#### Project report on

**HASHED PASSWORD CRACKER**

**A Dissertation submitted in partial fulfillment of the Academic requirements for the award of the degree of**

**Bachelor of Technology**

## In

**Computer Science & Engineering (Cyber Security)**

**Submitted by**

G.Nithin kumar (22H51A6220) K.Shivanand (22H51A6226) M.Naresh (22H51A6234)

**Under the esteemed Guidance of**

**Dr.R.Venkateswara Reddy (Associate Professor and HOD,CSC)**



**Department of Cyber Security**

#### CMR COLLEGE OF ENGINEERING & TECHNOLOGY

**(Autonomous)**

**(NAAC Accredited with ‘A+’ Grade & NBA Accredited) (Approved by AICTE, Permanently Affiliated to JNTU Hyderabad)**

**KANDLAKOYA, MEDCHAL ROAD, HYDERABAD-501401**

#### CMR COLLEGE OF ENGINEERING & TECHNOLOGY

**(Autonomous)**

**(NAAC Accredited with ‘A+’ Grade & NBA Accredited) (Approved by AICTE, Permanently Affiliated to JNTU Hyderabad)**

**KANDLAKOYA, MEDCHAL ROAD, HYDERABAD-501401**

**DEPARTMENT OF CYBER SECURITY**



#### CERTIFICATE

This is to certify that the Mini Project -1 report entitled “**Hashed password cracker**” being submitted by **K.Shivanand (22H51A6226), G.Nithin kumar (22H51A6220), M.Naresh (22H51A6234)** in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering (Cyber Security)** is a record of bonafide work carried out his/her under my guidance and supervision.The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree.

M.L.Saranya Dr. R. Venkateswara Reddy

Assistant Professor Associate Professor & HOD

Dept. of CSC Dept. of CSC

#### ACKNOWLEDGEMENT

With great pleasure I want to take this opportunity to express my heartfelt gratitude to all the people

`who helped in making this project a grand success.

I am grateful to **M.L.Saranya**, Assistant Professor, Dept. of Computer Science and Engineering for her valuable suggestions and guidance during the execution of this project.

I would like to thank **Dr. R. Venkateswara Reddy**, Head of the Department of Computer Science and Engineering, for his moral support throughout the period of my study in CMRCET.

I am highly indebted to **Major Dr. V.A. NARAYANA**, Principal CMRCET, for giving permission to carry out this project in a successful and fruitful way.

I would like to thank the Teaching & Non- teaching staff of the Department of Computer Science and Engineering for their co-operation.

Finally, I express my sincere thanks to **Mr. CH. GOPAL REDDY**, Secretary, CMR Group of Institutions, for his continuous care. I sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project work.

**G.Nithin kumar (22H51A6220)**

**K.Shivanand**

**(22H51A6226)**

**M.Naresh**

**(22H51A6234)**

#### ABSTRACT

* The hashed password cracker tool will be designed to crack hashed passwords often used for secure storage and authentication. It will utilize various techniques such as brute force, dictionary attacks, and rainbow table lookups to attempt to reverse-engineer the original password from its hash value. The cracker will provide a user-friendly interface for inputting hashed passwords and will employ advanced algorithms and optimization techniques to increase efficiency and speed. The project aims to assist users in recovering forgotten passwords or testing the strength of their hashed password implementations.

**Table Of Content**

|  |  |  |
| --- | --- | --- |
| **CHAPTERS** | **DESCRIPTION** | **PAGE NUMBERS** |
| 1 | INTRODUCTION | 2 |
| 1.1 | AIM | 3 |
| 1.2 | SCOPE | 4 |
| 2 | LITERATURE REVIEW | 6 |
| 3 | EXISTING SOLUTIONS | 8 |
| 4 | PROPOSED SYSTEM | 11 |
| 4.1 | REQUIREMENT ANALYSIS | 12 |
| 4.1.1 | HARDWARE REQUIREMENTS | 12 |
| 4.1.2 | SOFTWARE REQUIREMENTS | 12 |
| 4.2 | MERITS AND DEMERITS | 13 |
| 5 | DESIGN DESCRIPTION | 15 |
| 5.1 | CONCEPTUAL DESIGN | 15 |
| 6 | IMPLEMENTATION AND DISCUSSION | 16 |
| 6.1 | IMPLEMENTATION | 17 |
| 7 | RESULT | 20 |
| 8 | CONCLUSION AND FUTURE ENHANCEMENT | 21 |
| 8.1 | CONCLUSION | 21 |
| 8.2 | ENHANCEMENT | 21 |
| 8.3 | REFERENCES | 24 |

# CHAPTER 1

1

#### INTRODUCTION

* When a password is stored, it is often transformed using a hashing algorithm. Hashing is a one-way cryptographic function that takes an input (or message) and returns a fixed-size string of characters, which appears random. Common hashing algorithms include MD5, SHA-1, and SHA-256. For example, hashing the password "password123" with SHA-256 would yield a hash.
* Hashing is used to protect passwords because even if an attacker gains access to the hashed passwords, they cannot easily reverse the hash to discover the original passwords. Hashes are also useful for ensuring data integrity and verifying data authenticity.
* A hashed password cracker is a tool or technique used to recover passwords from their hashed forms. It aims to find the original password that corresponds to a given hash. Cracking hashed passwords typically involves trying various methods to guess the password, leveraging weaknesses in the hashing process or the user's choice of passwords.

#### AIM

* + The project involves addressing several crucial aspects. Firstly, the project aims to delve into the intricacies of cryptographic hash functions and their role in securing passwords. Understanding these functions—such as MD5, SHA-1, and SHA-256—is fundamental as they form the basis for generating unique hash values from plaintext passwords. The primary objective of the project lies in developing and evaluating techniques for cracking hashed passwords. This includes implementing various methods like brute force attacks, dictionary attacks, and rainbow table lookups to assess their effectiveness in recovering passwords from their hashed forms. Throughout the implementation phase, rigorous testing and validation protocols are applied to verify the accuracy and efficiency of the password cracker. The report further analyzes the performance metrics of these techniques, comparing their success rates and resource consumption. It also considers the broader implications for cybersecurity, highlighting vulnerabilities in password storage systems and recommending strategies to enhance overall security practices. Ethical considerations, such as privacy and legal implications associated with password cracking, are carefully discussed to ensure responsible research conduct. Ultimately, the project aims to contribute insights into password security mechanisms, offering practical recommendations for mitigating risks associated with hashed vulnerabilities.

#### SCOPE

The scope of a hashed password cracker project encompasses several key dimensions that define its boundaries, objectives, and areas of focus. Here’s a detailed outline of the scope:

1. **Technologic al Focus:**

* The project focuses on cryptographic hash functions used in password storage, such as MD5, SHA-1, SHA-256, etc. It involves understanding these functions, their properties, and their vulnerabilities.

1. **Password Cracking Techniques:**

* The scope includes exploring and implementing various password cracking techniques. This includes brute force attacks, dictionary attacks, rainbow table lookups, and possibly hybrid approaches combining these techniques.

1. **Software Development:**

* It encompasses the design, development, and implementation of a password cracker software or tool. This involves coding algorithms, designing user interfaces (if applicable), and integrating necessary libraries or frameworks.

1. **Testing and Validation:**

* The project involves rigorous testing to validate the effectiveness and efficiency of the password cracker against different types of hashed passwords. This includes testing with various hash functions and evaluating performance metrics like speed and success rates.

1. **Performance Evaluation:**

* The scope includes evaluating the performance of the implemented techniques. This involves measuring factors such as cracking speed, success rates, and resource consumption (CPU, memory usage).

# CHAPTER 2

#### LITERATURE REVIEW

Cryptographic hash functions play a pivotal role in modern cybersecurity by transforming passwords into fixed-length hashes, which are stored in databases to prevent plaintext exposure. The selection of hash algorithms, such as MD5, SHA-1, and SHA-256, significantly influences the security of password storage systems. MD5 and SHA-1, once prevalent, have been shown vulnerable to collision attacks, prompting a shift towards more secure alternatives like SHA-256 and SHA-3. Research has extensively explored the properties of these hash functions, emphasizing their one-wayness and resistance to pre-image attacks. Password cracking techniques leverage weaknesses in hash functions to recover plaintext passwords from their hashed counterparts. Brute force attacks systematically test all possible password combinations, while dictionary attacks exploit common password choices. Rainbow table attacks use precomputed tables to expedite the reverse hashing process. Hybrid approaches combine these methods, optimizing for efficiency and coverage of password spaces. Tools such as Hashcat and John the Ripper exemplify advancements in password cracking software, employing parallel processing to enhance speed and scalability. Ethical considerations surrounding password cracking include privacy infringements and legal implications, necessitating responsible research conduct and adherence to ethical guidelines. By understanding these facets, researchers aim to bolster password security measures and mitigate vulnerabilities in digital systems.

# CHAPTER 3

#### EXISTING SOLUTION

1. **Dictionary Attacks**

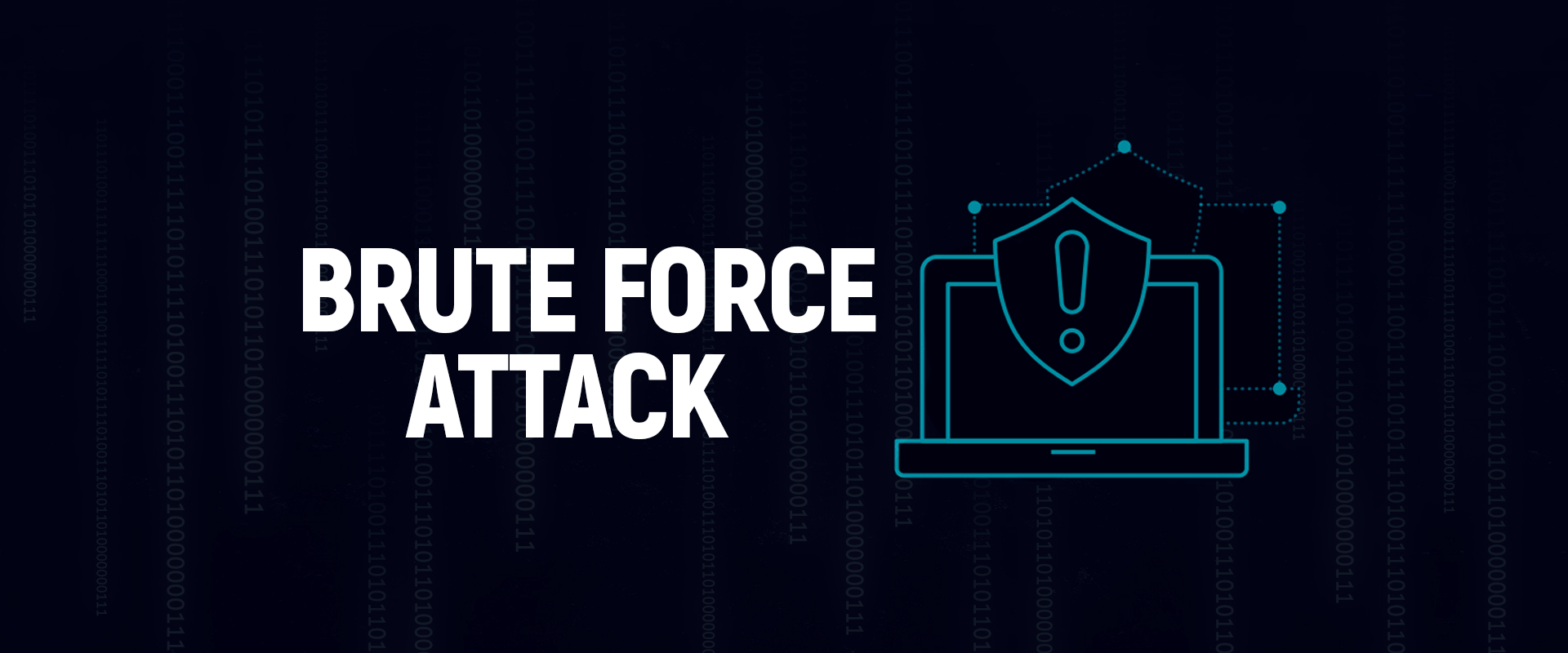
A dictionary attack uses a pre-arranged list of potential passwords, known as a dictionary, to crack the hashed password. This list contains common words and passwords that users might use.



**Fig 1:** Dictionary Attack

**2.Brute Force Attacks:**

A brute force attack attempts all possible combinations of characters until the correct password is found. This method is very time-consuming and computationally intensive.



**Fig 2:** Brute Froce Attaack

**3.Rainbow Tables**

Rainbow tables are precomputed tables for reversing cryptographic hash functions, usually for cracking password hashes. These tables provide a time-memory tradeoff, using more memory to reduce the timerequired to crack a hash.



**Fig 3:** Rainbow Table Attack

**4. John the Ripper:**

John the Ripper is a powerful and popular password cracking tool used by security professionals and researchers to test the strength of passwords and identify weak passwords in systems.



**Fig 4:**John the ripper tool

# CHAPTER 4

#### PROPOSED SYSTEM

Employing specialized password cracking tools such as Hashcat augmented with advanced techniques like GPU acceleration, rule-based cracking, and precomputed hash tables, provides an efficient approach to cracking hashed passwords. However, it's imperative to emphasize ethical usage and ensure compliance with legal regulations to avoid unauthorized access and maintain data security. To crack a hashed password using Hashcat, start by installing the tool on your system. Ensure you have a wordlist, such as rockyou.txt, and a file containing the hash you want to crack, e.g., hashes.txt. Identify the hash type (e.g., MD5, SHA1) and use the appropriate hash mode (0 for MD5, 100 for SHA1). Run Hashcat with a command like hashcat -m 0 -a 0 -o cracked.txt hashes.txt /path/to/wordlist.txt, where -m specifies the hash type, -a 0 indicates a dictionary attack, and -o defines the output file for cracked passwords. If desired, you can enhance the attack using rule files to transform the wordlist entries. Once the process is complete, check the cracked.txt file for the cracked passwords. This approach leverages Hashcat's powerful cracking capabilities to attempt various password combinations against the given hash.

#### REQUIREMENT ANALYSIS

###### Software Requirements

* + - * Windows 7 or later, Linux, or macOS
      * Eclipse
      * Java jdk

###### Hardware Requirements

* + - * System 32 or 64 bit with 4 GB or 8 GB RAM





##### MERITS AND DEMERITS

**MERITS**

* **Vulnerability Assessment**: Helps identify weaknesses in password storage systems by demonstrating potential for password recovery from hashes.
* **Improves Security Practices**: Encourages the development of stronger password security protocols and hashing algorithms.
* **Educational Tool**: Provides insights into password security risks, aiding in educating security professionals and system administrators.
* **Supports Security Audits**: Facilitates regular audits and assessments of password security measures to ensure robustness.
* **Promotes Research**: Stimulates research and development efforts towards enhancing password security and hash function resilience.

##### DEMERITS :

 **Ethical Concerns**: Raises privacy and ethical concerns due to potential for unauthorized access andmisuse.

 **Resource Intensive**: Effectiveness depends on computational resources available and the strength of hash functions used.

 **Dependence on Hash Strength**: Effectiveness varies based on the strength of the hash algorithm; weaker hashes are more susceptible.

 **Legal Implications**: In some jurisdictions, unauthorized use of hashed password crackers could violate laws governing data protection and cybersecurity.

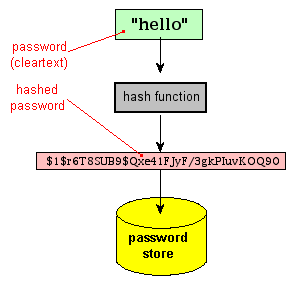
# CHAPTER 5

14

#### DESIGN DESCRIPTION

##### 5.1 CONCEPTUAL DESIGN

The diagram shows the steps involved in dictionary attack and using Hashcat Tool



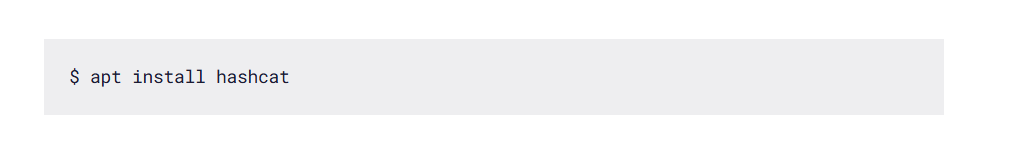
**Fig 6:** Steps for using hashcat tool

# CHAPTER 6

* 1. **IMPLEMENTATION AND DISCUSSION**

**3.1 IMPLEMENTATION**

## Install Hashcat:

Hashcat comes pre-installed in Kali and Parrot OS. To install it in Ubuntu / Debian-based systems, use the following command: 

Once the installation is done, we can check Hashcat’s help menu using this command:

### COMMAND: $ hashcat-h



**Fig 6:** Hashcat help menu

## World list:

In addition to Hashcat, we will also need a wordlist. A word list is a list of commonly used terms. This can be a password wordlist, username wordlist, subdomain wordlist, and so on.A popular password wordlist is rockyou.txt. It contains a list of commonly used passwords and is popular among pen testers. You can find the Rockyou wordlist under /usr/share/wordlists in Kali Linux.

### COMMAND: crunch 3 5 pass123 -0 wordlist

### 

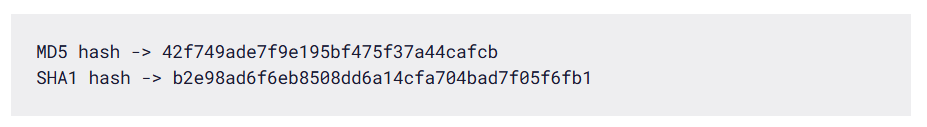
### 

**Fig 7:**creating wordlist

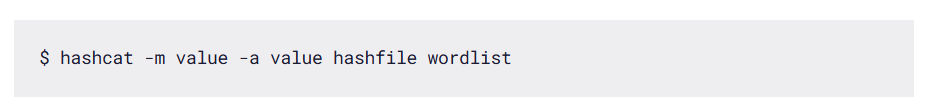
**Working with MD5 and SHA1:**

*Let’s create two hashes: A MD5 hash and a SHA1 hash for the string “Password123”.*

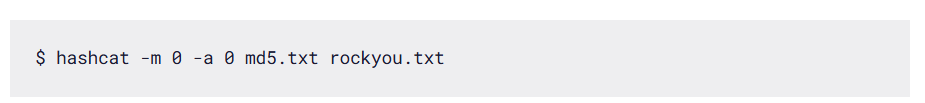
Here are the generated hashes for the input strings.

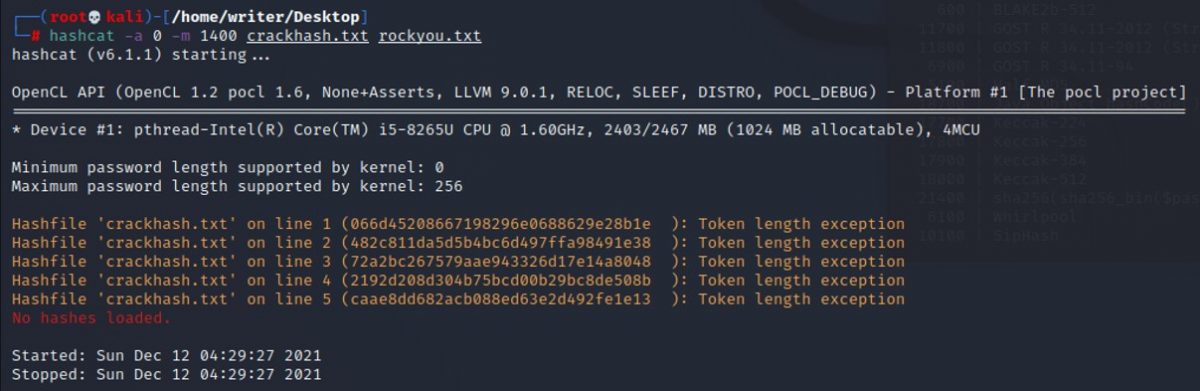


We can store these hashes under the names md5.txt and sha1.txt to use them when working with Hashcat. To crack a password using Hashcat, here is the general syntax.



For the attack mode, we will be using the dictionary mode (0) using the flag -a. Here is the full command:





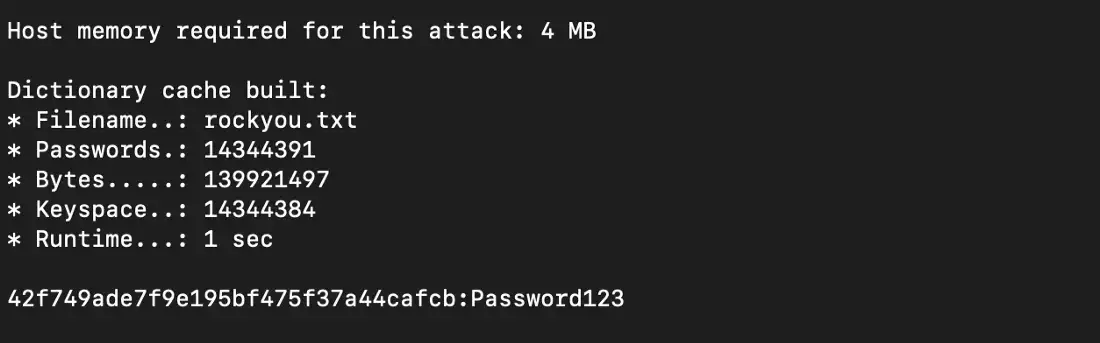
**Fig 8**

**COMMAND: *hash-identifier***



**Fig 9**

**COMMAND: *-m 0 MD5hash.txt wordlist.txt –show***

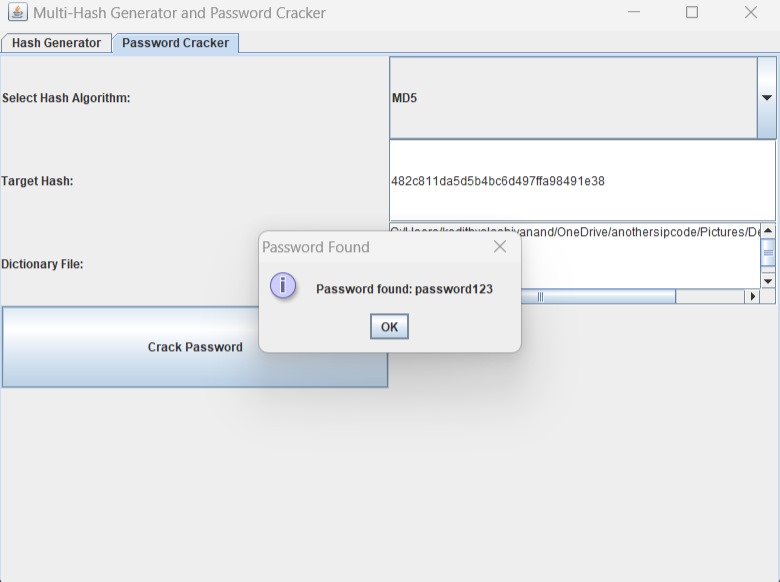


**Fig 10**

# CHAPTER 7

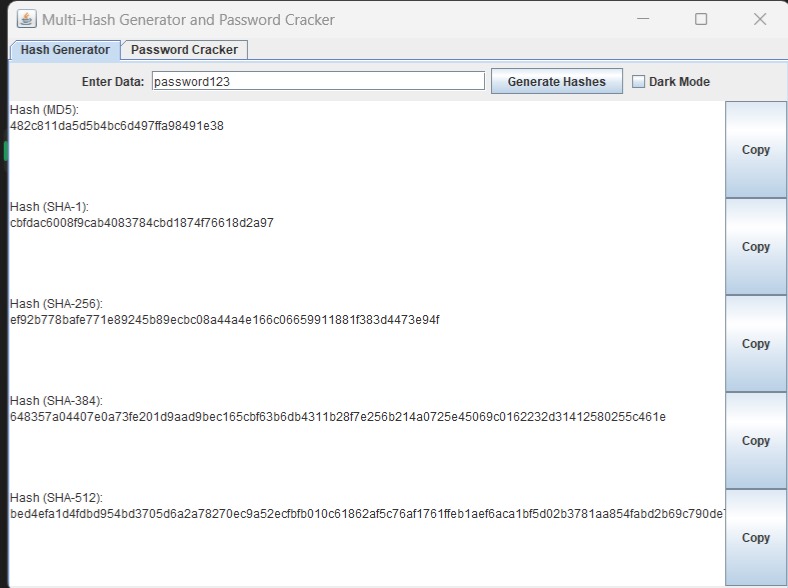
#### RESULT

We have successfully crack the password using algorithms MD5 AND SHA1



**Fig 11**

## Website interaface:



**Fig**

# CHAPTER 8

#### CONCLUSION AND FUTURE ENHANCEMEMT

##### CONCLUSION

* + - In conclusion, The project to create a website for demonstrating hashed password cracking using MD5 and SHA-1 algorithms aims to educate users about the vulnerabilities associated with these outdated hashing techniques and the importance of robust password security practices
    - This project serves as a foundation for different hashing algorithms, and the principles behind password cracking.
    - The full report is available on our website for reference.
    - Thanks to the team members and stakeholders for their dedicated efforts.

##### 8.2 FUTURE ENHANCEMENTS

* + - Expand educational resources, including tutorials, workshops, and documentation, to help users understand the ethical implications and best practices for using Hashcat responsibly in cybersecurity contexts.