A Framework for Securing Data at Rest in Big Data Domain

Salman Abdul Moiz

Abstract—Data is growing exponentially due to enormous use of smart devices, internet and social media. Big data is huge collection of data sets which is of the order of petabytes and exabytes. Since the traditional databases systems are not effective in managing huge voulmes of data, it is often stored and mainained by cloud service providers. Security during transmission of data is guaranteed using SSL (Secure Socket Layer). However, the issue is with securing the data at rest which is currently trusted by Service Level Agreement (SLAs). Encrypted database concepts were applied in cloud environment to transform the plain query into encrypted query and getting back the encrypted results. However, these solutions directly don't apply to big data domain. The Volume, Velocity, Variety and Veracity in big data domain adds extra challenges to existing encrypted cloud database solutions. The article presents a framework for securing data at rest in big data domain. The architecture of the proposed framework can be seen as an enahancement to the framework of of encrypted databases in cloud domain.

Index Terms—Big data, Cloud environment, Data at rest, Encrypted databases

1 Introduction

According to ISO 2015[3], the big data is characterized by 5V's. Data that is coming from several sources is huge (Volume) and often arrives at high speed (Velocity), which is of dierse types (Variety) with noise (Veracity) and which changes too fast (Variability). Traditional database systems can't manage data with these characteristic. Managing big data and realizing today's threat environment is a challenging issue.

The huge volumes of data are often stored on the cloud environment which is often untrusted as the key management is also realized by the service providers. The data which is stored on a device or backup medium in a digital form is often referred to as data at rest. Several solutions to secure data at rest using encrypted database mechanisms are proposed by researchers. These mechanisms often work well for the structured data. In which the request for data retrieval is sent to cloud service provider as an encrypted structured query. However the other dimensions such as Variety, Velocity, Veracity etc., requires addition of new wrappers to existing solution of data at rest in cloud environments. In this paper a framework to realize few charateristics of the big data environment is presented. The strategies that can be applied for key management is presented and certain open issues and challenges are discussed.

The remaining part of this paper is organized as follows: Section-2 gives a start of art of security in big data environments; section-3 presents a framework for securing data at rest in big data environments. Section-4 highlights the open issues and challenges and section-5 conclues the paper.

2 RELATED WORK

According to Trend Micro [7] todays threat environment deals with how security vendors manage threats in presence of Volume, Variety and Velocity of the data. From the year 1990 to 2010 the amount of spam messages increased from 2 to 200 billion per day. The malwares detected only in January 2008 were more than all malwares reported prior to 2008. With the increase of social media tools the volumes of data have increased exponentially. Since data is streamed from various devices at the same time, velocity becomes a challenging issue. The data generated can be in varied formats ranging from databases, documents, emails to transactions, audios, videos etc. There is a need to manage variety of forms of data.

The security and encryption mechanisms can be realized based on the state of digital data. There are three states of data: Data at rest, Data in transit and Data in use.

Data at rest is a state that indicates that the data is stored in repository in a digital form. This state of data basically refers to the data stored on cloud which is currently not being transmitted across the network.

Data in transit refers to the data that is travelling across network. It could be in transit from local to cloud storage or vice versa. The data should be in encrypted form and the Secure Socket Layer (SSL) ensures that data in transit remains integral.

Data in use refers to the data that is currently being processed. Data in use may also be prone to threats. However, device authentication and authorization mechanisms help in dealing with these threats.

2.1 Encrypting Critical Data Items

The entire data stored by cloud service provides is generally stored in encrypted form. Raghava et.al [1] proposed that the entire database need not be encrypted to allow efficient query processing. Instead only the critical data items may be encrypted. However the challenge is to dynamically select critical data items from the huge volumes of data. Secondly when more and more features are added to the repository the existing critical data items set may not be sufficient. Further the intruder can at least know as to which database objects and attributes are being accessed and may guess the critical data items from the other data items.

2.2 Security measures in Cloud Computing Environment

Venkat et.al [5] proposed security measures in Cloud Computing Environment (CCE) that would improve security of cloud environments using the mechanisms like File encryption, Network encryption, Logging, Node maintenance, Rigorous testing of map reduce jobs, layered framework for assuring cloud and third party secure data publication to the cloud.

These approaches may not be suitable for big data domain. There is a need for authentication of devices and the encryption mechanisms used may not directly be suitable for big data domain. There is a need to revisit these approaches by considering the characteristics of big data domain.

.

2.3 Data mining techniques for Preserving Privacy

Neetul et al [12] presented a survey of techniques used for preserving privacy using data mining approaches. The selection of a particular technique depends upon the type of data set used [4]. In the Anonymization technique, the sensitive attribute value(s) is replaced by other value(s) with an intention of not disclosing the sensitive data. Each tuple of a relation must be indistinguishably associated with a value to deal with identification or risks.

In randomization technique an extra field is added to the current field(s) known as noise. With the inclusion of the additional field the correct individual information can be presented but the combine effects are preserved. Individual fields of the relation are randomized but the aggregated values will yield correct results.

In “Probabilistic results for Queries” approach the results of queries can be null or it can be probabilistic replies instead of producing the actual results [4]. There is a need of modification of these approaches keeping in view the characteristics of big data environment.

Several strategies like encrypted database concepts help in managing data at rest but they are realized for the structured queries and don’t consider the velocity and veracity of data. Hence there is a need to include the wrapper modules to secure seamless processing, transmission and storage of data.

3 Framework for securing data at rest

Traditional database systems are suitable for management of structured data. In order to store and process huge volumes of data, the storage was outsourced to the data centers or cloud service providers. The data in transit is secured using SSL and data at rest is managed using encrypted databases. However these solutions are not effective in analyzing semi structured and unstructured data at the same time (Variety). Further the traditional database systems can’t manage data deluge. The data which is generated by humans and devices is referred as data deluge. Enormous amount of data is generated each hour from various devices and is stored in cloud databases. This data may be generated in various formats.

3.1 Cloud Database Service

Traditionally maintenance and management of data was the responsibility of the customer’s using their private servers. With the increase in volume of data, it was difficult to manage it. Hence the organizations started outsourcing the same. Cloud databases is an essential service of cloud computing environment as the operational and management cost increases exponentially with the size of the data. There are two approaches for deploying database in the cloud. The first mechanism is referred as “Do-it-yourself”. In the approach the consumer subscribes to IaaS (Infrastructure as a Service) and uses its own database instance to realize the velocity, variety and veracity characteristics. In the second approach the consumer deploys its database on to the cloud and it is managed by Cloud Service Provider (CSP).

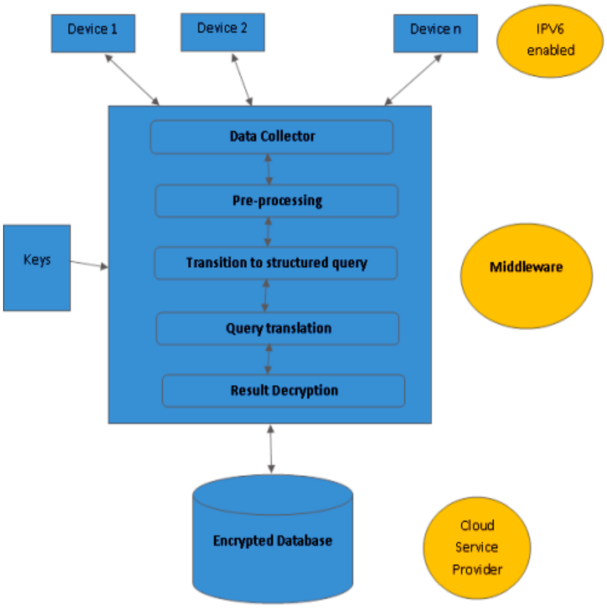
Cloud services provides two types of environments to realize database services viz., multi-instance model and multi-tenant model. In multi-instance model, each consumer is provisioned with unique DBMS running on a dedicated virtual machine subscribed by specific customer. In this environment, the consumers have better control over administrative and other security solutions. In multi-tenant model, tagging method is used which provides pre-defined database environment that is shared by many tenants. Data of each residing tenant is tagged by identifier that is used for unique identification.

3.2 Architecture for Managing Encrypted Databases

One of the effective techniques to ensure confidentiality of sensitive data in cloud environment is to use encryption of data in transit as well as data at rest.

Even if the database is outsourced to cloud service provider (untrusted servers), encryption can be used to improve database security. However encryption shouldn't hamper the normal functionality of databases. Ideally queries should be executed directly over encrypted databases. Database may be deployed on untrusted or compromised machine. The cloud service providers which are in untrusted domain shouldn't be in a position to hold the key for encrypting or decrypting the data. There is a need for a middleware or proxy, which is in trusted domain. The keys for encryption and decryption are available with the proxy or middleware. The proxy based solutions were realized using tools like CryptDB. However they don't directly apply to the big data domain as the data may not be always in the structured form.

The following architecture is used to manage encrypted databases in big data domain.



The data comes from various devices. There is a need for IPV6 enabled devices to realize device authentication. All the applications reside on these devices. These devices may range from desktop to mobile and smart devices which can be in the scope of Internet of Things. The cloud service providers stores the data in encrypted from. There is a need for realizing strong middleware which can handle huge volumes of data of various forms with noise and are captured through fast streaming.

The middleware is expected to realize five modules. The data collection is a module which is used to capture data. The data collector must be in a position to manage heterogeneous formats of data. The data be in varied forms and may also contain noise.

The functionality of the pre-processing module is to remove noise from the captured data. Effective preprocessing mechanisms may help in effective removal of noise from the captured data.

Since the data can be structured or semi-structured, No SQL databases like MongoDB can be used. The other approach is to convert the semi structured or unstructured data into the structured tuples.

As the data at the server of data center is stored in encrypted form, the plain query needs to be translated into cipher form. The respective keys for encryption and decryption are maintained at the middleware. The data is stored in encrypted form at the Cloud service provided. When the information has to be retrieved from the cloud service provider, the middleware receives the data in encrypted form. The middleware then can decrypt the data and sends it to the device which has requested the information.

The deterministic encryption [9], order preserving encryption [8] and homomorphism techniques can be applied to retrieve data, realize range queries and to realize the aggregate functions using the above architecture. The onion layers can be realized after implementation of preprocessing and transition modules. The onion layers [8,9] used in encrypted databases in cloud environments can be used in big data domain. However Data collector and preprocessing modules will act as outer layers of the onion.

3.3 Key Management and Fault tolerance

Key management is a challenging issue in the cloud and big data domain. The key may be stored at client, server, middleware or it may be shared among various devices. It may not be feasible to store the keys at clients as the devices may be prone to failures and may be lost. As the server is in untrusted domain, it is not desirable to store the keys at the server. Keys may be shared among several trusted devices where each device maintains a partial key. In order to encrypt or decrypt the data, the key has to be reconstructed by collecting the partial keys from all devices. If any of the device fails then it will be difficult to store or query the data.

In the proposed framework, it is proposed to store the keys on the middleware or proxy. The middleware or proxy is in trusted domain. However if the middleware fails, the system stops functioning. Hence there is a need to realize fault tolerance of middleware.

It is proposed that the middleware id to be connected with its mirror copy in synchronous mode. Both images are expected to be in active-active mode, such that if one of the middleware fails, the other middleware automatically starts realizing the transaction management without any delay. When the traffic over the network increases, it also helps as a load balancer. The logs which are maintained at various mirror copies helps in recovery of the failed middleware. Since the data is stored at cloud service provider, it is expected that only transaction replication will be used between the primary and backup copies of middleware.

4 Open issues and challenges

The big data threat model has certain open issues and challenges:

- The proposed framework has to be implemented on high volumes of data to know its robustness.
- Device authentication is needed before the data is processed at various devices. Since various forms of data re generated from multiple devices, the same has to be realized on the proposed framework.
- The two modules of middleware i.e. preprocessing and transition to structured query require design of optimization techniques as huge data in varied forms is collected.
- In order to guarantee integrity property, effective key management strategies such as key based encryption are expected to be included in the proposed framework.
- Novel approaches for key management are desirable.
- There is need for proposing mechanisms to implement failure recovery of middleware.

5 Conclusion

According to Gartner, "Big data information security is a necessary fight". The characteristics of big data environment pose new security challenges. The existing database encryption solutions are not sufficient as it is expected that the data is captured from several devices with varied forms. In this paper, a framework is presented which will be used to realize the security solutions in big data domain. The open issues and challenges highlights the enhancements that can be made to the proposed framework.

References

[1] Raghav Toshnival, Kanishka Ghost Dastidar, Asoke Nath, "Big Data Security Issues & Challenges", International Journal of Innovative Research in Advanced Engineering (IJIRAE), Issue 2, Vol. 22, pp.15-20, 2015

[2] Neetu Chaudhari, Satyajee Srivastava, "Big Data Security Issues and Challenges", International Conference on Computing, Communication & Automation (ICCCA 2016) pp. 60-64, 2016

[3] Big Data Preliminary Report 2014, ISO/IECJTC1, Information Technology, www.iso.org/iso/home/about/iso_members.htm, prelimnary report 2015.

[4] Arun Thomas George, Arun Viswanathan, Kiran N.G, Phil Shelley, Dough Cutting, S. Gopalkrishnan, Big Data Spectrum 2012.

[5] Venkata Narasimha Inukollu, Sailaja Arsi, Srinivas Rao Ravuri, "Security Issues associated with Big Data in Cloud Computing", Internatinal Journal of Network Security & its Applications (IJNSA), Vol. 6, No. 3. PP. 45-46, 2014.

[6] P. R Anisha, Kishore Kumar Reddy C, Srinvasulu Reddy K, Surender Reddy S. "Third Party Data Protection Applied to Cloud and Xacml Implementation in Hadoop Environment with Sparql",IOSR Journal of Computer Engineering, pp. 39-46, 2012.

[7] "Addressing Big Data Security Challenges: The Right Tools for Smart Protection". A Trend Micro white paper, pp.1-7, Sept 2012.

[8] Alexandra Boldyreva et al. "Order Preserving Encryption Revisited" Improved Security Analysis & Alternate Solutions", LNCS,Crypto 2011.

[9] Alexandra Bldyreva, Serger Fehr, Adam O Neill, "On Notions for Security for Deterministic Encryption & Efficent Construtions Without Random Oracles", CRYPTO 2008, LNCS pp. 335-339, 2008

Salman Abdul Moiz is working as an Associate Professor at School of Computer & Information Sciences, University of Hyderabad. His areas of interest includes Software Engineering, Distributed Databases, E-Learning and Fault tolerance. He did his Ph.D from Osmania University, M.Tech (CSE) from Osmania University, MCA from Osmania University and M.Phil(CS) from Madurai Kamaraj University. He is a Fellow of IETE, Senior Member of IEEE, Senior Member of ACM, Life Member CSI, Life Member EWB and Member ISRS.