UNIVERSITY OF PETROLEUM AND ENERGY STUDIES, DEHRADUN

Zip Guard

"Shield your data with Encryption"

Synopsis Report of the (Minor Project - 1) in Semester V

Prepared by Team No: 24

S. No	Students Name	Roll Number	Sap Id
1.	Akshat Pandey	R2142220306	500101788
2.	Madhav Madan	R214220265	500105699
3.	Dhruv Joshi	R2142220483	500106021
4.	Kavya Singh	R2142220538	500106891

BACHELOR OF TECHNOLOGY, COMPUTER SCIENCE ENGINEERING

With specialization in DevOps

Under the guidance of

Dr. Arjun Arora

INDEX

Zip Guard

“Shield your data with Encryption”

Introduction

The protection and management of data are critical in the digital age. "Zip Guard" is a tool with strong file compression and encryption features that is made to meet these objectives. This project allows users to secure their data via encryption and decryption, as well as zip and unzip files using Huffman coding. With the added convenience of auto-generated keys, users have a choice between two robust encryption algorithms: AES and RSA. Zip Guard’s main goal is to provide a straightforward but efficient method of safeguarding confidential information.

Background Information

Encryption and file compression are two essential methods for today's data management. A popular technique for efficient data compression that minimizes file size without sacrificing integrity is huffman coding. However, encryption makes sure that the data is compressed and protected from unwanted access. Two of the most reliable encryption algorithms out there are RSA and AES. AES is a symmetric encryption technique that uses a single key for both encryption and decryption, whereas RSA is an asymmetric encryption system that depends on two keys: a public key and a private key. These technologies are combined by "Zip Guard" to offer a complete data protection solution. Zip Guard not only protects data but also speeds up transfer times by compressing files before encryption. Its use of AES and Huffman coding allows for maximal compression with the least amount of risk to sensitive data. With the frequency of data breaches in the modern era, Zip Guard is a vital tool for safeguarding crucial information. This solution ensures efficiency and security for corporate data as well as personal files. Users may safely share and save data using Zip Guard without sacrificing function or privacy.

Motivation

This project is created in response to the growing demand for effective and safe data management solutions. Unauthorized access and data breaches are growing concerns along with the amount of digital information. For average users, traditional file compression and encryption techniques may be too complicated or challenging. By offering a straightforward interface that combines robust encryption settings with file compression, "Zip Guard" aims to make this process easier. The project satisfies the security requirements of a broad spectrum of users with its automatic key generation and customizable encryption options.

Additionally, the development of "Zip Guard" serves as an opportunity for our team to improve and test our programming skills. Working with advanced concepts like Huffman coding, RSA, and AES encryption in Core Java not only challenges our technical abilities but also helps us deepen our understanding of data security and software development practices. This project is a practical application of our learning, allowing us to enhance our coding proficiency while contributing to a real-world solution. The project builds collaboration and teamwork as we work together to overcome challenging issues with coding. Additionally, it helps in our major develop in improving speed and debugging for larger datasets. Our goal is to integrate user feedback into "Zip Guard" and make it able to satisfy increasing security requirements so that it will continue to be a reliable and useful tool in the future. We're sure that this experience will help us in the future when we work on projects that call for both software design and security skills.

Contribution

"Zip Guard" contributes to society by enhancing the security and privacy of digital data. In an era where data breaches are common, this tool offers a straightforward way to protect sensitive information. Whether for personal use or within organizations, "Zip Guard" empowers users to safeguard their files against unauthorized access. The project's focus on ease of use makes advanced encryption accessible to a broader audience, thereby promoting better data protection practices across society.

Related Work

Several projects and tools have addressed file compression and encryption, but few have combined these functionalities in a single package. Existing solutions like WinRAR and 7-Zip offer compression and basic encryption but lack the choice between advanced algorithms like RSA and AES. Academic research has explored the implementation of Huffman coding, RSA, and AES individually, but "Zip Guard" innovates by integrating these techniques into a cohesive tool. This project builds on previous work by offering a more versatile and user-friendly solution. "Zip Guard" not only offers more robust encryption solutions but also simplifies the user experience by automating several difficult tasks, including algorithm selection and key management. Customers may now enjoy speed, security, and ease of use all in one package, eliminating the need for them to pick between the three. Users can customize their security settings according to the sensitivity of their data thanks to the freedom to choose from a variety of encryption techniques.

Works have been done until now

For file compression and encryption, numerous projects and tools have been created, all of which include advanced data management solutions. Although well-known programs like WinRAR and 7-Zip have simple encryption and compression capabilities, they are frequently rigid and difficult to use. These solutions often only offer one encryption type and don't have sophisticated features like RSA, which is necessary for asymmetric encryption. Furthermore, handling and storing keys by hand can be challenging when using these tools for key management.

"Zip Guard" overcomes these drawbacks in comparison to these current methods by including Huffman coding for compression and providing an option between RSA and AES encryption. Zip Guard's automatic key creation capability streamlines the procedure and increases accessibility for people without technical knowledge. "Zip Guard" is a more complete and user-friendly solution for safe data management than earlier initiatives because of its versatility, advanced security options, and simplicity of use.

Problem statement

In today's digital world, secure and effective data management is essential. Many consumers find that the security and usability of existing solutions are compromised, leaving their sensitive data poorly protected. With an emphasis on user choice and convenience, "Zip Guard" aims to address this issue by offering an efficient solution that combines file compression with encryption.

Objective

- Develop a tool that combines efficient file compression using Huffman coding with secure encryption options.
- Provide users with a choice between RSA and AES encryption algorithms to meet varying security needs.
- Automate the key generation process to simplify encryption for users without technical expertise.
- Ensure the tool is user-friendly and effective in protecting sensitive digital data.

Problem in model

Current models for data compression and encryption often fall short in terms of user accessibility and flexibility. Many tools require users to have a deep understanding of encryption techniques, making them inaccessible to non-technical users. Additionally, the lack of choice in encryption algorithms can be a limitation for users with specific security needs. "Zip Guard" addresses these issues by offering an easy-to-use interface with customizable encryption options.

Example illustration

Consider a scenario where a user needs to send sensitive financial documents over the internet. Using "Zip Guard," they can compress the files to reduce their size and then choose either RSA or AES encryption to secure them. The tool automatically generates the necessary keys, ensuring that only the intended recipient can decrypt and access the files. This process not only protects the data but also ensures efficient file transfer.

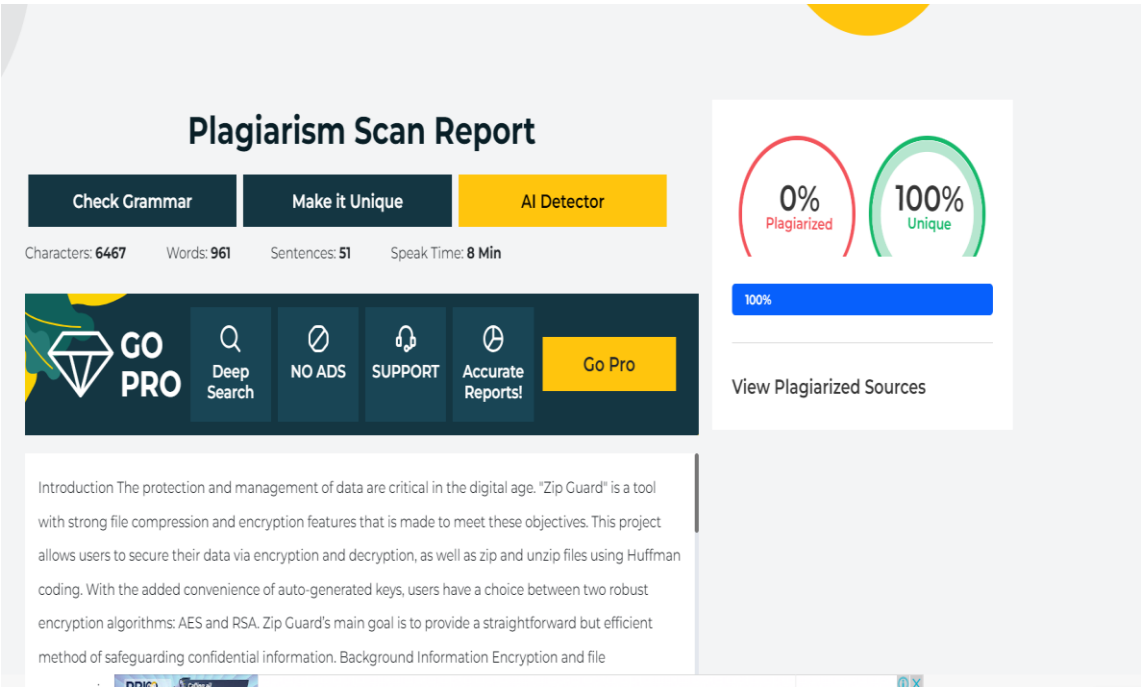
Proposed Method

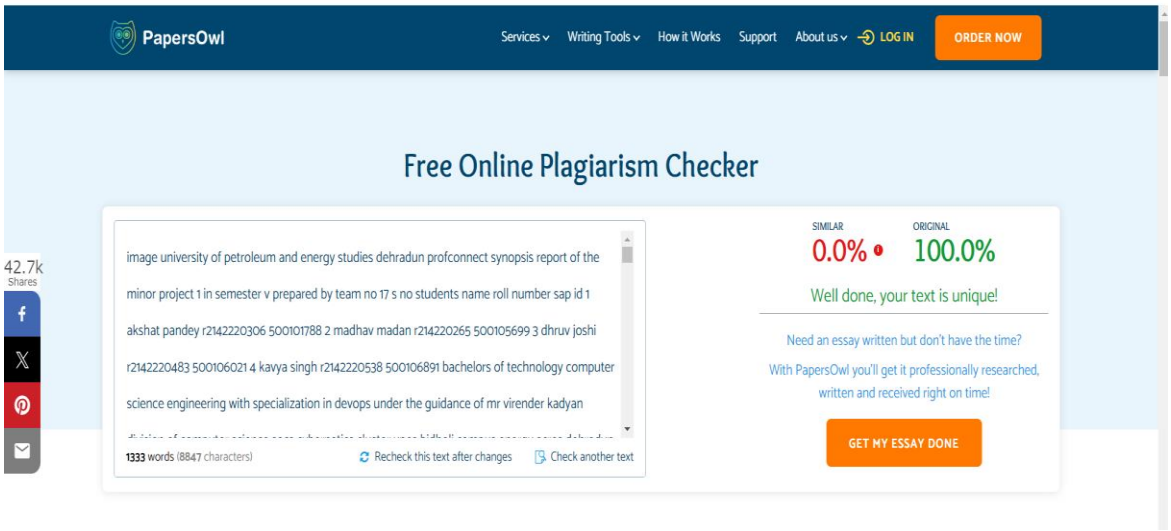
Zip Guard" compresses files using Huffman coding and encrypts them with RSA/AES. The first step in the procedure is for the user to choose which files to compress. After that, the utility uses Huffman coding to minimize the file size. The tool then automatically produces the necessary keys once the user chooses between RSA and AES encryption. After the files have been encrypted and compressed, they can be transferred or stored safely. This technique makes sure that data is safely stored and managed effectively.

What exactly is done

1. **Research About the Topic:** Extensive research is conducted on Huffman coding, RSA, and AES encryption algorithms to understand their implementation and best practices for combining them in a single tool.
2. **Schedule Team Meetings on Google Meet:** Regular meetings are organized using Google Meet to discuss progress, share updates, and coordinate development efforts among team members.
3. **Gathered Data and Created a Synopsis Report:** Collected and organized essential information related to file compression and encryption, outlining the project's scope and objectives.
4. **Created a Presentation Based on the Synopsis Report:** Developed a presentation summarizing the key points of the synopsis report for clarity and ease of communication.
5. R**eceived Mentor Feedback on the Synopsis Report:** Incorporated valuable feedback from the mentor, refining the project approach and enhancing the overall direction.

Plagiarism Scan Report





Approved By

(Mentor) (Panel Member)

References

[1] Oracle (2024) “Java™ Platform, Standard Edition Security Developer's Guide: Standard Names,” Available at: https://docs.oracle.com/en/java/javase/22/docs/specs/security/standard-names.html .

[2] Oracle (2024) “Java™ SE 8: Cryptography Roadmap,” Available at: https://www.java.com/en/jre-jdk-cryptoroadmap.html .

[3] Data Structures Project - File Compression Using Huffman Coding in JAVA (2023) “YouTube,” Available at: https://www.youtube.com/watch?v=S0Wua5WxKZI .

[4] IETF (2016) “PKCS #1: RSA Cryptography Specifications Version 2.2,” RFC 8017. Available at: https://www.ietf.org/rfc/rfc8017.html .

[5] Schneier, B. (1996) Applied Cryptography: Protocols, Algorithms, and Source Code in C. 2nd end. New York: John Wiley & Sons. Available at: https://mrajacse.wordpress.com/wp-content/uploads/2012/01/applied-cryptography-2nd-ed-b-schneier.pdf .