



# Encrypted Database using RAM

Harsh Singh<sup>1</sup> | Gaurav Bisht<sup>2</sup> | Dr. Shikha Gupta<sup>3</sup> | Dr. Sunil Maggu<sup>4</sup>

<sup>1,2</sup>Student, Information Technology, Maharaja Agrasen Institute of Technology, Rohini, Delhi

<sup>3,4</sup>Assistant Professor, Information Technology, Maharaja Agrasen Institute of Technology, Rohini, Delhi

## To Cite this Article

Harsh Singh, Gaurav Bisht, Dr. Shikha Gupta and Dr. Sunil Maggu. Encrypted Database using RAM. *International Journal for Modern Trends in Science and Technology* 2021, 7, pp. 192-194. <https://doi.org/10.46501/IJMTST0711032>.

## Article Info

Received: 15 October 2021; Accepted: 25 November 2021; Published: 28 November 2021

## ABSTRACT

This research paper focuses on a Database project which is made using JavaScript and Node.js. "Encrypted Database using RAM" is a CLI (Command Line Interface) app which is used to store the Data into the main memory of the system which helps to access the data quickly which in a trivial system takes too much time because of the use of secondary memory to store the data.

The data gets compressed and encrypted before getting stored into the memory with the help of Huffman Coding Algorithm which makes our database more secure and the data takes less space than the normal one because of the compression. Thus, our project improves the access speed, security and the storage used by the data.

**KEYWORDS:** Database, Database Management System, Huffman coding algorithm, Compression, Encryption, JavaScript, Node.js.

## INTRODUCTION

In the present scenario, the use of the internet by the people has reached a new height. In the near future also, the use of the internet is going to increase because a large number of the population is still untouched by the use of the internet. The integral part of the internet is the DATA, which has to be stored and accessed in the most secure and optimized way. An individual's privacy, the security of data and the faster access of it is the top priority of any average internet user. This research focuses on the very same concerns of an individual or any organization, and the institution. It uses the main memory of the system to store and access the data, ultimately it helps in the faster access of data. Encryption is used while storing the data, hence increasing the security. And the compression of data while storing it, enhances the space complexity.

This is where our project "Encrypted Database using RAM" aims to fix and improve this current system for good. Two major aspects of improving database performance and reliability is optimizing the access speed and the security of the data.

- The main concern in access speed is read/write speed of magnetic disks. Moore's law accelerates the performance of CPU and memory, but the disk's performance still lags behind. Encrypted databases using main memories that directly enter data into the main memory have been developed to perform manipulation because of this mismatch between the performance of CPUs and disks. Compared with disk, the read/write speed of memory is greater by magnitude of approx.  $10^6$  times.
- In order to increase the security of the data, the encryption can be achieved using Huffman coding. It is a lossless data compression algorithm. The idea is to

assign variable-length codes to input characters; lengths of the assigned codes are based on the frequencies of corresponding characters. The most frequent character gets the smallest code and the least frequent character gets the largest code. The variable-length codes assigned to input characters are Prefix Codes means the codes (bit sequences) are assigned in such a way that the code assigned to one character is not the prefix of code assigned to any other characters.

## OBJECTIVES

The main objectives of our project "Encrypted Database using RAM" are:

Provide a private and instant Database using the main memory of the user's system.

- To enhance the security of the Data and provide multi-layered protection by encryption of data using Huffman Coding Algorithm.
- To provide faster access to the data by providing speed of magnitude approx.  $10^6$  times than the trivial Database system.
- To provide the platform on which the risk of data breach and its consequences will be minimized.
- To remove the dependency on the external database systems where the data privacy is questionable.

## TECHNOLOGY USED

- JavaScript: JS is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js
- Node.JS: Node.JS is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.

## IMPLEMENTATION

Implementation of the project has been done in two following sections which comprises improving the Access speed of the data and the Compression and Encryption of the stored data:

- Increased Access speed: The Increase in speed of accessing the data has been achieved by using Hash Table data structure that offers an amortized  $O(1)$  time

access to data. The reason for using this data structure is, it uses key-value as the storage format. This database is using an exact key matching to access the desired data. There is no range base, pattern matching, sorted queries. Storage of data is being done in the main memory of the system. The main memory is faster than the disk memory by the magnitude of approximately  $10^6$  times. Hence, increasing the retrieval speed of data. JavaScript and its useful concepts were used to store the data and provide the management system of the database while using Main memory of the system. The data then is saved into secondary memory and whenever the use of that particular database is required by the user, it gets loaded into the main memory for the fast access.

- Increased Security and Compression: In our project, the encryption and compression of data has been achieved by using the Huffman algorithm, which is a technique of compressing data to reduce its size without losing any of its details. In general it is useful to compress the data in which there are frequently occurring characters but a small modification to the Huffman Algorithm resulted in compression and encryption of the data even having a single character. The compression and encryption process is done when the data is given by the user. The data after getting compressed and encrypted, gets stored in the main memory and remains unchanged even after getting saved into the secondary memory. When the data is requested by the user with the help of "get" function, it goes through the process of decompression and decryption and original data is shown to the user.

## FUTURE SCOPE

Our Encrypted Database using RAM possess a great scope for the future because of the immense potential it has and various ideas we plan to execute in near future:

- The security of the database can always be improved because of the various techniques and algorithms that can be merged. The access time is already many times faster than the trivial disk memory one but still there is a good scope for faster access because of many techniques that use different structures to store data for faster access. Such examples are B Trees and B+ Trees.

- It can be converted into a system application for different Operating Systems such as Windows, Linux, and MacOS that will ease its use.
- Since the data is now encrypted using Huffman Coding, we can also store it into cloud storages without any fear in case of data leak/breach since our data is already encrypted.
- Back-end could also be added so that it can ease the transfer of data from our system to cloud databases and vice versa.
- A login interface could be added for only authorized access to the data on the system itself.

## CONCLUSION

Our project is capable of making the data encrypted, hence enhancing the security. It is providing a multi layered security to the user. It is highly customizable in nature and can be modified further in accordance with the future requirements. The access time for the data has been increased by a magnitude of approximately  $10^6$  times. A decent front-end enhances the user interface along with the ease in selecting various operations on the database. This project has a high potential to be used for personal as well as industrial usage. Moderate-scale products will gain the maximum benefits from our project because of such quick transactions on data while stored on their machine only and even if they decide to store it onto the cloud databases, the encryption using Huffman Coding algorithm will make sure it can't be used by others in case of some data-breaches.

## ACKNOWLEDGMENT

Foremost, we would like to express our sincere gratitude to our mentors Dr. Sunil Maggu and Dr. Shikha Gupta for the continuous support during the project and research paper, for their patience, motivation, enthusiasm, and immense knowledge. Their guidance helped in all the time of research and writing of this paper. We could not have imagined having a better advisors and mentors for our project.

## REFERENCES

- [1] H. Garcia-Molina and K. Salem, "Main memory database systems: an overview," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 4, no. 6, pp. 509-516, Dec. 1992, doi: 10.1109/69.180602.
- [2] Medeiros, Henry Ponti, et al. "Lightweight data compression in wireless sensor networks using Huffman coding." *International Journal of Distributed Sensor Networks* 10.1 (2014): 672921.
- [3] Chauhan, Priyank, Sanjali Gupta, & Nikhil Shanker Mathur. "A New Approach to Encryption using Huffman Coding." *International Journal of Progressive Sciences and Technologies [Online]*, 2.2 (2016): 76-82. Web. 25 Nov. 2021.
- [4] <https://www.oracle.com/in/database/what-is-database/>
- [5] <https://www2.cs.duke.edu/csed/poop/huff/info/>
- [6] <https://developer.mozilla.org/en-US/docs/Web/JavaScript>