International Journal of Engineering Research and Development
e- ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 22, Issue 3 (March 2026), PP 51-58

An Intelligent Data Protection Model for Relational Database Systems through SQL Encryption

T.N.Vignesh[1], Mrs. K.Vasumathi[2], Dr. S.Selvakani[3]
[1]*PG Scholar, PG Department of Computer Science, Government Arts and Science College, Arakkonam, Tamil Nadu, India.*
[2]*Assistant Professor, PG Department of Computer Science, Government Arts and Science College, Arakkonam, Tamil Nadu, India.*
[3]*Assistant Professor and Head, PG Department of Computer Science, Government Arts and Science College, Arakkonam, Tamil Nadu, India.*

Abstract
The fact that the data owners outsource their data to external service providers introduces many security and privacy issues. Among them, the most significant research questions relate to data confidentiality and user privacy. Encryption was regarded as a solution for data confidentiality. The privacy of a user is characterized by the query he poses to the server and its result. We explore the techniques to execute the SQL query over the encrypted data without revealing to the server any information about the query such as the query type or the query pattern, and its result. By implementing all the relational operators by using the unique selection operator on the server-side database with a constant number of elements in each time of selection, our proposal can defeat against the statistical attacks of the untrusted server compromising data confidentiality and user privacy. Experimental evaluation demonstrates that our proposal less affects the system's performance and is applicable in the real world.

Keywords: *Database outsourcing, database encryption, user privacy, access pattern privacy, access privacy.*

Date of Submission: 01-03-2026 Date of acceptance: 10-03-2026

I. Introduction

Virtualized infrastructure in cloud computing has become an attractive target for Cyber attackers to launch advanced attacks. This paper proposes a novel big data based security analytics approach to detecting advanced attacks in virtualized infrastructures. Network logs as well as user application logs collected periodically from the guest virtual machines (VMs) are stored in the Hadoop Distributed File System (HDFS).

The extraction of attack features is performed through graph-based event correlation and Map Reduce parser based identification of potential attack paths. Next, determination of attack presence is performed through two-step machine learning, namely logistic regression is applied to calculate attack's conditional probabilities with respect to the attributes, and belief propagation is applied to calculate the belief in existence of an attack based on them. Experiments are conducted to evaluate the proposed approach using well-known malware as well as in comparison with existing security techniques for virtualized infrastructure. The results show that our proposed approach is effective in detecting attacks with minimal performance overhead.

This paper proposes an **intelligent data privacy model for relational databases using SQL encryption**, which selectively encrypts sensitive attributes while allowing secure query execution on encrypted data. The proposed model incorporates intelligent decision mechanisms to identify sensitive data elements, apply appropriate encryption techniques, and manage secure query translation without revealing plain-text information to unauthorized users or database servers. By combining encryption strategies with SQL-aware processing, the modelensures fine-grained privacy protection while maintaining compatibility with existing relational database systems.

The proposed approach aims to enhance data confidentiality, minimize information leakage, and support practical deployment in real-world database environments. Experimental evaluation demonstrates that the model provides effective privacy protection with acceptable performance overhead, making it suitable for applications that require strong data security without sacrificing usability.

It is followed by a **Material and Methods** that reviews existing research and related work to establish the foundation of the proposed approach. Next, the **Experimental Results** section presents the implementation details along with performance analysis and outcomes. Finally, the **Discussion andConclusion** summarizes the key findings and contributions of the study, and the **References** section lists the sources and materials consulted during the research.

II. Material And Methods

MATERIALS:

HARDWARE REQUIREMENTS

Processor: Pentium IV
RAM : 128 MB
Hard disk: 50 GB
FDD: 1.44MB
Monitor: 14 inch
Mouse: 3 Button scroll
CD Drive: 52 X
Keyboard: 108 keys

SOFTWARE REQUIREMENTS

Operating System: Windows XP/2007
Front End: J2EE
IDE: NET-BEANS IDE
Back End: MYSQL

MODULES DESCRIPTION:

1. Authentication Module

This module contains all the information about the authenticated user. User without his username and password can't enter into the login if he is only the authenticated user then he can enter to his login.Authentication is the process of verifying the identity of a user by obtaining some sort of credentials and using those credentials to verify the user's identity. If the credentials are valid, the authorization process starts. Authentication process always proceeds to Authorization process. Administrators assume these responsibilities as volunteers who go through a community review process. They are not acting as users. They are never required to use their tools, and must never use them to gain an advantage in a dispute whose need the access for their database in secured way of organization. Administrators should not be confused incoming user registration time and login time

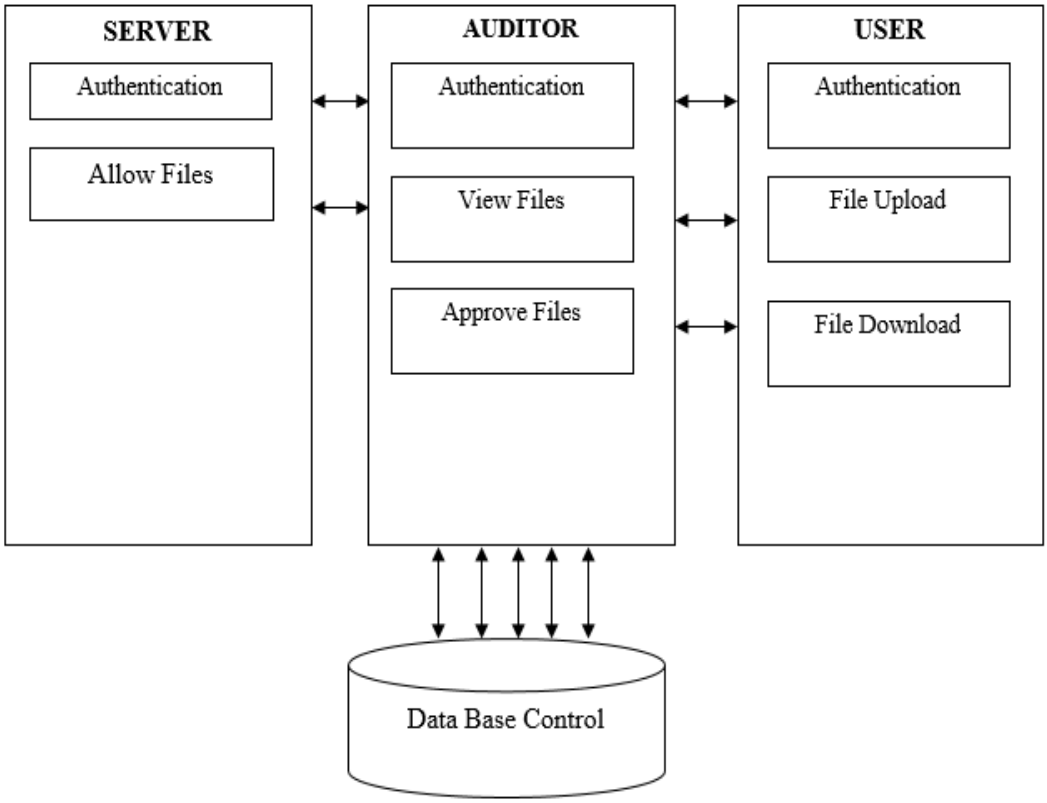

Figure 1: Architecture Diagram

2. Security Policies Module

Attribute-Based Encryption (ABE) has emerged as a promising technique to ensure the end-to-end data security in cloud storage system. It allows data owners to define the access policy and encrypt the data under the policy, such that only users whose attributes satisfying the access policies can decrypt the data. When more and more organizations and enterprises outsource the data into the cloud, the policy updating becomes a significant issue as the data access policies may be dynamically and frequently changed by data owners.

3. Data Auditor

Every authority is independent with each other and is responsible for managing attributes of users in its domain. It also generates a secret/public key pair for each attributes in its domain, and generates a secret key for each user according to their attributes.

4. Server Control Module

The cloud server stores the data of data owners and provides data access service to users. The server is also responsible for updating cipher texts from old access policies to new access policies.

5. Data Consumer

Each user is assigned with a global user identity and can freely get the cipher texts from the server. The user can decrypt the cipher text, only when the attributes satisfy the access policy defined in the cipher text.

III. Experiment And Results

1. EXPERIMENTS

Encrypted Data Storage Format

Encrypted values are stored in Base64 encoded format inside database columns.Example (Database Table View):

Table 1

ID	Name (Encrypted)	Email (Encrypted)
1	Q2F0NjlYV2x…	WUdkRjU4d…

Authorized User Output

When an authenticated user queries data: SELECT decrypt (name), decrypt (email) FROM users; System Output (User View):

Table 2

ID	Name	Email
1	John Doe	john@email.com

Unauthorized Access Output

If an attacker accesses the database directly: SELECT name, email FROM users;

Output:

Table 3

ID	Name (Encrypted Data)	Email (Encrypted Data)
1.	Q2F0NjlYV2x…	WUdkRjU4d…

Data remains unreadable without decryption key.

2. ALGORITHM

Advanced Encryption Standard (AES)

AES is a specification for the encryption of electronic data established by the U.S National Institute of Standards and Technology (NIST) in 2001. AES is widely used today as it is a much stronger than DES and triple DES despite being harder to implement.

Points to remember

- AES is a block cipher.
- The key size can be 128/192/256 bits.
- Encrypts data in blocks of 128 bits each.

That means it takes 128 bits as input and outputs 128 bits of encrypted cipher text as output. AES relies on substitution-permutation network principle which means it is performed using a series of linked operations which involves replacing and shuffling of the input data.

Working of the cipher:

AES performs operations on bytes of data rather than in bits. Since the block size is 128 bits, the cipher processes 128 bits (or 16 bytes) of the input data at a time.

The number of rounds depends on the key length as follows:

- 128 bit key – 10 rounds

- 192 bit key – 12 rounds
- 256 bit key – 14 rounds

Creation of Round keys:

A Key Schedule algorithm is used to calculate all the round keys from the key. So the initial key is used to create many different round keys which will be used in the corresponding round of the encryption.

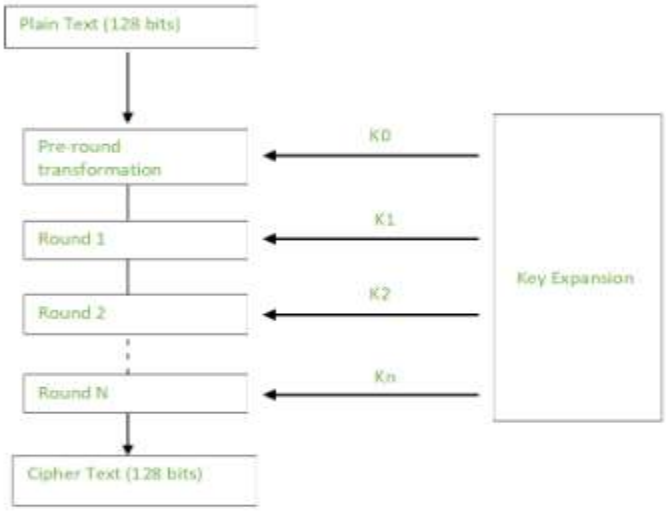

Figure 2: Round keys

Encryption:

AES considers each block as a 16 byte (4 byte x 4 byte = 128) grid in a column major arrangement.

```
[ b0 | b4 | b8 | b12 |
| b1 | b5 | b9 | b13 |
| b2 | b6 | b10| b14 |
| b3 | b7 | b11| b15 ]
```

Each round comprises of 4 steps:

- SubBytes
- ShiftRows
- MixColumns
- Add Round Key
- The last round doesn't have the MixColumns round.

The SubBytes does the substitution and ShiftRows and MixColumns performs the permutation in the algorithm.

SubBytes:

This step implements the substitution. In this step each byte is substituted by another byte. Its performed using a lookup table also called the S-box. This substitution is done in a way that a byte is never substituted by itself and also not substituted by another byte which is a compliment of the current byte. The result of this step is a 16 byte (4 x 4) matrix like before.

The next two steps implement the permutation.

Shift Rows:

This step is just as it sounds. Each row is shifted a particular number of times.

- The first row is not shifted
- The second row is shifted once to the left.

- The third row is shifted twice to the left.
- The fourth row is shifted thrice to the left.
- (A left circular shift is performed.)

```
[ b0  | b1  | b2  | b3  ]        [ b0  | b1  | b2  | b3  ]
| b4  | b5  | b6  | b7  |   ->   | b5  | b6  | b7  | b4  |
| b8  | b9  | b10 | b11 |        | b10 | b11 | b8  | b9  |
[ b12 | b13 | b14 | b15 ]        [ b15 | b12 | b13 | b14 ]
```

Mix-columns:
This step is basically a matrix multiplication. Each column is multiplied with a specific matrix and thus the position of each byte in the column is changed as a result.

This step is skipped in the last round.

```
[ c0 ]        [ 2  3  1  1 ] [ b0 ]
| c1 |  =     | 1  2  3  1 |    | b1 |
| c2 |       | 1  1  2  3 |    | b2 |
[ c3 ]       [ 3  1  1  2 ]    [ b3 ]
```

Add Round Keys:
Now the resultant output of the previous stage is XOR-ed with the corresponding round key. Here, the 16 bytes is not considered as a grid but just as 128 bits of data.

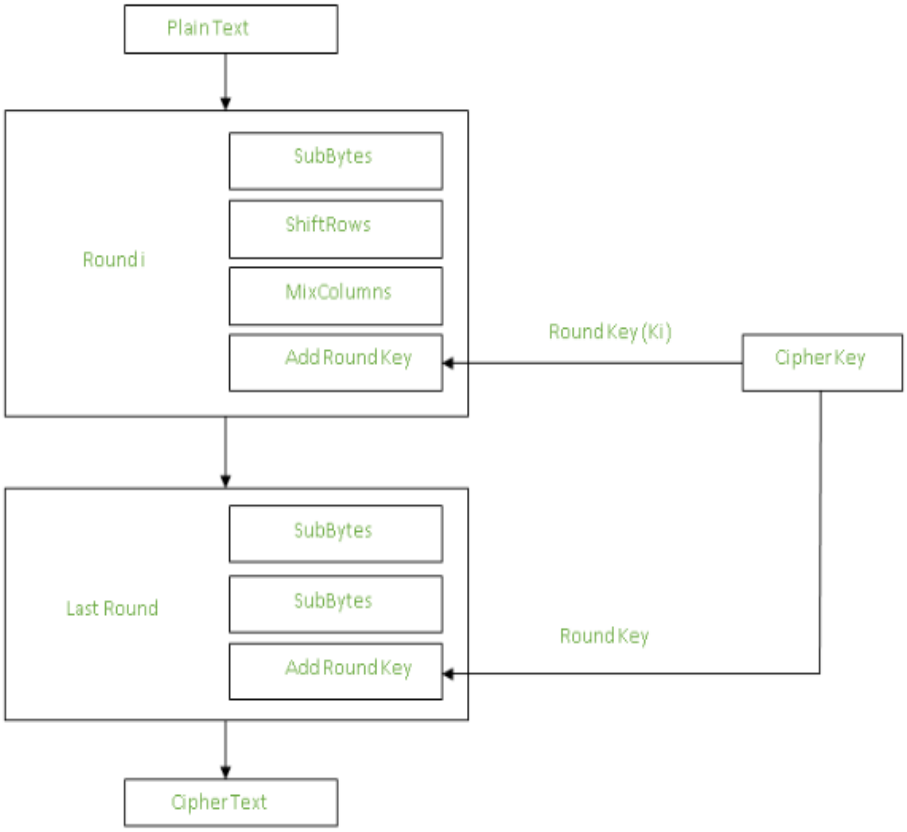

Figure 3: Add Round Keys

After all these rounds 128 bits of encrypted data is given back as output. This process is repeated until all the data to be encrypted undergoes this process.

Decryption:

The stages in the rounds can be easily undone as these stages have an opposite to it which when performed reverts the changes.Each 128 blocks goes through the 10,12 or 14 rounds depending on the key size.

The stages of each round in decryption is as follows:

- Add round key
- Inverse MixColumns
- ShiftRows
- Inverse SubByte
- The decryption process is the encryption process done in reverse so i will explain the steps with notable differences.

Inverse Mix Columns:

This step is similar to the Mix Columns step in encryption, but differs in the matrix used to carry out the operation.

$$\begin{bmatrix} b0 \\ b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} 14 & 11 & 13 & 9 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{bmatrix} \begin{bmatrix} c0 \\ c1 \\ c2 \\ c3 \end{bmatrix}$$

Inverse Sub-Bytes:

Inverse S-box is used as a lookup table and using which the bytes are substituted during decryption.

Summary:

AES instruction set is now integrated into the CPU (offers throughput of several GB/s)to improve the speed and security of applications that use AES for encryption and decryption. Even though its been 20 years since its introduction we have failed to break the AES algorithm as it is infeasible even with the current technology. Till date the only vulnerability remains in the implementation of the algorithm.

3. RESULTS

- **Register Pages:**

The registration page of the proposed intelligent data protection system is designed to ensure secure on boarding of users into the encrypted relational database environment. It serves as the first layer of defense by collecting essential user information while applying strong security controls during the account creation process.

Figure 4: Register Page

- **File Uploaded**

The file upload module of the proposed intelligent data protection system is designed to ensure secure storage and management of external data files within the encrypted relational database environment.

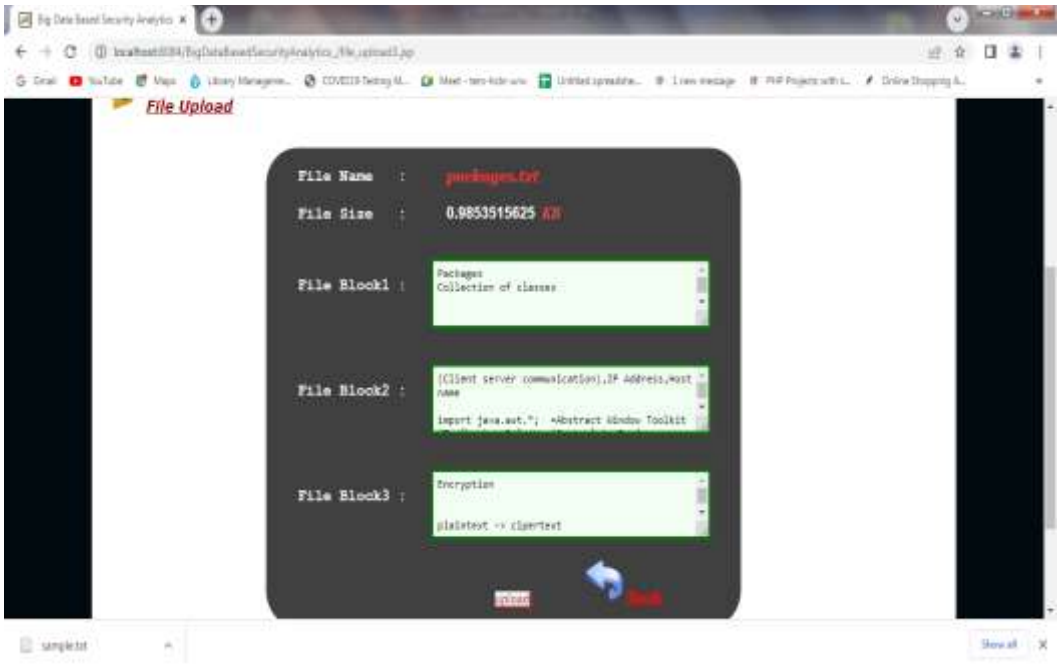

Figure 5: File Uploaded

- **Encrypted**

The encryption module is the core component of the proposed intelligent data protection model, designed to ensure confidentiality, integrity, and secure processing of data within relational database systems. This module integrates directly with the SQL query processing layer, enabling automatic encryption of sensitive data before it is stored in the database.

Figure 6:Encryption output

- **Decrypted**

The decryption module is an essential component of the proposed intelligent data protection model, responsible for securely converting encrypted data back into its original readable form for authorized users. This module operates at the application layer and is activated only after successful user authentication and authorization verification.

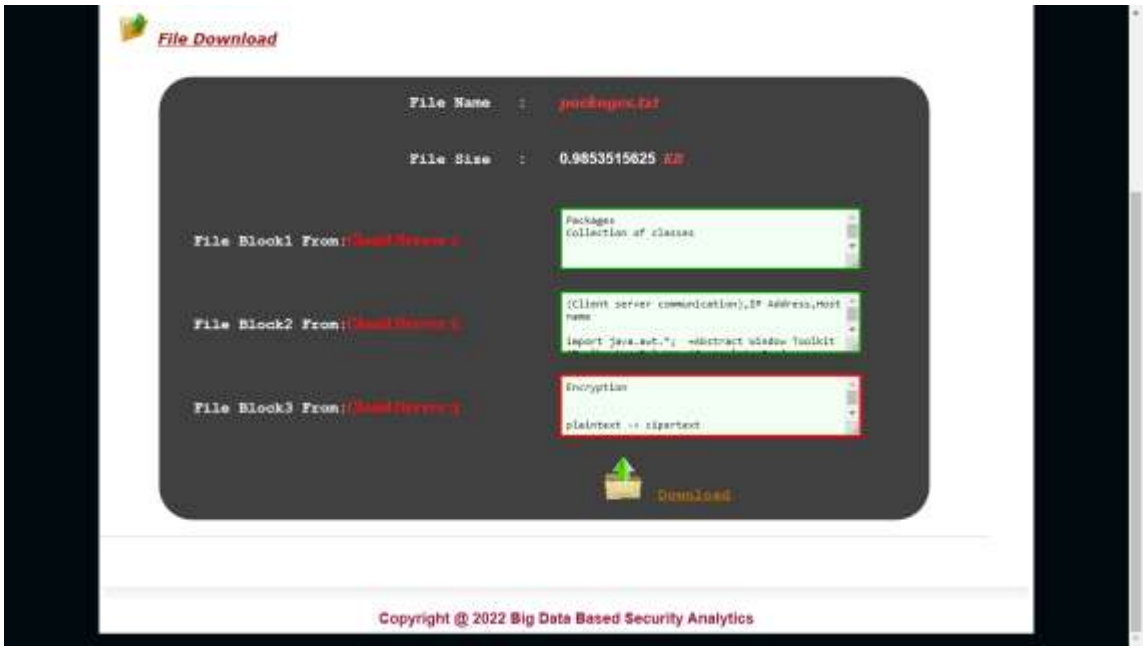

Figure 7: Decrypted Output

IV. Discussion And Conclusion

The Big Data Based Security Analytics for Protecting Virtualized Infrastructures in Cloud Computing" was successfully completed and then tested. Almost all the objectives of the project were meet and thereby a trial run of the project giving sample data gives good result.This system has been developed to reduce the manual work and to reduce the waste of time. This system can adopt any changes in the future also.

Thus this software is developed successfully to fulfill the objectives and satisfies the requirements of the user. And also this is efficient, reliable and secure to use and work.

In this paper, we analyze the existing solutions for protecting data confidentiality and user privacy in DAS. The recent and well-known proposal of all. It is expressive but cannot defeat against the statistical attacks of the untrusted server, which may violate the data confidentiality and the user privacy.

References

[1]. "A Generic Data Privacy Approach for Relational Databases and Data Warehouses" – Rahul Kumar Sharma & Vivek Kapoor (2023).
[2]. "J. H. Yu and Z. M. Zhou, Components and development in big data system" - J. Electr. Sci. Technol., vol. 17, no. 1, pp. 51–72, (2019)
[3]. "Experiments & Analysis of Privacy-Preserving SQL Query Sanitization Systems" – Loïs Ecoffet, Veronika Rehn-Sonigo, Jean-François Couchot & Catuscia Palamidessi (2025).
[4]. "CryptDB: Protecting Confidentiality with Encrypted Query Processing" – Raluca Ada Popa, Catherine M. S. Redfield, Nickolai Zeldovich & Hari Balakrishnan (2011).
[5]. "Unstructured data big data using map reduce"- V. Subramaniyaswamy, V. Vijayakumar, R. Logesh, and V. Indragandhi, vol. 50, pp. 456–465, (2015).
[6]. "Experiments & Analysis of Privacy-Preserving SQL Query Sanitization Systems" – Loïs Ecoffet, Veronika Rehn-Sonigo, Jean-François Couchot & Catuscia Palamidessi (2025).
[7]. "Executing SQL over encrypted data in the database-service-provider model" -Hacigümüs, H., Iyer, B.R., Li, C., Mehrotra,S.In: SIGMOD, pp. 216–227 (2002).
[8]. "Fuzzy C-Means Algorithms for Very Large Data Sets" - T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall & M. Palaniswami (2012).
[9]. "Secure and Efficient Models for Retrieving Data from Encrypted Databases in Cloud" – Sultan Ahmed A. Almakdi (2020).
[10]. "The rise of 'big data' on cloud computing: Review and open research issues" I. A. T. Hashem, I. Yaqoob, A. Gani, and S. U. Khan, Inform. Syst., vol. 47, pp. 98–115, (2015**).**