Security Implementation of Unstructured Data in NoSQL MongoDB
Deepak Gurmani, Prof. Mona Mulchandani
CSE Dept.
Jhulelal Institute of Technology
Nagpur, 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 However, traditional relational databases cannot address users' demands for quick data access and calculating and retrieval of information, since they cannot process data in a distributed way. To solve this problem, non-relational databases such as MongoDB have emerged up and been applied in various Scenarios. Nevertheless, it should be noted that most MongoDB products fail to consider user's data privacy. In this paper, we propose a practical encrypted MongoDB (i.e., CryptMDB). Specifically, we utilize an additive homomorphic asymmetric cryptosystem to encrypt user’s data and achieve strong privacy protection. Security analysis indicates that the CryptMDB can achieve confidentiality and privacy of user’s data and prevent adversaries from illegally gaining access to the database. Furthermore, extensive experiments demonstrate that the CryptMDB achieves better efficiency than existing relational database in terms of data access, retrieval and calculating.

Keywords— MongoDB, Big Data, Database Security, Privacy Protection, Additive Homomorphism

I. INTRODUCTION
With the rapid development of network and information technology, data are woven into every corner of our lives, such as websites, social networks, public platforms, etc. Along with the convenience and immediacy of data acquisition, big data technologies, and many other new data analysis tools [1], [2] have been prevailed and applied in various fields of people’s lives, which have generated enormous social impact and economic benefits [3], [4].

Generally, data should be stored in databases for easy access and utilization. Existing mainstream databases adopted by enterprises and individuals are relational databases, such as MySQL, Oracle, DB2, etc., in which data are stored as a item of tables and participated various Sql requests. However, with the data explosion, especially in big data era (data are growing at an alarming rate every day), traditional relational databases cannot address users’ demands for quick data access and calculating, since they cannot process data in a distributed way. To tackle this problem, MongoDB, a new non-relational database, which has been favored in more and more enterprises and individuals due to its strong ability in distributed data processing. It is generally known that MongoDB is one kind of non-relational databases based on documents. Different from other relational databases, arbitrary type data can be stored in a document in MongoDB. Therefore, compared with relational databases, data can be more easily stored on other servers. As a result, data can be processed in distributed servers for large-scale access and calculation tasks.

However, existing MongoDB products nearly fail to consider an urgent and practical issue in databases, i.e., privacy protection. It is widespread that data is stored without any safety measures on commonly used databases, which is vulnerable to attackers who are interested in users’ sensitive information if adversaries can compromise databases to steal private data. Besides, MongoDB server is suspected as honest but curious, which may malicious peep data stored in databases due to it has the full access permission
Therefore, it is urgent to propose a privacy-preserving approach which can ensure confidentiality of users’ information on MongoDB. In the last few years, several encryption schemes have been applied in relational databases. Raluca et al. design a CryptDB system in MySQL, which uses an onion encryption structure to support 99.5% operations over encrypted data. Deshmukh et al. propose a transparent data encryption scheme to provide high levels of security for columns, table and tablespace in Microsoft SQL Server 2008. Then Raluca et al. present an ideal-security protocol for order-preserving encoding over relational databases, which not only can provide ideal security but also demonstrate the higher performance comparing with previous order-preserving approach. However, few specific encryption tools which have been applied in non-relational databases. In this paper, we propose a practical encrypted MongoDB, i.e., CryptMDB, which can guarantee strong privacy protection and high performance in non-relational databases. In specific, the contributions of this paper can be summarized as follows:

- We leverage an additive homomorphic asymmetric cryptosystem to design an encrypted MongoDB (i.e., Crypt-MDB), which can achieve additive operations over en-crypted data.
- Security analysis shows that the CryptMDB can achieve strong privacy protection of users’ information stored in databases. Besides, extensive experiments indicate that the CryptMDB is better than existing relational database (such as MySQL) in terms of data access and calculating.
- The remainder of this paper is organized as follows. In Section II, we will describe the preliminaries. In Section III, we will propose a practical encrypted MongoDB and describe the details of our model. Then we carry out the security analysis and performance evaluation in Section IV and Section V, respectively. Finally, Section VII concludes the paper.

II. METHODOLOGY

In this section, we will introduce the CryptMDB architecture and analyze threats of CryptMDB. Besides, encrypted tool also will be brought in this part, which will be served as the basic of our proposed scheme.

A. CryptMDB Architecture

As shown in Fig.1, CryptMDB mainly contains three parts: User’s computers, CryptMDB proxy server and MongoDB server. Firstly, data provided by users will be encrypted by encryption tools and stored in MongoDB. When users want to query the contents of database, they should send some specific MongoDB query languages (Mql) to CryptMDB proxy server. Then these Mql queries will be rewritten by pre-set encrypted tools and sent to the MongoDB server. Next, the MongoDB server executes Mql to match corresponding ciphertexts which will be delivered to CryptMDB proxy server. Finally, the proxy server decrypts these ciphertexts and sends them to authorized users.

We can see that in CryptMDB where MongoDB server executes Mql queries and return corresponding ciphertexts to users, it cannot gain access to the sensitive data of users, which ensures that user’s private information cannot be leaked to any part in whole CryptMDB architecture. Besides, in CryptMDB, different users have their own disparate key to encrypt personal information. Therefore, even if the attackers full control the CryptMDB, they cannot get private data whose owner are not log in the CryptMDB systems. In this paper, although CryptMDB can protect the data confidentiality, it does not guarantee the data completeness, freshness, integrity and so on. Moreover, other attacks such as compromise user’s computers, gain user’s key, or a malicious DBA, are not the scope of our CryptMDB.

B. Threat 1: MongoDB Compromise

For the threat 2, when the MongoDB server and proxy server are compromised by attackers, they can use the proxy server to encrypt ciphertexts returned by MongoDB server and get the plaintexts. For this case, we adopt different keys to encrypt each user’s information in CryptMDB. In this way, user’s keys only be activated by user logged in MongoDB at that time. Thus, although the proxy server and MongoDB server have been compromised by attackers, they only can steal the information from current users, and other user’s data (which are not logging in MongoDB) do not reveal to any attacker.

III. RELATED WORK

With information explosion and development of network and information technology, encrypted database, which is designed to achieve user’s data privacy protection has drawn more and more attention [11]–[14], and has been proposed in various application scenarios. For example, Raluca et al. [7] design a CryptDB system based on MySQL, which uses an onion encryption structure to support 99.5% operations over encrypted data. Deshmukh et al. [8] propose a transparent data encryption scheme to provide high levels of security for columns, table and tablespace in Microsoft SQL Server 2008. Then Raluca et al. [9] present an ideal-security protocol for order-preserving encoding over relational databases, which not only can provide ideal security but also demonstrate
the higher performance comparing with previous order-preserving approach.

We know that user’s privacy have been proposed in many fields [15]–[18], also including relational database. However, few specific encrypted tools which have been applied in non-relational database. Recently, Tian et al. [19] proposes a transparent middleware implementation in MongoDB, which can efficiently encrypt sensitive data specified by users on a dataset level, however, it cannot achieve large scale aggregation operation. With the data explosion, especially in big data era (data is growing at an alarming rate every day), due to users’ demands for speed of data access and calculating [20] – [22]. Therefore, it is urgent to propose a privacy-preserving approach which can ensure confidentiality of users’ information over non-relational database. In this paper, we propose a practical encrypted MongoDB, i.e., CryptMDB, which can guarantee strong privacy protection and high performance in non-relational database. Furthermore, extensive experiments demonstrate that the CryptMDB is better than existing relational database in terms of data access and calculating.

IV. CONCLUSION

In this paper, we propose a practical encrypted MongoDB (i.e., CryptMDB) to achieve the privacy protection of user’s data stored in database. The key idea of the CryptMDB is utilizing an additive homomorphic asymmetric cryptosystem to encrypt user’s data. Security analysis demonstrates that the cryptMDB can achieve strong privacy protection for user’s data and prevent adversaries from illegally gaining access to the database. Furthermore, through the theoretical analysis and extensive experiments, we show that performance of the CryptMDB is better than relational database (such as MySQL) in terms of data access and calculating.

V. ACKNOWLEDGMENT

I would like to express my deep gratitude to my guide Prof. Mona Mulchandani for their valuable guidance and encouragement.

I would also like to sincerely thank all the faculty members of CSE Department and my colleagues for helping me throughout this project.

REFERENCES


