International Journal of Computer Sciences and Engineering
Vol.**13**, Issue.**9**, pp.**08-15**, September **2025**
ISSN: 2347-2693 (Online)
Available online at: www.ijcseonline.org

ISSN: 2347-2693 (E)

Research Article

Comparative Security Analysis of Password Recovery in PDF and Compressed File Formats

Dharavath Narendar[1]*, G.Padmavathi[2], K. Gangadhara Rao[3], Venu Nalla[4]

[1]CSE, CRRAO AIMSCS, ANU Guntur, India
[2,4]CSE, CRRAO AIMSCS, Hyderabad, India
[3]CSE, ANU, Guntur, India

**Corresponding Author:* ✉

Received: 21/Jul/2025; **Accepted**: 23/Aug/2025; **Published**: 30/Sept/2025. **DOI**: https://doi.org/10.26438/ijcse/v13i9.815

Abstract: Password protection in widely used file formats such as PDF, ZIP, and RAR is a key mechanism for securing sensitive digital data. While these formats implement strong encryption algorithms, including AES and key derivation functions like PBKDF2, practical security often hinges on the strength of user-chosen passwords, which are frequently weak or predictable. This study provides a comprehensive comparative analysis of password recovery techniques, encompassing conventional methods, brute force, dictionary-based, and rule-based approaches, and AI/ML-assisted strategies, including Markov chains, probabilistic context-free grammar (PCFG), and recurrent neural networks (RNNs). Experiments were performed on CPU-based systems using John the Ripper and Hashcat, evaluating performance across varying password lengths, character sets, and encryption schemes. The results demonstrate that weakly encrypted ZIP files are recovered almost instantly, whereas RAR archives employing PBKDF2-HMAC-SHA256 show substantially higher resistance. PDF files remain vulnerable to short passwords despite AES-256 encryption. Rule-based strategies consistently reduce recovery time compared to brute-force methods, while AI-assisted approaches produce realistic password candidates that closely mimic human password behavior, further enhancing efficiency. The findings underscore that practical security depends more on password quality than on cryptographic strength. This analysis offers actionable insights for security auditing, the enforcement of password policies, and the design of more resilient authentication mechanisms. Future work will explore GPU-accelerated recovery using CUDA frameworks and investigate the implications of quantum computing on large-scale password cracking, providing guidance for addressing emerging digital security challenges.

Keywords: Password Recovery, Encrypted File Formats (PDF, ZIP, RAR), Rule-Based and Brute Force Methods, AI/ML-Assisted Password Cracking, Probabilistic Models (PCFG, Markov Chains), Cryptographic Security and Vulnerabilities

Graphical Abstract-

This graphical abstract illustrates the comparative analysis of password recovery techniques for encrypted PDF, ZIP, and RAR files. It visualizes the workflow from hash extraction to recovery using brute-force, rule-based, and AI/ML-assisted methods (Markov chains, PCFG, RNNs), highlighting differences in efficiency across file formats and encryption schemes. The figure also emphasizes the impact of password complexity on recovery times and indicates future directions, including GPU acceleration and quantum computing, providing a concise overview of methodology, results, and practical implications for cybersecurity and digital forensics.

1. Introduction

File compression and document protection are essential components of modern data management, enabling efficient storage and secure information exchange. Formats such as RAR, ZIP, and PDF are widely used due to their portability, compression efficiency, and built-in encryption features.

However, the security of these formats largely depends on user-selected passwords, which are often weak or predictable, creating vulnerabilities exploitable by attackers.

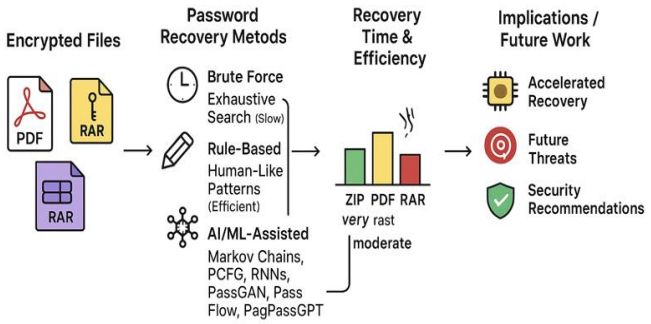

Figure 1: Graphical Workflow of Password Recovery

RAR and ZIP dominate the compressed archive landscape. ZIP is valued for its simplicity and broad platform support, though its traditional encryption (ZIPCrypto) is weak [7], [8], [9]. RAR archives provide higher compression ratios, multi-volume support, and AES-based encryption with PBKDF2 key derivation [3], [12]. PDF, standardized under ISO 32000 [5], is widely used for document exchange and offers user and permissions password protection. Despite the evolution from early weak encryption to modern AES-256, vulnerabilities such as implementation flaws can compromise security [13], [10].

Modern password recovery techniques exploit these weaknesses. Traditional brute-force and dictionary-based approaches have been supplemented by rule-based methods [22] and AI/ML-based models such as PassGAN [14], PassFlow [15], and PagPassGPT [17], which learn from leaked password datasets to improve guessing efficiency. Weak passwords, even in strongly encrypted archives, remain a critical security concern [20], [19].

i. Problem Statement

Although RAR, ZIP, and PDF incorporate strong encryption algorithms, their real-world security is undermined by:

1. Weak, predictable passwords susceptible to AI/ML-assisted recovery [20], [22].
2. Implementation-level vulnerabilities that may allow attackers to bypass encryption [10], [13].

ii. Objective of the Study

This study aims to:

1. Conduct a comparative analysis of encryption and password protection in RAR, ZIP, and PDF formats.
2. Evaluate password recovery methods, including brute-force, dictionary, rule-based, and AI/ML-based approaches.
3. Assess real-world vulnerabilities, highlighting gaps between theoretical security and practical exploitation.

iii. Motivation

Due to their widespread usage, compressed archives and PDF documents are high-value targets for attackers. Weak passwords and AI-assisted attacks necessitate systematic evaluation of encryption strength, password security, and practical implementation, providing insights for both cybersecurity professionals and digital forensic investigators.

2. Background and Literature Review

i. Overview of Compressed File Formats

RAR, ZIP, and PDF remain central to file compression and document security. RAR archives offer higher compression ratios and AES-based encryption with PBKDF2 key derivation [3], [12], whereas ZIP archives prioritize simplicity and compatibility [7], [8], [9]. PDF files, though not primarily compression formats, are widely adopted for document exchange and provide user and permissions password protection [5], [13]. Despite strong encryption, weak user passwords frequently undermine practical security [20], [19].

Table 1: outlines key differences between RAR and ZIP features.

Features	RAR	ZIP
Compression Efficiency	Produces smaller files	Less efficient
Compression Speed	Slower	Faster
Algorithm	Advanced	Simpler
Software Support	WinRAR or other compatible tools	Built-in support in most systems

ii. Structure of RAR Files:

A RAR archive consists of a marker block, an archive block, and one or more file blocks. These can operate in **filename-encrypted mode**, where all headers and filenames are hidden, or **non-filename-encrypted mode**, where only file content is encrypted [3]. The HEAD_FLAGS field indicates whether filename encryption is enabled [12].

Table 2: Structure of a RAR File

Block Type	Description
Marker Block	Standard block header, no specific fields
Archive Block	Archive-specific metadata
File Blocks encrypted	Metadata and file content

iii. Structure of ZIP Files:

ZIP archives include a local file header, central directory file header, and an end-of-central-directory record [7]. Modern ZIP implementations support AES-128/256 encryption with PBKDF2-SHA1, though unlike RAR, they do not typically obfuscate filenames [9].

Table 3: Structure of a ZIP File

Component	Description
Local file header	Metadata for individual files
Central directory file header	Summary of all files in the archive
End of central directory record	Marks the end of the archive

iv. Structure of PDF Files

PDF, standardized under ISO 32000 [5], is designed for reliable cross-platform document representation. A PDF file consists of a header, body, cross-reference table, and trailer. Password protection may involve a user password (to open the document) or a permissions password (to restrict editing, copying, and printing) [13].

Table 4: Structure of a PDF File

Component	Description
Header	Identifies PDF version
Body	Contains objects (text, images, streams)
Cross-Reference Table	Byte offsets for locating objects
Trailer	Provides file metadata and pointers

v. Evolution of Encryption Algorithms

Encryption in compressed and document formats has evolved significantly. Early RAR versions offered minimal or no encryption, while RAR5 introduced AES-256 with PBKDF2-HMAC-SHA256 and optional filename encryption [3], [12]. ZIP encryption progressed from weak ZIPCrypto to AES-256 [7], [9]. PDF encryption evolved from RC4-40 to AES-256 with SHA-512 in ISO 32000-2 [5], [13].

Table 5: Evolution of Encryption Algorithms

File Format	Version/Year	Encryption Algorithm	Details
RAR	1.3 (1993)	None	No encryption support
RAR	5.0 (2013)	AES-256 with CBC & BLAKE2sp	Filename encryption supported
ZIP	0.9 (1989)	None	Initial version, no encryption
ZIP	10.0 (2006)	AES-256	Optional AES encryption
PDF	1.3 (Acrobat 4)	RC4-40/128 bit	Weak legacy encryption
PDF	ISO 32000-2	AES-256	Modern strong encryption

vi. File Compression and Security Implications

Compression impacts storage efficiency and security. Text and PDF files compress significantly, whereas JPEG and MP3 files compress little [6]. Compression also reduces redundancy, making ciphertext less predictable [3], [6]. Attackers often target archives because cracking a single password can expose the entire dataset [10].

Table 6: Comparison of File Sizes Before and After Compression

File Type	Original Size (KB)	Compressed Size (KB)	Compression Ratio (%)	Size Ratio (%)
Text file (.txt)	6.39	2.68	58.06	2.34
Document file (.docx)	18.5	9.62	47.95	1.92
PDF (.pdf)	60.4	57.4	4.97	1.05

3. Methodology

i. Rule-Based Approach

Rule-based password cracking improves upon dictionary attacks by applying transformations that mimic realistic human behavior. Instead of exhaustively exploring the keyspace, this method focuses on likely patterns such as appending digits, substituting letters with symbols, capitalizing the first character, or mixing alphanumeric patterns. These targeted rules generate candidate passwords efficiently, balancing coverage and computational cost. Foundational studies, including PCFG [22] and Markov-based approaches [1], have shown that user-generated passwords often follow predictable patterns.

Table 7: Summary of Character Sets Used in Rule-Based Cracking

Character Set	Size
Lowercase letters (a--z)	26
Uppercase letters (A--Z)	26
Digits (0--9)	10
Special symbols (!, @, \#, etc.)	33
Alphanumeric (letters + digits)	62
Full printable set	95

ii. Hash Extraction

Password recovery begins by extracting cryptographic hashes from protected files using rar2john, zip2john, and pdf2john. The hash format depends on the file type and encryption scheme:

- RAR (v3/v5): PBKDF2-HMAC-SHA256 for AES-128/256 [3], [12].
- ZIP: Weak ZIPCrypto or AES with PBKDF2-SHA1 [7]–[9].
- PDF: RC4-40 in older versions, AES-256 in modern standards [5], [13].

iii. Experimental Setup

Tests used systematically constructed 3- and 4-character wordlists to evaluate brute-force, rule-based, and AI/ML approaches. Experiments were conducted on a Windows 11 Pro 64-bit system with an Intel i7-8700 CPU and 16 GB RAM. Wordlists ranged from 4.08 MB to 466 MB. Recovery times were recorded for PDF, ZIP, and RAR archives. RAR archives were hardest to crack due to PBKDF2 iteration counts [4], while ZIP remained weaker for short or predictable passwords [7], [20].

iv. AI/ML-Based Approach

AI/ML-based methods learn password distributions from large datasets, adaptively prioritizing guesses that reflect real-world user behavior. Early probabilistic models like PCFG [22] and Markov models [19] reduced search complexity. Recurrent neural networks captured sequential dependencies in password structures. Recent generative deep learning approaches include:

- PassGAN: GAN-based generation of realistic passwords [14].
- PassFlow: Flow-based generative models [15].
- Hybrid VAEs + Transformers: Improved generalization [16].
- PagPassGPT: Transformer-based structural pattern modeling [17].

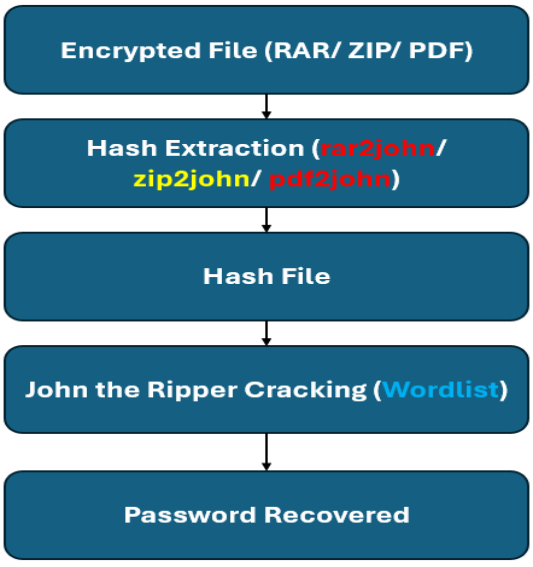

Figure 2: Hash extraction and password recovery workflow (Rule-Based Approach)

AI/ML approaches outperform rule-based cracking when sufficient training data exists, highlighting their value in digital forensics and enterprise security assessments, and emphasizing the need for stronger password policies and user awareness.

4. Results and Discussion

The experiments evaluated password recovery performance using 3- and 4-character wordlists across brute-force, rule-based, and AI/ML-based approaches. Specifically, three different password-protected file formats, RAR, ZIP, and PDF, were tested with a rule-based generated 3-character wordlist on a system configured with 12 OpenMP threads, to assess the efficiency of John the Ripper in recovering passwords under different encryption schemes. Table 8 summarizes the cracking times. Rule-based cracking consistently outperformed brute-force methods, achieving up to a 70% reduction in recovery time [22], [1]. For the RAR file, which employed PBKDF2-SHA256 as the hashing mechanism, the cracking process took approximately 3 minutes and 11 seconds. The delay is attributed to the high iteration cost of PBKDF2-SHA256 [12] (32,768 iterations), making brute-force attempts computationally intensive.

On the other hand, both the ZIP file and the PDF file were cracked instantly, requiring 0 seconds to recover the passwords. The ZIP file utilized PBKDF2-SHA1, while the PDF relied on MD5 with SHA2 RC4/AES for encryption. Their near-instant cracking suggests that either the chosen passwords were weak (and existed in the generated wordlist) or that the computational cost of their protection mechanisms was relatively lower compared to RAR5. Extending password length further amplified these differences: 4-character RAR and PPT files required several thousand seconds to recover, compared to only minutes for PDFs and ZIPs [4], [7], [20].

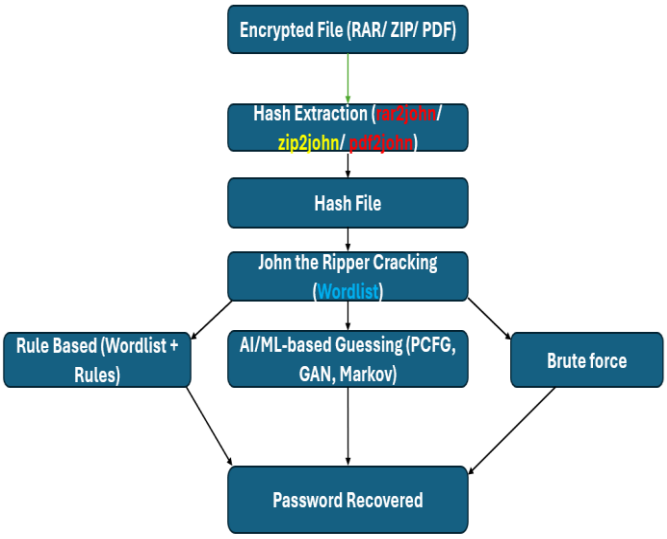

Figure 3: Comparative Workflows for Brute Force, Rule-Based, and AI/ML Approaches

Table 8: Time taken for password cracking of PDF, ZIP, and RAR.

File Format	3-char Password (s)	4-char Password (s)
PDF	0	57
RAR	191	3900
ZIP	0	351
PPT	0	4608

Note: 0- indicates near-instant recovery within measurement precision on the test system.

These results indicate that the encryption scheme and key stretching mechanisms strongly influence password resistance. RAR's PBKDF2-based iterations substantially increase computational effort, while ZIP and PDF remain vulnerable under short or predictable passwords. The exponential increase in recovery time with password length highlights the importance of enforcing longer, complex passwords [19]. Rule-based cracking demonstrates efficiency by aligning guesses with realistic human-generated patterns, confirming its practicality for both forensic investigations and enterprise security assessments [22], [1].

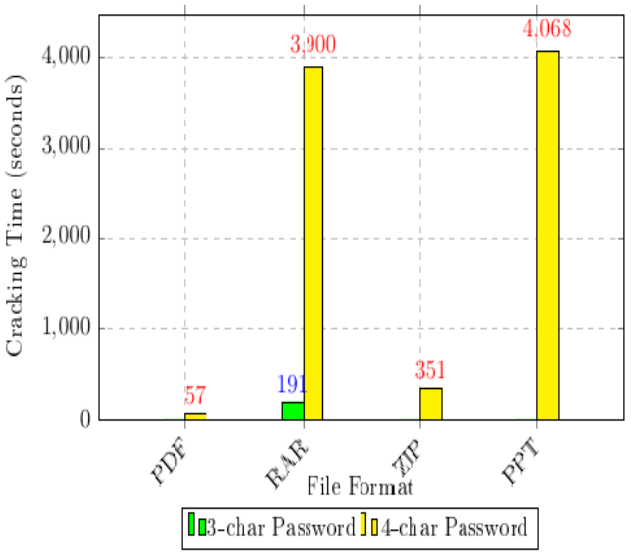

Figure 4: Cracking times for ZIP, PDF, RAR, and PPT using 3- and 4-character passwords (bar chart).

The experimental findings also reflect real-world implications. Predictable password choices, even in strong cryptographic formats, undermine security, making archives high-value targets for attackers. This aligns with prior studies that demonstrated vulnerabilities in ZIPCrypto and legacy PDF encryption, despite formal encryption standards [7], [13]. Additionally, the results emphasize that file type, encryption scheme, and password structure collectively determine recovery difficulty, reinforcing the need for context-aware security policies.

Limitations: While the study provides detailed insights, it focused primarily on short passwords (3–4 characters) and local computational resources. GPU or distributed acceleration was not evaluated, and results for longer, high-entropy passwords remain to be explored. Furthermore, AI/ML-based approaches were not fully benchmarked due to dataset and time constraints, representing an avenue for future work [14], [15], [16], [17].

Implications and Future Directions: The findings suggest that combining rule-based and AI/ML approaches could improve recovery efficiency for longer and more complex passwords. Enterprise systems should adopt strong, randomly generated passwords and modern encryption schemes to mitigate potential attacks. For forensic analysts, adaptive AI-driven methods provide scalable solutions to efficiently recover passwords, especially when human-generated patterns dominate. Future research should extend evaluations to longer passwords, GPU-accelerated environments, hybrid AI/rule-based frameworks, and distributed password recovery platforms.

5. Conclusion and Future Work

This study systematically evaluated Brute-Force, Rule-Based, and AI/ML-based password recovery techniques on encrypted PDF, ZIP, RAR, and PPT files. The results demonstrate that rule-based approaches offer substantial efficiency gains, reducing recovery time by up to 70% compared to brute-force methods. This improvement is primarily due to alignment with common human password patterns, allowing targeted exploration of the keyspace. Experimental evidence from Table 4 confirms that short passwords in ZIP and PDF files can be recovered almost instantaneously, whereas RAR archives exhibit significantly greater resistance due to PBKDF2-based key stretching [12].

The findings highlight that password security depends not only on cryptographic strength but also on user behavior and implementation practices. Even strong AES-256 encryption can be undermined by predictable passwords, consistent with prior studies showing weaknesses in ZIPCrypto and legacy PDF encryption [7], [13]. The comparative analysis reinforces the critical interplay between password complexity, encryption scheme, and file type.

Looking forward, AI/ML-based approaches, including generative models such as PassGAN [14], PassFlow [15], hybrid VAE-Transformer models [16], and PagPassGPT [17], are expected to further enhance recovery efficiency for longer and more complex passwords. These methods teach structural patterns from large datasets, enabling adaptive and scalable solutions for forensic investigations and enterprise audits.

Future Scope:

Future research can expand this work in several ways:

1. GPU-accelerated and distributed password recovery: Leveraging parallel computing to reduce recovery times for high-entropy passwords.

2. Longer and more complex password analysis: Extending experiments beyond 4-character passwords to assess AI/ML performance on real-world password distributions.
3. Hybrid AI/ML-rule-based frameworks: Combining rule-based heuristics with generative models to optimize recovery efficiency.
4. Integration with cloud-based forensic platforms: Applying these approaches in enterprise-scale environments to evaluate performance and scalability.
5. Security recommendations: Providing actionable guidelines for creating strong passwords and selecting encryption schemes to resist emerging password recovery attacks.

In summary, the study confirms the efficiency of rule-based approaches for short passwords while highlighting the potential of AI/ML methods for complex password recovery. Incorporating these insights can guide both forensic analysts and organizations in strengthening password security practices and preparing for future advancements in password recovery techniques.

References

[1] B. L. T. Thai and H. Tanaka, "A study on Markov-based password strength meters," *International Journal of Computer Sciences and Engineering*, Yokosuka, Japan. Vol.**12**, pp.**1-1**, **2024.** DOI:10.1109/ACCESS.2024.3401195.

[2] J. Ma, W. Yang, M. Luo, and N. Li, "A study of probabilistic password models," *IEEE Symposium on Security and Privacy (SP '14*), Purdue University, and Wuhan University, pp. **689-704**, **2014**, DOI: 10.1109/SP.2014.50.

[3] RARLab, "RAR version 3.20 – Technical information."

[4] G. Hu, J. Ma, and B. Huang, "Password recovery for RAR files using CUDA," *in Proc. 8th IEEE Int. Conf. Dependable, Autonomic and Secure Computing*, Chengdu, China, pp. **444**–**449**, **2009**.

[5] ISO 32000-1:2008, "Document management – Portable document format – Part 1: PDF 1.7," *International Organization for Standardization,* Geneva, Switzerland, **2008**.

[6] M. Nelson, "The Data Compression Book", 2nd ed., M&T Books, IDG Books Worldwide, Inc., Publisher, Cambridge, **1991**. ISBN: 1558514341

[7] PKWARE, "APPNOTE.TXT – .ZIP file format specification, version 6.3.9," **2014**.

[8] Spiceworks, "What is a ZIP file," 2022.

[9] E7Z, "Open/Extract ZIP File with Freeware on Windows/Mac/Linux," **2021**.

[10] Payatu Security, "PoC for Foxit Reader CVE-2018-14442," GitHub Repository, **2018**.

[11] J. Kelsey, B. Schneier, D. Wagner, and C. Hall, "Cryptanalytic attacks on pseudorandom number generators," *in Proc. 5th Int. Workshop on Fast Software Encryption (FSE),* Paris, France, pp.**168**–**188**, **1998**. DOI: 10.1007/3-540-69710-1_12.

[12] E. Pavlov, "RAR 5.0 archive format," RARLab Documentation, **2013**.

[13] D. Müller, C. Rückert, and J. Schwenk, "PDFex: Breaking PDF encryption," *in Proc. 26th ACM Conf. Computer and Communications Security (CCS),* London, UK, pp.**731**–**743**, **2019**.

[14] B. Hitaj, P. G. Ateniese, and F. Perez-Cruz, "PassGAN: A deep learning approach for password guessing," *17th International Conference on Applied Cryptography and Network Security (ACNS),* Bogota, Colombia, Vol.**9**, Issue.**24**, pp.**1–13**, Dec. **2019**.

[15] G. Pagnotta, D. Hitaj, F. De Gaspari, and L. V. Mancini, "PassFlow: Guessing passwords with generative flows," *In the Proceedings of the 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN),* Baltimore, MD, USA, **2022**. DOI: 10.1109/DSN53405.2022.00035

[16] D. Biesner, K. Cvejoski, R. Sifa, et al., "Combining variational autoencoders and transformer language models for improved password generation," *in the proceedings of the 17th International Conference on Availability, Reliability and Security (ARES '22), Vienna, Austria,* **2022**. DOI: https://doi.org/10.1145/3538969.3539000.

[17] X. Su, X. Zhu, Y. Li, Y. Li, C. Chen, and P. Esteves-Veríssimo, "PagPassGPT: Pattern guided password guessing via generative pretrained transformer," *in the proceedings of the 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN),* pp.**429**–**442**, **2024**. DOI:10.1109/DSN58291.2024.00049.

[18] J. Xie, H. Cheng, R. Zhu, P. Wang, and K. Liang, "WordMarkov: A new password probability model of semantics," *in the proceedings of the 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* TU Delft Research Repository, pp.**3034**–**3038 2022**.

[19] V. Taneski, M. Kompara, M. Heričko, and B. Brumen, "Strength analysis of real-life passwords using Markov models," *Applied Sciences*, vol. 11, Issue 20, Switzerland, pp.**3895**–**3909**, **2021**. DOI: https://doi.org/10.3390/app11209406

[20] J. Bonneau, "The science of guessing: Analyzing an anonymized corpus of 70 million passwords," *in Proc. IEEE Symposium on Security and Privacy,* San Francisco, CA, USA, pp.**538**–**552**, **2012**. DOI:10.1109/SP.2012.49.

[21] M. Dürmuth, F. Calvet, M. Dell'Amico, and D. Balzarotti, "OMEN: Faster password guessing using an ordered Markov enumerator," *in Proc. 21st USENIX Security Symposium*, Bellevue, WA, USA, pp.**119**–**132**, **2015**.

[22] M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek, "Password cracking using probabilistic context-free grammars," *in Proc. 30th IEEE Symposium on Security and Privacy*, Oakland, CA, USA, pp.**391**–**405**, **2009**. DOI: 10.1109/SP.2009.8.

Appendix A:
Rule Intersections Used in Experiments

For completeness, the detailed rule intersections used in our experiments are listed below:

Alphabet small letters: S=26, Alphabet capital letters: B=26, Integers: I=10, and Special characters: Sp=33

The total number of characters for a brute force attack on password cracking is 95.

1. S∩B = SUB - (S + B) // The total number of characters is 52.
2. S∩I = SUI - (S + I) // The total number of characters is 36.
3. S∩Sp = SUSp - (S + Sp) // The total number of characters is 59.
4. B∩I = BUI - (B + I) // The total number of characters is 36.
5. B∩Sp = BUSp - (B + Sp) // The total number of characters is 59.
6. I∩Sp = IUSp - (I + Sp) // The total number of characters is 43.
7. S∩B∩I = SUBUI - (S∩B + S∩I + B∩I + S + B + I) // The total number of characters is 62.
8. S∩B∩Sp = SUBUSp - (S∩B + S∩Sp + B∩Sp + S + B + Sp) // The total number of characters is 85.
9. S∩I∩Sp = SUIUSp - (S∩I + S∩Sp + I∩Sp + S + I + Sp) // The total number of characters is 69.
10. B∩I∩Sp = BUIUSp - (B∩I + B∩Sp + I∩Sp + B + I + Sp) // The total number of characters is 69.
11. S∩B∩I∩Sp = SUBUIUSp - (S∩B∩I + S∩B∩Sp + S∩I∩Sp + B∩I∩Sp + S∩B + S∩I + S∩Sp + B∩I + B∩Sp + I∩Sp + S + B + I + Sp) // Total number of characters is 95.

Appendix B:
Extracted Hash Formats by File Type

1. **RAR Files:**
   - RAR3 (AES-128, PBKDF2-SHA1): $rar3$*0*ad4b28f813c7c2f0b98ec5e9a00c6f2d*5ae6c46db9b3c298a61bf3eef0b73e8d
   - RAR5 (AES-256, PBKDF2-HMAC-SHA256, 32768 iterations):$rar5$16$2b1d2b9e7c84c219a5b08f7c321e84a3$15$00453a9dcabe4a5f0f0a7e31e27b7e5f$8$00000000$32768$16$6c77e2ab21340e6ad7bb9a7c
2. **ZIP Files**
   - ZIPCrypto (legacy, weak):$pkzip$1*2*2*0*8*24*ab12cd34ef56gh78*88*ffffabcd1234*...
   - AES-256 (modern, PBKDF2-SHA1): $zip2$*0*3*256*1000*16*4a8b30f1a3c8720fbc213c24dfae04d5*32*29a0f0b7f33a9a728d0b4dbf9d8d2b05...
3. **PDF Files**
   - PDF 1.1–1.3 (RC4-40-bit, Hashcat mode 10400): pdf1*2*40*1*0*16*51726437280452826511473255744374*32*9b09be05c226214fa1178342673d86f273602b95104f2384b6c9b709b2cbc058*32*0000000000000000000000000000000000000000000000000000000000000000
   - PDF 1.4–1.6 (RC4/AES-128, mode 10500): pdf2*3*128*1028*1*16*da42ee15d4b3e08fe5b9ecea0e02ad0f*32*c9b59d72c7c670c42eeb4fca1d2ca150000000000000000000000000000000000*32*c4ff3e868dc87604626c2b8c259297a14d58c6309c70b00afdfb1fbba10ee571
   - PDF 1.7 Level 3 (AES-256, mode 10600): pdf5*5*256*1028*1*16*2058381440218422686648533275431 5*127*f95d927a94829db8e2fbfbc9726ebe0a391b22a084ccc2882eb107a74f788481…
   - PDF 1.7 Level 8 / PDF 2.0 (AES-256 SHA-384/512, mode 10700): pdf5*6*256*4*1*16*381692e488413f5502fa7314a78c25db*48*e5bf81a2a23c88f3dccb44bc7da68bb5606b653b733bcf9adaa5eb2c8ccf53abba66539044eb1957eda68469b1d0b9b5*48*b222df06deb308bf919d13447e688775fdcab972faed2c866dc023a126cb4cd4bbffab3683ecde243cf8d88967184680

AUTHORS PROFILE

Dharavath Narendar is presently working as a Research Associate at CRRAO AIMSCS, Hyderabad. He did his M. Tech. in Computer Science & Engineering (Information Security) at NITK Surathkal in 2015. He is pursuing his Ph.D (Computer Science & Engineering Dept.) from Acharya Nagarjuna University, Guntur. His areas of interest include password recovery, Cryptography Algorithms, and Network Security—research experience of more than 9 years and teaching experience, more than 3 years.

Padmavathi is working as an assistant professor in CRRAO AIMSCS, Hyderabad. She received a Gold medal in M.Sc. (Maths) from Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. Awarded Ph.D. in Mathematics from JNTUH University, Hyderabad. She has published research papers in reputed international and national journals and conferences, including IEEE, and they're also available online. She holds one patent publication derived from her research. She has 20 years of combined experience in teaching & Research. Her main research interests include Cryptology, Machine learning, Modeling, and Analysis.

Gangadhara Rao Kancherla is a Professor at Acharya Nagarjuna University, Guntur, India. He has published research papers in reputed international and national journals and conferences, including IEEE, and they're also available online. He has 30 years of combined experience in teaching & Research. His areas of interest in Automation, Cloud Computing, Computer Networking, and Information and Communication Technology.

Venu Nalla is presently working as a research associate in CRRAO AIMSCS, Hyderabad. He did his M.Tech (VLSI & Computer Engineering) from IIIT Hyderabad. He did his Ph.D. (Computer Science & Engineering Dept.) from Acharya Nagarjuna University, Guntur. He is also a programmer with proficiency in C & Python. His areas of interest are Cryptology, Side Channel Cryptanalysis, and High Performance Computing.

