Journal of Engineering and Applied Sciences 12 (Special Issue 5): 7069-7073, 2017
ISSN: 1816-949X

Data Security in Hadoop: A Technological Perspective and Review

Varsha Dipak Tayde and Praveen Kaushik
Department of Computer Science and Engineering,
Maulana Azad National Institute of Technology (MANIT), 462003 Bhopal, India

Abstract: Today, world is going to become more digital. As everyone using internet today, large amount of data gets generated every day. Every single person has his own account on different social sites, shopping sites, commercial sites. Billion kilobytes of data get generated every single day. This data may important for user. We need to store those data for future use. There is need to provide privacy and security to such data. Hadoop is used to store and analyse the data. There are different tools which research on the top of Hadoop stack to provide security to data. The motive of this study to provide review of existing privacy and security methods for Hadoop security.

Key words: Bigdata, Hadoop, HDFS, MapReduce, risk, security, privacy

INTRODUCTION

Apache Hadoopis used for storing and processing large amount of data. Hadoop contain major 2 parts: HDFS and MapReduce. HDFS stand for Hadoop Distributed File System (Shvachko *et al*., 2010). It is related to storing of data. MapReduce is another phase of Hadoop. It is related to processing of data. It supports parallel processing. MapReduce contains two parts JobTracker and TaskTracker. JobTracker is master node while TaskTracker is slave node. The MapReduce framework contains a single master JobTracker and one TaskTracker per cluster-node. Resource management is done by master. The TaskTracker execute the tasks as directed by the master and send status information to the master periodically (Zhang *et al*., 2014). Figure 1 shows structure of Hadoop cluster.

Key challenges in Hadoop: Hadoop is a distributed process framework and it was not originally developed for security. It was meant to operate in trusted environments (Derbeko *et al*., 2016).

Confidentiality: Confidentiality means data can see by those people who are actually supposed to see.

Security: Data stored in databases should be protected, it cannot be access by unauthorized person.

Privacy: Privacy means controlling spreading of information to unauthenticated person.

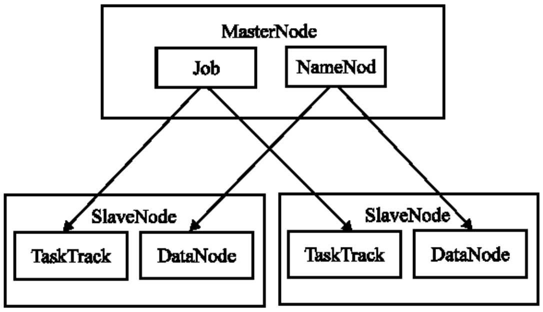

Fig. 1: Structure of Hadoop cluster with master and slave nodes (Shvachko *et al*., 2010)

Access control: Access control means controlling access of sensitive data. Some challenges are overcome as (Samuel *et al*., 2015):

Authentication: Authentication is a process in which it provide three way handshake to identify the identity of legitimate user. Kerberos is a computer network authentication protocol. It is used by Hadoop security models to provide security to data. It allows different nodes in network to communicate and share data. Kerberos uses a delegation token. A delegation token allows users authenticate themselves with the Namenode using Kerberos.

Encryption: It is a process of converting a text form into encoded format so it can't read by outsiders. There are many different algorithms that encrypt data and can retrieve it back.

Corresponding Author: Varsha Dipak Tayde, Department of Computer Science and Engineering,
Maulana Azad National Institute of Technology (MANIT), 462003 Bhopal, India

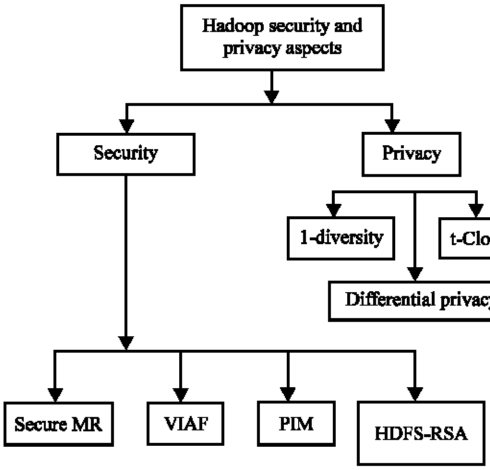

Fig. 2: Hierarchical diagram of Hadoop privacy and security algorithms

Watermarking: It is process of probing a bit of information into data as a copy write, so that any unauthorized person cannot use that data.

Anonymization: It is a process of adding a noise to original data and converting the data into non-readable format so that any unwanted person cannot access original data.

LITERATURE REVIEW

Security in Hadoop can be achieved in two ways: providing security to Hadoop and applying privacy to the computational data. Some of existing privacy and security solutions for MapReduce and HDFS are as follows (Fig. 2):

SecureMR (secureMapReduce): It is a decentralized verification based framework. It is made up of five security components: Secure manager, scheduler, task executor, committer and verifier. Manager and scheduler are research on master process and perform task duplication and assignment. The commitment protocol is used for communication between master process and reducers. The verification protocol is used for communication between mapper and reducers (Wei *et al*., 2009).

Communication design: The communication between master process and reducers is carried out using commitment protocol. The communication between mapper and reducers is carried out using verification protocol (Fig. 3).

Commitment protocol

Assign: The Assign message includes a monotonically increasing identity $ID_{Map}$of a map task and an input data block location $Data_{Loc}$ which is signed by master and encrypted using K_{pubM}, public key of the mapper.

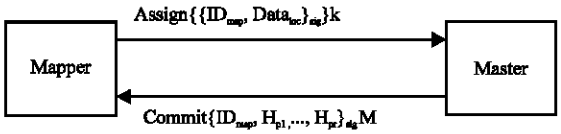
$Assign\{\{ID_{map}, Data_{loc}\}_{sig}\}k$
Mapper
Master
$Commit\{ID_{map}, H_{p1},..., H_{pr}\}_{sig}M$

Fig. 3: Commitment protocol

Commit: Manager verifies the signature using K_{pubM} public key. In this master process assign task to Mapper using encrypted message and allow Mapper to send commit message along with hash value. It uses public key encryption techniques.

Verification protocol: It contains 5 steps: Assign, Notify, Request, Response, Report. In verification protocol, reducers verify output and check hash value which is submitted.

Assign: Master send assign message to reducer using K_{pubM}.

Notify: When manager receive completion event then master send notify message to reducer.

Request: Verifier send data request to committer of the mapper along with $Ticket_M$.

Response: Committer verifies the ticket and reducers signature and send response message to verifier.

Report: Verifier send final report to manager whether there is consistency or not.

VIAF (Verification-based Integrity Assurance Framework): It verifies replication of tasks and builds trust between master and mapper process. This framework contains trusted workers and untrusted workers. The verification task, reduce task and master process are execute on trusted workers. The map task executes on untrusted workers (Wang and Wei, 2011).

The survival chance can calculate by assigning replicate task to collusive workers. With survival chance, cheat probability can derive overhead can calculated by deriving number of rescheduling activities. Verification over head is nothing but execution of verification task. Figure 4 shows flow of VIAF in detail (Algorithm 1).

Algorithm 1; Verification task:

```
1. Master process assign task to different workers
2. Each worker perform mapping task and return hash value of result to master
3. Master process verifies the hash value
4. If hash values are different then it concludes the malicious worker else store the result of both workers
```

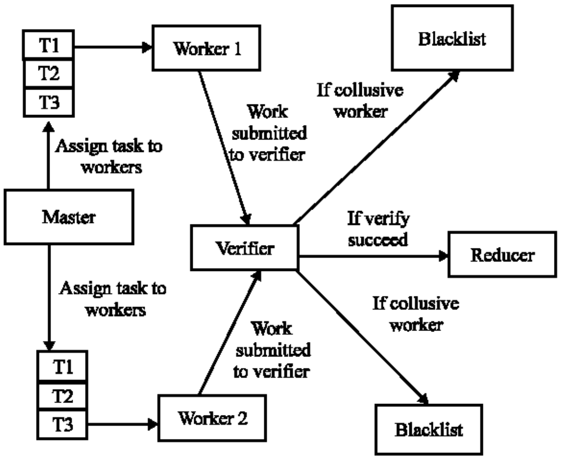

Fig. 4: Data flow of VIAF

5. Master process store task information in cache
6. Master process execute verification task to verify consistent task
7. Finally, conclude the result as workers are non-malicious if verification results are different

In VIAF the verification overhead calculated as:

$$V0=\frac{v\left(1\text{-}m\left(1\text{-}c\right)r\right)}{1\text{-}m\left(1\text{-}c\right)r\text{-}m^2c^2\left(1\text{-}\Delta^2\right)\left(1\text{-}v\right)}$$

Where:

v = The probability that a task returning consistent results is verified by verifier
m = The malicious worker ratio out of all workers
r = The probability that a non-collusive worker determine to commit a cheat when assigned a task
c = Collusive worker ratio out of malicious workers

iBigtable: It is a enhance form of Bigtable. Bigtable is a distributed storage system which is mainly designed for structured data where each column store key-value pair. A table in Bigtable is multidimensional sorted map. It contains Root table, Metadata table and User table. It is integrity based model. It ensures integrity of data by using centralized and distributed authenticated data structure. It uses Markle Hash Tree (MHT) which is one type of data structure contains hash values. The root hash of hash table is stored on user side. Whenever any data is updated corresponding Verification Object (VO) gets generated and data structure also gets updated. iBigtable supports single-level MHT, Multi-level MHT and two-level MHT. Data insertion contains 2 steps: tablet splits and tablet merge. Tablet gets divided into two sub trees, data get inserted after range query, VO get generated and both sub trees get adjusted then get merge together (Wei *et al*., 2013). Communication between user and tablets are shown in Fig. 5 (Algorithm 2).

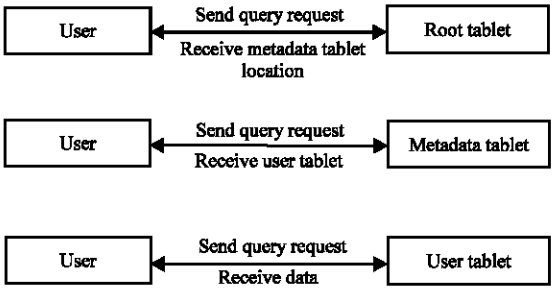

Fig. 5: Bigtable: tablet and user communication

Algorithm 2; Merge algorithm of iBigtable:

```
k ← the least key in Tr
hl ← GetHeight(Tl)
hr ← GetHeight(Tr)
hmin ← GetM in(hl , hr)
if hl≤hr then
    plm ← the leftmost node in Tr at hmin
    add k to plm
    pmerged ← merge the root of Tl and plm
    ifIsValidNode(pmerged) is false then
        run a node split process for pmerged
    end if
    return Tm ← Tr
else
    prm ← the rightmost node in Tl at hmin
    add k to prm
    pmerged ← merge the root of Tr and prm
    ifIsValidNode(pmerged) is false then
        run a standard node split process for pmerged
    end if
    return Tm ← Tl
end if
```

Integrity verification Algorithm of iBigtable:

1. Client request for data to tablet server
2. Tablet server generate VO for data sent to client
3. Tablet server sends data with VO to client
4. Client runs verification algorithm to check integrity of data

HDFS-RSA: It is a hybrid method enhances HDFS by embedding encryption technique into reading writing operations of the file system. It encrypts data at time of storing and decrypts it while retrieving. Hybrid encryption uses symmetric key and public algorithms. It uses AES as block cipher and RC4 as stream cipher. In this method data is divided to form fixed-sizes blocks, d1, d2, d3, ..., dn. Then theseblocks get encrypted using a random key k and stream cipher then encrypted using public key scheme. This method has two parts: one is HDFS-RSA for authentication and another one is HDFS-Pairing for data to store (Lin *et al*., 2012). This method consists 4 Algorithms:

Setup: In this μ is generated as:

$$\mu=\left(g, h, \bar{e}, G_1, G_2, p\right)$$

Key generation algorithm: For generating the key pair, we randomly select the value a from the Z_p and set the value of pk = g^a and sk = a.

Encryption algorithm (pk, f): For generating the cipher text of the message fЄG is generated as:

$$C = (\alpha, \beta) = (g^r, F\bar{e}(g^a, h^r))$$

where, r is randomly selected number from Z_p. DEC algorithm (sk, C): Let C = (α, β) by using the secret key (sk, F) is obtained as follows:

$$F = \frac{\beta}{\bar{e}(\alpha, h^{sk})}$$

PIIM (Probe-Injection-based Identification Method): It is one of watermark based algorithm. A probe is data injected into original dataset. The result of probe set is computed before input data set. The probe has two attributes: data value (value in data set) and location (position after injection) (Ding *et al.*, 2014; Karthik and Rangaswamy, 2015). PIIM contains following steps:

Probe injection: Probes are injected into original data. The probe data $D_{p\ is}$generated.

Computation: D' is submitted to MapReduce framework and computation is implemented.

Result analysis: After the computation, R' is traversed and all Result of Probes (RoPs) in R' are checked to determine if they are identical to the result from previous computation:

- If all RoPs are correct, then the total result of the computation is correct
- If some RoPs are computed incorrectly, the entire result may be incorrect and may be rejected

Result recovery: For accepted result R', influence of injected probes on final result must be cleaned up. Result R which is actually result of original input data, is obtained.

Differential privacy: It is a privacy algorithm. It can define as probability of output does not depend on any individual in data. A randomized function k gives ε differential privacy if, for all data sets D1, D2 such that one can obtained from the other by modifying a single record. It added noise to data at certain threshold. It maximizes accuracy of queries from databases while minimizing disclosure of identity (Fung *et al.*, 2010; Dwork, 2006).

$$Dp = f(x)+Lap(\Delta(f)$$
$$\Delta f = max/f(D1)-f(D2)|$$

Where:

D_p = Differential privacy
F(x) = Requested data
Lap = Laplace distribution
Δ(f) = Function sensitivity
D1 = Data Block 1
D2 = Data Block 2

t-Closeness (privacy beyond k-anonymity and l-diversity): K-anonymity and L-diversity is privacy preserving algorithms. In k-anonymity, data contain k rows cannot be different from k-1 rows. The l-diversity, each equivalence class must have at least l distinct sensitive values. It's difficult to achieve. It is not sufficient to prevent from attribute disclosure. t-Closeness can be defined as distance between distribution of a sensitive attribute and distribution of attribute in whole tablenot more the threshold value (Zhou *et al.*, 2008; Zhang *et al.*, 2012). Let P = (p1, p2, p3, ...,) and Q = (q1, q2, q3, ...,) are two distributions. Distance can be measure as:

$$D[P, Q] = \sum_{i=1}^{m} \frac{1}{2} |p_i - q_i|$$

Let, E1 and E2 are two equivalence classes and P, P1, P2 are distribution of sensitive attributes in E1 and E2 then:

$$D(P, Q) \le \frac{|E1|}{|E1|+|E2|} D[P1, Q] + \frac{|E1|}{|E1|+|E2|} D(P2, Q)$$

COMPARISON AND ANALYSIS

The comparison of these algorithms on important parameters as security, privacy, access control and confidentiality.

SecureMR is verification based algorithm. It ensures integrity and prevents reply attacks. VIAF provides security and integrity. It is redundancy based approach. HDFS-RSA is a hybrid algorithm. It based on encryption. It provides security and integrity. PIIM secure data by probing extra information into data. Differential privacy adds noise to data and provides privacy but it lacks to provide security. t-Closeness is a privacy algorithm but it doesn't provide data integrity (Table 1).

Table 1: Comparison of different algorithms

Algorithems	Security	Privacy	Confidentiality	Access control
SecureMR	High	Low	Decent	Easy
VIAF	High	Low	Better	Medium
iBigTable	Medium	Low	Good	Easy
HDFS-RSA	High	Low	Decent	Easy
PIIM	High	Low	Better	Easy
Defferential privacy	Medium	Medium	Better	Easy
T-Closeness	Medium	High	Good	Medium

CONCLUSION

In this study, various privacy and security methods are observed which provide security to the data. Today, where big data performs vital role, security of data is a measure issue. As Hadoop doesn't contain any fixed security mechanism and it is adopted by various industries to process data, it is requirement to provide strong security solution to Hadoop.

RECOMENDATION

For the future research, we can apply such algorithm which full fill all aspects of Hadoop security such as fine anonymization techniques, hybrid of generalization and bucketization or k-anonymity, etc.

REFERENCES

Derbeko, P., S. Dolev, E. Gudes and S. Sharma, 2016. Security and privacy aspects in MapReduce on clouds: A survey. Comput. Sci. Rev., 20: 1-28.

Ding, Y., H. Wang, S. Chen, X. Tang and H. Fu and P. Shi, 2014. PIIM: Method of identifying malicious workers in the MapReduce system with an open environment. Proceedings of the 2014 IEEE 8th International Symposium on Service Oriented System Engineering (SOSE), April 7-11, 2014, IEEE, Oxford, UK., SBN:978-1-4799-3616-8, pp: 326-331.

Dwork, C., 2006. Differential privacy. Proceedings of the 33rd International Colloquium on Automata, Languages and Programming, July 10-14, 2006, Venice, Italy, pp: 1-12.

Fung, B., K. Wang, R. Chen and P.S. Yu, 2010. Privacy-preserving data publishing: A survey of recent developments. ACM. Comput. Surv. CSUR., 42: 515-556.

Karthik, K. and M.D. Rangaswamy, 2015. A novel three-tier protection for digital images using blind watermarking scheme. Intl. J. Adv. Trends Comput. Sci. Eng., 4: 15-19.

Lin, H.Y., S.T. Shen, W.G. Tzeng and B.S.P. Lin, 2012. Toward data confidentiality via integrating hybrid encryption schemes and Hadoop distributed file system. Proceedings of the IEEE 26th International Conference on Advanced Information Networking and Applications (AINA), March 26-29, 2012, IEEE, Fukuoka, Japan, ISBN:978-1-4673-0714-7, pp: 740-747.

Samuel, S.J., K. RVP, K. Sashidhar and C.R. Bharathi, 2015. A survey on big data and its research challenges. ARPN. J. Eng. Appl. Sci., 10: 3343-3347.

Shvachko, K., H. Kuang, S. Radia and R. Chansler, 2010. The hadoop distributed file system. Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), May 3-7, 2010, IEEE, Nevada, USA, ISBN:978-1-4244-7152-2, pp: 1-10.

Wang, Y. and J. Wei, 2011. Viaf: Verification-based integrity assurance framework for mapreduce. Proceedings of the 2011 IEEE International Conference on Cloud Computing (CLOUD), July 4-9, 2011, IEEE, Washington, USA., ISBN:978-1-4577-0836-7, pp: 300-307.

Wei, W., J. Du, T. Yu and X. Gu, 2009. SecureMR: A service integrity assurance framework for mapreduce. Proceedings of the Annual Computer Security Applications Conference, December 7-11, 2009, Hawaii, USA., pp: 73-82.

Wei, W., T. Yu and R. Xue, 2013. IBigTable: Practical data integrity for bigtable in public cloud. Proceedings of the 3rd ACM Conference on Data and Application Security and Privacy, February 18-20, 2013, ACM, New York, USA., ISBN:978-1-4503-1890-7, pp: 341-352.

Zhang, X., C. Liu, S. Nepal, C. Yang and J. Chen, 2014. Privacy Preservation Over Big Data in Cloud Systems. In: Security, Privacy and Trust in Cloud Systems, Nepal, S. and P. Mukaddim (Eds.). Springer, Berlin, Germany, ISBN:978-3-642-38586-5, pp: 239-257.

Zhang, X., C. Liu, S. Nepal, W. Dou and J. Chen, 2012. Privacy-preserving layer over MapReduce on cloud. Proceedings of the 2012 2nd International Conference on Cloud and Green Computing (CGC), November 1-3, 2012, IEEE, Xiangtan, China, ISBN:978-1-4673-3027-5, pp: 304-310.

Zhou, B., J. Pei and W.S. Luk, 2008. A brief survey on anonymization techniques for privacy preserving publishing of social network data. ACM SIGKDD Explo Rations Newslett., 10: 12-22.