ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Homomorphic Elgamal Encryption in NoSQL for Secure Cloud

Shagufta Praveen*

M.Tech Scholar, Dept. of Computer Science, Uttaranchal University, Dehradun, India*

ABSTRACT: NoSQL adopted by number of leading organizations. RDBMS doesn't fit with today's Big Data scenario. Rapid growths of data not only ask for storage but ask for its security too. This paper provides the specific method to secure data on cloud with the help of Homophormic Elgamal Encryption (HEE). To implement our work, column oriented database (HBase), an awesome storage explanation for NoSQL database is chosen. Data encrypted using proposed HEE scheme, using Microsoft Azure toolkit and basic .NET package with SQL server. Thus, only authentic user can access the NoSQL database by following this proposed method. Cause of the proposed functionality, HEE algorithm allows users to send encrypted data, which would compute the cipher text without decrypting it and send back same encrypted effect to a server or user.

KEYWORDS: NoSQL; Column-Oriented Databases; HEE algorithm.

I. INTRODUCTION

Relational Database system ruled for around 40 years which based on relational model. It became the priority for storing information from business records, personal information and much more. According to IBM, "everyday, we create 2.5 quintillion bytes of data –so much that 90% of the data in the world today has been created in the last two years alone"[1]. As organizations know RDBMS doesn't fit in the criteria of bulk data storage. They can see the pace of heterogeneity, amount and velocity of data building and its consumption. This results into Big Data, where you definitely need NoSQL. This is not about major industries like Facebook, Google, Foursquare but any organization when feel that its data is not fit for server and asking for distributed system. These companies also opt for NoSQL.
Microsoft Azure chains an assortment of NoSQL technologies that provides well managed services for running with relational and non-relational data. Apart from the services, other new technologies can also be used by running them on Virtual machines of Azure. Figure 1 represents some of the NoSQL technologies that are able to use on Azure. As presented in the figure, NoSQL technologies are categorized into two groups of data technologies that can be run on Azure. The options contain the following:

Document Stores - DocumentDB, MongoDB, CouchDB,etc.

Key/Value Stores - LevelDB, Redis, Riak, Table,etc.

Column Family Stores – HBase, Cassandra,etc.

Big-Data analytics - Hadoop, HDInsight,etc

Azure's four NoSQL managed services: DocumentDB, Tables, HBase, and HDInsight [2]. ***In this paper we are concentrating just on (Column-Oriented Database) and implementing a security algorithm (HEE) for a cloud system***

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

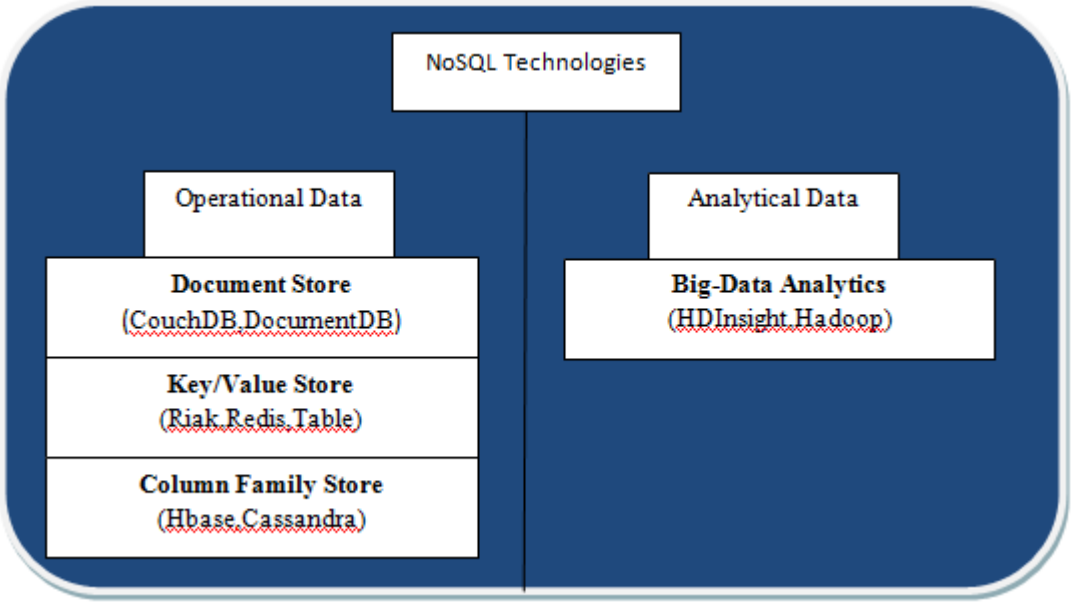

Figure 1: Azure data technologies categorized into two groups for NoSQL[2].

1.1 COLUMN-ORIENTED DATABASES (COLUMN FAMILY STORE): HBASE

HBase belongs to column family stores. It contains rows and column like relational database still it is very different from the relational database. As in HBase there are column families. They are called so because they hold number of columns. When this database store information of a web application, the column family of the user has a column that hold a unique row key for each row. The schema of column is such that, a user can add any number of columns in the family. The columns of column family can store the data of the user last access and the first he used it. Due to adding any number of columns, table can be large and can have millions of columns and rows, yet many of the cells in the table might be vacant. For this kind of situation column family store is most favorable.

For example, a user wants to create a table with information about all the web pages present on the internet. Page could be described by each row while some aspects of that would be described by each column. With this concept there would be lots of column but many of the cells in this table would be vacant. Because most pages will have only a subset of possible attributes [2].

HBase hold no data type, in this way too, it is different from relational database. Instead of data type, it stores byte strings. Each cell of the table also has time stamp to access the older information from the application. User has not to use any query language in HBase. He can access its information about a specific cell through three things (column family, column qualifier and row key).

II. RELATED WORK

Information leakage are one of the big issue in cloud storage, encryption is the most common way to hide data so that confidential data leakage can be avoid. Function of homomorphic encryption was suggested for aggregation queries SUM and AVG [3]. A scheme OPES was proposed through which cipher text can be indexed and can be directly handled without decrypting it. But order preserving Encryption (OPES) is not a better solution for preserving data as it would leak information [4]. A smart model was made for the analysis outsourced software. And safe protocol is given for processing of k-nearest-neighbor queries on Rtree index [5]. Reverse encryption algorithm was proposed to measure the performance of query processing. but privacy of query and information was not consider[6]. Encryption called privacy homomorphism was proposed that makes a cipher text on which complex computation can be done without decrypting it [7]. Another encryption scheme to handle encrypted aggregation queries but it some of it attributes are encrypted and some are not. Which describe the vulnerability to cipher text [8]. With data privacy, user privacy is also

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

considered in [9][10][11]. A protected transversal framework proposed with privacy homomorphism based encryption scheme in [11]. For better aggregation operation in database addition of keywords in block is proposed. Rather than adding them one after one. But in this method range have not considered by author. In order to provide security to data and user, block structure are used to hide real structure. For better key search, comparison k-in-1 is done where k is the number of key in a block, which is decided by size of key and block [9]. In this framework, index transversal path cannot be trace by service provider during evaluation. Hence, privacy of users can be kept in [10]. Paillier and ElGamal encryption are the best schemes to provide security to the cloud. As they have very good possible implication on the outsourcing of private computation [12]. For database outsourced services, data privacy and user privacy is discussed in [13].

III. GENERATION OF PLAN WORK

Here we are using a homomorphic encryption technique and implementing HEE algorithm. *Homomorphic encryption* is one of the most stimulating new themes for security of communication data, which assure that data stored in the cloud is perfectly secure. With this technique, an encrypted data would be send by user to a server and it would compute it without decryption. And at last encrypted message would return to user. With this technique, server would never aware about original data.

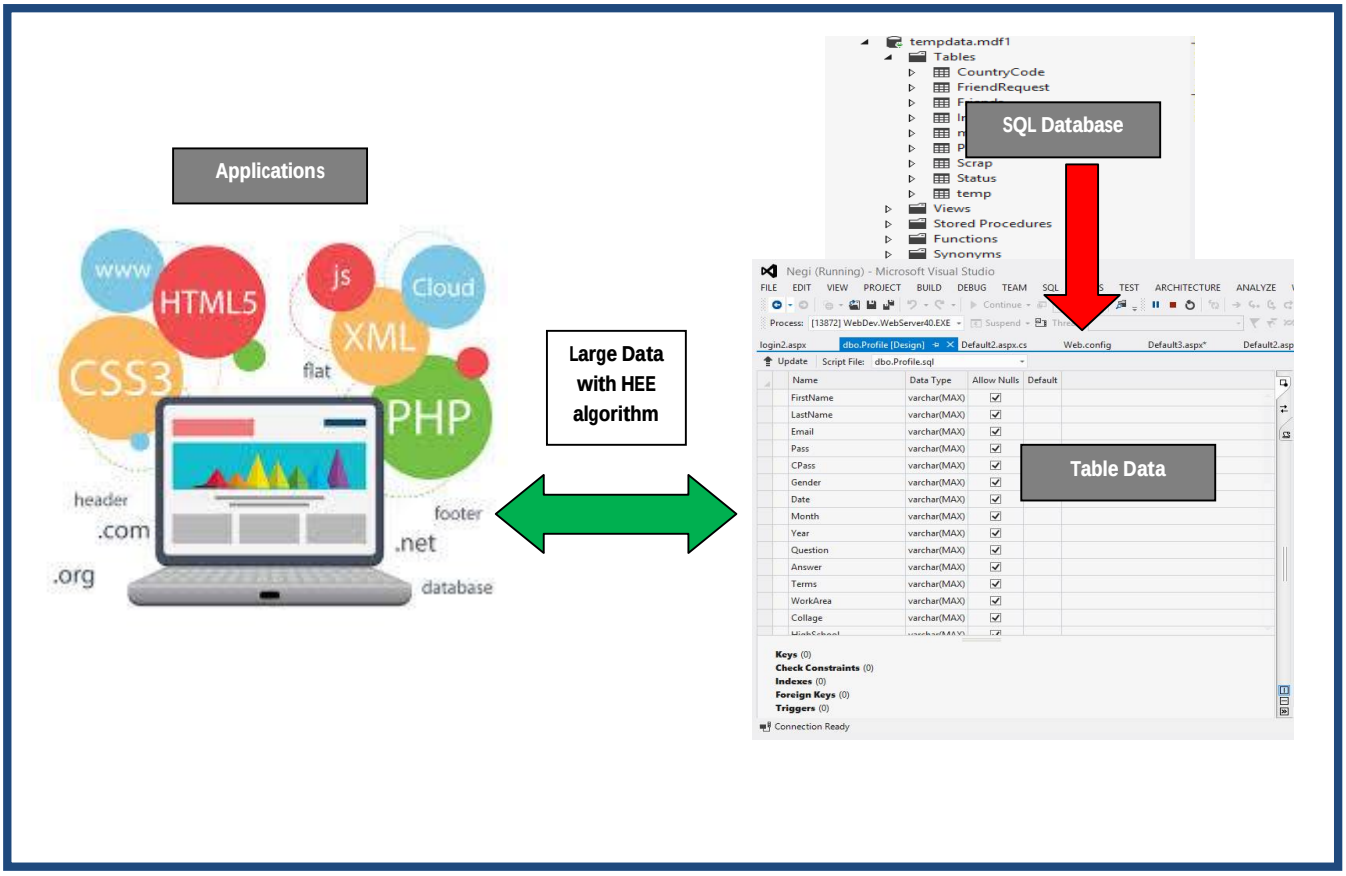

Figure 2: NoSQl Database interacting with cloud storage (Web-Application) and sharing data of users. Source: http://www.mirrorminds.in/

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Just for this proposed work we don't have the funds for implementation, so here we are showing a demo project of a broad database of web application with the help of SQL-server and Microsoft visual studio .net 2012. In this demo project SQL-server used for HBase and Microsoft visual studio .net 2012 used for Microsoft Azure.
HBase provides transaction, cells present is the same row of a single table get modify. HBase can store same information at multiple server which increases availability and by default HBase provide strong consistency too. HBase is part of Hadoop family, when it works on Azure, it provides HDInsight service, Instead of providing complete services like Tables. HBase needs a user to mention the number of virtual machines he requires for his HBase cluster [2].
Homomorphic ElGamal Encryption (HEE) algorithm provides security through calculation. In figure 2 we have 5 different ids (keys) for storage of user's data, *here we will apply Homomorphic ElGamal encryption algorithm for enhancing security of big-data or cloud network.*

IV. METHODOLOGY

Client to server and server to client, request and response it's all basic and always discussed in books, Internet, articles and research papers. In this work we are doing same but for an expansive encrypted data which managed at cloud end as NoSQL data. As we know, web application starts by user through web browser. Details filled by user for identification, these details converted to an encrypted message in the course of HEE algorithm. These details forwarded to server database. At server end user_id matches with details and filter it to authenticate the user. After authentication, user can communicate with other users and send data. Then, proposed HEE algorithm works successfully and messages will send to one another in encrypted form.

A. *Flowchart for a single user*

This work presented by a demo project in Microsoft visual studio for a single user with his huge data. Web-Application is a part of cloud computing so absolutely it's worked under a cloud provider like Google, Amazon etc. If a broad amount of data handled at cloud end by Google that means cloud provider using big-table or other source according to requirement for controlling it. So, nowadays most of the broad data managed on cloud in the form of NoSQL

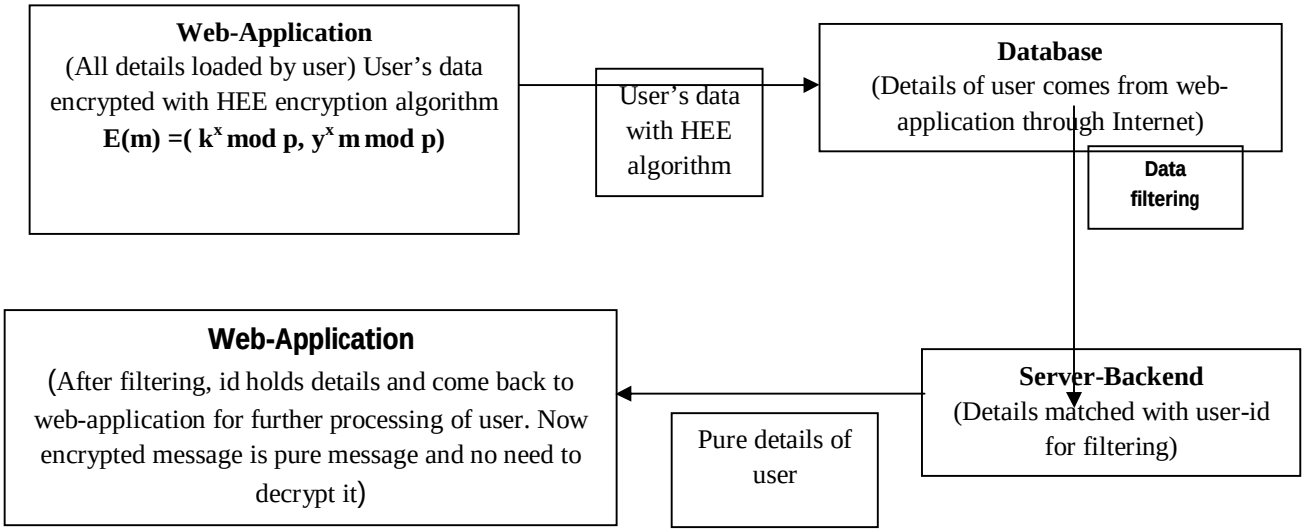

Figure 3: Flow of encrypted data from web-application to server

B. *Homomorphic Elgamal encryption algorithm*

Encryption key for Elgamal algorithm are created as follows: P chosen as large prime number, then k is chosen and at last ,b (an integer) is chosen and compute $y = k^b \pmod p$, where (p,k,y) are public keys whereas b kept as private key. Key b doesn't help in encrypting message but in decrypting it.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

For Example: B has public key (p,k,y) and A want to send an encrypted message m to B, using Elgamal encryption. A asked for public keys (p,k,y) and choose a random number x and compute $s=k^x \pmod p$.

1. Using same random number , A computes $t = y^x m \pmod p$.
2. C(s,t) is A's encrypted message
3. The encryption of a message **m** is: $\mathbf{E(m_1,x_1)=(C_1,C_2) = (k^{x}_1 \bmod p, y^{x}_1 m_1 \bmod p)}$
4. $\mathbf{E(m_2,x_2)=(C_3,C_4)=(k^{x}_2 \bmod p,\ y^{x}_2 m_2 \bmod p)}$
5. The Homomorphic property of Elgamal: $\mathbf{(C_1,C_2).(C_3,C_4)=(k^{x}_1{}^{+x}_2,\ m_1 m_2 y^{x}_1{}^{+x}_2)}$
6. B can decrypt it using a through $m= ts^{-b} \pmod p$

C. *Proposed Homomorphic Elgamal encryption (HEE) algorithm*

1. $S = k^x \pmod p$ where x is a random number.
2. The encryption of a message **m** is: $\mathbf{E(m,x) = (k^x \bmod p\ , y^x m \bmod p)}$
3. In this example we are using 5 random numbers keys. $C(m_1).C(m_2).C(m_2).C(m_4).C(m_5)= (k^{x}_1{}^{+x}_2{}^{+x}_3{}^{+x}_4{}^{+x}_5\ ,(m1.m2.m3.m4.m5)y^{x}_1{}^{+x}_2{}^{+x}_3{}^{+x}_4{}^{+x}_5)= C\ (m_1.m_2.m_3.m_4.m_5)$. In this example we are using only 5 keys, Algorithm is also useful for multiple keys {0, 1, 2, n}.

V. CONCULSION AND FUTURE WORK

Through HEE algorithm and Column-Oriented Databases we can secure user data and easily use cloud storage or NoSQL databases. This paper presented an independent study of Column-Oriented Databases (HBase). This work provides the specific method to secure data on cloud with the help of HEE algorithm. The authorized user can access the NoSQL database by this proposed method. Homomorphic technique never let unauthorized users to extract original form of data. Users can send and store their critical message and data in encrypted form, which would be operated without decryption process and same encrypted result received back by a server or user. Hence forth, data security is provided by implementing algorithm. Other NoSQL are DocumentDB, Tables, and HDInsight. These services are also possible implement same with same or different cloud security algorithms like symmetric (AES, DES) or asymmetric (DSA, RSA, ElGamal) algorithms.

ACKNOWLEDGEMENT

I am thankful to Dept. of Computer Science, Uttaranchal University, Dehradun, for providing me research facilities.

REFRERENCES

[1] ""IBM,What is big data?", Internet: http://www-01.ibm.com/software/au/data/bigdata/,2013.
[2] "Understanding NOSQL on Microsoft Azure", Internet: http://www.davidchappell.com/,2014.
[3] H. Hacigumus, B. R. Iyer, and S. Mehrotra (2004). 'Efficient execution of aggregation queries over encrypted database'.
[4] R. Agrawal, J. Kierrnan, R. Srikant, and Y. Xu.(2004), 'Order preserving encryption for numeric data'. Proceedings of the ACM SIGMOD international conference on Management of data (SIGMOD '04), p.p. 563–574, New York, NY, USA, 2004. ACM.
[5] Y. Hu, X. Mo, X. Zhang, Y. Zeng, J. Du, and K. Xie (2012). 'Intelligent analysis model for outsourced software project risk using constraint-based bayesian network',JSW, 7(2), p.p. 440–449.
[6] A. Mousa, E. Nigm, El-Sayed El-Rabaie, O. S. Faragallah, and O. S. Faragallah (2012), 'Query processing performance on encrypted databases by using the rea algorithm',p.p. 280–288.
[7] R. Rivest, L. Adleman, and M. Dertouzos (1978), 'On data banks and privacy homomorphisms', p.p 169–177. Academic Press.
[8] E. Mykletun and G. Tsudik(2006), 'Aggregation queries in the database-as-a-service model', E. Damiani and P. Liu, editors, 'Data and Applications Security', Springer Berlin / Heidelberg, (volume 4127 of Lecture Notes in Computer Science), p.p 89–103.
[9] T. Ge, Stanley B. Zdonik, and S. B. Zdonik (2007), 'Answering aggregation queries in a secure system model'. In VLDB, p.p 519–530
[10] H. Hu and J. Xu. Non-exposure location anonymity (2009), Y. E. Ioannidis, D. L. Lee, and Raymond T. Ng, editors, ICDE, .p.p 1120–1131. IEEE, 2009.
[11] H. Hu, J. Xu, C. Ren, B. Choi, and B. Choi (2011), 'Processing private queries over untrusted data cloud through privacy homomorphism'. In ICDE, p.p 601–612, 2011.
[12] D. Micciancio (2010), ' A first glimpse of cryptography's holy grail', p.p 96, 2010.
[13] Y. Yu and W. Bai (2011), 'Enforcing data privacy and user privacy over outsourced database service', JSW, 6(3), p.p. 404–412, 2011.