increases demonstrates why modern security practices emphasize using long, complex passwords to withstand advanced cracking techniques.

4.1 Optimization algorithm

Optimization algorithms play a crucial role in enhancing the efficiency of computational tasks, including password cracking. These algorithms aim to minimize computational complexity and maximize the success rate of cracking attempts. Traditional brute-force attacks, which involve testing every possible combination, are highly inefficient and time-consuming. Optimization techniques such as heuristic algorithms, genetic algorithms, and machine learning models improve performance by narrowing down the search space and prioritizing probable password patterns. Heuristic-based approaches analyze commonly used password structures, enabling attackers to focus on high- probability candidates. Genetic algorithms simulate evolution by selecting the best candidates, applying crossover and mutation, and iterating until an optimal solution is found. Deep learning models leverage vast datasets to predict password tendencies, refining attack strategies dynamically. Rainbow tables optimize attacks by precomputing password hashes, allowing quick lookups instead of real-time computations. Markov models generate probable passwords based on statistical likelihood, reducing unnecessary attempts. Parallel computing and distributed processing use multiple CPU or GPU cores to execute large-scale attacks efficiently.

4.2 Object detection and tracking

Object detection and tracking is a hot topic in computer vision, and is applied in surveillance, autonomous vehicle, robot, augmented reality, etc.

Object detection requires the recognition as well as localization of objects in an image or video frame, while object tracking deals with the association of objects over multiple frames in a video sequence. State-of-the-art object detection methods are based on deep learning approaches: Convolutional Neural Network (CNN), Region Convolutional Neural Network (R-CNN) and You Only Look Once (YOLO). These models fuse object classification and localization very effectively and can run in real time with very high object classification accuracy. Faster R- CNN and Single Shot MultiBox Detector (SSD) strike a balance between detection accuracy and speed that are appealing to practical use. On the other hand, object tracking maintains a tracking record of the detected objects through the frames. Tracking algorithms consist of filters, e.g., correlation filters such as MOSSE and deep learning-based trackers such as DeepSORT, methods are based on optical flow and etc. The former predict the motion of the object by predicting its future location, the latter cope with both simple and complex motion patterns.

Challenges in object detection and tracking include occlusion, variations in lighting, motion blur, and real-time constraints. Advanced solutions integrate multiple sensors, improve feature extraction, and apply reinforcement learning to enhance performance. These techniques are crucial in applications like automated surveillance, intelligent transportation, and interactive human-computer interfaces.

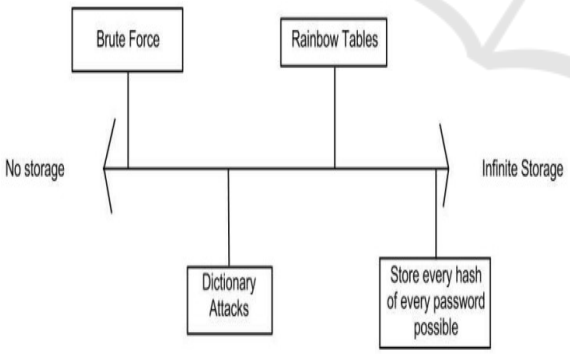

Figure 3: The spectrum of possibilities for password cracking attacks.

Figure 3 demonstrates the idea of applying a time-memory trade-off to the problem of cryptanalysis was first proposed. Though his approach attacks a cipher text encrypted using the Data Encryption Standard (DES), it requires only minor modifications to instead attack a password hashing scheme. Given a precomputed table smaller than what would be employed in an attack.

4.3 Redundancy Measures

Redundancy measures in password cracking and authentication systems refer to strategies that enhance security by making password cracking more difficult and protecting authentication mechanisms against attacks. These measures are crucial in preventing unauthorized access, reducing vulnerabilities, and increasing computational effort for attackers.One common redundancy measure is password salting, where a unique random value is added to each password before hashing. This prevents attackers from using precomputed hash tables, such as rainbow tables, to crack passwords efficiently. Hashing algorithms with high computational cost, like bcrypt, PBKDF2, and Argon2, introduce redundancy by requiring multiple iterations, making brute-force attacks slower.Another redundancy measure involves multi-factor authentication (MFA), where additional verification steps, such as one-time passwords (OTPs), biometrics, or security tokens, complement traditional password-based authentication. This reduces reliance on passwords alone and significantly enhances security.Account lockout mechanisms and rate limiting add redundancy by restricting repeated login attempts, preventing brute-force and dictionary attacks. CAPTCHA challenges further increase complexity for automated password-cracking bots.

4.4 Integration System

Integration in the context of security and authentication refers to the seamless combination of multiple security measures, systems, or technologies to enhance overall protection. In password authentication and cybersecurity, integration plays a crucial role in improving efficiency, user experience, and resilience against attacks. One key aspect of integration is combining authentication mechanisms such as multi-factor authentication (MFA), biometric authentication, and token-based authentication to provide layered security. This reduces reliance on a single method, making it harder for attackers to gain unauthorized access.Another significant integration approach involves the synchronization of authentication systems across multiple platforms and services. Single Sign-On (SSO) allows users to access multiple applications with a single set of credentials, reducing password fatigue and minimizing the risk of weak passwords.Additionally, cybersecurity systems integrate threat intelligence, anomaly detection, and behavioral analysis to identify potential breaches. Advanced monitoring tools and artificial intelligence-driven security solutions work alongside traditional

authentication mechanisms to detect and prevent suspicious activities in real-time.

5 EXPERIMENTAL ANALYSIS

Experiment analysis evaluates password cracking efficiency, optimization algorithms, security measures, and redundancy techniques to enhance authentication system performance and resilience.

Table 5 demonstrates the dictionary data is viewed to compare the efficiency of the data. Table 4 demonstrates the brute force attack works by trying every possible combination of the users.

Table 4: Brute Force Data.

Password	Length	Attempts/Hashes	MD5 Time (Seconds)	SHA-1 Time (Seconds)
a	1	2	0.001	0.001
z	1	27	0.001	0.001
an	2	380	0.003	0.005
or	2	502	0.004	0.003
a1	2	1038	0.010	0.006
z9	2	2159	0.017	0.017
aA1	3	216029	1.705	1.714
z9	3	249381	2.126	1.966
aZ16	4	14970377	114.995	119.043
16Za	4	460207	3.617	4.257
abcd	4	80975	0.663	0.679
9999	4	14641	0.109	0.123
fast	4	407545	3.698	4.077
Slow	4	3466988	25.294	29.198
apple	5	290459	2.224	2.781
zebra	5	887355	6.493	7.301
fast1	5	53515623	394.666	440.198
Slow9	5	982495000	7310.119	8498.992
abcde	5	2738180	20.580	23.8
quick	5	5912046	45.004	49.023
pass1	5	53464980	419.349	454.406
abcdef	6	88831622	672.138	747.456
aaaaaa	6	214985	1.648	1.775
passwd	6	70006643	539.175	616.408
Total:		1295527188	9697.017	11147.821
Average(hashes/second):			133600.5895	116213.4903

Table 5: Dictionary data.

Password	Length	Attempts/Hashes	MD5 Time (Seconds)	SHA-1 Time (Seconds)
a	1	2	0.001	0.001
l	1	1	0.001	0.001
an	2	7	0.001	0.001
or	2	65	0.001	0.001
and	3	124	0.001	0.001
try	3	763	0.008	0.008
test	4	3471	0.026	0.028
pass	4	2791	0.021	0.024
apple	5	4162	0.031	0.034
zebra	5	9967	0.073	0.078
kitten	6	14710	0.108	0.115
hacker	6	13901	0.100	0.136
balloon	7	21030	0.152	0.194
puppies	7	300018	0.214	0.263
password	8	44857	0.321	0.364
computer	8	37618	0.303	0.299
Total:		453448	1.362	1.484
Average(hashes/second):			332928.047	305557.9515

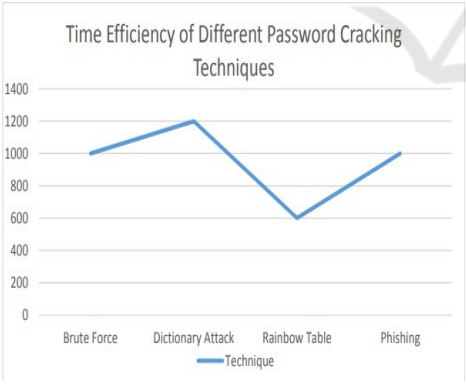

Figure 4: Time Efficiency.

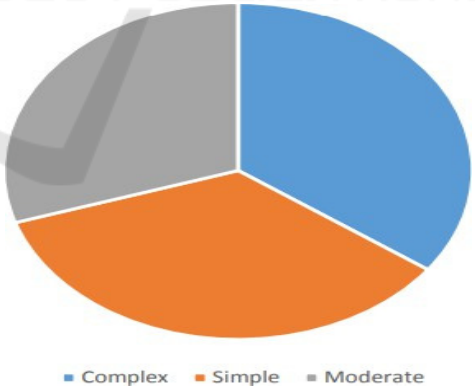

Figure 5: Password Complexity.

The Figure 4 demonstrates the speed at which password cracking techniques operate is a critical factor in cybersecurity risk assessment. A rapid crack can enable swift unauthorized access, posing severe threats to personal and organizational security.

Figure 5 demonstrates the password complexity by 'Simple,' 'Moderate,' and 'Complex' based on their length and character variety, we aim to unravel the prevailing trends in user- generated passwords. This exploration into password complexity not only illuminates the existing vulnerabilities but also informs security strategies, aiding in the creation of stringent password policies.

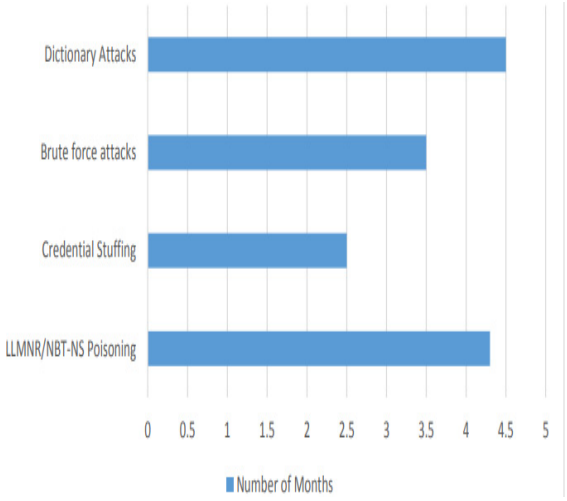

Figure 6: Success and Failure Bar Chart.

Figure 6 demonstrates the Success and Failure Bar Chart. By visualizing our findings in a bar chart, we aim to illuminate the patterns and trends underlying the success and failure rates of these exploitation techniques over a specific period.

6 CONCLUSIONS

The Optimized Password Cracker project highlights the importance of password security, ethical hacking, and penetration testing in modern cybersecurity. By leveraging advanced password-cracking techniques, AI-based predictions, GPU acceleration, and cloud computing, this project significantly enhances the efficiency and speed of password recovery while analyzing authentication vulnerabilities. Through this system, cybersecurity professionals can identify weaknesses in password-based authentication methods, assess the strength of hashing algorithms, and reinforce security policies to mitigate cyber threats. The project also emphasizes ethical and legal considerations, ensuring that the tool is used strictly for authorized security audits and educational purposes. Ultimately, the project serves as a powerful cybersecurity tool that contributes to strengthening digital security by raising awareness about password vulnerabilities and promoting best practices for secure authentication. Future enhancements may include improving AI-driven password prediction models, integrating real-time threat analysis, and expanding cloud-based distributed computing capabilities for large-scale security assessments.

REFERENCES

Arjovsky, M.; Chintala, S.; Bottou, L. {W}asserstein Generative Adversarial Networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Volume 70, pp. 214–223. [Google Scholar]

Cracking. IEEE Trans. Inf. Forensics Secur. 2015, 10, 1776–1791. [Google Scholar] [CrossRef]

CyberWarzonehttp://cyberwarzone.com/massive-collection- password- wordlists- recover lost-password/, last accessed on 22 March 2015

D. Pahuja and P. Sidana, "Implementing and comparing different password cracking tools", International Research Journal of Engineering and Technology (IRJET), vol. 8, no. 5, pp. 2089-2095, 2021.Yazdi, S.H. Probabilistic Context-Free Grammar Based Password Cracking: Attack, Defense and Applications. Ph.D. Thesis, Department of Computer Science, Florida State University, Tallahassee, FL, USA, 2015. [Google Scholar]

Dell'Amico, M., Mihajlin, I., Carminati, B., Ferrari, E. (2010). Password Strength: An Empirical Analysis. IEEE Transactions on Information Forensics and Security.[DOI:10.1109/TIFS.2010.2046759].Discusses password entropy and cracking feasibility based on real-world datasets.

Gasti, P., Rasmussen, K. (2012). On the Security of Password Manager Database Formats. ACM Conference on Computer and Communications Security.Evaluates vulnerabilities in password management tools.

Gershon Kedem and Yuriko Ishihara. Brute force attack on unix passwords with simd computer. In Proceedings of the 8th conference on USENIX Security Symposium - Volume 8, pages 8–8, Berkeley, CA, USA, 1999. USENIX Association.

He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recogni-tion. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pat-tern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. Neural Com-put. 1997, 9, 1735–1780. [Google Scholar] [CrossRef] [PubMed].

Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. Neural Comput. 1997, 9, 1735–1780. [Google Scholar] [CrossRef] [PubMed]

Houshmand, S.; Aggarwal, S. Using Personal Information in Targeted Grammar-Based Probabilistic Password Attacks. IFIP Adv. Inf. Commun. Technol. 2017, 511, 285–303. [Google Scholar] [CrossRef] [Green Version]

Houshmand, S.; Aggarwal, S.; Flood, R. Next Gen PCFG Password

Ibrahim Alkhwaja, Mohammed Albugami, Ali Alkhwaja, Mohammed Alghamdi, Hussam Abahussain, Faisal Alfawaz, et al., "Password Cracking with Brute Force Algorithm and Dictionary Attack Using Parallel Programming", Applied Sciences, vol. 13, no. 10, pp. 5979, 2023.

Katz, J., Lindell, Y. (2020). Introduction to Modern Cryptography. CRC Press.Covers theoretical foundations of cryptographic security, including hash functions used in password storage.

Kaufman, C., Perlman, R., Speciner, M. (2016). Network Security: PrivateCommunication in a Public World. Prentice Hall.Covers network security protocols and authentication mechanisms.

Li Zhuang, Feng Zhou, and J. D. Tygar. Keyboard acoustic emanations revisited. ACM Trans. Inf. Syst. Secur., 13:3:1–3:26, November 2009.

Narayanan, A., Shmatikov, V. (2005). ”Fast Dictionary Attacks on Passwords Using Time-Memory Tradeoff.” Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS).

National Institute of Standards and Technology. Digital Identity Guidelines. 2004. Available online: https://pages.nist.gov/800-63- 3/ (accessed on 10 June 2017).

S.Aggarwal, S. Using Personal Information in Targeted Grammar- Based Proba-bilistic Password Attacks. IFIP Adv. Inf. Commun. Technol. 2017, 511, 285–303.[Google Scholar] [CrossRef] [Green Version]

Weir, M., Aggarwal, S., Collins, M., Stern, H. (2009). ”Testing Metrics for Password Creation Policies by Attacking Large Sets of Revealed Passwords.”ACM Conference on Computer and Communications Security (CCS).

William, M.; Blase, U.; Sean M., S.; Saranga, K.; Lujo, B.; Nicolas, C.; Lorrie Faith, C. Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks. In Proceedings of the 25th {USENIX} Security Symposium ({USENIX} Security 16), Austin, TX, USA, 10–12 August 2016; pp. 175–191. [Google Scholar]

Z. Zhang and P. Liu, "A hybrid-cpu-FPGA-based solution to the recovery of sha256crypt-hashed passwords", IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 1-23, 2020.

Z. Zhang, P. Liu, W. Wang, S. Li, P. Wang and Y. Jiang, "High-performance password recovery hardware going from gpu to hybrid cpu-FPGA platform", IEEE Consumer Electronics Magazine, vol. 11, no. 1, pp. 80-87, 2022.