Optimizing Rainbow Tables for MD5 Hash Algorithm: An In-depth Analysis of Search Time Efficiency and Probability of Decryption

Bharat Sharma

Master of Science Data Science

Vellore Institute of Technology – Chennai, Tamil Nadu, India

Abstract

Today everything is done through internet, Data and information is travelling through networks, which are very easy to steal. Attackers perform various attacks just to disturb the flow of data, they try to steal, corrupt, edit, replace or add noise in the information flow. This can be done during the transmission of information through networks or with the unauthorized privileges. Hashing plays a major role in preventing the data security. MD5 and SHA-1 are the most popular algorithms used in encryption. These are very difficult to decrypt, but brute force attack can be utilized here. It is mostly time Consuming to perform a Brute force attack so huge tables of hashes are generated and are stored; these tables are also known as rainbow tables. Encrypted hash will be searched in these tables. Here Python language is used for Generating rainbow tables and libraries like hashlib and time is cast-off. Taking MD5 algorithm, this paper analyzes time and space complexity for different character set with different String length.

Key words: ***Encryption, Cryptography, MD5, Brute force, Rainbow tables***

Introduction

Internet is the most important tool and visible resource used by almost all people around the world. A global network of interconnected devices, seamlessly links together millions of computers, web pages, websites, and servers, creating a vast digital landscape that transcends geographical boundaries. With this remarkable technology, we gain the power to effortlessly transmit a wide array of media, from heartfelt emails to cherished photos, engaging videos, and meaningful messages, allowing us to stay connected with our loved ones regardless of the physical distance that separates us. Everything is done on/by/through internet. Data, which is basically formatted information, is travelling through this network that is the most important treasure for hackers. Hacker's Main Focus Is try to Steal, Corrupt, edit or replace the data. so, Information/Data security is important to organizations in all industries around the world.

Most people would not think twice about an account whose password they have forgotten or they have not bothered to open it in a long time, however the personal information linked to that account including email address is something which can lead to harmful consequences if that account is compromised. Once an account has been compromised, malicious actors can exploit that information to target other users by reusing the password of that compromised account and use it to find another account with the same or similar password to compromise that as well. The hackers may also use it to damage the victim's reputation by sending fake mails to the victim's loved ones, friends as well as colleagues, in a few extreme cases the hackers may even blackmail the victim to do something illegal or shameful. According to the Federal Trade Commission (FTC) identity theft cases have seen an exponential increase in the recent years, making it a serious problem.

Cryptography a technique to maintain Confidentiality, Integrity and Authentication while transferring information. The data is encrypted into hashes of fixed length so that, if it is steeled then also the attacker or unauthorized user can't be able to retrieve information from it. there are many algorithms for encryption of string like SHA-1, SHA256, etc. but most popular one is MD5(message Digest 5) and SHA1which are used in many Mobile based Application to encrypt API's.

Review of Literature

Najib et al. (2019, July) highlighted the use of hashing , its importance, uses and ways with different types of hashing algorithm, most popular were MD5 i.e., Message Digest 5 and SHA1 i.e., Secure Hashing Algorithm-1 which are used in many mobile based Application to Encrypt API's analysis of these 2 algorithm are done for both encryption and decryption and results shows that SHA1 is better to applied in the securing RESTAPI as compared to brute force attack results how that the MD5 because brute force attack results shows that MD5 algorithm is decrypting faster when captured to the SHA1 algorithm, while in case of encryption , they don't have that enough difference.

Al-Hammadi et al. (2019) described the time complexity of different hashing algorithms like MD5 and SHA1 can be reduced, while maintaining integrity and restriction of unauthorised data modification, also proves that the same result is obtained by the original function. This paper reduces the time complexity for a single starting step for MD5 and SHA1 and found that it can be reduced up to 50-70% for initial only.

Amoordon et al. (2022) described that the smart era with everything smart require a major component for performing communication between each other and the user through signals sent over the networks and due to which they are more vulnerable to some specific attacks like jamming, DE authentication and Fake Access point which can be easily penetrated against IEEE 802.11 as compared to device which uses wired connection for communications. This research helps in detection of these attack by analysing the packets send over the network with a specific frame length, size, gap and droppage in signal strength between them and compared them to the original packets send, with the help of machine learning models and result shows that Random Forest technique and K-nearest neighbour predictor have best exactness of 96% for detection of fraud AP and 81% for DOS attacks and Intentional Jamming.

Musthyala et al. (2021), described the wireless network types and standards like IEEE 802.11 WLAN standard, etc which are commonly used as of affordability, handiness and mobility. Various Protocols for security purposes are introduced like WEP, WPA, WPA2, WPA3, which are having their own advantages and disadvantages according to the encryption techniques. This paper also discussed about many hacking tools, phishing attacks and platforms like air crack-ng, airgeddon, wifite, wifite2, kali-Linux, black-arc Linux, etc and provides a script for phishing the saved Wi-Fi passwords with the help of Python also suggested some measures how to defend from these kinds of attacks.

Ferro et al. (2005) highlighted the 2 main standard Bluetooth and IEEE 802.11 for wireless communication only for a short range. Other standards are also discovered and the features are

compared in terms of different matrices like capacity, topology used, security, power consumptions, etc. and conclude that both have plenty of room for improvement.

Hager et al. (2003) described the Bluetooth system with the main focus on its known vulnerability and its security features, VERDICT and its key features with its history all points are covered then comparison of Bluetooth with IEEE 802.11 LAN vulnerability and VERDICT are done, results show some randomness are concluded that Bluetooth with IEEE 802.11 LAN shoes some improvement in terms of security features but also have some of the weakness that are present in most of the wireless system today.

Kissi et al. (2020) Described how the use of wireless network for different purposes is increasing day by day, sharing valuable information over this network in open air gives attackers a good choice to sniff and capture data packets, WEP, WPA/WPA2-psk are vulnerable to attacks, cracking of WEP is dependent on the generating of more weak IV's, once enough IV's are generated the key will be successfully cracked, while in case of WPA/WPA2-PSK, a four way handshake is required which can be easily captured with any of the device connected to this network and can be cracked if the Passphrase is prevented in the attacker's wordlist.

Noman et al. (2015) mentioned that wireless networks are more insecure as compared to wired networks because the data is transmitted using radio waves. This paper also described how an attacker performs DE authentication attacks on a network, by using an IJAM tool written in python. It also mentions how to detect these kinds of attacks. Tests are performed to prevent these if possible as they are active attacks, also invent an automated method to detection and prevention from these.

Nguyen et al. (2008) highlighted the fact that in a wireless network if any wireless station needs to disconnect itself from the network it must send 2 frames which are a DE authentication or disassociation frame to the access point which are basically unencrypted. So, the attacker can launch a Denial of Service attack and spoof the important messages and disable the transmission or communication going on between the device and the access point. This paper also suggested a function which can authenticate whether the frames are sent by the device or by an attacker, so as to prevent these kinds of DE authentication attacks.

Mar et al. (2010) realised IDS (Intrusion Detection System) which is built on adaptive neuro-fuzzy inference system to minimise the time taken to detect the DE authentication Denial-of-Service DoS attack on a wireless network. The SNG (Sequence Number Gap) between the packets which are sent and the statistical value of the DE authentication packets received by the AP are used for detection of the DoS attack, and the result of the Intrusion Detection System is set side by side with the algorithm which is used for the detection of non-parametric sequential change point.

Ramachandra et al. (2017) described how Biometric technologies like face recognition, iris, fingerprint, voice ,veins, keystrokes, gait and others are rapidly gaining popularity as it can automatically recognized individual based on their biological characteristics, here mainly focused on face recognition which are now used in international border controls, Forensic labs and even in many E-Gov Applications, but in some cases where bypass of these technologies and experimented or executed illegally where spoofing of data and modelling it in physical

state with the help of plastic surgery and other medical facility. This paper explains some of the ant spoofing techniques which can be effective in many of the cases for detection of these attacks and report the performance of face PAD algorithms but still not 100% Effective.

Lee et al. (2019) describe how Evolution meaning is changing, and lead to a competition of being more developed. Smartness in every device and everything being automated, controlled directly by ICT and others, these devices are monitored 24 hrs and also control their movements and outputs, if necessary, but smart cities are facing increased cyber security attacks. This paper helps to analyse the threats and architecture of devices in smart cities for security updates in future. 5 attacks are possible for different components of smart cities and can be chosen as a reference for possible scenarios.

THAKAR et al. (2020) highlighted how the use of IP Cameras is increasing day by day which results in low manipulation of CCTV cameras. IP cameras can be monitored and manipulated over the network while CCTV can't, as they are connected to monitored system but they are not secure as the company doesn't concern about the security features and its surveillance. surveys and various tests are performed like how attacks can be performed for video surveillance so for analysing the security mechanism and finding any vulnerabilities which can be valuable for future security updates.

Research Objective

This study seeks to provide a thorough evaluation of the use of rainbow tables in collaboration with the MD5 hash method for cryptographic analysis. The main objective here is to evaluate the decryption time needed to restore the original plaintext from MD5 hash values using rainbow tables, for various key space sizes and levels of complexity, to assess the likelihood of successfully decrypting MD5 hashes using rainbow tables while taking into account a variety of variables like table size, reduction functions, and chain length, to examine the time and space difficulty of creating rainbow tables for the MD5 hash method, paying particular attention to elements like table size, chain length, and reduction functions, in order to find the best settings.

By focusing on these goals, this study hopes to promote cryptographic analysis and help in the creation of more secure hashing techniques by offering insightful information about the usefulness and efficiency of rainbow tables in respect to the MD5 hash algorithm.

Research Methodology

User machine provides MD5 HASH String to decrypt with Expected Length and character set used, this hash String is then searched in the rainbow tables of the provided details on the server machine. If the hash provided matches with any of the hash in the rainbow table, then, the corresponding text will be the result of decryption.

For example: if the user machine provided a hash [ef775988943825d2871e1cfa75473ec0] and provided the expected length to be 5 with character set to be numeric {0,9,8,7,6,5,4,3,2,1}.

Then corresponding rainbow table of {0,9,8,7,6,5,4,3,2,1} with text length 5 is generated in which all Possible Combination of 5-Digits numeric String is generated. With total of One hundred thousand unique combinations, these hashes of the unique string is then stored in a file called rainbow tables. And after that the user's provided hash will be searched in the table, if the hash of the string (say PASSWORD) provided by the user machine is actual of 5-digit length and can be formed with the provided character set, it will be successfully retrieved from this method.

Generating Rainbow tables

Python Library hashlib has Predefined function md5(INPUTSTRING) to generate MD5 hash of INPUTSTRING variable, which is faster, the generated output is stored.

Also, md5("02") is not equal to md5("2") as length varies therefore the hash will be differ from each other.

Let's Say

the Time taken by md5 functions of hashlib library to generate hash be "n"

length of password be "l"

and the length of character set used to be "m"

then, the time complexity of algorithm which is used to generate the Rainbow table would be given by

Time complexity: - O (**(n*m) ^l)**

Space complexity would be

Space Complexity: - O (**1**)

Length	Character Set	Possible Combination	Time(ms)	Size(KB)
4	{0,9,8,7,6,5,4,3,2,1}	10000	41.84770584106445	333
5	{0,9,8,7,6,5,4,3,2,1}	100000	396.9910144805908	3321
6	{0,9,8,7,6,5,4,3,2,1}	1000000	3656.588077545166	33204
7	{0,9,8,7,6,5,4,3,2,1}	10000000	32658.249616622925	332032
8	{0,9,8,7,6,5,4,3,2,1}	100000000	317350.2595424652	3320313

Table 1.1 Time and Space for Different rainbow tables

Let's Say

the number of hash present in rainbow table be "n"

time Complexity for best case: - Hash matches at 1st entry

Time complexity: - O (**1**)

time Complexity for worst case: - Hash matches at last entry

Time Complexity: - O (**n**)

Length	First Entry of Rainbow table	Time taken to Decrypt (min time ms)	Last Entry of Rainbow table	Time taken to Decrypt (max time ms)
4	0000	0.0	9999	10.40506362915039
5	00000	0.9925365447998047	99999	68.55559349060059
6	000000	1.0001659393310547	999999	670.4027652740479
7	0000000	1.1429786682128906	9999999	6029.212474822998
8	00000000	1.789093017578125	99999999	56355.24582862854

Table 1.2 Time required searching 1st and last entry in Different rainbow tables

Result:

The relationship between the length of plain text (x) and the corresponding space required to store the rainbow table (y) shows an exponential growth pattern. As the length of plain text increases, the space required to store the rainbow table also increases significantly. Additionally, the time required to generate the rainbow table (z) exhibits a similar exponential growth pattern to the length of the plaintext. As the length of plain text increases, the time required to create the rainbow table increases significantly. These results highlight the important consideration of computational resources and time investment while implementing a rainbow table for hash decryption, especially in scenarios with longer plaintexts. This information will help make informed decisions about the feasibility and effectiveness of using Rainbow tables with the MD5 hash algorithm.

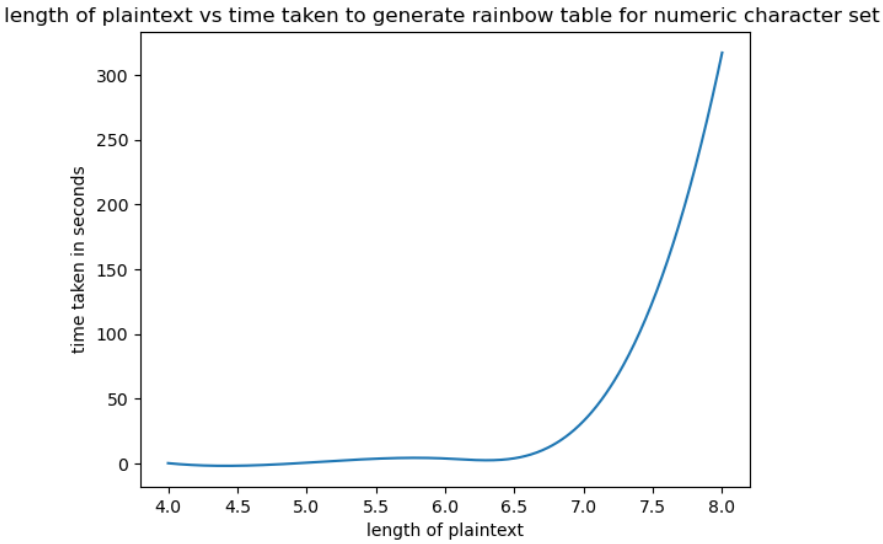

Figure 1.1 relationship between length of plain text vs time taken to generate the rainbow table in seconds

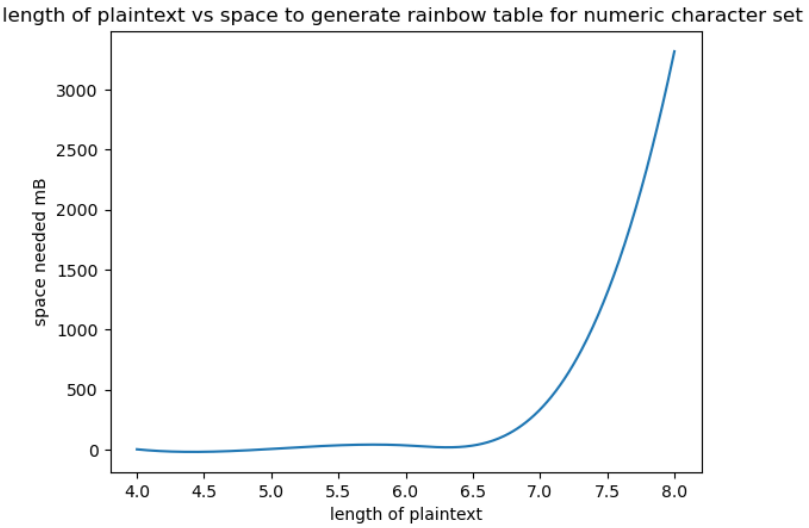

Figure 1.2 relationship between length of plain text vs space needed to store the rainbow table in megabytes

Length(x)	Time taken to Decrypt (min time ms)	Time taken to Decrypt (max time ms)
4	0.0	10.40506362915039
5	0.9925365447998047	68.55559349060059
6	1.0001659393310547	670.4027652740479
7	1.1429786682128906	6029.212474822998
8	1.789093017578125	56355.24582862854

These results represent the minimum and maximum times taken (in milliseconds) to search for the hash in the rainbow table for different lengths of plaintext (x). The search times demonstrate the efficiency and scalability of the rainbow table approach with respect to the MD5 hash algorithm.

Conclusion

By observing the tables, it is known that

- the Time taken for Generating and searching in rainbow table depends upon the size of string and as well as the length of character set used.

- The Probability of Finding the hash Would depends upon the Accuracy of Details Provided for Generating the Corresponding Rainbow table.
- Encryption and searching speed also depend upon the hardware Configuration and as well as OS used, for this (Microsoft Window 11 Home Single language, 8GB DDR4 Ram, SSD, AMD Ryzen 3 5300u with Radeon Graphics, 2600Mhz, 4Cores(s), 8 Logical Processor(s)) is used.

languages like Mojo which claims that it is 35000 times faster than Python, as it is using swift programming language development so it can also be analyzed in future. Rainbow table has a lot of room for growth and is evolving; numerous scholars are still engaged in this field.

References:

[1] Najib, A. F., Rachmawanto, E. H., Sari, C. A., Sarker, K., & Rijati, N. (2019, July). A comparative study MD5 and SHA1 algorithms to encrypt REST API authentication on mobile-based application. In *2019 International Conference on Information and Communications Technology (ICOIACT)* (pp. 206-211). IEEE.

[2] Al-Hammadi, Y. A., & Fadl, M. F. I. (2019). Reducing hash function complexity: MD5 and SHA-1 as Examples. *IJ Mathematical Sciences and Computing, 1*, *5*(1), 1-17.

[3] Amoordon, A., Deniau, V., Fleury, A., & Gransart, C. (2022). A single supervised learning model to detect fake access points, frequency sweeping jamming and DE authentication attacks in IEEE 802.11 networks. *Machine Learning with Applications*, *10*, 100389.

[4] Musthyala, H., & Reddy, P. N. (2021, May). Hacking wireless network credentials by performing phishing attack using Python Scripting. In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 248-253). IEEE.

[5] Ferro, E., & Potorti, F. (2005). Bluetooth and Wi-Fi wireless protocols: a survey and a comparison. *IEEE Wireless Communications*, *12*(1), 12-26.

[6] Hager, C. T., & MidKiff, S. F. (2003, March). An analysis of Bluetooth security vulnerabilities. In *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.* (Vol. 3, pp. 1825-1831). IEEE.

[7] Kissi, M. K., & Asante, M. (2020). Penetration testing of IEEE 802.11 encryption protocols using Kali Linux hacking tools. *International Journal of Computer Applications*, *975*, 8887.

[8] Noman, H. A., Abdullah, S. M., & Mohammed, H. I. (2015). An automated approach to detect DE authentication and disassociation dos attacks on wireless 802.11 networks. *International Journal of Computer Science Issues (IJCSI)*, *12*(4), 107.

[9] Nguyen, T. D., Nguyen, D. H., Tran, B. N., Vu, H., & Mittal, N. (2008, August). A lightweight solution for defending against DE authentication/disassociation attacks on 802.11 networks. In *2008 Proceedings of 17th International Conference on Computer Communications and Networks* (pp. 1-6). IEEE.

[10] Mar, J., Yeh, Y. C., & Hsiao, I. F. (2010, October). An ANFIS-IDS against DE authentication DOS attacks for a WLAN. In *2010 International Symposium On Information Theory & Its Applications* (pp. 548-553). IEEE.

[11] Ramachandra, R., & Busch, C. (2017). Presentation attack detection methods for face recognition systems: A comprehensive survey. *ACM Computing Surveys (CSUR)*, *50*(1), 1-37.

[12] Lee, J., Kim, J., & Seo, J. (2019, January). Cyber Attack scenarios on smart city and their ripple effects. In *2019 international conference on platform technology and service (PlatCon)* (pp. 1-5). IEEE.

[13] THAKAR, D. (2020). Survey on IP Camera Hacking and Mitigation. *Multidisciplinary International Research Journal of Gujarat Technological University*, *2*(2), 28-33.

[14] https://www.mail.com/blog/posts/why-do-hackers-steal-data/178/