MongoDB with Encryption Techniques

Ch. Srilatha[1], **P. Asha**[2]

[1, 2]Assistant Professor, Dept. of Computer Science, Dr. Lankapalli Bullayya College, Visakhapatnam, India

Abstract: ***In order to provide high security for the confidential data, there is a need for proper encryption techniques that are to be followed by the concerns. This paper presents an analysis of the various encryption algorithms and their performance on handling the private data with authentication, access control, secure configuration and data encryption. It consists of the enhancement of the MongoDB level based access protected model along with privacy keys for security and monitor. The, NoSQL data stores, namely highly compress data on non-relational database management systems, which provides data management of internet user program, still do not provide support.***

Keywords: Mongodb, data stores, NoSQL, Encryption algorithms, data security

1. Introduction

MongoDB is documents based databases. Different from other relational databases, arbitrary type data can be stored in a document in MongoDB. However, existing MongoDB products provide poor privacy and security protection. In this, we proposed a privacy access policy, by taking some credential from user and encrypting it which guarantee strong security for user sensitive information and high performance in MongoDB. So data security has become one of the key requirements for all the users who share their data on any media. The technology behind information security in various fields such as computer science, information technology and e-commerce is cryptography. Cryptography is the art of combining some input data, called the plain text with a user defined password or key to generate an encrypted output, called the cipher text. The key is a sequence of symbols that controls the cryptographic operations such as encryption, decryption and signature generation or verification .

1.1 Encryption Techniques

In order to provide secure data transmission, various security algorithms are used along with the information that is transferred. Encryption is the process of converting plain text into unreadable cipher text format by applying some mathematical transformation techniques. The real security exists on the secrecy of the key rather than the encryption algorithm used according to the Kircchoff's statement. In this paper, we have analyzed the various encryption algorithms available to establish a confidential data transmission such as DES, triple DES, RSA, AES, ECC, BLOWFISH AND RC5 algorithms. Among these encryption algorithms RSA and ECC are asymmetric key algorithms and the remaining are symmetric key cryptographic algorithms. In symmetric algorithms, both sender and receiver share the same key for encryption and decryption whereas in asymmetric key algorithms, two keys are used, public key for encryption and private key for decryption.

1.2 Analysis and Design

Use of query optimizer to select the final recommended indexes. Our approach to create virtual indexes which removes any modification in the database. Applying the approach to a document-based NoSQL database. The typical setting involves two user: one that gets information from the other that is either to share the requested information. Finally there is a conflict between information sharing and privacy. Whereas the, sensitive data needs to be kept confidential as the owners may be willing, or forced, to share inform. The general approach to the rule of privacy-aware access control into NoSQL data stores a very important goal. Users can to execute for access purposes for which they have a proper authorization. Purpose authorizations are granted to users as well as to roles. In the MongoDB data storage and network transfer type for documents, simple and fast.

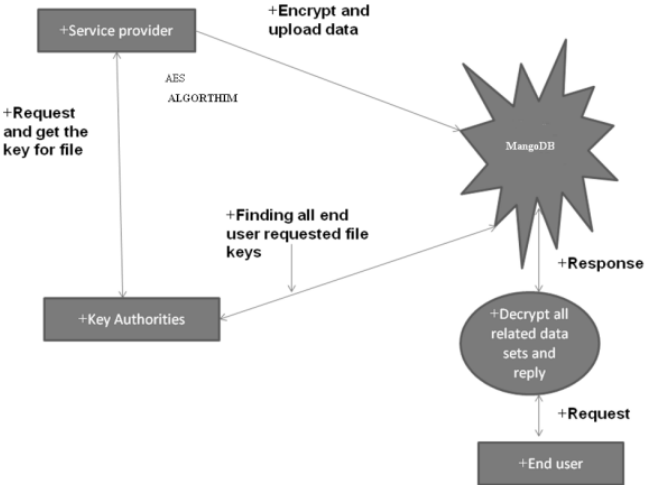
+Service provider
+Encrypt and upload data
AES ALGORTHIM
+Request and get the key for file
MangoDB
+Finding all end user requested file keys
+Response
+Key Authorities
+Decrypt all related data sets and reply
+Request
+End user

Figure 1: Data Encrypted Key System

2. Implementation Details

2.1 Steps involved in MongoDB

Mongo DB stores its data in BSON. The server has many databases, for each database has a many of collections. They are like tables in a relational store. We only need a single collection to model our data. If we were to query the Post collection from the shell (after inserting some data), we'd see BSON come back representing our data.

Data flow

Step 1: Start mongo server from command prompt, go to bin directory where the mongo server start the port. Then the monog.exe will start the mongo server.

Step 2: At second step client will log using user id and password in the system and authenticate itself. Application server checks the client is authorized or not and grant permission to the client to access the database.

Step 3: The step 3 provide two types of access from where we can upload image with access control and other types of file. This also gives the access to admin panel.

Step 4: For image uploading the required parameter is taken from client and by applying algorithm the encrypted key isgenerated. Then file breaks into chunks and stored in mongo server.

Step 5: For insertion operation, application server store the encrypted key for data into one collection of database and retrieve the encrypted key for data from another collections from mongo database.

Step 6: For the other file format same steps are repeated but these are directly converted to document type.

2.2 Encryption procedure

On the one hand, sensitive data needs to be kept confidential; on the other hand, data owners may be willing, or forced, to share information. The general technique to the rule of security is very important objective in NoSQL data stores

Jaishree Singh et al. [3] proposed a technique to secure data or message with authenticity and integrity. In their work, the secret message is encrypted before the actual embedding process starts. The hidden message is encrypted using tiny algorithm using secret key and DCT technique is used for embedding and extracting file.

Er. ManpreetKauret al. [4] discussed on the different encryption algorithms like RSA (Rivest Shamir and Adleman) Algorithm, Digital Signature Algorithm, Diffie–Hellman Algorithm, Data Encryption Standard and AES (Advanced Encryption Standard). The main security goals like confidentiality, integrity, authentication and non-repudiation are achieved by following some real time encryption techniques. Each technique has its own applications and might be suitable for the particular environment in order to have high rate of security.

The general technique to the rule of privacy-aware access control is very important objective in NoSQL data stores. Gurpreet Singh et al. [11] emphasize the importance of securing data during transmission and studied the encryption algorithms like, RSA, DES, 3DES andAES. They described that each algorithm is unique in its own way and may be suitable for different applications. They compared these algorithms in terms of speed, time, and throughput and avalanche effect; found that AES algorithm is most efficient and also suggested to use more than one algorithm to have secure transmission of data. For the future work, they recommend a combination of algorithms in sequential or parallel in order to have more secure environment for data storage and retrieval

3. NOSQL Database Encryption

3.1 Security for NOSQL databases

To improve security controls of various NoSQL databases, the sharding architecture of various existing databases namely, MongoDB, Redis, HBase, Cassandra, CouchDB and Couch base were compared on the basis of defined assessment criteria.

Anam Zahid et al. Analysed the various security features offered by NoSQL databases and proposes an assessment criterion which comprises of various security features. To improve security controls of various

The authors compared the NoSQL databases on the assessment criteria based on authentication, Access Control, Secure configuration, data encryption and auditing. They found from their analysis that all the assessment criteria provides low and medium support except access control is the only high factor that is provided by MongoDB

3.2 MONGODB Server and Security

MongoDB server front-end interacts, through message exchange, with multiple MongoDB clients. Mem operates as a proxy in between a MongoDB server and its clients, monitoring and possibly altering the flow of messages that are exchanged by the counterparts. In case the intercepted message encodes a query, it writes it in such a way that it can only access documents for which the specified policies are satisfied. The integration of data into a MongoDB deployment is straightforward and only requires a simple configuration. No programming activity is required to system administrators. Additionally, Meme has been designed to operate with any MongoDB driver and different MongoDB versions. The experiments conducted on a MongoDB server of realistic size have given a low enforcement overhead which has never compromised query usability

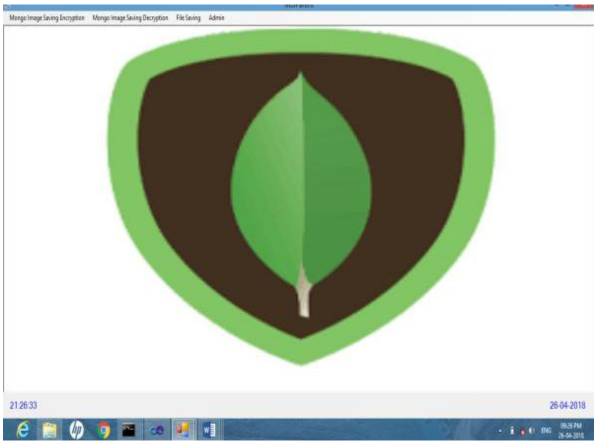

Figure 2: Multiple Document Interface

Saurabh Singh et al [14] discussed on the security vulnerabilities of Mongo DB databases and proposed some cryptographic techniques using elliptic curve and RSA for data encryption and decryption to reduce the security breaches. They introduced a hybrid protocol architecture, in which the client requests server authentication, then SSH protocol uses RSA for authentication and in parallel it uses ECC to provide integrity and confidentiality for the transaction of the data. Performance of ECC depends on efficient computation which is known as elliptic curve discrete logarithm problem or scalar multiplication. The new security protocol has been designed for better security. It is a combination of both the symmetric and asymmetric cryptographic techniques. The protocol provides three cryptographic primitives such as Integrity, Confidentiality and Authentication. These three primitives can be achieved with the help of ECC, Dual-RSA and Message Digest MD5 and from their results; the time required for encryption/decryption of ECC is less than RSA and its improved version. They have also used the ECC technique to create the secure shell while transmission of data in communication channel. ECC also provide confidentiality and integrity. Elliptic curves are believed to provide good security with smaller key sizes, something that is very useful in many applications.

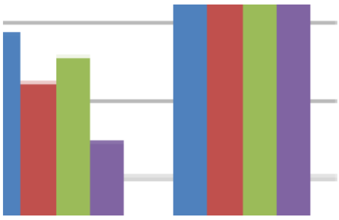

Figure 3: Decryption time of ECC and RSA with its improved version [13]

3.3 Open source Encryption

Preecha Noiumkar et al [15] compared the security level of the top 5 open source databases namely, MongoDB, Cassandra, CouchDB, Redis and Hypertable. They were compared on the security issues namely; data file encryption, client/server authentication/encryption, inter cluster encryption and script injection and Denial of Service attacks. They found from their research that MongoDB, CouchDB and Cassandra were the databases that are safe from data capturing and sniffing during the communication from the servers and Redis and Hypertables are safe from attacks that are launched by internet users as shown in Table 2. The researchers found that all these databases do not perform data file encryption and have suggested some useful methods like encrypting sensitive data in the application level by creating stunnels for making safer communication.

Sr. No.	Heading1	Heading2	Heading3	Heading 4	Heading5

3.4 Mongo DB Encryption

In Mongo DB Denial of Service (DoS) attacks are the only attacks that are not possible in the database. It is suggested that before recoding the sensitive data like passwords and credit card numbers, the application must perform data encryption at the application level itself in order to protect the data from hacking. Running MongoDB in standalone mode or replica-set mode is more secured than in shared mode because the authentication with pre-shared secret is activaHowever, hackers with an access to the system files can crack the pre-shard secret. Thus, to make the key file more secured, permission in an OS level should be determined to suit the key file (e.g. using chmod command). In the same way, as per the author’s statement, we can prevent this kind of attack on MongoDB by terminating the following symbols: (:), ({), and (}), in order to stop the attacking input from getting into the web server, which is the frontage of the database server. For these reasons, the developers should write an extra script to detect and delete these extra symbols before they can get into the database.
Matthew Trudeau et al [16] discussed the probability of hacking in NoSQL database technologies. They focus on Mongo DB database and their security features that are built in including authorization, authentication and TLS/SSL encryption. They insisted the importance of using the built in security features otherwise major security risks will be attempted on the confidential data as attacked in January 2017.

3.5 Mongodb Features Encryption standards

Mongo DB (version 3.4) has built in features in order to provide authentication, authorization, encryption, auditing, network exposure, injection prevention etc., but all these features are not effective when they decrease the speed of database

Mongo DB supports two type of encryption standards namely, AES256-CBC, which is the Advanced Encryption Standard running in Cipher Block Chaining mode. Additionally, MongoDB supports AES256-GCM, which is known as Galois/Counter Mode. Master keys and database keys are used to encrypt, The data within the database is encrypted using the database keys, and the database keys are in turn encrypted with the master key. The authors have also analyzed that MongoDB does not offer any in-house features for application level encryption. To encrypt each field or document, MongoDB documentation suggests writing a custom encryption/decryption methods or using solutions created by one of their partners [18]. MongoDB also supports transport encryption, such as TLS/SSL, to encrypt network traffic. The implementation of TLS/SSL makes use of Open SSL libraries, only using SSL ciphers that use a key that is at least 128-bit in length.

Kusum Kakwani et al [19], in their work presented an enforcement monitor called Mem (MongoDB enforcement monitor) to implement security by acting as a proxy between the MongoDB user and server and enforce access control. P. R. Hariharan et al [20] proposed a survey on various schemes for database encryption and the future need for the complete solution of providing better secured environment for the data transmit

4. Results and Discussion

When the first time user interacting with the system this window will appear A Multiple Document Interface (MDI) programs can have many child windows inside them. While in single document interface (SDI) applications, one document at a time can manipulate. Notepad is an example of an SDI application and visual Studio is an example of Multiple Document Interface (MDI). MDI applications have a Window menu item for switching between windows or documents.

After the data has been saved, the next step is to read data from the database table, save it as a bitmap again, and view the bitmap on the form. By using the Graphics we can view an image. Draw Image method or using a picture box

This paper also analyzes the various NoSQL databases like MongoDB, Cassandra, Redis, CouchDB, Hypertable etc. Since they follow unstructured format of data which are available in the form of documents, emails etc. Most of the NoSQL databases are susceptible to external security attacks by the intended or unintended intruders and found to be weak in some aspects like authentication, script injection, DoS attacks etc. Almost all the NoSQL databases do not follow proper encryption/decryption techniques to authenticate the user data which is at rest or in transit. From the survey, it is clear that there is a serious need to handle the confidential data safely without any loss in transit or susceptible to sniffing or injection attacks by providing a suitable secured environment by following encryption and compression techniques. When using MongoDB database, implementing SHA-3 algorithm or using enterprise edition are possible options to avoid hacking. Application level encryption must be implemented to avoid interception of data. There are certain suggestions to encrypt all the fields and follow the best practices in order to provide safe environment to the database users and implementing all built-in security features is a must for any successful database. Security attacks that are occurred in early 2017 are due to questionable selection of default settings and not following the best practices like opt-out.

5. Conclusion and Scope

The Purpose concepts and related give mechanisms to regulate the access at document level on the basis of purpose and key based policies. An enforcement monitor, has been designed to implement the proposed security. It operates as a between MongoDB user and a MongoDB server, and enforces access control by monitoring and possibly manipulating the flow of exchanged messages. Furthermore, we plan to generalize the presented approach to the support for multiple NoSQL data stores. Enhancing data security is one of the important aspects of the transmission of data among the users. While the usage of data in unstructured format has been increased in various fields, there is a significant need for providing access criteria such as authentication, access control, data encryption, secure configuration and auditing. Data security is provided by proper encryption methods of various important fields of data without affecting the performance of the database in the aspects of speed and memory usage. Since the usage of document oriented, unstructured data are often handled by the NoSQL databases such as MongoDB, Cassandra, CouchDB, Redis, Hypertable etc. Due to their open source nature, there is a serious requirement in providing high security and safeguard the user’s confidential data without any tampering during at rest or in transit. There is a demand for a single solution to enhance the secure transmission of data by providing improved encryption method that minimizes the process delay and memory usage in handling databases.

References

[1] K. Browder and M. A. Davidson. The Virtual Private Database inOracle9iR2. Technical report, 2002. Oracle Technical White Paper.

[2] J. Byun and N. Li. Purpose based access control for privacy protection in relational database systems. The VLDB Journal, 17(4), 2008.

[3] R. Cattell. Scalable SQL and NoSQL Data Stores. SIGMOD Rec., 39(4):12–27, May 2011.

[4] Gurpreet Singh, Supriya Kinger, “Integrating AES, DES, and 3-DES Encryption Algorithms for Enhanced Data Security”, International Journal of Scientific & Engineering Research, Volume 4, Issue 7, July-2013, pp- 2058-62

[5] Jaishree Singh, Dr. J.S. Sodhi ,”Secure Data Transmission using Encrypted Secret Message”, International Journal of Computer Science and Information Technologies, Vol. 4 (3) , 2013, pp-522-525.

[6] Symmantec, “White paper: Keeping your private data secure”, http://www.symantec.com/encryption.

[7] 6. Er. ManpreetKaur , Er. Jasjeet Kaur, “Data Encryption Using Different Techniques: A Review”, International Journal of Advanced Research in Computer Science, Volume 8, No. 4, May 2017 (Special Issue),pp-252-255.

[8] 7. ArpitAgrawal, Gunjan Patankar , “Design of Hybrid Cryptography Algorithm for Secure Communication”, International Research Journal of Engineering and Technology (IRJET, Volume: 03 Issue: 01 pp- 1323-28.

[9] Cavoukian. Privacy by Design: leadership, methods, and results. In S. Gutwirth, R. Leenes, P. de Hert, and Y. Poullet, editors, European Data Protection: Coming of Age. Springer, 2013.

[10] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. Bigtable: A distributed storage system for structured data. ACM Transactions on Computer Systems (TOCS), 26(2):4, 2008.

[11] P. Colombo and E. Ferrari. Enforcement of purpose based access control within relational database management systems. IEEE Transactions on Knowledge and Data Engineering (TKDE), 26(11), 2014.

[12] Nikita D. Dongare, Prof. V. T. Gaikwad, Prof. H. N. Datir, “Secure Data Transmission Scheme by Using Encryption Based Technique: A Review”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 6, Issue 3, March 2016 pp-25-28

[13] Prateek Kumar Singh, Pratikshit Tripathi, Rohit Kumar, Deepak Kumar, ” Secure Data Transmission”, International Research Journal of Engineering and Technology (IRJET), Volume: 04 Issue: 04,| Apr -2017, pp-217-222.

[14] Gurpreet Singh, Supriya, “A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security”, *International Journal of*

Computer Applications (0975 – 8887) Volume 67– No.19, April 2013, pp-33-38

[15] Anam Zahid, Rahat Masood, Muhammad Awais Shibli, “Security of Sharded NoSQL Databases:***A Comparative Analysis”,*** Conference on Information Assurance and Cyber Security (CIACS), 2014 IEEE

[16] Charmi Pariawala, and Ravi Sheth, “Encrypting Data of MongoDB at Application Level”, Advances in Computational Sciences and Technology,Volume 10, Number 5 (2017) pp. 1199-1205

[17] Saurabh Singh, Karamjit Kaur ,”Comparative analysis of ECC and RSA for Document-oriented database MongoDB”, International Journal of Computer Technology & Applications,Vol 5 (4), April 2014, pp-1555-1560

[18] Preecha Noiumkar, and Tawatchai Chomsiri, “A Comparison the Level of Security on Top 5 Open Source NoSQL Databases”, The 9th International Conference on Information Technology and Applications (ICITA2014), At Sydney, Australia

[19] Matthew Trudeau, Joshua Kolodny “An Analysis and Overview of MongoDB Security,51st Hawaii International Conference on System Sciences (HICSS 2018), Waikoloa Village, Hawaii, USA, 2 - 6 January 2018, Volume 1 of 8, ISBN: 978-1-5108-5655-4.

[20] Hou, Boyu, et al. “Towards Analyzing MongoDB NoSQL Security and Designing Injection Defense Solution.” *2017 IEEE 3rd International Conference on Big Data Security on Cloud,* July 2017.