Security in Distributed System: A Review Perspective

Chukwuemeka Obasi[1], Ikharo A. B.[1], Victor Oisamoje[2]
[1]*Department of Computer Engineering, Edo State University Uzairue, Edo State, Nigeria*
[2]*Department of Electrical Electronic Engineering, Edo State University Uzairue, Edo State, Nigeria*

***Abstract*: Computer over the years has grown from single user to multi-user systems and then presently to distributed system, implying that computing resources can be distributed to different host and can be access by multiple users. This scenario is required for horizontally scaling of systems' requirement against vertical scaling now known as distributed systems, which cannot be improved beyond certain threshold. Security in such a heterogeneous system becomes a paramount issue of concern. This paper looks at distributed system, while highlighting the security challenges inherent in such system and reviewing the various progress in providing solution to the security challenges.**

***Index Terms*: Distributed System, Computing, Security, Attack, Database, Network**

I. INTRODUCTION

Distributed System (DS) is one of the advancements made in communication technology, in which access to information resources is independent of the host. In other words, it is a system of interoperable resources (hardware and software) among heterogeneous hosting devices or environments [2, 19]. This definition suggests that DS could be a set of configurable hardware, forming networked environment, a software system, or a combination of hardware and software system [10, 11]. One of the major challenges of DS is security [1, 2, 5, 7, 12] and is diverse in nature. Distributed system creates an illusion that a user has dedicated access to resources [9]. Many top companies have created complex distributed systems to handle billions of requests and upgrade without downtime. Data security and sharing have increased risks in distributed computer systems. Similarly, bugs are harder to detect in systems that are spread across multiple locations. The network has to be secured, and users must be able to safely access replicated data across multiple locations [37].

A comparative analysis conducted in [17] reveals that out of the four algorithms, the Optional Practical Training (OPT) yielded better results than the rest in the control of concurrency. Security issues in cluster computer was addressed in [12, 13 and 14], where cycle stealing, inter-node communication, snooping and cluster service disruption or DoS attacks were suggested as the types of attacks on cluster distributed system. To achieve security here, multilevel secure mechanism was used to assign security levels to each transaction and data item, which restrict database operation based on the access level [12, 15].

A security model for distributed system should be based on the framework definition of the distributed system. The argument leads to the development of security model, which considered the internal framework of the DS and mapping each functionality into the proposed security solution [16].

Distributed system does not only end with database systems and its allies, this concept is applicable in a number of operations, such as in power distribution [18], with possibilities of threat attacks. [36] presented the theory needed to design models of real-world large-scale systems and described the whole design process of such model. International Electrotechnical Commission's (IEC – 61850), which is an international standard defining communication protocols for Intelligent Electronics Devices (IED) shows that power network systems are susceptible to attack and would require middleware and or Net filter to defend the standard.

This paper looks at distributed system, while highlighting the security challenges inherent in such system and reviewing the various progress in providing solution to the security challenges.

II. REVIEW OF DISTRIBUTED SYSTEMS

A. Distributed System Scalling

Distributed system as a concept of horizontally scaling system deploys resources to improve the processing power and overall performances [6], for example bioinformatics system hosting both hardware and software systems in [4]. Figure 1 shows an illustration of typical distributed system.

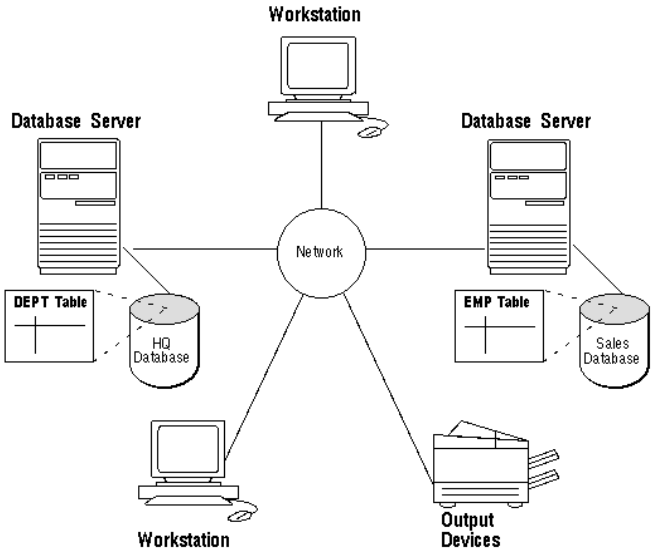

Figure 1. A Distributed Environment [25]

B. Cluster Computing

Cluster Computing, which is a set of computing systems in a network that are configured in a way that they appear to the users as a single computing node [12]. Such a system works in a manner that when a task is executed in a cluster will be accessible by all the nodes in the cluster [27]. This kind of configuration is necessary when we need to enhance computing speed and capacity. Figure 2 shows an illustration of cluster computing environment.

C. Grid Computing

Grid Computing is distributed system in which case, sets of independent computing nodes form a virtual super computer to enable them perform tasks that are ordinarily too large for a single computing node [12] as illustrated in Figure 3. Such an environment encourages hug resource sharing which are not centralised where each node can serve an entry or exit point to the grid computing system [29].

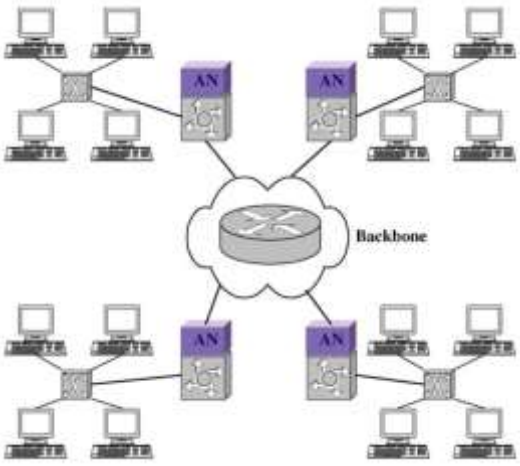

Figure 2. Cluster Computing Environment [28]

Figure 3. Grid Computing Environment [29]

D. Distributed Databaases

Distributed Database system which is a configuration of distributed system where a set of independent databases are running on multiple nodes simultaneously so that users can access stored data from any node irrespective of were the data was stored [12, 30, 32]. Distributed database system is illustrated in Figure 4.

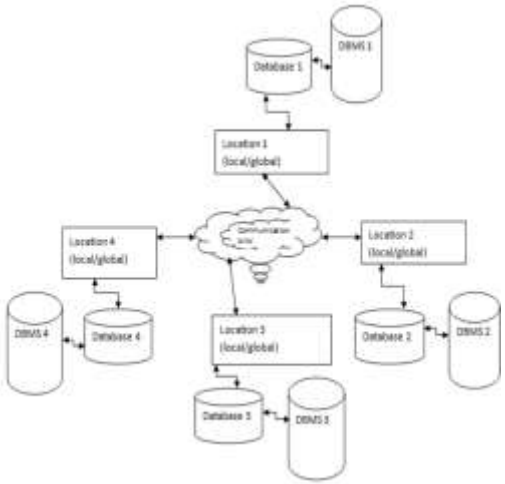

Figure 4. Distributed Database system [31]

E. Distributed Storage

Distributed Storage system represents a system with multiple storage locations across a network environment such that there is a shared access to the stored data as shown in Figure 5. The primary purpose for this being to prevent data loss in the event of system failure [22, 23]. Technologies used for the implementation of this include Redundant Array of Independent Disk (RAID), Centralized RAID, Network-Attached Storage (NAS) and Storage Area Network (SAN) [24].

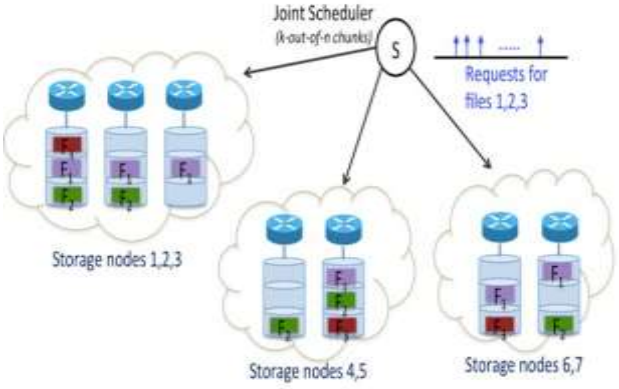

Figure 5. An Illustration of Distributed Storage System [26]

Typically, the nature of the distributed system is, interacting with a number of hosts in order to share resources. Four cardinal security issues have been identified in DS, to include: Concurrency, Fault Tolerance and Failover Recovery, and Naming [1]. Among these, the most common is concurrency and a number research-based solution have been contributed by several authors in an attempt to proffer solution in this regard.

III. SECURITY HOLES IN DISTRIBUTED SYSTEMS

A. Concurrency

This is a situation where by two or more users attempt to perform the same process simultaneously, and is often a problem of distributed database systems and distributed operating systems. The process in itself is not a threat, but can be compromised into being a threat. Concurrency can occur in a number of ways such as concurrent update, which is a fall through fore replay attack, deadlock, non-convergence and secure time.

B. Fault Tolerance and Failover Recovery

This security model tries to make a distributed system more resilient to failure or attack [1]. The approach to this has been sub-divided into Failure Model (Byantine Failure and Iteration with Fault Tolerance), Resilience, Redundancy and Service-Denial Attack.

C. Naming

Naming is an important aspect of distributed system, in which case all the devices and objects or entities in the system must be identified by a unique identity in the system. Name conflict causes some sort of security issue that calls for concern [1]. A name therefore is a set of bit strings or groups of character strings or both that is used to refer to an object, device or entity [25].

IV. REVIEWN ON DISTRIBUTED SYSTEM SECURITY

Fundamentally, security in distributed systems can be divided into two parts: the first part is concerned with the communication between users or processes, which dwell on distinct machines. The primary technique for making sure such communication is secure is using a secure channel. The second part concentrates on authorization that deals with making sure that a process attains access rights to those resources in a distributed system for which it is assigned to [38]. Security is paramount in a distributed system. The main idea of providing security is to ensure information and data protection, integrity, user access and authentication to information system are not compromised [3]. This section presents the reviewed work on distributed system security issues. In taking a more detailed view of the two parts [19], namely:

1. User level security, which ensure that users are authenticated before gaining access to the resources,
2. Resources level security, which ensures that each user only gain access to the resources assign to him.

Denial of service (DoS) attack is one of the most common type of security challenge in a distributed system as identified in [1]. DoS was identified by [7] as a problem of distributed urban traffic control system and hence suggested a solution by developing a security system and isolation model that is built from service-oriented access control methodologies. Development of a model for large scale distributed application system in an attempt to provide high level system perform and application security was in [8], in which database security was the focus. Concurrency issue was addressed using the mechanism of serialization of tasks, while suggesting the following algorithms for addressing concurrency issues:

1. Basic Timestamp Ordering Algorithm: this is an algorithm that is used to preserve consistency in DS [33]. While timestamp uniquely identifies transactions, the algorithm serialises the transactions [8] to ensure order in execution, hence ensuring that conflicts among concurrent tasks do not occur since each task will only be allowed to occur in its own timestamp. The rule for this operation is presented here:

Transaction T issue R (A) operation.

If WTS1>TS (T)

Then rollback T

Else

Perform R (A) successfully

set RTS (A) = max {RTS (A), TS (T)}.

Transaction T issues W (A) operation.

If RTS1 (A) > TS (T)

Then rollback T

elseIf WTS (A) > TS (T)

Then rollback T

Else

Perform W (A) successfully

set WTS (A) =TS (T)

end

2. Distributed Two-Phase Locking (2PL): this protocol allows transactions to acquire a lock and unlock permission in two phases as shown in Figure 6, which include;
    1. *Growing Phase*, in which a lock may be acquired but none can be released.
    2. *Shrinking Phase*, in which previously acquired lock may be releases but new ones not acquired.

In any case, all locks are held until transactions are aborted or executed successfully.

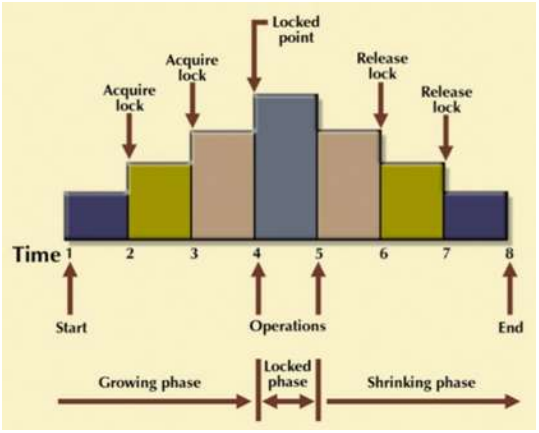

Figure 6 Two Phase Locking (2PL)

3. Distributed Optimistic Protocol (DPT): this algorithm certifies a read request if the version that was read is still the current version of the item or no write with a newer timestamp has already been locally certified, and certifies a write request if no later reads have been certified and subsequently committed or no later reads have been locally certified already

4. Wait-Die and Wound-Wait Algorithms: Deadlock often leads systems to halt, resulting from indefinite waiting of two concurrent transactions. Two algorithms can be used to prevent or avoid the occurrence of deadlock. These are known as Wait-Die and Wound-Wait algorithms.

A. Die-Wait Algorithm

Here a transaction with an older timestamp is made to wait until a requested resource is available. In a case of two transactions, T1 and T2, and timestamp of any transaction be TS (T). If a transaction has locked T2 with a resource and T1 is requesting the same resource, the following actions will be performed:

if TS (T1) < TS (T2) –

if T1 is older than T2 and T2 is wit resource

then T1 should wait until resource is available for execution

This implies that if a younger transaction has locked some resource and older transaction is waiting for it, then older transaction is allowed wait for it till it is available.

elseIf T1 is older than T2 and T2 is with resource

if T2 is waiting for resource then

Kill

Wait for random time

Restart process

B. Wound-Wait Scheme

Here, a younger transaction is made to wait if it has requested a resource that is currently being held by an older transaction but is reverse is the case, the older transaction forces the younger to kill the transaction and restart after some interval of delay.

V. PROPOSED SOLUTIONS

A. Hadoop Distributed File System

Distributed system in our modern world of computing has open the door for massive data generation [3], which requires special attention. In so doing, Hadoop provided the specialised file system known as Hadoop Distributed File system (HDFS). In such environment, [3] used real-time encryption techniques. The objective was to control access to data using authentication algorithm. In addition, a secure algorithm known as blockchain algorithm was suggested in [20]. It is a shared, distributed fault resilient database that every participant in the network can share, but no entity can control it. The distributed database in blockchain contains continuously growing list of records that cannot be altered by any user as illustrated in Figure 7.

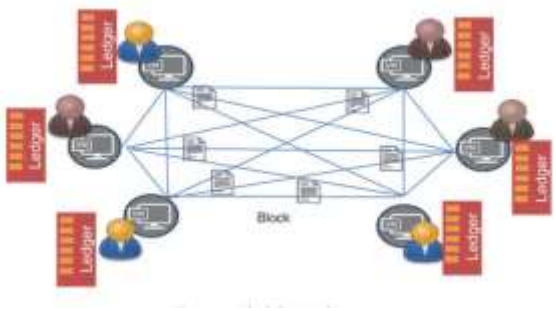

Figure 7. Blockchain Architecture

The key features of Blockchain include: Decentralised Network, Distributed Consensus, and Cryptographically secure algorithm.

B. Defence Against Attach

Although multiple defences may be necessary to withstand an attack, these defences should be based on five fundamental security principles: layering, limiting, diversity, obscurity, and simplicity. These principles provide a foundation for building a secure system.

C. Layering Approach

Layered security refers to network security systems that use multiple components to protect operations on multiple levels or layers in a distributed network [33]. If one layer is penetrated several more layers must still be breached, and each layer is often more difficult or complicated than the previous [34]. A layered approach has the advantage of creating a barrier of multiple defences that can be coordinated to thwart a variety of attacks. Layered security provides the

most comprehensive protection. [33] suggested some essential security layers: Firewall - acts as a barrier between a trusted network and an untrusted network, only allowing into your network traffic that has been defined in the security policy. Patch Management - refers to the process of distributing and applying updates to software and firmware. They address functionality errors or bugs, boost performance, and close the security gaps that would otherwise leave your systems, software, and applications vulnerable to cyberattacks. Multi-Factor Authentication - requires multiple forms of verification to access an application, account, or corporate network and prevent hackers from exploiting weak or compromised end-user credentials to access your network. Endpoint Protection- Every device connected to your network is a potential entry point for hackers. All of these entry points, known as "endpoints," need to be included in your organization's cybersecurity plan. Web Content Filtering - blocks users' access to websites and online content deemed inappropriate or dangerous. Sophisticated Password Policy - prevent password re-use, prohibit weak passwords, and improve your network security. Phishing Simulations - test users on their vigilance in recognizing suspicious emails, further strengthening your defences. Email Filtering- Filtering emails at the gateway reduces this risk and helps to protect your users and your business from email-borne cyber threats such as phishing attacks, ransomware, viruses and malware, and business email compromise. Dark Web Monitoring - Dark web monitoring tools scan the dark web for email addresses and passwords associated with your company's domain so you can identify and address these vulnerabilities before they can be exploited by a hacker. Physical Security - restrict access to and protect your on-premises infrastructure and spaces in which data is stored. Examples of such measures include access control systems, key cards and door locks, security cameras and surveillance, and security personnel.

D. Limiting Approach

Limiting access to information decreases the threat against it which implies that only those people assigned must only gain access to it [35]. In addition, access type should be limited to what those people need in order to perform their specific jobs. Some ways to limit access are technology-based (such as assigning file/data permissions so that a user can only read but not modify a file), while others are procedural (prohibiting an employee from removing a sensitive document from the premises). Rate limiting is used to control the amount of incoming and outgoing traffic to or from a network reason being to allow for a better flow of data and to increase security by mitigating attacks such as Distributed Denial of Service (DDoS) [35]. The key is that access must be restricted to the barest minimum.

According to [35], two implementation approaches were highlighted in integrate rate limiting either via Nginx or Apache. This does not allows not more than 2 request per second at an average, with bursts not exceeding 5 requests.

E. Nginx Approach

Using Nginx as your web server, taking advantage of the *ngx_http_limit_req_module* and implementing directly within the Nginx configuration file, the Nginx rate limits based on the user's IP address will be thus:

http {*limit_req_zone $binary_remote_addr zone=one:10m rate=2r/s; ... server* {*... location /promotion*/ {*limit_req zone=one burst=5;*}}

Apache method

Using Apache to implement rate limiting within the Apache configuration file, the module mod_ratelimit must be used in order to limit client bandwidth. Throttling is applied to each HTTP response instead of being aggregated at the IP/client level.

<Location "/promotion">

SetOutputFilter RATE_LIMIT

SetEnv rate-limit 400

SetEnv rate-initial-burst 512

</Location>

The values in the snippet above are defined in KiB/s. Therefore, the rate-limit environment variable, used to specify the connection speed to be simulated is 400 KiB/s while the initial amount of burst data is 512 KiB/s. [35] also noted that rate limiting can be a great method to help fight against infrastructure attacks as well as block other types of suspicious activity.

F. Diversity Approach

Diversity is another aspect of layering. As it is important to protect data with layers of security, the layers also must be different (diverse) in security architecture. This implies that breaking or penetrate one layer does not mean using the same techniques to break through all other layers. Using diverse layers of defence means that breaching one security layer does not compromise the whole network system [34].

G. Obscurity Approach

Security is sometimes compromised during a shift change of the security pickets. This approach tries to enforce secrecy and confidentiality of the system's internal design architecture. Security through obscurity aims to secure a system by deliberately hiding or concealing its security flaws [39]. This technique is sometimes called security by obscurity, that is obscuring to the outside world what is on the inside there by making attacks much more difficult. An example of obscurity in network security would be not to reveal the type of computer, version of operating system, or brand of software that is used [34]. An attacker who knows this information could use it to determine the vulnerabilities of the system.

Obscuring distributed network can be an important means of protection [34].

H. Simplicity Approach

Simplicity approach becomes necessary because attacks can come from a variety of sources and in many ways. Distributed network security is by its very complex in nature. Yet the more complex it becomes, the more difficult it is to understand. Complex security systems can be difficult to understand, troubleshoot, and be sure of safety [34]. Its simplicity should be simple for those within to understand and use. Complex security schemes are sometimes found to be compromised to make them easier for dependable users to work with and yet easier for the attackers. Major benefits are achieved if the distributed system is made simple from the inside but complex on the outside.

VI. CONCLUSION

Security of distributed system comes with its goals depending on its architecture outlook and design algorithm. Security of distributed system is paramount to data and users' protection. The study presents the basic introduction to distributed system with brief review of the security application of distributed system. State of the art solutions to the security challenges in distributed system and as key element in its applications and usage were discussed. Multiple defence systems were similarly proffered as mitigating models for the numerous attacks in DS networks that are necessary antidotes for seamless operations and optimal performances. This work would help to lay foundations to others who would like to carry out some research on Distributed Networks.

REFERENCES

[1] R. J. Anderson, "Security Engineering: A Guide to Building Dependable Distributed Systems", Second Edition, Wiley Publishing, Inc. 10475 Crosspoint Boulevard Indianapolis, IN 46256, 2008.

[2] M. A. Bauer, N. Coburn, D. L. Erickson, P. J. Finnigan, J. W. Hong, P. A. Larson, J. Pachl, J. Slonim, D. J. Taylor, T. J. Teorey, "A Distributed System Architecture for a Distributed Application Environment", IBM System Journals, Vol. 33, No. 3, 1994.

[3] S. Y. Inamdar, A. H. Jadhav, R. B. Desai, P. S. Shinde, I. M. Ghadage, A. A. Gaikwad, "Data Security in Hadoop Distributed File System", International Research Journal of Engineering & Technology Vol. 3 No. 4, 2016.

[4] S. Krishnaswamy, & T. M. Mohan, "The largest distributed network of bioinformatics centres in the world: Biotechnology Information System Network (DBT-BTISNET)", Current Science, Vol. 110, No. 4, 25 February 2016 (available at doi: 10.18520/cs/v110/i4/556-561).

[5] S. G. Wasnik, J. Pimple, "Distributed Cloud based Business Management System", International Journal for Innovative Research in Science & Technology, Vol 2, No 11, 2016.

[6] Song Jiajia, "Computer Network Performance Optimization Approaches based on Distributed System with the Cloud Computing Environment", International Journal of Science and Research, Vol 5, No 2, 2016.

[7] A. M. N. Mocofan, R. Ghită, V. R. Tomás López3, F. C. Nemţanu, "Quality of Services Solution for Efficient Communication within a Distributed Urban Traffic Control System", U.P.B. Sci. Bull., Series C, Vol. 78, Iss. 1, 2016.

[8] N. McDonald & W. J. Dally, "Sikker: A High-Performance Distributed System Architecture for Secure Service-Oriented Computing", available at https://pdfs.semanticscholar.org/989e/6e77b2e9c60e408 e4e4e72e37cd6a933fd07.pdf on November 26 2019.

[9] A. M. Gupta & Y. R. Gore, "Concurrency Control and Security Issue in Distributed Database System", International Journal of Engineering Development and Research, Vol. 4 No. 2, 2016.

[10] F. Ali & R. Z. Khan, "Distributed Computing: An Overview", International Journal of Advanced Networking and Applications, Vol. 7, No. 1, pp. 2630 – 2635, 2015.

[11] M. Rathi & M. Lohia, "Research Paper on Distributed Operating Systems", International Journal of Innovative Research in Technology, vol. 1No. 5, pp. 128 – 132, 2014.

[12] M. Firdhous, "Implementation of Security in Distributed Systems – A Comparative Study", International Journal of Computer Information Systems, Vol. 2, No. 2, 2011.

[13] T. Xie & X. Qin, "Security-Aware Resource Allocation for Real-Time Parallel Jobs on Homogeneous and Heterogeneous Clusters", IEEE Transactions on Parallel and Distributed Systems, Vol. 19, No. 5, 2008.

[14] J. Zhongqiu, Y. Shu & W. Liangmin, "Survivability Evaluation of Cluster-Based Wireless Sensor Network under DoS Attacks", 2009 5th International Conference on Wireless Communications, Networking and Mobile Computing, 2009.

[15] N. Kaur, R. Singh, A. K. Sarje, & M. Misra, "Performance evaluation of secure concurrency control algorithm for multilevel secure distributed database system," in International Conference on Information Technology: Coding and Computing (ITCC 2005), Las Vegas, NV, USA, 2005, pp. 249-254.

[16] V. Krasnoproshin & T. Galibus "Conceptual Distributed System Models and Organization of Security Mechanisms", 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2015.

[17] M. Nasseri & S. M. Jameii, "Concurrency Control Methods in Distributed Database: A Review and Comparison", 2017 International Conference on Computer, Communications and Electronics (Comptelix), 2017.

[18] Y. Wang, Y. Zhou, X. Wang, J. Ruan & C. Yang, "Vulnerability Analysis of IEC61850 Standards in Distributed Energy System", 2018 International Conference on Power System Technology, 2018.

[19] R. N. Shahabi, "Security Techniques in Distributed Systems", Computer Science and Information Technology Vol. 3, No. 2, pp 49-53, 2015. Available at DOI: 10.13189/csit.2015.030203.

[20] S. S. Shetty, C. A. Kamhoua & L. L. Njilla, "Blockchain for Distributed System Security", IEEE Computer Society, 2019.

[21] Marchese F. T., http://csis.pace.edu/~marchese/CS865/Lectures/Chap5/Chapter5.htm

[22] Jepsen T. C., "Distributed Storage Networks: Architecture, Protocols and Management", John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, ISBN: 9780470850206,0470850205, 2003.

[23] Yuchong H., Yinlong X., Xiaozhao W., Cheng Z., & Pei L., "Cooperative Recovery of Distributed Storage Systems from Multiple Losses with Network Coding," IEEE Journal on Selected Areas in Communications, vol. 28, no. 2, pp. 268- 276, February 2010.

[24] Xiao G. Y. & Wei X. L., "A new network storage architecture based on NAS and SAN," in 10th International Conference on Control, Automation, Robotics and Vision (ICARCV 2008), Hanoi, Vietnam, 2008, pp. 2224 - 2227.

[25] Oracle7 Server Distributed Systems Manual, Vol. 1, https://docs.oracle.com/cd/A57673_01/DOC/server/doc/SD173/ch1.htm

[26] Aggarwal V., Al-Abbasi A. O., & Fan J., "Taming Tail Latency for Erasure-coded, Distributed Storage Systems", IEEE

INFOCOM 2017 - IEEE Conference on Computer Communications, 2017. Available at 10.1109/INFOCOM.2017.8056997

[27] Baker M. & Buyya R., "Cluster Computing at a Glance," in High Performance Cluster Computing: Architectures and Systems - Volume 1, Rajkumar Buyya, Ed. Upper Saddle River, NJ, USA: Prentice Hall, 1999, Ch. 1, pp. 3-47.

[28] Bouhafsa F., Gelasa J. P., Lefevre L., Maimoura M., Phama C., Vicat-Blanc P. P., Tourancheau B., "Designing and evaluating an active grid architecture", Future Generation Computer Systems Vol. 21, pp. 315–330, 2005.

[29] Ku-Mahamud K. R. & Nasir H. J. A., "Ant Colony Algorithm for Job Scheduling in Grid Computing", IEEE Computer Society 2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation, pp. 40 – 45, 2010.

[30] Mamaghani A. S., Mahi M., Meybodi M. R., & Moghaddam M. H., "A Novel Evolutionary Algorithm for Solving Static Data Allocation Problem in Distributed Database Systems", Second International Conference on Network Applications, Protocols and Services (NETAPPS), Alor Setar, Kedah, 2010, pp. 14-19.

[31] Kumar N, Kumar A. & Alam I. "Enhanced " One Phase Commit Protocol " In Transaction Management", International Journal of Soft Computing and Engineering, vol. 3, No. 4, pp. 221 – 225, 2013.

[32] Subir Verma, "Performance Evaluation of the Timestamp Ordering Algorithm in a Distributed Database", IEEE Transactions On Parallel And Distributed Systems, Vol. 4, No. 6, 993.

[33] https://blog.totalprosource.com/what-is-layered-security-how-does-it-defend-your-network#:~:text=Layered security is a network security approach that, environment where a breach or cyberattack could occur.

[34] Ciampa, M. "CompTIA Security+ Guide to Network Security Fundamentals" 5th edition, Cengage Learning, Boston, USA. 2014.

[35] https://www.keycdn.com/support/rate-limiting. What Is Rate Limiting?

[36] Stefan, M., Arben, C. and Zdenek, B. "Distributed Systems – A brief review of theory and practice". IFAC-Papers-On-Line, 49-25, pp. 318–323, 2016.

[37] Fawcett, A. "What are distributed systems?" A quick introduction. Access from https://www.educative.io/blog/distributed-systems-considerations-tradeoffs, 2022.

[38] Priyadarshini, S. B. B., Bagjadab, A. B. and Mishra, B. K. "Security in Distributed Operating System: A Comprehensive Study." In Cyber Security in Parallel and Distributed Computing (Ed) pp.221-230, 2019.

[39] Technopedia. Security Through Obscurity. Accessed from https://www.techopedia.com/ definition/21985/security-through-obscurity-sto, 2013.