Enhanced Database Encryption System

Piyush Kumar
Department of Networking and Communications, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulatur, India
pk0669@srmist.edu.in

Laraib Hanif
Department of Networking and Communications, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulatur, India
lh9435@srmist.edu.in

L.N.B Srinivas*
Department of Networking and Communications, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulatur, India
srinival@srmist.edu.in

***Abstract*- This study, in particular, highlights the critical nature of Impacts of the Enhanced Database Encryption. System within contemporary enterprises. It places significant focus on critical goals, such as data consolidation [1]. Preservation of integrity, and enhanced security controls. Moreover, It emphasizes the essential nature of Database Management System. (DBMS) in streamlining various operational domains including customer relationship management, supply chain optimization and advanced analytics. Furthermore, the paper emphasizes the importance of dealing with such data privacy issues. regulatory compliance, as well as setting up a strong data governance frameworks. These efforts collectively contribute to the development of stable and highly reliable databases that are necessary for contemporary organizations that want to protect their strategic resources and preserve the faith in the digital-centered landscape.**

Keywords—Integrity preservation; Advanced analytics; Data governance framework; Data consolidation

I. INTRODUCTION

With more developments in the field of data management and security taking place, encryption for securing sensitive information has become crucial. The annotated project entitled "Improving Database Encryption Systems"[2] discusses the area of data security that concentrates on enhancing and strengthening encryption methods operating within databases. This study seeks to underscore the pivotal nature of strong database encryption systems in an age characterized by unending cyber threats and tight regulatory frameworks.

The main goal of this project is to analyze and improve current database encryption systems in order to strengthen data protection, which provides a guarantee for confidentiality, integrity, and accessibility of private information. This research seeks to set a new standard for data security, both in an organizational setting and on the individual level using sophisticated encryption methods and creative solutions. In addition, the initiative acknowledges the ever-changing landscape of cyber dangers and seeks to predict new developments in the field of data protection. Through the integration of adaptable measures and keeping up with technological developments, the research aims to provide a robust framework that can change with changing security environments. Fostering user-friendly implementations of these upgraded security measures is also crucial in order to guarantee smooth integration and user acceptability, which in turn encourages widespread adoption and adherence to strong data protection standards. By using this diverse strategy, the project hopes to improve both the security and user-friendliness of the digital environment while also raising the bar for existing data security standards.

The paper starts by conducting a comprehensive analysis of the current state of database encryption, revealing its strengths and weaknesses as well as identifying potential opportunities for development. This study then sets off on the path of innovation and evolution with a goal of producing advanced encryption algorithms, key management models, and access control materials. These improvements are not only intended to strengthen data security but also optimize database performance and compliance with new data protection legislation.

Since these questions are complex in nature and require a lot of experimentation and analysis, this project aims at not only answering the above questions but also provides practical peace-of-mind solutions that can be widely applicable in real world database encryption scenarios. The long-term objective is to assist in the continued initiatives of securing data, protecting confidential information, and helping organizations and individuals leverage technology without fear.

In the following parts of our research paper, we are going to discuss the methods, results and recommendations based on investigation of advanced data encryption systems that suggest ways towards safer data driven future.

II. LITERATURE REVIEW

A. *Advancing database security*

Under CC BY 4.0, the article "Advancing Database Security [3] A Comprehensive Systematic Mapping Study of Potential Challenges," published in the July 2023 issue of Wireless Networks presents a thorough systematic mapping study dedicated to addressing challenges aimed at strengthening database security that is currently requested by rapidly increasing number of data breaches. It provides key insights into different areas of security such as phases, threats, and issues that could enhance strategies on how to protect data better.

But there is also an important limitation to consider in the approach of this study. The research mainly uses a dataset for its conclusions. Though this approach has its advantages, it may not be able to fully describe the nuances and subtleties that are characteristic of many real-life situations. However, nothing could be further from the truth in practice as advanced database security needs to adjust to a variety of contexts and dynamic threats. Future studies in this domain must include more pluralistic and total sources of information, providing a broader perspective of the intricate issues surrounding protecting confidential information within the dynamic and globalized digital age.

B. *KMF Algorithm in Database System*

The establishment of a strong and private communication channel is contingent upon the secure exchange of a security key between a server and a client. Cryptographic techniques and algorithms are usually used in this process to protect the integrity and confidentiality of the transferred key. In order to mutually authenticate and establish a shared secret key, the server and client engage in a secure handshake during this transaction. The key exchange is guaranteed to take place in a secure manner by utilising protocols such as Secure Sockets Layer (SSL) or Transport Layer Security (TLS), or its predecessors. The resistance of this procedure against eavesdropping and unauthorised access is increased by the use of robust authentication protocols and encryption mechanisms.

Furthermore, it is essential to securely exchange the security key in order to keep malicious parties from intercepting and decoding private information as it is being transmitted. This procedure is an essential part of the larger plan to strengthen the privacy of channels of communication. A safe environment is created for the smooth movement of information when a secure key exchange is established, giving the client and server confidence that the shared secret is known only to them.

C. *Query Optimization in Database System*

The paper, authored by Mathias Jarke of The Graduate School of Business Administration at New York University and Jürgen Koch from Johann Wolfgang Goethe- Universität in Frankfurt, West Germany describes a development history on database management systems (DBMS) and focuses attention to query optimization techniques [4,9] within the relational calculus framework. The purpose of this analysis is to enhance the effectiveness of dealing with complicated query operations, which is one of the main aspects related to database administration.

Nevertheless, though, it is necessary to note the paper's boundaries which may make its use in modern and more sophisticated database systems problematic. One significant weakness is that distributed query optimization receives a very short description. In modern networked and distributed computing, where data is scattered through numerous locations and systems, improving queries that span such distributed resources is a current issue.

The utilisation of indexes, join order, and the choice of suitable algorithms for tasks like sorting and filtering are all taken into consideration during query optimisation. The need of effective query optimisation techniques is highlighted by the complexity of contemporary database systems, which have massive datasets and intricate queries. Advanced optimisation tactics, such dynamic programming and heuristic algorithms, further contribute to reaching optimal query performance. To ensure continuous efficiency in database operations, it is imperative to conduct regular monitoring, analysis, and optimisation strategy adjustments in response to shifting data distributions and query patterns. Essentially, query optimisation is an ongoing procedure that is essential to preserving a database system's overall responsiveness and speed.

D. *Indexing Techniques in Database System*

In the field of information retrieval, indexing strategies are essential for improving the effectiveness and performance of a database system. Their importance cannot be emphasised enough. The main objective of indexing in a database system is to speed up the retrieval of data by offering an organised and efficient method of information access. Advanced indexing strategies, such as hash-based and B-tree approaches, help reduce query response times and enhance overall system responsiveness. Effective indexing is essential for database administration since it not only speeds up data access but also helps with query optimisation, which enables complex queries to be executed quickly. The investigation and improvement of indexing strategies remain important as database systems continue to handle enormous volumes of data.

Effective indexing is essential for improving query execution plans in addition to speeding up data retrieval. It makes a substantial contribution to decreasing query response times, which improves the database system's overall responsiveness and performance—especially in situations involving big amounts of data. To be on the cutting edge of data retrieval efficiency in the ever- changing field of database management, indexing strategies must be continuously explored and refined.

E. *Differential Data Recovery Algorithm*

In terms of data restoration and retrieval, the Differential Data Recovery Algorithm is a state-of-the-art method. This algorithm, in contrast to traditional recovery techniques, minimises the time and resources needed for the recovery process by identifying and restoring just the modified or altered data. The technique effectively identifies modifications made to a dataset since the last backup by utilising the idea of data differentials, enabling a more focused and rapid recovery.

This method greatly lowers the recovery time and storage space needs, making it especially useful in situations involving huge datasets. The Differential Data Recovery Algorithm is a significant development in the field of data recovery techniques that shows promise for improving data resilience and reducing downtime in important systems.

F. *Query Optimization in Database System*

The paper, authored by Mathias Jarke of The Graduate School of Business Administration at New York University and Jürgen Koch from Johann Wolfgang Goethe- Universität in Frankfurt, West Germany describes a development history on database management systems (DBMS) and focuses attention to query optimization techniques [4,9] within the relational calculus framework. The purpose of this analysis is to enhance the effectiveness of dealing with complicated query operations,

which is one of the main aspects related to database administration.

Nevertheless, though, it is necessary to note the paper's boundaries which may make its use in modern and more sophisticated database systems problematic. One significant weakness is that distributed query optimization receives a very short description. In modern networked and distributed computing, where data is scattered through numerous locations and systems, improving queries that span such distributed resources is a current issue.

III. Motivation

To achieve system optimization and scalability as main objectives are to improve the capabilities of the system to process increasing load and demand from users. It includes a methodical approach characterized by code optimization, careful tuning of databases, workload balance and prudent use of scalable structures. It includes a methodical approach characterized by code optimization, careful tuning of databases, workload balance and prudent use of scalable structures. This requires a planned method that includes the use of efficient coding, careful adjustment of databases, balance of workload and intelligent utilisation of scalable infrastructure This means a balancing way to utilize the code, fine-tune databases in detail, balance load works and properly use scalable infrastructure. The all-encompassing objective is to maintain the system's receptiveness and effectiveness despite growth, thereby guaranteeing uninterrupted user experience.

At the same time, the second crucial aspect focuses on strengthening security [5] of the system by implementing robust and trustworthy authentication schemes. This refers to the use of strong encryption technology, multi-factor authentication and other security mechanisms all aimed at unquestionably identifying users. The system aims at securing entry to authorized individuals only by implementing effective authentication measures in order to curb unauthorized access.

In addition to performance and security aspects, the other area of focus emphasizes developing an interface that is user-friendly and understandable. This includes the creation of an interface that makes it very easy for a user to navigate, having clear layouts, easily navigable menus and straightforward features. A human-centered design reduces user frustration, encourages involvement, and increases user satisfaction, thus focusing primarily on the heterogeneous users.

IV. Design Methodology

Efficient data management and the assurance of reliable data are central tenets facilitated by a robust Database Management System (DBMS). Serving as the organizational hub, the DBMS simplifies the consolidation of data integrity from diverse sourceslike sales, marketing, and finance. Through meticulous validationchecks, elimination of duplicates [6], and standardization of data formats, the DBMS ensures data fidelity, resulting in a high- quality dataset that is both dependable and precise. This consolidated and refined data provides companies with a comprehensive understanding of their operations, forming the basis for well-informed decision-making rooted in a thorough comprehension of the business environment.

Beyond streamlining data administration, a proficiently managed database system acts as a catalyst for optimized decision-making by granting real-time access to vital business data. This empowers employees to make swifter and more informed decisions, facilitating in-depth data analysis to identify trends, customer preferences, and market dynamics. Armed with these insights, businesses can make well-considered decisions, thereby enhancing overall operations and strategic planning. Additionally, a centralized database promotes collaboration among different departments, leading to more synchronized and unified decision-making processes throughout the organization.

Addressing paramount concerns in today's business landscape, data privacy, and regulatory compliance, a dependable DBMS comprehensively handles these issues. Robust user access controls ensure that sensitive data remains accessible only to authorized personnel, a fundamental aspect of protecting data privacy. Features such as data encryption, both at rest and during transmission, are crucial for data security and meeting stringent privacy regulations like GDPR and HIPAA. Furthermore,comprehensive audit trails facilitated by the DBMS empower organizations to meticulously monitor data access and alterations, playing an essential role in compliance reporting and showcasing transparency in data management.

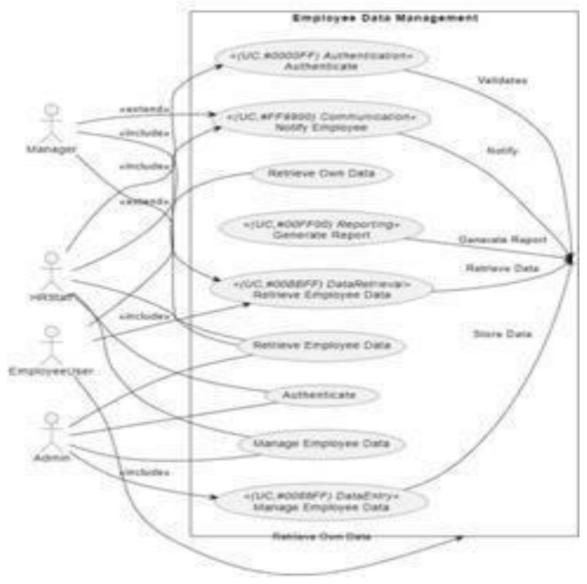


Fig. 1. Use Case Diagram

In a rapidly changing business environment, agility is crucial,and a flexible DBMS framework provides scalability to accommodate the increasing data volumes characteristic of the present data-intensive landscape. This adaptability is particularly important for companies managing expanding data loads efficiently. Moreover, the DBMS is positioned for the future, seamlessly integrating with emerging technologies such as artificial intelligence, the Internet of Things, and blockchain. This forward-thinking adaptability positions the DBMS as a dynamic asset in the digital era, allowing businesses to adopt cutting-edge technologies without the need for a comprehensive overhaul of their current database infrastructure.

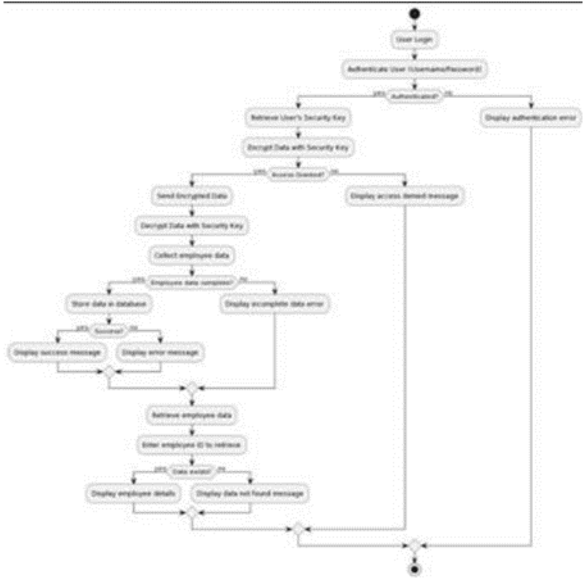
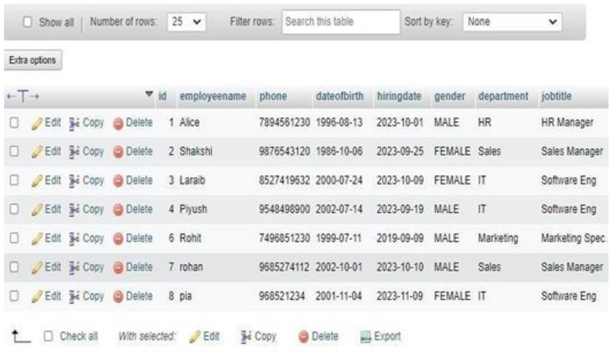

User Login
Authenticate User (Username/Password)
yes
Authenticated?
no
Retrieve User's Security Key
Display authentication error
Decrypt Data with Security Key
yes
Access Granted?
no
Send Encrypted Data
Display access denied message
Decrypt Data with Security Key
Collect employee data
yes
Employee data complete?
no
Store data in database
Display incomplete data error
yes
Success?
no
Display success message
Display error message
Retrieve employee data
Enter employee ID to retrieve
yes
Data exists?
no
Display employee details
Display data not found message

Fig. 2. .Flow Chart Diagram

Fig. 3. Local server



Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

				id	employeename	phone	dateofbirth	hiringdate	gender	department	jobtitle
☐	Edit	Copy	Delete	1	Alice	7894561230	1996-08-13	2023-10-01	MALE	HR	HR Manager
☐	Edit	Copy	Delete	2	Shakshi	9876543120	1986-10-06	2023-09-25	FEMALE	Sales	Sales Manager
☐	Edit	Copy	Delete	3	Laraib	8527419632	2000-07-24	2023-10-09	FEMALE	IT	Software Eng
☐	Edit	Copy	Delete	4	Piyush	9548498900	2002-07-14	2023-09-19	MALE	IT	Software Eng
☐	Edit	Copy	Delete	6	Rohit	7496851230	1999-07-11	2019-09-09	MALE	Marketing	Marketing Spec.
☐	Edit	Copy	Delete	7	rohan	9685274112	2002-10-01	2023-10-10	MALE	Sales	Sales Manager
☐	Edit	Copy	Delete	8	pia	968521234	2001-11-04	2023-11-09	FEMALE	IT	Software Eng

Check all With selected: Edit Copy Delete Export

Fig. 4. Entries in server

A Use Case Diagram (fig.1) for a Database Management System (DBMS) provides a visual representation of how users and external systems interact with the DBMS highlighting various use cases or functions offered by the system. In the simplified diagram provided, the central element is the "Database Management System" itself.

A flowchart for a Database Management System (DBMS) (fig.2) showing various processes and interactions pertaining to databases. The flowchart demonstrates a systematic diagrammatic picture of the workflow system through major actions and decision points. The "Database System" is the central element of this simplified flowchart where we begin at a "Start" point and end at an "End," with the user interacting with the DBMS performing specific use cases depicted as distinct branches from his or her interactions.

V. Implementation

A. Modules

Indexing: This algorithm allows to build well- balanced indexing structures [7], for instance, B-trees(fig.3), which playa significant role in acceleration of query operation in large databases with logarithmic time complexity. The structured approach guarantees that information is preserved in an organized manner (fig.4), shortening search times, and improving the overall refectiveness of the database.

Handshake Protocol [8]: During the initializing phase of TCP communication, the crucial three-way handshake takes place. It begins with the client initiating the process by sending a synchronize packet (SYN) to the server. This packetincludes a randomly generated sequence number.

In response, the server counters by sending a SYN-ACK packet back to the client. The SYN-ACK packet is self-generated and contains an acknowledgment (ACK) number corresponding to the sequence number sent by the client.

Once the client receives the server's SYN-ACK message, it responds with an ACK message, acknowledging the sequence number provided by the server. This exchange ensures that the sequence numbers on both sides are now aligned, effectively synchronizing the communication. With this synchronization accomplished, independent transmission is facilitated for both the client and the server, setting the stage for reliable and efficient data exchange.

KMF: The Key Management Framework (KMF) algorithm simplifies and secures the exchange of cryptographic keys between servers and clients during secure communications, as commonly seen in SSL/TLS.It ensures that keys are generated, distributed, and maintained securely, reducing the risk of key-related security breaches.

B. Implementation

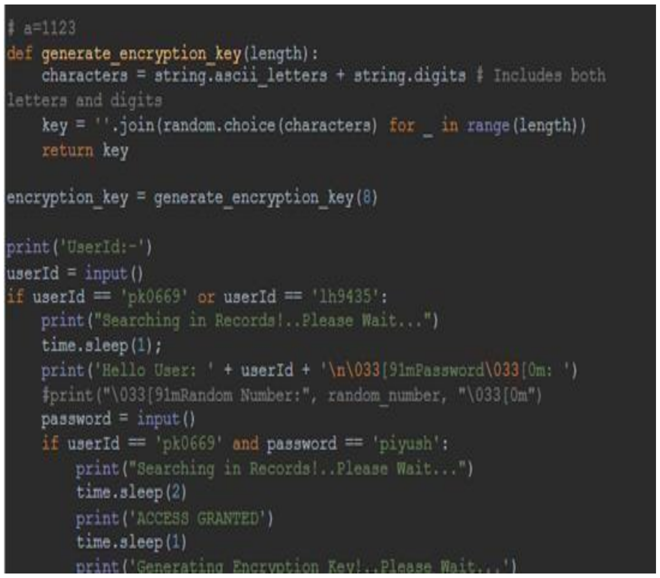
In the encryption phase [10], the passwords for users is critical, and a number of actions are taken to preserve this. sensitive information. Rather than using user passwords as littering the database with plain text (fig.5), they go through a process password hashing. This involves employing one-way cryptographic function like bcrypt, scrypt, or Argon2. that change the original password to a fixed length. string of characters. This transformation is aimed at impossible to undo computationally, hence the irreversibility of original password cannot be easily recovered. ensuring that the original password cannot be easily obtained.

To further enhance security, a technique known as salting is employed. For each user, a random salt is generated and added to the password before hashing. This random value ensures that even if two users have the same password, their hashed values will be distinct due to the unique salts associated with each user. This serves as a defense mechanism against attacks like rainbow tables.

In some cases, an additional layer of security called "peppering"is applied. Peppering involves the use of a secret value that is combined with the password before hashing. This secret value, The First step will be to generate the sensitive key using KMF or pepper, is typically stored separately from the database and, algorithm with the help of authenticated ID & password and keep adding an extra layer of security and making it more challenging for attackers to compromise passwords.

In the password decryption phase of database password management, Decryption of passwords does not occur during authentication (fig.6). Otherwise, when a user enters password, the entered password is hashed employing the same salt and pepper (if present) as well as comparing with stored hash in the database. The authentication process involves retrieving the user’s salt and pepper, mixing them with. takes the entered password, hashes the result and compares it with a checksum. stored hash. When hashes are successfully authenticated.

Match, providing authorization though failed authentication results in denied access. This security approach avoids storing plain text passwords, an additional layer of security. making it difficult for attackers to compromise user. credentials. it challenging for attackers to compromise user credentials.

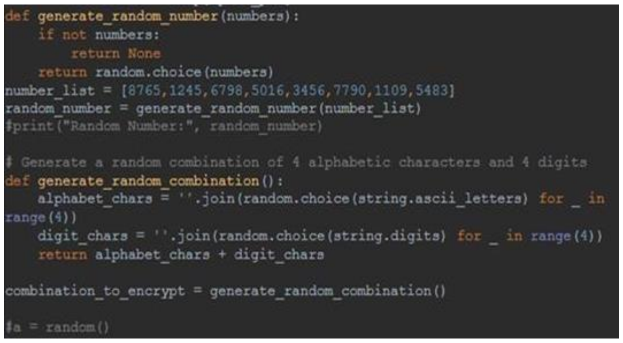


```
# a=1123
def generate_encryption_key(length):
    characters = string.ascii_letters + string.digits # Includes both
letters and digits
    key = ''.join(random.choice(characters) for _ in range(length))
    return key

encryption_key = generate_encryption_key(8)

print('UserId:-')
userId = input()
if userId == 'pk0669' or userId == 'lh9435':
    print("Searching in Records!..Please Wait...")
    time.sleep(1);
    print('Hello User: ' + userId + '\n\033[91mPassword\033[0m: ')
    #print("\033[91mRandom Number:", random_number, "\033[0m")
    password = input()
    if userId == 'pk0669' and password == 'piyush':
        print("Searching in Records!..Please Wait...")
        time.sleep(2)
        print('ACCESS GRANTED')
        time.sleep(1)
        print('Generating Encryption Key!..Please Wait...')
```

Fig. 5. Encryption phase



```
def generate_random_number(numbers):
    if not numbers:
        return None
    return random.choice(numbers)
number_list = [8765,1245,6798,5016,3456,7790,1109,5483]
random_number = generate_random_number(number_list)
#print("Random Number:", random_number)

# Generate a random combination of 4 alphabetic characters and 4 digits
def generate_random_combination():
    alphabet_chars = ''.join(random.choice(string.ascii_letters) for _ in
range(4))
    digit_chars = ''.join(random.choice(string.digits) for _ in range(4))
    return alphabet_chars + digit_chars

combination_to_encrypt = generate_random_combination()

#a = random()
```

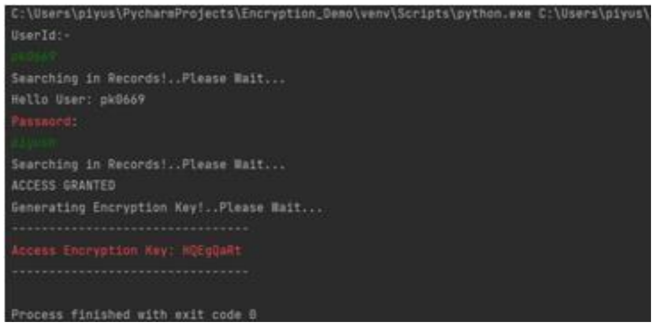
Fig. 6. Decryption phase

VI. EXPERIMENTAL RESULTS

A. *KMF Encryption Key Generation*

In the password decryption phase of database password management, Decryption of passwords does not occur during authentication (fig.6). Otherwise, when a user enters password, the entered password is hashed employing the same salt and pepper (if present) as well as comparing with stored hash in the database. The authentication process involves retrieving the user’s salt and pepper, mixing them with. takes the entered password, hashes the result and compares it with a checksum. stored hash. When hashes are successfully authenticated.

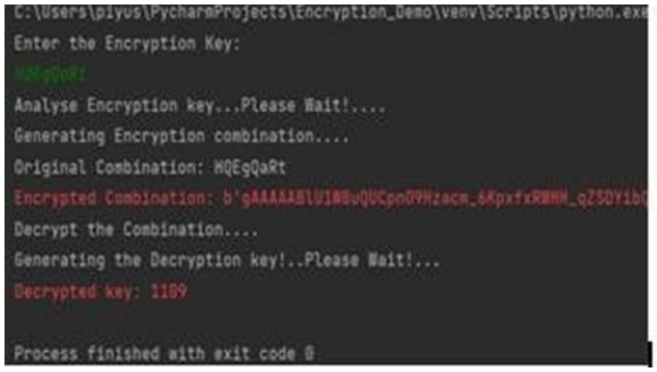
Match, providing authorization though failed authentication results in denied access. This security approach avoids storing plain text passwords, an additional layer of security. making it difficult for attackers to compromise user. credentials. it challenging for attackers to compromise user credentials.



```
C:\Users\piyus\PycharmProjects\Encryption_Demo\venv\Scripts\python.exe C:\Users\piyus\
UserId:-
pk0669
Searching in Records!..Please Wait...
Hello User: pk0669
Password:
piyush
Searching in Records!..Please Wait...
ACCESS GRANTED
Generating Encryption Key!..Please Wait...
--------------------------------
Access Encryption Key: HQEgQaRt
--------------------------------

Process finished with exit code 0
```

Fig. 7. .Encrypted Key



```
C:\Users\piyus\PycharmProjects\Encryption_Demo\venv\Scripts\python.exe
Enter the Encryption Key:
HQEgQaRt
Analyse Encryption key...Please Wait!....
Generating Encryption combination....
Original Combination: HQEgQaRt
Encrypted Combination: b'gAAAAABlU1B8uQUCpn09Hzacm_6KpxfxRBHM_qZSDYib
Decrypt the Combination....
Generating the Decryption key!..Please Wait!...
Decrypted key: 1109

Process finished with exit code 0
```

Fig. 8. Decrypted Key

B. Decryption KeyGeneration

After, getting the required encrypted key from the user, we will be using it and generate Decrypted key using cryptography and use it to verify our authenticity, so that we can login our database company portal, based on the above decrypted sensitive key(shown in fig.8)

VII. CONCLUSION

Therefore, "Enhanced Database Encryption System" stands continuous growth to meet the increasing needs of data security and privacy. These include homomorphic computations on encrypted data are performed with the help of encryption. with decryption, however quantum-resistant encryption to protect along with quantum threats, and multi- party computation for secure collaborative data analysis. Additionally, improved ZK Proofs, block integration and refined security will be buttressed through access control mechanisms security.

Post-quantum cryptography will become a necessity for security and privacy-preserving analytics will be long term permits analysis of data while preserving privacy. The constantly shifting environment of data security will require continuous accommodation to new threats technological advancements to safeguard databases and everything.

REFERENCES

[1] Cockshott, W. P., et al. "Data Compression in Database Systems." Proceedings Ninth International Workshop on Database and Expert Systems Applications (Cat. No.98EX130), 1998, pp. 981–90

[2] Hwang, Min-Shiang, and Wei-Pang Yang. "A Two-Phase Encryption Scheme for Enhancing Database Security." Journal of Systems and Software, vol. 31, no. 3, Dec. 1995, pp. 257–65. Semantic Scholar, https://doi.org/10.1016/0164-1212(94)00102-2.

[3] Iqbal, Asif, et al. "Advancing Database Security: A Comprehensive Systematic Mapping Study of Potential Challenges." Wireless Networks, July 2023.

[4] Bertino, Elisa. "Data Security and Privacy: Concepts, Approaches, and Research Directions." 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), IEEE, 2016, pp. 400–07

[5] Barbella, Marcello, and Genoveffa Tortora. "A Semi-Automatic Data Integration Process of Heterogeneous Databases." Pattern Recognition Letters, vol. 166, Feb. 2023, pp. 134–42.

[6] Gupta, M. K., & Badal, D. (2012). Comparative study of indexing techniques in DBMS, available as a conference paper with DOI: 10.5120/333844844.

[7] Hsu, Fu-Hau, et al. "TRAP: A Three-Way Handshake Server for TCP Connection Establishment." Applied Sciences, vol. 6, no. 11, Nov. 2016, p. 358. www.mdpi.com, https://doi.org/10.3390/app6110358.

[8] Kossmann, J., Papenbrock, T., & Naumann, F. (2021). Data dependencies for query optimization: a survey. The VLDB Journal, 30(5), 761-789.

[9] Al-hazaimeh, Obaida M., et al. "Analytical Approach for Data Encryption Standard Algorithm." International Journal of Interactive Mobile Technologies (iJIM), vol. 17, no. 14, Aug. 2023, pp. 126–43.

[10] Eken, Gorkem, et al. "A Lessons Learned Database Structure for Construction Companies." Procedia Engineering, vol. 123, Jan. 2015, pp. 135–44.

[11] Susanto, A., & Meiryani. (2019). Database management system. International Journal of Scientific & Technology Research.

[12] Dixit, Anil, and Dr Suchithra R. "Advanced Database Security and Encryption." International Journal of Engineering Research & Technology, vol. 4, no. 21, Apr. 2018.

[13] Eken, Gorkem, et al. "A Lessons Learned Database Structure for Construction Companies." Procedia Engineering, vol. 123, 2015, pp. 135–44.

[14] Ge, Tingjian, and Stan Zdonik. "Fast, Secure Encryption for Indexing in a Column-Oriented DBMS." 2007 IEEE 23rd International Conference on Data Engineering, Apr. 2007, pp. 676–85.

[15] Liu, Lianzhong, and Jingfen Gai. "A New Lightweight Database Encryption Scheme Transparent to Applications." 2008 6th IEEE International Conference on Industrial Informatics, 2008.

[16] Department of Information Technology, KLN College of Engineering, Madurai, India, et al. "A Secured Database Monitoring Method to Improve Data Backup and Recovery Operations in Cloud Computing." BOHR International Journal of Computer Science, vol. 2, no. 1, 2023, pp. 1–7.

[17] Ordonez, Carlos. "Can We Analyze Big Data inside a DBMS?" Proceedings of the Sixteenth International Workshop on Data Warehousing and OLAP, Association for Computing Machinery, 2013, pp. 85–92. ACM Digital Library