Applied Computer Science, vol. 15, no. 4, pp. 63–74
doi:10.23743/acs-2019-30

Submitted: 2019-11-04
Revised: 2019-12-07
Accepted: 2019-12-24

BigData, Hadoop, RSA, Paillier, Cryptography

Shadan Mohammed Jihad ABDALWAHID[*], *Raghad Zuhair YOUSIF*[**] , *Shahab Wahhab KAREEM*[*]

ENHANCING APPROACH USING HYBRID PAILLER AND RSA FOR INFORMATION SECURITY IN BIGDATA

Abstract

The amount of data processed and stored in the cloud is growing dramatically. The traditional storage devices at both hardware and software levels cannot meet the requirement of the cloud. This fact motivates the need for a platform which can handle this problem. Hadoop is a deployed platform proposed to overcome this big data problem which often uses MapReduce architecture to process vast amounts of data of the cloud system. Hadoop has no strategy to assure the safety and confidentiality of the files saved inside the Hadoop distributed File system (HDFS). In the cloud, the protection of sensitive data is a critical issue in which data encryption schemes plays avital rule. This research proposes a hybrid system between two well-known asymmetric key cryptosystems (RSA, and Paillier) to encrypt the files stored in HDFS. Thus before saving data in HDFS, the proposed cryptosystem is utilized for encrypting the data. Each user of the cloud might upload files in two ways, non-safe or secure. The hybrid system shows higher computational complexity and less latency in comparison to the RSA cryptosystem alone.

1. INTRODUCTION

Cloud computing has attracted increasing attention since the last few years. Cloud computing provides users with a wide range of resources, such as computing platforms, storage, computing power, and internet applications. Amazon, Google,

[*] Erbil Polytechnic University, Erbil Technical Engineering College, Department of Information System Engineering, Erbil, Iraq, shadanbaban@yahoo.com, shahab.karim@epu.edu.iq
[**] Catholic University in Erbil, Information Technology Department, Erbil, Iraq, raghad.yousif@cue.edu.krd

IBM, Microsoft, etc. are the biggest cloud available in the markets now. With a growing number of companies utilizing resources in the cloud, data from different users need to be protected. Cloud computing is presently used in a tremendous amount in various fields. In daily life, huge amounts of data produced. Consumers use cloud computing services to store this huge amount of data. Some of the major challenges cloud computing faces are to secure, protect and process the data that is the user's property (Merla & Liang, 2017; Kareem, 2009). Big data refers to the processing and retrieval of massive data collection. Big data must also be concerned with the collection of essential and sensitive data from social sites and issues of government and hence, security. This collected data has to encrypt by using appropriate algorithms to secure them. The features of Big Data can be identified in term of four V's (Hilbert, 2016): Volume, Velocity, Variety and Veracity. Every subject holds its job of remaining in Big data. Thus, volume: the amount of data produced and might be stored it could be in the level of different size terabytes rather Petabytes. Variety: which are the data forms and its kinds, structure, unstructured and semi-structured. Velocity: which indicates an input and the output rates of data streams generated and stored in the system. In this context, an abstraction provided in a way that the systems within big data can eventually, independently collect data from the outgoing or incoming clip. Veracity: It's a term of data quality; this context is also Refers to data confidentiality, data privacy, integrity, and availability. Establishments must be grantee that the data and the analyses conducted on the data are precise. Big data processing has become almost pivotal for many governments and business applications with an incredible rate of data generated, collected and analyzed by computer systems (Amrulla, Mourya, Sanikommu & Afroz, 2018). Thus, many factors have participated in data huge increment like the emerge of IoT, object localization and tracking, besides the growing adoption of healthcare devices which gather personal statistics. This prevalence of big data has some disadvantages. The data collected usually involves some personal information about persons, or it is including secrets that would be problematic if the opponent discovers them. Criminal groups create underground markets for the possession and purchase of stolen personal information (Motoyama, McCoy, Levchenko, Savage & Voelker, 2011). Government intelligence services rely on personal, corporate and adverse government eavesdropping and competitive advantage systems. Most recent, highly publicized cyber-attacks against commercial attacks demonstrate this potential for damage, and government targets, it pays millions of dollars to these organizations and causes severe damage to the affected individuals and organizations (Kareem & Hussein, 2017). Furthermore, protection across cloud services is under its developing stage; a huge quantity about safety vulnerabilities would risk data in the cloud. The cloud administrators have no clue as to where and in what format the data is stored. Thus, adequate security measures must be modified to preserve the data, essentially of information leakage plus manipulation. Also, processing/analyzing enormous data in the data center is a dangerous problem in the cloud. Different

spread structures like HADOOP have recently been available (Li, Wang, Zhao, Pu, Zhu & Song, 2015; Ahamad, Akhtar, Hameed, 2019), like Google File System (Yang, Lin & Liu, 2013), which is developed to store and process Big Data. Still, the spread HADOOP structure is common with manufacturing and investigation centres. HADOOP holds pair organizations of functionalities, (i) For storage of large and unstructured data sets (HDFS), has been employed, and (ii) Map-Reduce framework for hug data manipulation. HADOOP usually serves among applying that have huge of data links also petabytes. As a literature survey Chao YANG et al. (Yang, Lin & Liu, 2013). Suggest a triple encryption scheme for enhancing the security of Hadoop. Thus the encryption of HDFS files is achieved by using DEA (Data Encryption Algorithm), whereas RSA has been used in the encryption of data key. Eventually, the RSA private key is secured using the IDEA (International Data Encryption Algorithm). Huixiang Zhou et al. (Zhou & Wen, 2014) They present CP-ABE (Ciphertext policy attribute-based encryption) scheme for access control instead of the traditional schemes like PKI, which requires all relevant customer data to be sent to the resource provider, thus destroying the privacy of the user, and takes more bandwidth and overhead processing. Masoumeh Rezaei Jam et al. (Jam, Khanli, Akbari & Javan, 2014) point out that currently, the core technology of cloud computing are services security and data privacy. A security mechanism based on Kerberos protocol for authentication firewalls of perimeter level security was presented (Ismael, Youail & Kareem, 2014). Security leak was handled by implementing the Apache sentry for access control, triple encryption of data using RSA, DES, IDEA algorithms, was proposed in protecting file system based on fully homomorphic encryption. R. PARMAR1 et al. (Parmar, Roy, Bhattacharyya, Bandyopadhyay & Kim, 2017), proposed a novel method which can be used to secure Hadoop, a cost-effective technique works in Hadoop cluster to give it 3-D security. Muhammad Usama, et al. (Usama & Zakaria, 2017), proposed Data compression and encryption for Hadoop. Hence a combined compression and encryption scheme was presented based on Tent Map and Piece-wise Linear Chaotic Map (PWLM), the proposed approach implements a masking pseudorandom keystream that strengthens the encryption process. The proposed algorithm, providing robust encryption and compression schemes.

HADOOP does not incorporate security mechanisms. The Application of ciphering algorithms in HADOOP data encryption, then storing them at HDFS has reported in several works. Ciphering schemes perform different replacements and do some manipulation on the clear message to transforms it into ciphertext, which must be random and incomprehensible. Different ciphered schemes were developed and employed for the sake of information security. Hence the two main categories are: (i) Symmetric-key (secret key) cryptosystems (Chandra, Bhattacharyya, Paira & Alam, 2014) like Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Triple DES (ii) Asymmetric-key (public key) algorithms (Chandra, , Alam, Paira & Sanyal, 2014) like Elliptic Curve Diffie-Hellman (ECDH) and RSA. The proposed approach is considered as an

attempt to improve what was presented by the paper (Usama & Zakaria, 2017) at both of encipherment /decipherment procedures for securing files of Big Data-based Hadoop-integrated AES and OTP algorithms (Mahmoud, Hegazy & Khafagy, 2018). An architecture to secure Hadoop was examined in paper (Park & Lee, 2013). Thus for data encryption and decryption, AES encryption/decryption classes are added. Implement two HDFS pairing integrations and HDFS-RSA (Shetty & Manjaiah, 2016) applied since various amazing kinds of extensions from HDFS. Analyses demonstrated adequate expenses for understanding processes also significant overhead for recording actions (Yang, Lin & Liu, 2013). Three encryption scheme (Inukollu, Arsi & Ravuri, 2014) integrated with cloud data storage system depending on Hadoop to encrypt files in HDFS based on DES and RSA then refer to IDEA for securing the RSA private key for the users. The encryption of the HDFS files implemented when they stored in a buffer after uploading data to HDFS. In this work, a modified asymmetric key cryptosystem is being presented to secure Big data. The following is the organization of this paper: Section II outlines the security framework. Section III, based on HDFS and MapReduce, presents the Big Data at HADOOP. Section IV discusses the proposed optimized hybrid encipherment algorithm and compare it with the classical public-key cryptosystems before applying it to secure Big Data at HADOOP. Section V presents the discussion of the simulation results. Finally, section VI list the conclusions.

2. SECURITY ISSUES

Big data is about data storage, data processing, data recovery. Many technologies, such as memory management, transaction management, visualization and networking, are used for these purposes. These technologies security issues are also applicable to big data. Big data's four major security issues are authentication, data level, network level and generic matters (Bhandarkar, 2010; Raghad, Kareem & Hasan, 2016).

2.1. Authentication Level Issues

A lot of clusters and nodes are present. Each node has priorities or rights that are different. Administrative nodes can access any data. But sometimes it will steal or manipulate the critical user data if any malicious node has organizational priority. Many nodes are joining clusters for faster execution with parallel processing. Any malicious node can disturb the group in the event of no authentication. Logging in big data plays an important role. If logging not provided, no activity that modifies or deletes data will record. If the new node joins the cluster, the absence of logging will not recognize it. Users may also sometimes use malicious data unless the log provided.

2.2. Data level issues

Data is an essential part of big data and also plays a vital role. Data is nothing but some of the government or social networking sites necessary and personal information about us. The main issues that could be handled by the data level are integrity and availability of data like protection and distribution of data. Big data environments such as Hadoop store the data as it is without encryption to improve efficiency. If the hacker accesses the machines, he/she cannot be stopped. Information stored in a distributed data store for quick access in many nodes with replicas. But if hacker deletes or manipulates any reproduction or information from another node, then it will be difficult to recover that data.

2.3. Network-level issues

There are many nodes in clusters, and these nodes are used to compute or process data. This data processing can be done anywhere between the cluster nodes. It is, therefore, difficult to determine which node data is processed. It will be complicated because of this difficulty on which node safety should be provided. Two or more nodes can communicate or share their data/resources via the network. RPC (Remote Procedure Call) often used for network communication. But until and unless it is encrypted, RPC will not be secure.

2.4. General level issues

Many technologies are also used in the big data environment to process the data for some traditional security tools for security purposes. Over the years, traditional tools have been developed. Thus with the new distributed form of big data, these tools may not be performed well. As big data uses many data storage, data processing and data recovery technologies, there may be some complexities due to these different technologies.

3. BIG DATA AND HADOOP

Hadoop architecture consists mainly of two primary components which are: (HDFS) to store Big Data and MapReduce to analyze Big Data (Bhardwaj, Singh, Vanraj & Narayan, 2015). HDFS is a file management system used for the distributed storage of massive datasets on the Hadoop cluster in with a default block size of 64 MB (Dubey, Jain & Mittal, 2015). After storing the input files in HDFS, then it manipulated with MapReduce software. Eventually, the results moved to the output folder of HDFS (Dean & Ghemawat, 2008). MapReduce in Hadoop is an application software designed for processing huge volumes of data sets over machine set (Zhou & Wen, 2014). MapReduce is the core scheme used by the Hadoop system

for spreading a bunch of work. Each input data, which inhabits throughout the cluster on a distributed file system, is divided into groups of equal size to facilitate and simplify in a suitable, and almost error-free manner the enormous volumes from processing the data under parallel at huge organizations regarding tools. As specified by the name, MapReduce involves two –stages like data calculation within Hadoop, the initial stage is the map, and the other stage signifies reducing, i.e. a huge amount from data sets is transformed inside structured key-value pairs and provided since inputs (Dean & Ghemawat, 2008).

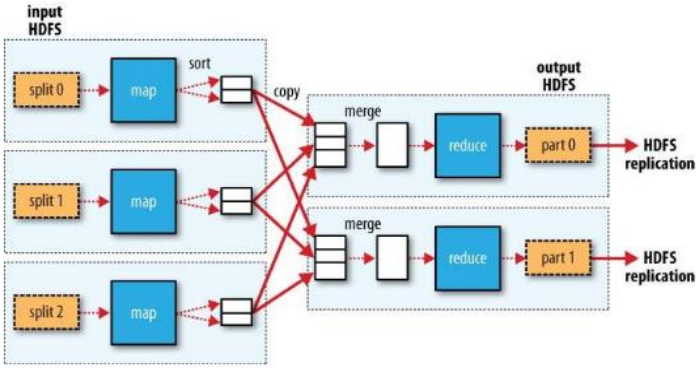

Fig. 1. MapReduce Data Stream

Figure 1 shows the MapReduce computation data flow. The mapper doesn't write directly on disk but uses the benefit from buffering some writings. Every mapper becomes a round buffer of memory among default size is 100 MB which can do modified through improving each property of (io. sort. mb). That makes a rapid flush. If the buffer is loaded up before specific inception, it initiates the transfer to the disk the content of the barrier. Before each spill appears on the drive, each thread separations these data based on the reducers that require ongoing background thread performs any sort of in-memory within the key-based partition before the spill takes place to the disk. If a mixer is started, it applies the output of the in-memory kind (Dean & Ghemawat, 2008).

4. PROPOSED ALGORITHMS

Hadoop is the primary provider of large-scale cloud data processing and storage, and is, therefore, uses some techniques of encryption to ensure security. This paper introduces new technology – this technique based on cascading two public-key cryptosystems (RSA and Paillier) (Kareem, 2009). Hybridization's a way to over-come the limitations of using each cryptosystem individually and to improve security. It is considered that all the files written to HDFS must be previously encrypted.

The HDFS client is responsible for keys generations (public and private keys). Then the proposed hybrid system is employed While the file caching in HDFS encrypted it utilizing the unstructured data for the file. The HDFS starts sending an encrypted file on the data nodes. These stages shown in Figure 2.

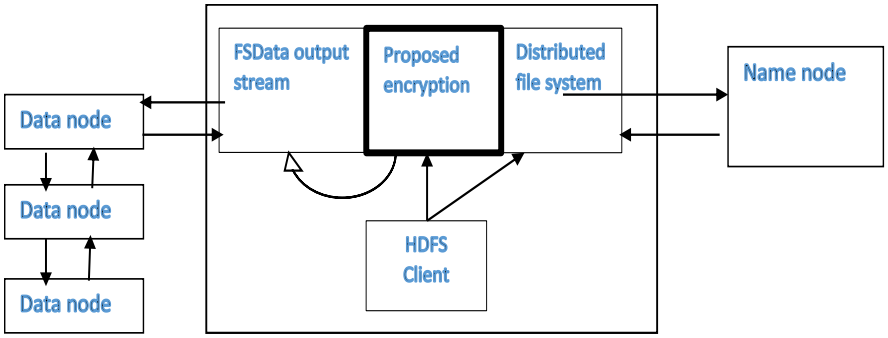

Fig. 2. Encryption procedure in HDFS

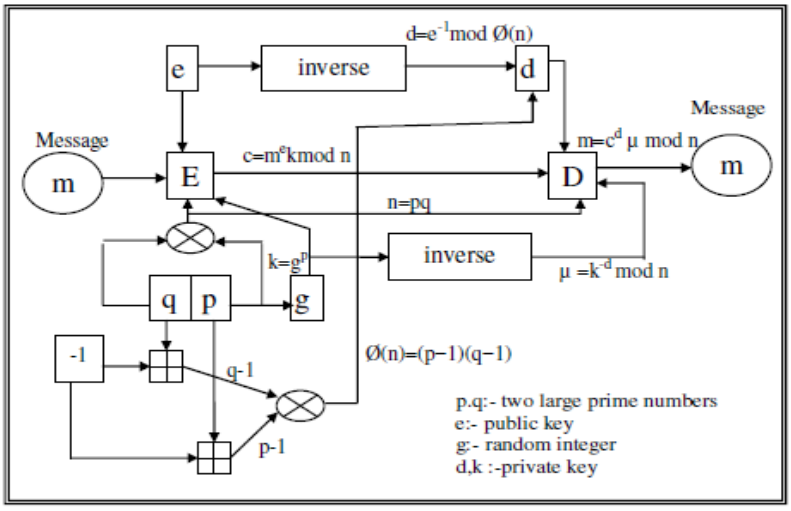

Fig. 3. Process of hybrid public key algorithm

HDFS consists of a Name Node that stores Metadata which manages the namespace the file system and monitor clients obtain for the files that encrypted. The files that encrypted is made up regarding one or higher blocks collected within a collection of data nodes. This proposed hybrid system described in Figure 3.

From Figure 3. the keys (public and private) generation procedure is based on the mechanism used by the RSA cryptosystem and its depicted by Algorithm1 below:

Algorithm 1*: – Key Generation of the proposed algorithm*
INPUT*: Choose big prime random numbers p and q*
OUTPUT: *A private key (p,q,d) and a public key, (n; e),*
User B received message from user A.

1. *Select two large random (and distinct) primes p and q, each roughly the same size.*
2. *Compute n = p*q and Ø(n) = (p − 1)(q − 1).*
3. *Select a random integer e, 1 < e < Ø , such that gcd(e; Ø) = 1.*
4. *Select a random integer g and compute* $k=g^p \bmod n$
5. *Use the extended Euclidean algorithm to compute the unique integer d, 1< d < Ø, such that ed ≡ 1 (mod Ø).*
6. *User's public key is (n; e); the user's private key is (d,k).*

The encryption process takes the scenario of Paillier algorithm to encrypt the message (m); thus, the cypher text computed as (C), with an augmented parameter $k = g^p$.

$$C = m^{e*k} \bmod n \tag{1}$$

Algorithm2 below shows in detail the encryption procedures:
Algorithm2*: – Encryption process of the proposed algorithm*
INPUT*: Plaintext to encrypt, and receiving the user's public key (n; e).*
OUTPUT*: Encrypted ciphertext.*
User A sends the message to user B.
To encrypt B should do the following:

(a) Obtain A's authentic public key (n; e).
(b) Represent the message as an integer m in the interval [0; n − 1].
(c) Compute $c = (m^e*k) \bmod n$.
(d) Send the ciphertext c to A.

This parameter transmitted along with the ciphertext, this parameter is used in decryption process to recover (m) to compute k then k^{-1}, While. This method explained in the following algorithms:

$$m = c^d * k^{-1} \bmod n \tag{2}$$

To recover the message, m four messages generated m. So the correct plain text is one of them. This procedure explained in (Algorithm3) below:
Algorithm3*:-Decryption process of the proposed algorithm*
INPUT*: Received encrypted ciphertext and the receiver's private key a.*
OUTPUT*: Original plaintext.*
To recover plaintext m from c, B should do the following:

(a) Compute $\mu = k^{-d} \bmod n$.
(b)Compute $m = c^d * \mu \bmod n$.

After applying the proposed encryption scheme, data stored in the cloud. Thus via HADOOP File System (HDFS), data will be stored in a cluster. Whenever the user requests data, the server will introduce the encrypted data to the decryption procedure. The user then uses the private key to retrieve the decrypted data using a hybrid system which is the proposal of this paper.

5. EXPERIMENTAL RESULTS AND ANALYSIS

HDFS and MapReduce functions have used for performance evaluation of encrypted HDFS. Each node has i3 core, four processors, 4 GB of memory, and 750 GB of the hard disk. Encryption Time: The time is taken by the RAS alone or the hybrid algorithm to encrypt the Hadoop divided dataset files into ciphertext using a key. It is calculated in milliseconds. Decryption Time: The time taken by the RAS alone or the hybrid system to decrypt the Hadoop split dataset files back into the plaintext using the private key. It calculated in milliseconds. Thus the Encryption Time is equivalent to system current time before encryption subtracted from it the system current time after encryption. Whereas the Decryption Time is equal to the system current time before decryption subtracted from it the system current time after decryption.

Figure 4 depicts the results of the comparison between encryption schemes, the RSA alone and the Hybrid system with different file sizes. It's clear that the proposed method showed efficient time consumption compared to the RSA for files size stars from 100 MB and ends with 1 GB with a step size of 100 MB.

And hence, the proposed method (Hybrid system) in the encryption stage is faster than the default RSA. Figure 5 shows the running time for RSA and the proposed method in the decryption stage. The encrypted files applied to this stage are of different sizes. By utilizing both of RSA and the hybrid system (the proposed method), it's evident that decryption time needed by the hybrid ciphered method is shorter than that required by RSA. Table I. Shows the computational complexity of the Hybrid cipher (proposed method) with, RSA and Paillier cryptosystems from which it's clear that the proposed method has doubled the computational complexity as compared to the individual systems (RSA or Paillier).

Tab. 1 Computational Complexity of the Proposed Method, RSA and Paillier

Method	Encryption	Decryption
RSA	$T(c) = O (\log n)^3$	$T(M) =O (\log n)^3$
Paillier	$T (c) = 2O (\log n)^2$	$T(M) =O (\log n)^3$
Hybrid system	$T(c) =2O (\log n)^3 +O (\log n)$	$T(M)= 2 O (\log n)^3 +3+O (\log n)$

Tab. 2. Time of encryption Process the file size in MB and time in second

File size in MB	RSA Encryption	Paillier Encryption	Hybrid method
100	220.5882	444.85287	224.26467
200	235.9926	550.6494	247.2303
300	265.7913	911.7259	324.5126
400	285.9564	1106.6512	304.7076
500	300.3254	1162.2592	320.0188
600	310.3456	1201.0374	330.6961
700	327.2813	1266.5786	348.7423
800	345.0357	1335.2881	367.6609
900	359.7902	1392.3880	383.3838
1000	368.5446	1426.2676	392.7114

Tab. 3. Time of encryption Process the file size in MB and time in second

File size in MB	RSA	Paillier	Hybrid Method
100	47.6470	78.5294	61.7646
200	133.6091	234.9738	181.2826
300	686.2018	788.6856	788.6856
400	955.9391	1493.3747	1031.6014
500	1391.3672	2186.2474	1526.0225

6. CONCLUSION

While Hadoop allows overcoming the difficulties confronted by big data in businesses and organisations, it has no security mechanism. An attacker or eavesdropper may compromise the data stored in Hadoop. The authenticity of data is always at stake, while Hadoop takes not implement any protection tool. Before storing it in HDFS, the proposed Hybrid asymmetric key algorithm encrypts the file content by obtaining that of the various network attacks. The file or data can therefore now collected under Hadoop without troubling on protection problems through utilizing the encryption methods to the records before it saved in Hadoop. The proposed Hybrid system supports most cloud computing system service models such as Service Software (SaaS), Service Infrastructure (IaaS), and Service Platform (PaaS). It also supports data management and security issues (Authentication, Integrity, Availability, and Confidentiality) in security and key management for data transfer.

The proposed method showed excellent time consumption with different file sizes in the encryption and decryption stages with higher complexity (double the computational complexity in decryption stages). The future work would be integrating both of ElGamal and RSA asymmetric key cryptosystem. The limitation of the proposed hybrid system is the time taken by the decryption procedure to discover the correct plaintext form the four alternatives messages resulted by Paillier method decryption.

REFERENCES

Bhardwaj, A., Singh, V. K., Vanraj, & Narayan, Y. (2015). Analyzing BigData with Hadoop Cluster in HDInsight Azure Cloud. *Annual IEEE India Conference (INDICON).* India: IEEE. doi:10.1109/INDICON.2015.7443472

Ahamad, D., Akhtar, M., & Hameed, S. A. (2019). A Review and Analysis of Big Data and MapReduce. *International Journal of Advanced Trends in Computer Science and Engineering, 8*(1), 1–3.

Amrulla, G., Mourya, M., Sanikommu, R. R., & Afroz, A. A. (2018). A Survey of : Securing Cloud Data under Key Exposure. *International Journal of Advanced Trends in Computer Science and Engineering, 7*(3), 30–33.

Bhandarkar, M. (2010). MapReduce programming with apache Hadoop. *International Symposium on Parallel & Distributed Processing (IPDPS)* (pp. 1-2). Atlanta: IEEE .

Chandra, S., Alam, S. S., Paira, S., & Sanyal, G. (2014). A comparative survey of symmetric and asymmetric key cryptography. *International Conference on Electronics, Communication and Computational Engineering (ICECCE)* (pp. 83–93). IEEE.

Chandra, S., Bhattacharyya, S., Paira, S., & Alam, S. S. (2014). A Study and Analysis on Symmetric Cryptography. *ICSEMR* (pp. 1–8). IEEE.

Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified Data Processing on Large Clusters. *6th Symposium on Operating Systems Design and Implementation* (pp. 107–113). ACM.

Dubey, A. K., Jain, V., & Mittal, A. P. (2015). Stock Market Prediction using Hadoop Map-Reduce Ecosystem. *2nd International Conference on Computing for Sustainable Global Development* (pp. 616–621). IEEE.

Hilbert, M. (2016). Big Data for Development: A Review of Promises and Challenges. *Development Policy Review*, *34*(1), 135–174.

Inukollu, V. N., Arsi, S., & Ravuri, S. R. (2014). Security Issues Associted With big Data In Cloud Computing. *International Journal of Network Security & Its Applications (IJNSA)*, *6*(3), 45–56.

Ismael, R. S., Youail, R. S., & Kareem, S. W. (2014). Image Encryption by Using RC4 Algorithm. *European Academic Research, II*(4), 5833–5839.

Jam, M. R., Khanli, L. M., Akbari, M. K., & Javan, M. S. (2014). A Survey on Security of Hadoop. *4th International Conference on Computer and Knowledge Engineering (ICCKE)* (pp. 716–721). IEEE.

Kareem, S. W. (2009). *Hybrid Public Key Encryption Algorithms For E-Commerce.* Erbil: University of Salahaddin–Hawler.

Kareem, S. W., & Hussein, Y. T. (2017). Survey and New Security methodology of Routing Protocol in AD-Hoc Network. *The 1st International Conference on Information Technology* (pp. 452–464). Erbil.

Li, B., Wang, M., Zhao, Y., Pu, G., Zhu, H., & Song, F. (2015). Modeling and Verifying Google File System Modeling and Verifying Google File System. *16th International Symposium on High Assurance Systems Engineering* (pp. 207–214). IEEE.

Mahmoud, H., Hegazy, A., & Khafagy, M. H. (2018). An approach for Big Data Security based on Hadoop Distributed File system. *International Conference on Innovative Trends in Computer Engineering (ITCE 2018).* Aswan: Aswan University.

Merla, P., & Liang, Y. (2017). Data analysis using hadoop MapReduce environment. *IEEE International Conference on Big Data (Big Data)* (pp. 4783–4785). Boston: IEEE.

Motoyama, M., McCoy, D., Levchenko, K., Savage, S., & Voelker, G. M. (2011). An analysis of underground forums. *ACM SIG- COMM Conference on Internet Measurement Conference IMC '11* (pp. 71–80). New York: ACM.

Park, S., & Lee, Y. (2013). Secure Hadoop with Encrypted HDFS. *International Conference on Grid and Pervasive Computing* (pp. 134–141). Springer.

Parmar, R. R., Roy, S., Bhattacharyya, D., Bandyopadhyay, S. K., & Kim, T.-H. (2017). Large-scale encryption in the Hadoop environment: Challenges and solutions. *IEEE Access*, *5*, 7156–7163.

Raghad, Z. Y., Kareem, S. W., & Hasan, A. O. (2016). *Design Security System Based on AES and MD5 for Smart Card*. Sulaimanyia: Charmo university.

Shetty, M. M., & Manjaiah, D. H. (2016). Data security in Hadoop distributed file system. *IEEE Int. Conf. Emerg. Technol. Trends Comput. Commun. Electr. Eng. ICETT 2016* (pp. 939–944). IEEE.

Usama, M., & Zakaria, N. (2017). *Chaos-Based Simultaneous Compression and Encryption for Hadoop*. PLoS One.

Yang, Ch., Lin, W., & Liu, M. (2013). A Novel Triple Encryption Scheme for Hadoop-based Cloud Data Security. *Fourth International Conference on Emerging Intelligent Data and Web Technologies* (pp. 437–442). IEEE.

Zhou, H., & Wen, Q. (2014). A new solution of data security accessing for Hadoop based on CP-ABE. *5th International Conference on Software Engineering and Service Science* (pp. 525–528). IEEE.