IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

Secure Big Data Processing Through Homomorphic Encryption in Cloud Computing Environments

J. Josepha Menandas[1], J. Jakkulin Joshi[2]

Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India[1, 2]

Abstract: With the fast development of rising applications like informal community examination, semantic Web investigation and bioinformatics system investigation, an assortment of information to be handled keeps on seeing a speedy increment. Viable presents a few major information handling techniques from framework what's administration furthermore, investigation of vast scale information represents a fascinating however basic challenge. As of late, Big Data has pulled in a great deal of consideration from the scholarly world, industry and additionally government. This paper more, application viewpoints. To start with, from the perspective of cloud information administration and enormous information preparing systems, we show the key issues of enormous information preparing, including distributed computing stage, cloud structural planning, cloud database and information stockpiling plan. Taking after the MapReduce parallel preparing structure, we then present MapReduce streamlining techniques furthermore, applications reported in the writing. Then we use standard encryption methods when we transferred to the cloud to secure the operations and the storage of data. Homomorphic Encryption is used to perform functions on encrypted data. Here we provide various homomorphic encryption schemes available on cloud computing environment for secure data processing. At last, we discuss and find the best homomorphic encryption method in cloud computing environment in big data.

Keywords: Big Data, Map Reduce, Homomorphic Encryption.

I. INTRODUCTION

Big data is a term that describes the complex or large volume of structured and unstructured data. It is difficult to process using on-hand database management tools. To extract important quality from Big Data, we require ideal processing techniques, analytic abilities and aptitudes. There are several big data processing techniques including cloud environment available. In the most recent two decades, the consistent increment of computational force has created a staggering stream of information. Enormous information is turning out to be more accessible as well as more justifiable to PCs. For instance, the celebrated interpersonal organization Website, Facebook, serves 570 billion site visits for every month, stores 3 billion new photographs each month, and oversees 25 billion bits of content. Google's inquiry and notice business, Facebook, Flickr, YouTube, and Linkedin utilize a heap of manmade brainpower traps; require parsing unlimited amounts of information and settling on choices immediately. Multimedia data mining stages make it simple for everyone to accomplish these objectives with the base measure of exertion as far as programming, CPU and system.

Every one of these illustrations demonstrated that overwhelming enormous information challenges and critical assets were dispensed to bolster these information serious operations which prompt high stockpiling and information handling expenses. The present advances, for example, grid and cloud computing have all expected to get to a lot of aggregating so as to register force assets and offering a solitary framework view. Among these advances, cloud computing is turning into a capable construction modeling to perform substantial scale what's more, complex computing, and has upset the way that registering foundation is preoccupied and utilized. Also, an essential point of these advances is to convey processing as an answer for handling enormous information, for example, large scale, multi-media and high dimensional information sets.

Big data and cloud computing are both the speediest moving advancements distinguished in Gartner Inc's. 2012 Hype Cycle for Emerging Technologies. Cloud computing [1] is connected with new worldview for the procurement of figuring foundation and enormous information handling system for a wide range of assets. Also, some new cloud-based innovations must be embraced on the grounds that managing huge information for simultaneous preparing is troublesome.

Big data is a massive volume of both structured and unstructured data that is so large to process with traditional database and software techniques [2]. The definition of big data as also given by the Gartner: "Big Data are high-volume, high-velocity, and/or high-variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization [3]. According to Wikimedia, "In information technology, big data is a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools".

The objective of this paper is to give the status of huge information studies and related works, which goes for

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

giving a general perspective of enormous information administration advances and applications. We give a diagram of major methodologies and arrange them as for their procedures including enormous information administration stage, disseminated document framework, huge information stockpiling, MapReduce application and advancement. Be that as it may, keeping up and handling these substantial scale information sets is regularly past the span of little organizations and it is progressively posturing difficulties notwithstanding for extensive organizations and foundations. Then, we examine the open issues and difficulties in preparing enormous information in three critical viewpoints: huge information stockpiling, investigation and security.

Finally we discuss several security mechanisms in big data. We use standard encryption methods when we transferred to the cloud to secure the operations and the storage of data. Here data has been encrypted before move it to the cloud and execute operations on encrypted data without decrypt, this provides same results after calculations as if we worked directly on the plain data. Homomorphic Encryption is[4] used to perform functions on encrypted data without known the private key, thus generating an encrypted result which, when decrypted, matches the result of operations performed on the plaintext, the client only holds the secret key to be decrypted. Here we provide various homomorphic encryption schemes available on cloud computing environment for secure data processing. The general view of Homomorphic encryption as shown in the Fig. 1

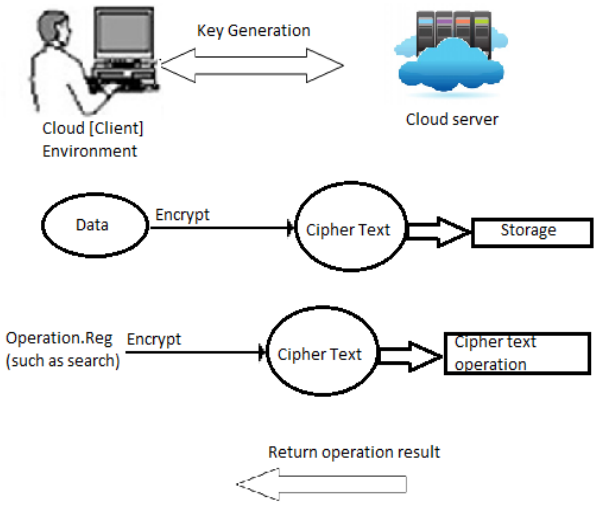

Fig. 1 Data Security scheme for cloud Computing

II. BIG DATA MANAGEMENT SYSTEM

Numerous analysts have proposed that business DBMSs is not suitable for handling greatly large scale information. Exemplary building design's potential bottleneck is the database server while confronted with top workloads. One database server has limitation of versatility and expense, which are two essential objectives of enormous information handling. Keeping in mind the end goal to adjust different vast information handling models, D. Kossmann et al. displayed four unique architectures in light of excellent multi-level database application Architecture which are apportioning, replication, disseminated control and reserving architecture. It is clear that the option suppliers have diverse plans of action and target various types of uses: Google is by all accounts more keen on little applications with light workloads though Azure is at present the most reasonable administration for medium to huge administrations. The greater part of late cloud administration suppliers are using mixture building design that is fit for fulfilling their genuine administration prerequisites. In this area, we for the most part talk about huge information construction modeling from three key viewpoints: disseminated record framework, non-auxiliary and semi-organized information stockpiling and open source cloud stage.

A. Non-structural and semi-structure Data Storage

With the accomplishment of the Web 2.0, more IT organizations have expanding needs to store and break down the steadily developing information, for example, hunt logs, slithered web substance, and snap streams, ordinarily in the scope of petabytes, gathered from an assortment of web administrations. In any case, web information sets are normally non-social or less organized and preparing such semi-organized information sets at scale postures another test. Besides, basic disseminated document frameworks said above can't fulfill administration suppliers like Google, Yahoo!, Microsoft and Amazon. All suppliers have their motivation to serve potential clients and own their applicable state of-the-craft of huge information administration frameworks in the cloud situations. Bigtable is a dispersed stockpiling arrangement of Google for overseeing organized information that is intended to scale to a huge size (petabytes of information) crosswise over a great many thing servers. Bigtable does not bolster a full social information model. Then again, it gives customers a basic information show that backings element control over information design and configuration. PNUTS is a massive scale facilitated database framework intended to backing Yahoo's! web applications. The principle center of the framework is on information serving for web applications, instead of complex questions. Upon PNUTS, new applications can be constructed effortlessly and the overhead of making and keeping up these applications is not a lot. The Dynamo is a profoundly accessible and adaptable conveyed key/quality based information store manufactured for supporting interior Amazon's applications. It gives a straightforward essential key just interface to meet the prerequisites of these applications. On the other hand, it contrasts from key-esteem stockpiling framework.

Facebook proposed the configuration of another group based information distribution center framework, Llama, a mixture information administration framework which consolidates the components of line shrewd and section astute database frameworks. They additionally portray another section savvy document design for Hadoop called CFile, which gives preferable execution over other record groups in information examination.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

B. Distributed File System

Distributed File System (DFS) is an arrangement of customer and server benefits that permit an association utilizing Microsoft Windows servers to sort out numerous dispersed SMB document offers into an appropriated document framework. DFS gives area straightforwardness and excess to enhance information accessibility despite disappointment or substantial burden by permitting shares in numerous diverse areas to be sensibly gathered under one organizer, or DFS root. Google File System (GFS) is a lump based disseminated record framework that backings adaptation to non-critical failure by information apportioning and replication. As a fundamental stockpiling layer of Google's distributed computing stage, it is utilized to peruse info and store yield of MapReduce. Additionally, Hadoop[14] likewise has a conveyed record framework as its information stockpiling layer called Hadoop Distributed File System (HDFS), which is an open-source partner of GFS. GFS and HDFS are user level file systems that don't execute POSIX semantics and intensely enhanced for the instance of substantial records (measured in gigabytes). Amazon Simple Storage Service (S3) is an online open stockpiling web administration offered by Amazon Web Services. This record framework is focused at bunches facilitated on the Amazon Elastic Compute Cloud server-on-interest foundation. S3 means to give adaptability, high accessibility, and low dormancy at ware costs. ES2 is a versatile stockpiling arrangement of epiC, which is intended to bolster both functionalities inside of the same stockpiling. The framework gives effective information stacking from distinctive sources, adaptable information dividing plan, list and parallel successive sweep. What's more, there are general file systems that have not to be tended to, for example, Moose File System (MFS), Kosmos Distributed File system (KFS).

C. Open source and cloud platform

The primary thought behind server farm is to influence the virtualization innovation to expand the use of registering assets. Subsequently, it gives the fundamental fixings, for example, stockpiling, CPUs, and system transfer speed as a thing by particular administration suppliers at low unit cost. For coming to the objectives of enormous information administration, the greater part of the examination foundations and undertakings bring virtualization into cloud architectures. Amazon Web Services (AWS), Eucalptus, Opennebula, Cloudstack and Openstack are the most famous cloud administration stages for framework as an administration (IaaS). AWS is not free but rather it has immense utilization in flexible stage. It is anything but difficult to utilize and just pay-as-you-go. The Eucalyptus works in IaaS as an open source.

It utilizes virtual machine as a part of controlling and overseeing assets. Since Eucalyptus is the most punctual cloud administration stage for IaaS, it consents to API good arrangement with AWS. It has a main position in the private cloud market for the AWS natural environment. OpenNebula has reconciliation with different situations. It can offer the wealthiest highlights, adaptable ways and better interoperability to fabricate private, open or cross breed mists. OpenNebula is not a Service Oriented Architecture (SOA) outline and has powerless decoupling for registering, stockpiling and system free segments. CloudStack is an open source cloud working framework which conveys open distributed computing like Amazon EC2 yet utilizing clients' own equipment. CloudStack clients can exploit distributed computing to convey higher effectiveness, boundless scale and speedier sending of new administrations and frameworks to the enduser. At present, CloudStack is one of the Apache open source ventures. It as of now has full grown capacities. Be that as it may, it needs to further reinforce the freely coupling and part plan. OpenStack is an accumulation of open source programming activities planning to assemble an open-source group with specialists, designers and ventures. Individuals in this group share a typical objective to make a cloud that is easy to convey, enormously adaptable and loaded with rich elements. The building design and parts of OpenStack are clear and stable, so it is a decent decision to give particular applications to endeavors. In current circumstance, OpenStack has great group and natural environment. Then again, despite everything it have a few inadequacies like fragmented capacities furthermore, absence of business backings.

III. APPLICATIONS AND OPTIMIZATION

A. Application

In this time of information blast, parallel preparing is fundamental to perform a huge volume of information in an auspicious way. The utilization of parallelization methods and calculations is the way to accomplish better adaptability and execution for preparing enormous information. At present, there are a great deal of well known parallel preparing models, including MPI, General Purpose GPU (GPGPU), MapReduce and MapReduce-like. MapReduce proposed by Google, is an extremely prominent enormous information preparing model that has quickly been contemplated and connected by both industry and the scholarly world. MapReduce has two noteworthy focal points: the MapReduce model conceal subtle elements identified with the information stockpiling, conveyance, replication, burden adjusting et cetera. Besides, it is simple to the point that software engineers just indicate two capacities, which are guide capacity and diminish capacity, for performing the preparing of the huge information. We separated existing MapReduce applications into three classifications: apportioning sub-space, decaying sub-procedures and estimated covering computations.

While MapReduce is referred to as a new approach of processing big data in cloud computing environments, it is also criticized as a “major step backwards” compared with DBMS. We all know that MapReduce is schema-free and index-free. Thus, the MapReduce framework requires parsing each record at reading input. As the debate continues, the final result shows that neither is good at the other does well, and the two technologies are complementary HadoopDB[15] is a hybrid system which efficiently takes the best features from the scalability of

MapReduce and the performance of DBMS. The result shows that HadoopDB improves task processing times of Hadoop by a large factor to match the shared nothing DBMS. Lately, J. Dittrich et al. introduced a new type of system named Hadoop++ which indicates that HadoopDB has also severe drawbacks, including forcing user to use DBMS, changing the interface to SQL and so on. MapReduce has gotten a considerable measure of considerations in numerous fields, including information mining, data recovery, picture recovery, machine learning, and example acknowledgment. For instance, Mahout is an Apache venture that goes for building versatile machine learning libraries which are all actualized on the Hadoop. In any case, as the measure of information that should be prepared develops, numerous information handling strategies have gotten to be not suitable or restricted. As of late, numerous exploration endeavors have misused the MapReduce system for taking care of testing information handling issues on extensive scale datasets in diverse areas. For instance, the Ricardo is delicate framework that incorporates R measurable instrument and Hadoop to support parallel information examination. Rank Reduce splendidly joins the Local Sensitive Hashing (LSH) and MapReduce, which adequately performs K-Nearest Neighbors look in the high dimensional spaces. F. Cordeiro et al. proposed BoW system for bunching expansive and multi-dimensional datasets with MapReduce which is a hard-grouping strategy and permits the programmed, and element exchange off between circle defer and system delay. MapDupReducer is a MapReduce based framework fit for identifying close copies over gigantic datasets effectively. Moreover, C. Officer et al. actualize the MapReduce structure on various processors in a solitary machine, which has increased great execution. As of late, B. He et al. create Mars, a GPU-based MapReduce system, which increases preferable execution over the cutting edge CPU-based structure.

B. Optimization

In this area, we display points of interest of ways to deal with enhance the execution of handling Big data with Map Reduce.

1) Data Transfer Bottlenecks: It is a major test that noisy clients must consider how to minimize the expense of information transmission. Thusly, scientists have started to propose assortment of methodologies. Map Reduce[13] presented another model called Map-Reduce-Merge. It adds to Map-Reduce a Merge stage that can productively blend information as of now divided and sorted or hashed by guide and diminish modules. Guide Join-Reduce is a framework that amplifies and enhances Map Reduce runtime system by including Join stage before Reduce stage[12] to perform complex information examination undertakings on extensive groups. They show another information preparing methodology which runs separating join collection errands with two back to back MR occupations. It embraces one-to-numerous rearranging plan to keep away from successive check directing and rearranging of middle of the road results. In addition, distinctive occupations frequently perform comparative work; in this manner having comparative work diminishes general measure of information exchange between employments. MRShare is a sharing system proposed by T. Nykiel et al. that changes a clump of questions into another cluster that can be executed all the more proficiently by consolidating employments into gatherings and assessing every gathering as a solitary inquiry. Information skew is likewise a critical variable that influences information exchange cost. Keeping in mind the end goal to defeat this lack, we propose a technique that partitions a MapReduce occupation into two stages: testing MapReduce employment and expected MapReduce occupation. The principal stage is to test the input data, assemble the intrinsic dissemination on keys' frequencies and afterward make a decent segment plan ahead of time. In the second stage, expected MapReduce occupation applies this allotment plan to each mapper to gathering the moderate keys rapidly.

2) Interactive Optimization: MapReduce likewise is a prominent stage in which the dataflow takes the type of a coordinated non-cyclic diagram of administrators. Be that as it may, it requires bunches of I/Os and pointless calculations while taking care of the issue of cycles with MapReduce. Twister proposed by J. Ekanayake et al. is an upgraded MapReduce runtime that backings iterative MapReduce calculations productively, which includes an additional Combine stage after Reduce stage. In this way, information yield from join stage streams to the following cycle's Map stage. It abstains from instantiating specialists over and again amid cycles and beforehand instantiated laborers are reused for the following emphasis with diverse inputs. HaLoop is like Twister, which is an adjusted variant of the MapReduce structure that backings for iterative applications by including a Loop control. It additionally permits to store both stages' info and yield to spare more I/Os amid emphasess. There exist heaps of emphasess amid chart information preparing. Pregel actualizes a programming model roused by the Bulk Synchronous Parallel(BSP) model, in which every hub has its own particular information and exchanges just a few messages which are required for the following cycle to different hubs.

3) Online: There are a few occupations which need to process online while unique MapReduce cannot do this exceptionally well. MapReduce Online is desgined to backing online conglomeration and consistent questions in MapReduce. It raises an issue that regular checkpointing and rearranging of middle results limit pipelined handling. They alter MapReduce structure by making Mappers push their information briefly put away in neighborhood stockpiling to Reducers preiodically in the same MR work. What's more, Map-side pre-collection is utilized to decrease correspondence. Hadoop Online Prototype (HOP) proposed by Tyson Condie is like MapReduce Online. Bounce is an altered adaptation of MapReduce system that permits clients to ahead of schedule land comes back from a position as it is being registered. It likewise bolsters for ceaseless questions which empower MapReduce projects to be composed for applications, for

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

example, occasion observing and stream handling while holding the adaptation to internal failure properties of Hadoop. D. Jiang et al. found that the union sort in MapReduce costs loads of I/Os and truly influences the execution of MapReduce. In the study, the outcomes are hashed and pushed to hash tables held by reducers when every guide undertaking yields its halfway results. At that point, reducers perform total on the qualities in every basin. Since every container in the hash table holds all qualities which relate to an unmistakable key, no gathering is required. Moreover, reducers can perform accumulation on the fly notwithstanding when all mappers are not finished yet.

4) Join Query Optimization: Join Query is a well known issue in huge information territory. However a join issue needs 20 more than two inputs while MapReduce is contrived for preparing solitary information. R. Vernica et al. proposed a 3-stage approach for end-to-end set-comparability joins. They productively parcel the information crosswise over hubs keeping in mind the end goal to adjust the workload and minimize the requirement for replication. Wei Lu et al. research how to perform kNN join utilizing MapReduce. Mappers bunch objects into gatherings, then Reducers perform the kNN join on every gathering of articles independently. To decrease rearranging and computational costs, they desgine a viable mapping system that adventures pruning principles for separation sifting. What's more, two estimated calculations minimize the quantity of reproductions to decrease the rearranging expense.

IV. CHALLENGES

We are currently in the times of Big Data. We can assemble more data from day by day life of each person. The main seven major information drivers are science information, Internet information, account information, cell phone information, sensor information, RFID information and gushing information. Combined with late advances in machine learning and thinking, and also fast ascents in figuring force and capacity, we are changing our capacity to comprehend these inexorably vast, heterogeneous, loud and inadequate information sets gathered from an assortment of sources. We consider there are three vital viewpoints while we experience with issues in preparing Big Data, and we exhibit our perspectives in points of interest as takes after.

A. Big Data Storage and Management

Current innovations of information administration frameworks are not ready to fulfill the necessities of enormous information, and the expanding rate of capacity limit is a great deal not exactly that of information, therefore an upset re-development of data system is urgently required. We have to plan a progressive stockpiling structural engineering. Also, past PC calculations are not ready to viably stockpiling information that is specifically procured from the genuine world, because of the heterogeneity of the huge information. On the other hand, they perform fabulous in preparing homogeneous information. Along these lines, how to re-compose information is one major issue in huge information administration. Virtual server innovation can compound the issue, raising the possibility of over conferred assets, particularly if correspondence is poor between the application, server and capacity executives. We additionally need to explain the bottleneck issues of the high simultaneous I/O and single-named hub in the present Master-Slave framework model.

B. Big Data Computation and Analysis

While preparing a question in huge information, velocity is a huge interest. On the other hand, the procedure might require significant investment on the grounds that generally it can't cross all the related information in the entire database in a brief timeframe. For this situation, list will be an ideal decision. At present, records in huge information are just going for straightforward kind of information, while enormous information is turning out to be more entangled. The mix of suitable record for huge information and state-of-the-art preprocessing innovation will be an attractive arrangement when we experienced this sort of issues. Application parallelization and partition and-overcome is regular computational ideal models for drawing closer enormous information issues. In any case, getting extra computational Assets are not as straightforward as simply moving up to a greater and all the more capable machine on the fly. The conventional serial calculation is wasteful for the enormous information. On the off chance that there is sufficient information parallelism in the application, clients can exploit the cloud's lessened cost model to utilize several PCs for a brief span costs.

C. Big Data Security

We use standard encryption methods when we transferred to the cloud to secure the operations and the storage of data. Here data has been encrypted before move it to the cloud and execute operations on encrypted data without decrypt, this provides same results after calculations as if we worked directly on the plain data. Homomorphic

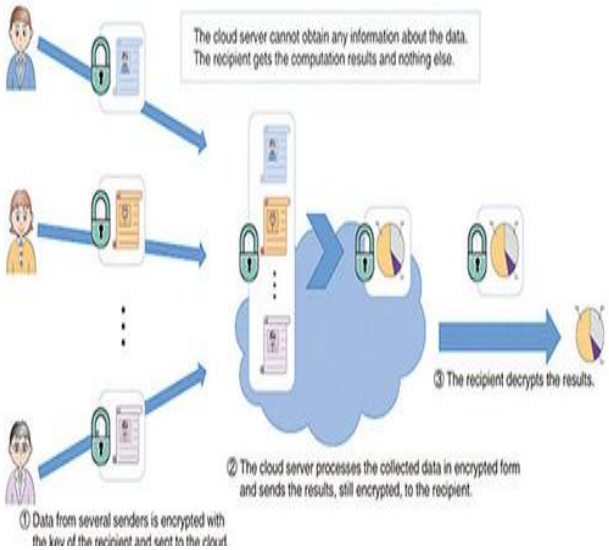

Fig. 2 Homomorphic Encryption

DOI 10.17148/IJARCCE.2016.5347

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

Encryption is used[6] to perform functions on encrypted data without known the private key, thus generating an encrypted result which, when decrypted, matches the result of operations performed on the plaintext, the client only holds the secret key to be decrypted. Fig 2 shows the pictorial representation of homomorphic encryption.

Definition : An encryption is homomorphic if : it is possible to compute Enc(f(x1, x1)) from Enc (x1) and Enc (x2) where f can be both + and x

- $Enc(x_1) + Enc(x_2)$ in R = $Enc(x_1 + x_2 \bmod 2)$
- $Enc(x_1)$ x $Enc(x_2)$ in R = $Enc(x_1 \text{ x } x_2 \bmod 2)$

The first property is called additive homomorphic encryption and the second is multiplicative homomorphic encryption. Here we provide various homomorphic encryption schemes available on cloud computing environment for secure data processing.

1) Unpadded RSA [5]: Let n = pq where p and q are primes. Pick a and b such that ab ≡ 1(mod ϕ(n)). N and b are public while p,q and a are private.
ek (x) = xb mod n
dk (y) = ya mod n
this is also called as multiplicative homomorphic encryption. Suppose x1 and x2 are plain texts, then ek(x1)ek(x2) = x1b x2b mod n = (x1x2)b mod n = ek(x1x2)

2) Paillier Cryptosystem: if the public key is the modulus m and the base g, then the encryption of a message x is ek (x) = gxrm mod m2 , for some random r € { 0,. . . , m-1}. The homomorphic property is then
ek(x1)ek(x2) = (gx1 r1 m) (gx2 r2 m) = gx1+x2(r1r2)m = ek(x1 + x2 mod m2).
This is also called additive homomorphic encryption.

3) El Gamal Cryptosystem [10]: Let p be a prime and pick α € Z*p such that α is the generator of Z*p. Pick a and β such that β ≡ αa (mod p). p, α and β are public; a is private. Let r € Zp-1 be a secret random number. Then, ek(x,r) = (αr mod p , xβr mod p).
This performs multiplicative homomorphic encryption propriety.

- $Z_p^* = \langle g \rangle$, m ∈ Z_p message
- B encrypts a message to A.
- Alice: a random, $h = g^a$, public key = (p, g,A)
- Bob: k random (ephemeral key), $c_1 = g^k$, shared key $K = A^k = g^{ak}$
- $E_A(m) = (c_1,c_2)$, c_2=mK mod p.
- $D_A((c1,c2)) = c_2*(1/K) \bmod p$, $K = c_1^a = g^{ak}$

4) El Gamal Digital Signature :

- $Z_p^* = \langle g \rangle$, m ∈ Z_p message
- A signs message m.
- Alice: A = ga, public key = (p, g,A), secret key = x.
- Alice: k random with gcd(k,p-1)=1
- r = gk (mod p)
- s = (m – xr)(1/k) mod p-1 [m = sk + xr (mod p-1)]
- Signature = (r,s) Verify gm=rshr

5) Okamoto Uchiyama Cryptosystem: It is the multiplicative group of integers modulo n, $(Z/nZ)^*$, where n is of the form p^2q and p and q are large primes. Like many public key cryptosystems, this scheme works in the group $(Z/nZ)^*$. A fundamental difference of this cryptosystem is that here n is a of the form p^2q, where p and q are large primes. This scheme is homomorphic and hence malleable.

Key generation
A public/private key pair is generated as follows:

- Generate large primes *p* and *q* and set $n = p^2q$.
- Choose $g \in (\mathbb{Z}/n\mathbb{Z})^*$ such that $g^p \neq 1 \mod p^2$.
- Let $h = g^n \bmod n$.

The public key is then (*n*, *g*, *h*) and the private key is the factors (*p*, *q*).

Message encryption
To encrypt a message *m*, where *m* is taken to be an element in $\mathbb{Z}/p\mathbb{Z}$

- Select $r \in \mathbb{Z}/n\mathbb{Z}$ at random. Set

$$C = g^m h^r \mod n$$

Message decryption

$$L(x) = \frac{x-1}{p}$$

If we define , then decryption becomes

$$m = \frac{L\left(C^{p-1} \mod p^2\right)}{L\left(g^{p-1} \mod p^2\right)} \mod p$$

6) Naccache –Stern Cryptosystem: It is a homomorphic public-key cryptosystem whose security rests on the higher residuosity problem. Like many public key cryptosystems, this scheme works in the group(Z/nZ)* where n is a product of two large primes. This scheme is homomorphic and hence malleable

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

Key Generation

- Pick a family of k small distinct primes $p_1,...,p_k$.
- Divide the set in half and set $u = \prod_{i=1}^{k/2} p_i$ and $v = \prod_{k/2+1}^{k} p_i$ $\sigma = uv = \prod_{i=1}^{k} p_i$.
- Choose large primes a and b such that both $p = 2au+1$ and $q=2bv+1$ are prime.
- Set $n=pq$.
- Choose a random g mod n such that g has order $\varphi(n)/4$.

The public key is the numbers σ,n,g and the private key is the pair p,q.

When $k=1$ this is essentially the Benaloh cryptosystem.

For Encryption: Pick a random $x \in \mathbb{Z}/n\mathbb{Z}$.

- Calculate $E(m) = x^{\sigma} g^{m} \mod n$

Then *E(m)* is an encryption of the message *m*

For Decryption: To decrypt, we first find *m* mod p_i for each *i*, and then we apply the Chinese remainder theorem to calculate *m* mod σ.

Given a ciphertext *c*, to decrypt, we calculate

- $c_i \equiv c^{\phi(n)/p_i} \mod n$. Thus

$$\begin{aligned} c^{\phi(n)/p_i} &\equiv x^{\sigma\phi(n)/p_i} g^{m\phi(n)/p_i} \mod n \\ &\equiv g^{(m_i+y_i p_i)\phi(n)/p_i} \mod n \\ &\equiv g^{m_i\phi(n)/p_i} \mod n \end{aligned}$$

where $m_i \equiv m \mod p_i$.

- Since p_i is chosen to be small, m_i can be recovered by exhaustive search, i.e. by comparing c_i to $g^{j\phi(n)/p_i}$ for *j* from 1 to p_i-1.
- Once m_i is known for each *i*, *m* can be recovered by a direct application of the Chinese remainder theorem.

7) Fully Homomorphic Encryption: This is used to solve the central open problem in cryptography. Given Encryption $E(m_1),\ldots\ldots E(m_t)$, one can compute the compact ciphertext that encrypt$f(m_{1,\ldots} m_t)$ with efficient computable function f.

Otherwise called Ring Homomorphism, with two operators + and * are satisfied by ring axioms like associative, additive identity, additive inverse, commutative, multiplicative distribution, etc…

Consider the function

$$f = Z_2 \rightarrow Z_2 \text{ given by } f(x)=x^2$$

where x=0 or x=1.

$f(x+y) = (x+y)^2 = x^2 + 2xy + y^2 = x^2 + y^2 = f(x) + f(y)$, where 2xy = because $f(xy) = (xy)^2 = x^2 y^2 = f(x)f(y)$.

KeyGen(m,λ):

1. Choose 2m odd numbers p_i and q_i, $1 \leq i \leq m$, which are mutally prime and of size λ/2 bits.
2. Let fi = piqi and N = i=1m π fi
3. Pick an invertible matrix as K of size 4, K € M(Zn)
4. Compute its inverse as K-1 modulo Zn
5. Output ‹ fi, N, K, K-1 › as K tuple.

Enc(x,Ktuple):

6. Choose a random value r € Zn, r ≠ x.
7. Construct a matrix x(m×3) such that each row has only one element equal to x and other two equal to r.
8. Using Chinese remainder theorem set xj , $1 \leq j \leq 3$ to be solution to the simultaneous congruences xj ≡ xij mod fi where $1 \leq i \leq m$.
9. Ciphertext C = K-1 * diag(x.x1, x2, x3)*K.

Dec(C, KTuple):

Output the plaintext as x =(KCK-1)11.

Table 1 shows the various existing homomorphic encryption cryptosystem according to the following characteristics such as platform, type of homomorphic encryption, privacy and security, applications etc.

DOI 10.17148/IJARCCE.2016.5347

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

TABLE.1 COMPARISON OF VARIOUS HOMOMORPHIC ENCRYPTION ALGORITHM

Characteristics	**Unpadded RSA**	**ElGamal**	**Goldwasser Micali**	**Paillier**	**Benaloh**	**Niccache stern**	**Damgard Jurik**	**Ocamotto-Uchiyama**	**Boneh-Goh Nissm**	**Ishai Paskin**
Property	Multilicative	Multilicative	Single bit additive encryption	Additive	Multiplicative and Additive	Multili -cative	Multiplicative and Additive	Multilicative	Multiplicative and Additive	Multiplicative and Additive
Platform	Cloud environ -ment	Cloud environ -ment	Cloud environ -ment	Cloud environ-ment	Cloud environ -ment	Cloud environ -ment	Cloud environ -ment	Cloud environ -ment	Cloud environ -ment	Cloud environ -ment
Encryption Keys	Different keys for encryption and decryption	Different keys for encryption and decryption	Different keys for encryption and decryption in bitwise order	Different keys for encryption and decryption	Different keys for encryption and decryption in blockwise	Different keys for encryption and decryption used only in the group $(Z/nZ)^*$	Different keys for encryption and decryption	Different keys for encryption and decryption used only in the group $(Z/nZ)^*$	Different keys for encryption and decryption	Different keys for encryption and decryption
Security and Seclusion of data applied to	Secure storage and communication process provided to cloud server	Secure storage and communication process provided to cloud server	Secure storage and communication process provided to cloud server	Secure storage and communication process provided to cloud server	Secure storage and communication process provided to cloud server	Secure storage and communication process provided to cloud server	Secure storage and communication process provided to cloud server	Secure storage and communication process provided to cloud server	Secure storage and communication process provided to cloud server	Secure storage and communication process provided to cloud server
Type of Homomorphic Encryption type (Partially or Fully)	Fully (Somewhat) Homomorphic crypto system	Fully (Somewhat) Homomorphic crypto system	Fully (Somewhat) Homomorphic crypto system	Fully (Somewhat) Homomorphic crypto system	Partially Homomorphic crypto system	Partially Homomorphic crypto system	Partially Homomorphic crypto system	Partially Homomorphic crypto system	Partially Homomorphic crypto system	Partially Homomorphic crypto system

V. CONCLUSION

Big Data is not new idea but rather exceptionally difficult. It calls for versatile stockpiling list and a dispersed way to deal with recover required results close ongoing. Data is too huge to process traditionally. By and by, big data will be unpredictable and exist persistently amid every single enormous test, which are the huge open doors for us. This paper depicted a methodical stream of overview on the huge information handling in the connection of distributed computing. We separately examined the key issues, including distributed storage and figuring construction modeling, prominent parallel preparing structure, real applications and enhancement of Map Reduce. Finally we described several homomorphic encryption techniques in secure big data processing in cloud environments. In future, we will analyze the performance of Homomorphic Encryption cryptosystems with the complexity behavior of public key length and time depends of on the encrypted messages from cloud provides.

REFERENCES

[1] Changqing Ji, Yu Li, Wenming Qiu, Uchechukwu Awada, Keqiu Li "Big Data Processing in Cloud Computing Environments", International Symposium on Pervasive Systems, Algorithms and Networks, vol. 9, pp.17-23, 2012

[2] "Big data: science in the petabyte era," Nature 455(7209): 1,2008.

[3] Douglas and Latency, "The importance of 'big data': A definition," 2008.

[4] Maha TEBAA and Said EL " Secure Cloud Computing through Homomorphic Encryption" International Journal of Advancements in Computing Technology, vol. 5, no.16, 2013.

[5] Sigrun Goluch, "The development of homomorphic cryptography from RSA to Gentry's privacy homomorphism", http://dmg .tuwien. ac.at.

[6] Ronald L.Rivest, Leonard Adleman, and Michael L. Dertouzos, "On Data Banks and Privacy Homomorphism", chapter on Data Banks and Privacy Homomorphisms, pages 169-180. Academic Press, 1978.

[7] Pascal Paillier, "Public-key cryptosystems based on composite degree residuosity classes", In 18th Annual Eurocrypt Conference(EUROCRYPT'99), Prague, Czech Republic, vol 1952, 1999.

[8] Kun Peng, Colin Boyd, Ed Dawson, "A Multiplicative Homomorphic Sealed-Bid Auction Based on Doldwasser-Micali Encryption", vol. 3650, pp 374-388, Springer, 2005.

[9] R.Rivest, A.Shamir, and L.Adleman. " A method for obtaining digital signatures and public key cryptosystems", Communications of the ACM, 21(2): 120-126, 1978. Computer Science, Pages 223-238, Springer, 1999.

[10] Osman Ugus and al. "Performance of Additive Homomorphic EC-ElGamal Encryption for TinyPeds", 6th Fachgesprach Drahtlose Sensornetze, pp. 55-58, July 2007.

[11] Craig Gentry, "A Fully Homomorphic Encryption Scheme", 2009.

[12] J.Dean and S.Ghemawat, "Mapreduce: Simplified data processing on large clusters." Communications of the ACM, vol.51, no.1, pp 107-113, 2008.

[13] D.Jiang, A.Tung, and G.Chen, "Map-Join-Reduce: Toward scalable and eficent data analysis on large clusters,"Knowledge and data engineering, IEEE transactions on, vol. 3, no. 1-2, pp. 494-505, 2010.

[14] D.Bothakur, "The Hadoop Distributed File System: Architecture and design," Hadoop Project Website, vol. 11, 2007.

[15] Y. Xu, P.Kostamaa and L. Gao, "Integrating hadoop and parallel dbms", in Proceedings of the 2010 international conference on Management of data. ACM, 2010, pp.969-974