

Encrypting Data of MongoDB at Application Level

¹Charmi Pariawala*, and ² Ravi Sheth

¹ Student, M.Tech In Cyber security, Raksha Shakti University, Gujarat, India

² Assitant Professor, Department of Information Technology and Telecommunication,
Raksha Shakti University, Gujarat, India

Abstract

In recent years, in distributed web application and cloud computing need to store large amount of data in relational and distributed database. So now days, the upcoming and growing companies moved into non-relational database like NoSQL. Due to the more use of NoSQL database much sensitive information and large amount of data are stored in NoSQL database. Because of a lack of encryption in MongoDB (open source), data and database may not more secure. This paper discusses security features of MongoDB and proposes a Encryption security features at application level with implementation. The analysis result of this implementation shows that how mongo encryption library works in php.

Keywords: MongoDB, Encryption, NoSQL, Database, Document-Oriented database

I. INTRODUCTION

In recent days in cloud computing and distributed web application need to store large amount of data in distributed databases that provides high availability and scalability. In recent year, a growing number of companies used various types of non relational databases, commonly referred as NoSQL. Different NoSQL databases take different approaches. The main advantage is that NoSQL handle unstructured database like document, email and social media efficiency.

NoSQL referred as “Not Only Structure Query Language” and “Not only SQL [3]” NoSQL is class of “schema-less” database management system that has been

designed with more relaxed data model as compared to RDBMS. There are several categories of NoSQL database. Four main are Key-values store, Column Family, Document Oriented and Graph database ^[1]. The common feature of NoSQL are summarized as: high scalability and reliability, very simple data model, very simple query language, lack of mechanism for handling and managing data consistency and integrity and almost no support for security at database level.

A. Security Issues in NOSQL Database

Given all these more chances brought by NoSQL, more and more undertakings and government agencies turn to NoSQL and increasing sensitive data are stored in these databases rightly, which takes the safety Issue of NoSQL databases into the public's attention. Currently most NoSQL databases existence without of natural to safety apparatuses.

A key feature of NoSQL is “Share nothing” horizontal scaling-replicating and partitioning data over many servers ^[2]. Due to this feature, NoSQL can support a large number of simple read/write operations per second. NoSQL systems don't provide ACID (Atomicity, Consistency, Isolation and Durability) guarantees but follow BASE. BASE is acronym for Basically Available, Soft state and eventually consistent ^[7].

In 2000, Prof. Eric Brewer introduces the CAP theorem. CAP theorem namely called as Consistency, Availability and Tolerance of network ^[4]. The main idea of CAP theorem is a distribute the system which cannot meet the three district need simultaneously, but it can be meet only two system like CA (consistency and Availability), AP (Availability and Partition Tolerance) ^[5].

B. Current Research

NoSQL database is existence without of taking care of expertly and business managers of knowledge for computers persons of representative and true, good nature and does not make ready database-level safety support, which leads to great safety dangers. Reference ^[2] discussed about NoSQL database and their security features and then also proposed a transparent middleware and its implantation which is done using JAVA class library in Linux Operating System. Reference ^[4] proposed the method of encryption, Because of the lack of encryption support at data at rest and weak authentication at database level. To overcome these issues security mechanism can be implemented at middleware layer with small changes in existing system. Reference ^[3] discussed about the different type of NoSQL and also discussed about the pros and cons of NoSQL database and also describes the advantages and disadvantages of each of the store data and cases when a particular data can be used. Reference ^[6] discussed about the storage mechanism of MongoDB.

II. PRINCIPLES

A. Analysis of MongoDB operation in php

Before Operating MongoDB, We need to achieve the connection to it. For connection, open a console and go into the MongoDB where you store your database. `mongod.exe -dbpath "h:\mongoDB\data"` and next open another console terminal and write a command like this `"h:\mongoDB\bin> mongo.exe"` MongoDB is run on its default port number 27017. In MongoDB, the records are called as documents and schemas are called as collections. To use the database, we have to write: use database name; after getting the database, we can get collection by doing: `db.createCollection()`.

BSON (Binary Serialized Document Format) is the binary form for representing simple data structure and associate arrays (called objects or documents in MongoDB). Typically a BSON is consist of keys and values like this: `{"key1": "value!", "key2": "value2",... "keyn": "value" }`.

B. Data modeling features in MongoDB

Unlike SQL database, data in MongoDB has flexible schema. For MongoDB collection, it is unnecessary to determine and declare a table's schema before inserting data. Every document can compare the data fields of the represented entity, even if data has substantial variation ^[13].

Document in MongoDB are stored in key-value pattern. Document in same collection in same collection can be in different storage areas. Values are mapped to their keys. This characteristic makes it possible that we can encrypt any dataset in a document.

C. Encryption Algorithm

Encrypting data will act on the operation and memory use, so it is important to get at the details of different algorithms and selecting the one which has least possible on having existence system. The security level of encryption algorithm is also important to consider. It is the most important parameter of cryptography because an algorithm is said to be better if they give a strong safety level. in addition limitations of the encryption algorithms is need to be consider, such as DES can be crack easily by brute force attack, IDEA and Blowfish are vulnerable to attack.

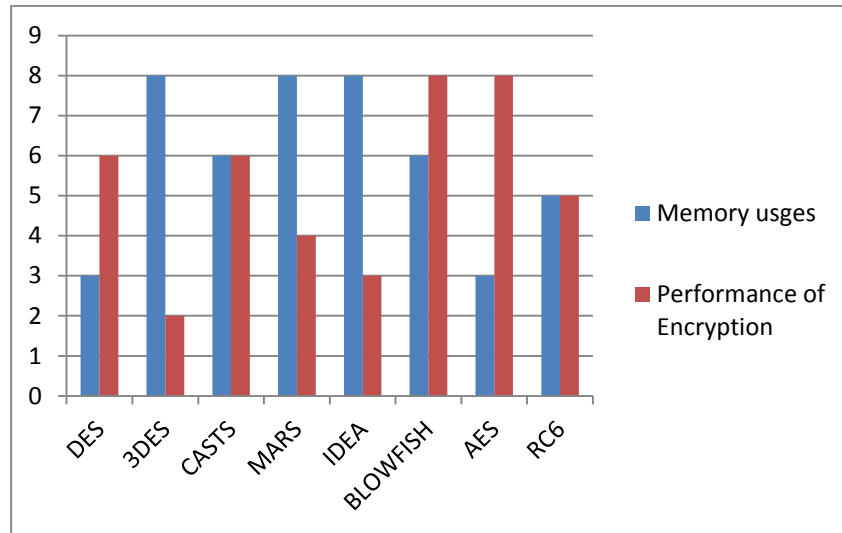


Figure 1: Comparison of Algorithm based on scalability ^[4]

As figure 1 analyzes various encryption algorithms on the basis of performance and space required by the encryption algorithm. From the figure it is clear that AES is best among all these related algorithms based on scalability. After various literature survey comparisons, it is describe that AES is secure, fast, better and effective encryption algorithm among all these encryption algorithms with less storage space, high encryption performance.

III. PROPOSED METHOD

Data flow

Step 1: At first step client will log in the system and authenticate itself using user id and password. Application server checks for the client's access permissions and grant the access of database to the client.

Step 2: After the authentication client can access the database. He can perform insert the data, update the existing data or delete the data from database. Client will send the data to application server in plain text format. And at retrieval of the data server will provide plain data to client.

Step 3: Application server then apply AES algorithm on data which is send by the client. On Client's request for data retrieval application server fetch the encrypted data from the database and decrypt that data using AES algorithm and plain data will send back to client.

Step 4: Application server store the encrypted data into specific collection of particular database on insertion operation and retrieve the encrypted data from specific collections from particular mongo database.

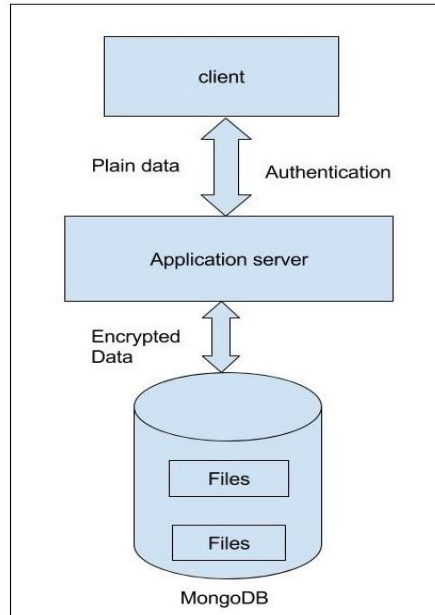


Figure 2: Data inserted into MongoDB via application Interface

IV. EXPERIMENT AND ANALYSIS

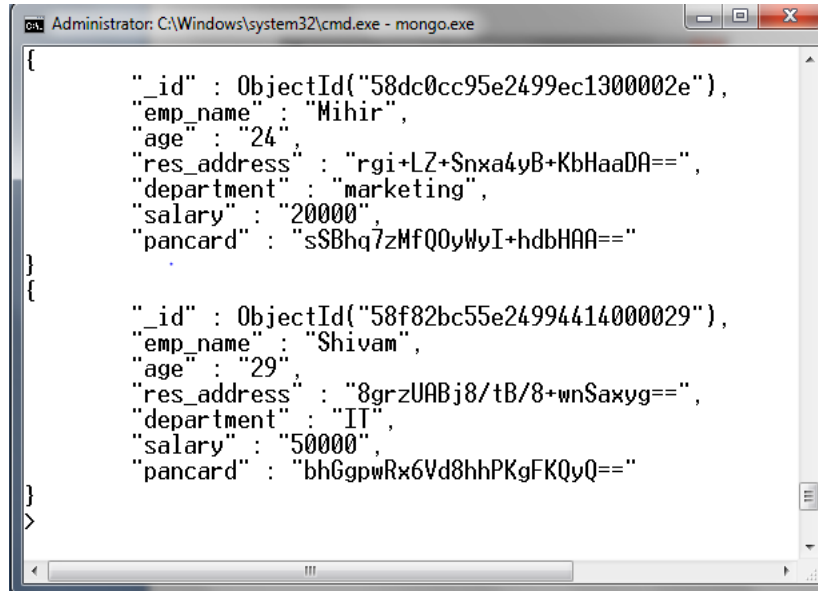
Experiment was conducted on PC with Intel i3 processor, 4GB random memory and Windows 7 64-bit operating system. From the screen shot fig. 1, we can find that some filed which are important is in encrypted form and data which are less important that are shown in plain text.

```

Administrator: C:\Windows\system32\cmd.exe - mongo.exe
> db.employees.find().pretty()
{
  "_id" : ObjectId("58cbba3c5e24993c10000029"),
  "emp_name" : "Charmi",
  "age" : "26",
  "res_address" : "rgi+LZ+Snxa4yB+KbHaADa==",
  "department" : "IT",
  "salary" : "25000",
  "pancard" : "01BV6V6fDKqparBc/v1MyQ=="
}
{
  "_id" : ObjectId("58cbba875e24993c1000002a"),
  "emp_name" : "Vishwa",
  "age" : "25",
  "res_address" : "sQsppr9zNHXmjKzSs2ZZFQ==",
  "department" : "IT",
  "salary" : "21000",
  "pancard" : "tRTh45xR00hwpmWCmLijda=="
}
{
  "_id" : ObjectId("58cbbb0f5e24993c1000002b"),
  "emp_name" : "Bhavik",
  "age" : "33",
  "res_address" : "/IXzPZxDgvv/5Xa0ia00vQ==",
  "department" : "sales",
  "salary" : "30000",
  "pancard" : "msk8q5gtfScPQcpjhFFeiQ=="
}
{
  "_id" : ObjectId("58cbbb805e24995c10000029"),
  "emp_name" : "Soham",
  "age" : "26",
  "res_address" : "rgi+LZ+Snxa4yB+KbHaADa==",
  "department" : "finance",
  "salary" : "36000",
  "pancard" : "b521hIpmiUwsHAn9oF02Rg=="
}
  
```

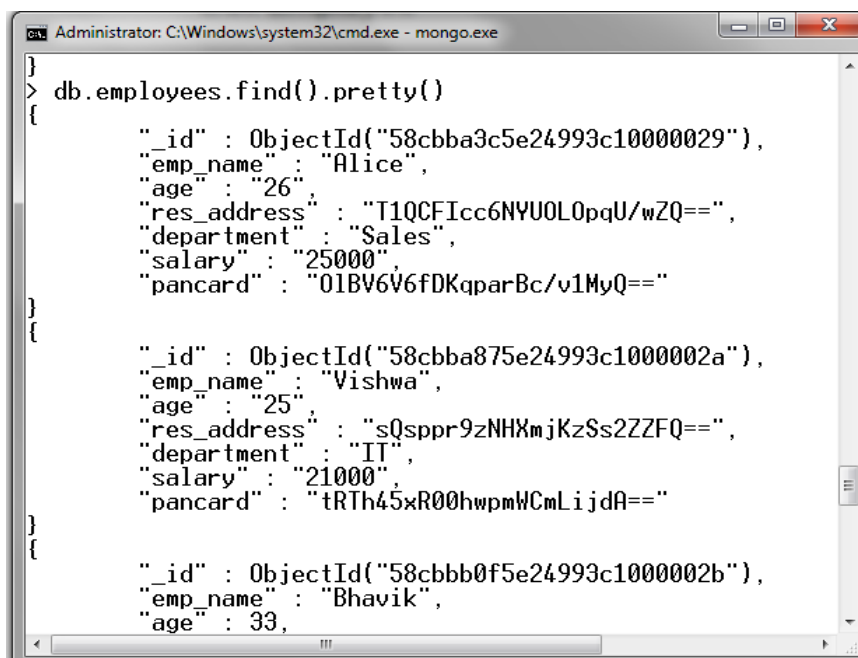
Figure 3: Important fields are in encrypted form (BSON format)

The data are inserted in collection also in encrypted form. The fig. 2, we can show that inserted some data in encrypted form. In fig. 3 we can also show updated data are also in encrypted form. Id is not change when update the data. We can verify with the help of id. We can also delete field



```
Administrator: C:\Windows\system32\cmd.exe - mongo.exe
{
  "_id" : ObjectId("58dc0cc95e2499ec1300002e"),
  "emp_name" : "Mihir",
  "age" : "24",
  "res_address" : "rgi+LZ+Snx4yB+KbHaaDA==",
  "department" : "marketing",
  "salary" : "20000",
  "pancard" : "sSBhq7zMfQ0yWyI+hdbHAA=="
}
{
  "_id" : ObjectId("58f82bc55e24994414000029"),
  "emp_name" : "Shivam",
  "age" : "29",
  "res_address" : "8grzUABj8/tB/8+wnSaxyg==",
  "department" : "IT",
  "salary" : "50000",
  "pancard" : "bhGgpwRx6Vd8hhPKgFKQyQ=="
}
>
```

Figure 4: Inserted data shown in encrypted form



```
Administrator: C:\Windows\system32\cmd.exe - mongo.exe
]
> db.employees.find().pretty()
{
  "_id" : ObjectId("58cbba3c5e24993c10000029"),
  "emp_name" : "Alice",
  "age" : "26",
  "res_address" : "T1QCFIcc6NYUOL0pQU/wZQ==",
  "department" : "Sales",
  "salary" : "25000",
  "pancard" : "01BV6V6fDKqparBc/v1MyQ=="
}
{
  "_id" : ObjectId("58cbba875e24993c1000002a"),
  "emp_name" : "Vishwa",
  "age" : "25",
  "res_address" : "sQsprr9zNHXmjKzSs2ZZFQ==",
  "department" : "IT",
  "salary" : "21000",
  "pancard" : "tRTh45xR00hwpmWCmLijdA=="
}
{
  "_id" : ObjectId("58cbbb0f5e24993c1000002b"),
  "emp_name" : "Bhavik",
  "age" : "33",

```

Figure 5: Updated first field data

CONCLUSION

In this paper, we discuss about NoSQL database and their security features. Then we propose a method to encrypt data into database and also implement its process. Experiment results confirm that if data are not store in encrypted form in MongoDB open source at database level then we can do encrypt the data at application level and protect our sensitive information of user.

REFERENCES

- [1] LiorOkman, Nurit Gal-Oz, YaronGonen, Ehud Gudes, JennyAbramov, "Security Issues in NoSQL Databases", 2011 International Joint Conference of IEEE TrustCom-11
- [2] XingbangTian, Baohua Huang,0 MinWu, "A Transparent Middleware for Encrypting Data in MongoDB", 2014 IEEE Workshop on Electronics, Computer and Applications.
- [3] Ameya Nayak, Anil Poriya, Dikshay Poojary, "Type of NOSQL Databases and its Comparison with Relational Databases", 2013 IJAIS
- [4] Karan Patel, Kirti Sharma, Mosin Hasan. "Encrypting MongoDB Data using Application Level Interface". *Discovery*, 2015, 46(214), 164-169
- [5] Jing Han, Haihong E, Guan Le, Jian Du. "Survey on NoSQL Database". 2011 International Joint Conference of IEEE TrustCom-11
- [6] Yunhua Gu¹, Xing Wang¹, Shu Shen¹, Jin Wang¹, Jeong-Uk Kim². "Analysis of Data Storage Mechanism in NoSQL Database MongoDB", 2015 IEEE
- [7] Rupali Arora, Rinkle Rani Aggarwal, "Modeling and Querying Data in MongoDB", IJSER-2013
- [8] [OL]<http://forums.asp.net/t/1205083.aspx?database+level+security+Vs+application+level+security>
- [9] [OL] <http://blog.websecurify.com/2014/08/hacking-nodejs-and-mongodb.html>
- [10] [OL] <http://blogs.perficient.com/dataanalytics/2015/06/22/nosql-nosecurity-security-issues-with-nosql-database>
- [11] [OL] <https://www.idontplaydarts.com/2010/07/mongodb-is-vulnerable-to-sql-injection-in-php-at-least/>
- [12] [OL] <https://scalegrid.io/blog/10-tips-to-improve-your-mongodb-security/>
- [13] MongoDB [OL] <http://www.mongodb.org/>

