International Research Journal of Natural and Applied Sciences
ISSN: (2349-4077)
Impact Factor 5.46 Volume 6, Issue 5, May 2019
Website- www.aarf.asia, Email : editor@aarf.asia , editoraarf@gmail.com

SECURITY FRAMEWORK FOR FILE TRANSFER IN A DISTRIBUTED ENVIRONMENT

[1*] Edward N. Udo, [1]Ini J. Umoeka and [1]Ndifreke A. Johnson

[1]Department of Computer Science, University of Uyo, Uyo, Akwa Ibom State, Nigeria

* Corresponding Author: Email- edwadudo@uniuyo.edu.ng. Phone: +234 8023339501

ABSTRACT

In today's world, computers in corporate organizations, institutions and establishments are working collaboratively with other computers for the purpose of data transfer, processing, and storage. As these collaborations are expanding daily, the infrastructures within a distributed environment are exposed to multiple attacks such as leakage, tampering, vandalism and spoofing. Security therefore continues to remain a major challenge in a distributed environment, even when several available security models are already developed and deployed. This work therefore develops and implements a security framework, at the middleware layer of the distributed system layered structure, which creates a database for storing users login details, authenticate and authorize the system users at every login phase. The user login keys are encrypted using Advanced Encryption Standard (AES) that adopts 192 bits key size for encryption and are given to the users at every login. The encrypted keys can last for only 20 minutes before expiration. After expiration, another encrypted key has to be generated for that user. The encrypted keys are matched, with the already stored plaintext, by a key matcher embedded within the file server. This approach shields the distributed system from spoofing attack and makes the distributed system more secured.

Key Words: Advanced Encryption Standard (AES), Distributed System, Distributed Environment, File Transfer and Security

1. INTRODUCTION

The era of static communication in a strictly closed network is now a thing of the past. The adoption of the internet and other communication media has brought about a more dynamic

communication approaches that make the computer network open. These approaches have been widely accepted as it has grown rapidly in every aspect of human life.

In today's world, computers in corporate organizations, institutions and establishments are working collaboratively with other computers for the purpose of data transfer, processing, storage etc. Systems working in collaboration and are scattered over a geographical space are called distributed systems (Firdhous, 2011). There is various definition of distributed system in literature, but the earliest definition by Tanenbaum and Steen (2007) will be adopted in this work. Distributed system is "a collection of independent computers that appears to its users as a single coherent system". This definition highlights two important aspects; first is that distributed systems, including computers are autonomous and second aspect is that users (humans or programs), think they are dealing with a system exclusively. The web is a clear example of a distributed system because there are various components that help browsers display web contents but the user feels the contents of the web pages are accessible through the browser.

Distributed computing systems allow homogeneous and heterogeneous computers to act as a computing environment (Lunstovskyy and Spillner, 2017). Users in this environment can equally access resources (both locally and remote) without the knowledge of the computers which their processes are running on.

The client/server computing model is therefore a major framework for a distributed environment as users are allowed to request for services, through the client computer, from other computers that provide the service (servers). Figure 1 shows the architecture of a distributed system based on a workstation-server model.

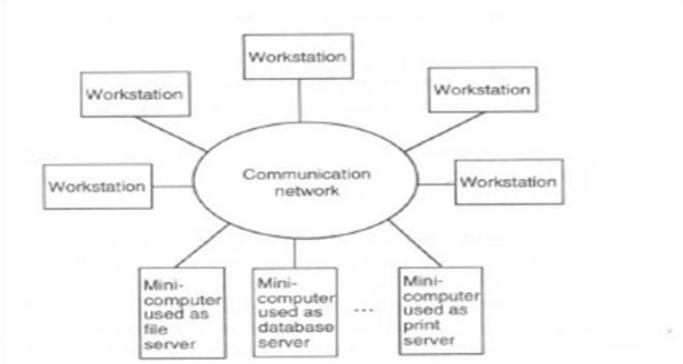


Figure 1 – Distributed System based on a Workstation-Server Model

Information sharing (file transfer) is the main essence of a distributed system. This phenomenon poses complicated security challenges even when the ultimate goal of distributed computing is to enhance performance by connecting users and resources in a cost-effective, transparent and reliable way. Other advantages of distributed computing include fault tolerance, transparency, openness, scalability and resource accessibility in an event of a failed node (Coulouris et al., 2012).

As distributed system is increasing, file transfer processes is also increasing thereby making distributed systems more susceptible to security threats like denial of service, information leakage and unauthorized access. Data services, such as file transfer services, must be preserved against security threats such as interception, interruption, modification and fabrication (Shahabi, 2015). Securing a distributed system is an important principle and is also the most difficult, because security must be maintained throughout the system. Security should therefore be given adequate attention because it is one of the fundamental issues in distributed systems (Shen and Wu, 2010)

Figure 2 shows the layered organization of a distributed system. In this layered organization, security mechanism is usually placed in the middleware, which is the distributed system layer. Middleware is considered as the bridge used to connect distributed applications across different physical locations, with different hardware platforms, network technologies, operating systems, as well as different programming languages.

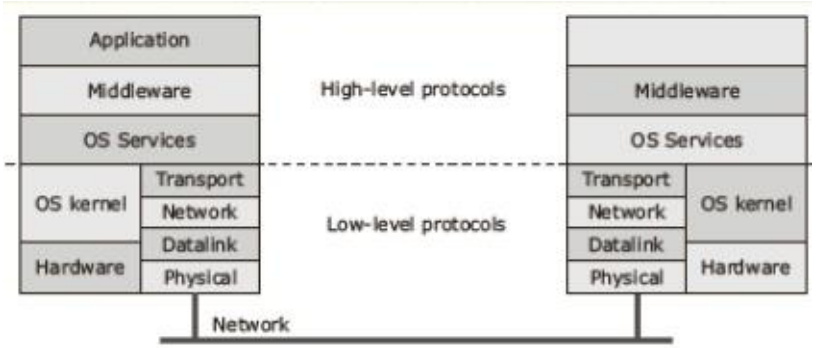


Figure 2 – Layered Organization of a Distributed System (Source: Tanenbaum and Steen, 2007)

There are many distributed systems in operation, especially as the age is characterized by cloud services. Some of such distributed systems are cluster, grid computing etc (Firdhous, 2011).

Cluster is a set of computers that are grouped together in such a manner that they form a single resource pool. Any task that has been assigned to the cluster would run on all the computers in the cluster in a parallel fashion by breaking the whole task into smaller self contained tasks. Then, the result of the smaller tasks would be combined in order to form the final result (Kaur and Rai, 2014).

Grid computing is defined as a parallel and distributed system that is capable of selecting, sharing, and aggregating geographically distributed resources dynamically at runtime based on their availability, capability, performance, and cost meeting the users' Quality of Service (QoS) requirements (Buyya and Venugopal, 2010; Dabas and Arya, 2013).

Security framework and approaches are put in place to secure these existing distributed systems. Researchers have developed approaches to secure cluster computing using authentication, integrity check and confidentiality framework. Li and Vanghu (2006) studied security vulnerabilities of computing clusters and modeled security mechanism for confidentiality, integrity and availability (CIA). Xi and Qin (2008) developed two schemes to ensure parallel applications executed on computing clusters meet the security requirements of CIA as well as the deadline for execution.

Approaches, in literature, for securing grid computing, cover authentication (using public key infrastructures), delegation, authorization and certificates. Xing and Xue (2010) created an authentication module to prevent external users from accessing internal grid and protect the grid from unauthorized access.

Liu et al.,(2008) used certificate authority to solve the problem of identity authentication in a distributed networks by adopting a binary tree code algorithm and presented a distributed authentication model based on public keys.

Shahabi (2015) reviewed the security techniques used in distributed systems and these include authentication based on shared keys authorship, authentication using a key distribution centre, authentication using public key encryption, the use of session keys and message encryption to ensure confidentiality and message integrity.

Despite the fact that researchers have come up with a lot of techniques to ensure secure file transfer in a distributed environment, security threats still remain and attackers continue to attack. This development therefore poses this big question "how secure is the system?"

In an attempt to answer this big question, this work designs and implements a security framework for file transfer in a distributed environment by creating a database for users login

details, authenticate as well as authorize the users for login using encrypted keys generated based on Advanced Encryption Standard (AES). The encrypted keys are matched, with the already stored plaintext, by a key matcher embedded within the file server. This approach shields the distributed system from spoofing attack; an attack resulting from a disguise communication from an unknown source that acts like a known and trusted source. This impersonator (user or device) launches attacks against network hosts, steal data, spread malware or bypass access controls to gain illegitimate access to the distributed system.

2. SYSTEM ARCHITECTURE

The architecture of the security framework for file transfer in a distributed system is depicted in Figure 3.

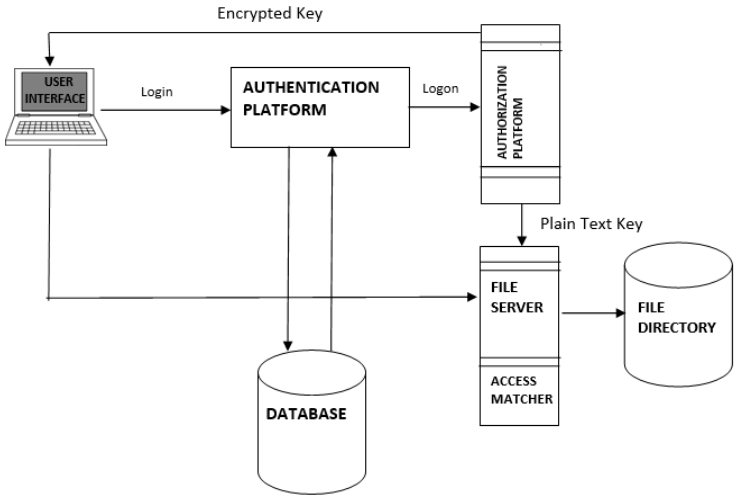


Figure 3 - System Architecture for a Secured File Transfer

The components of the architecture are:

1. **User Interface**: This is the link that enables the user to communicate with the system.
2. **Authentication Platform**: When the user attempts to login into the system, this platform confirms his login details in the user database before granting the user access.
3. **Authorization Platform**: The access granted the user after proper login, takes the user to the authorization platform where a randomly generated key is created and the encrypted equivalent of the key is sent back to the user while the plain key (unencrypted) is sent to the file server. The user now logins with the encrypted key. The encryption of the key is done with AES.
4. **Access Matcher**: This is a component of the file server, which matches the encrypted key used by the user to login with the equivalent plaintext. That is the encrypted key

is decrypted and compared with the plain key that was kept in the file server. When there is a match, the users are granted access to the file server for upload and download services accordingly.

5. **Database**: The database stores the login details for each user which was captured by the administrator during user's registration phase.

There are five major classes that make up the system (distributed environment) and the interactions between these classes are depicted in Figure 4

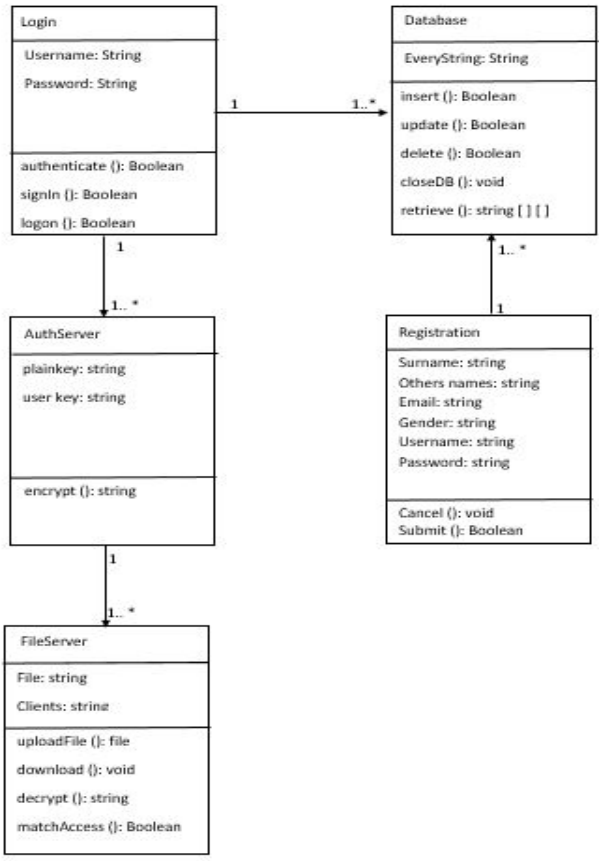


Figure 4 – UML Class Diagram

3. **ADVANCED ENCRYPTION STANDARD (AES)**

This encryption technique is adopted because it is a widely accepted and considered the global standard of encryption. This acceptability is due to its fast and secure form of encryption which keeps prying eyes away from data.

AES has a fixed block size of 128 bits and key sizes of 128, 192 and 256 bits. The encryption processes of AES are in three stages:

i. Initial Round

ii. Main Round

iii. Final Round

All these phases follow the same basic operations which are combined differently. The operations are illustrated in Figure 5.

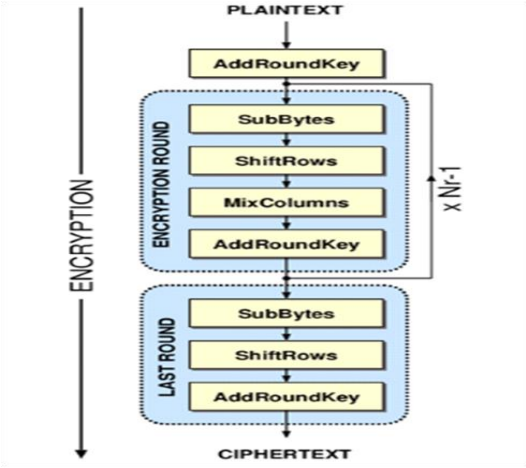


Figure 5 – Basic Structure of AES

The encryption rounds (N_r) are repeated depending on the key size used: 128 key size (9 times), 192 key size (11 times) and 256 key size (13 times).

In this work, 192 bits key size is used, giving a total of 12 rounds (including the last round). The sub-operations of AES are:

i. **SubBytes** – This means byte substitution which involves splitting the input into bytes and passing each through a substitution Box or S-Box that implements inverse multiplication in Galois Field 2^8.

ii. **ShiftRows** - The rows in this stage refer to the standard representation of the internal state in AES, which is a 4x4 matrix where each cell contains a byte. In this operation, each of these rows is shifted to the left by a set amount: their row number starting with zero. The top row is not shifted at all, the next row is shifted by one and so on.

iii. **MixColumns** - This phase provides diffusion by mixing the input around. Unlike ShiftRows, MixColumns performs operations splitting the matrix by columns instead of rows. The result is another matrix consisting of 16 bytes.

iv. **AddRoundKey** – This operation is the only phase of AES encryption that directly operates on the AES round key. In this operation, the input to the round is XORed with the round key. The last round gives the ciphertext.

4. IMPLEMENTATION

The security framework was implemented in Netbeans 8.0 Integrated Development Environment (IDE) using Java Programming Language (SE binary version 8.0.50.13) for the coding.

The system administrator logins into the system with a given admin code to create account for all the users of the distributed system by clicking the account button in the home screen. This is shown in Figure 6

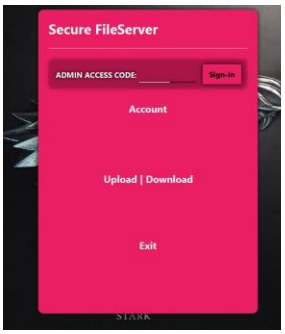
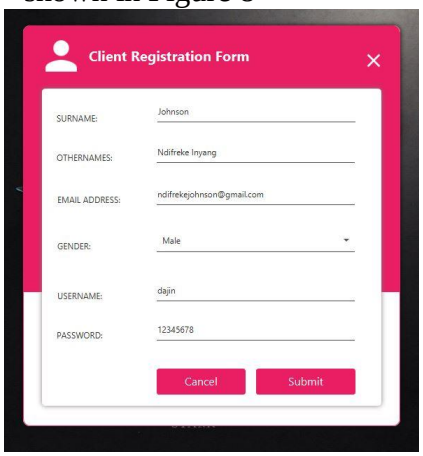
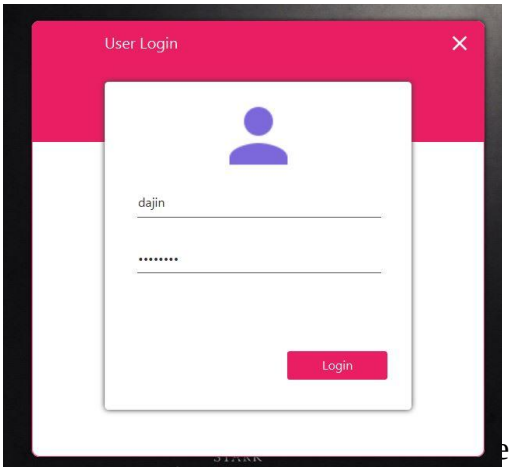


Figure 6 - Login Home Screen

After successfully logging-in, the administrator goes to the registration page, which he alone can access to register users. The registration form is depicted in Figure 7. A user that is successfully registered can now login into the system through the user login interface as shown in Figure 8





When a user logins, an encrypted key is send to the user from the authorization platform. This key changes anytime the user wants to login. This encrypted key last for only 20 seconds, else the user has to login again and another encrypted key will be sent. The plain text of the key is sent to the file server by authorization module. The key will be decrypted by the matcher module in the file server for comparison. Figure 9(a) and (b) shows the encrypted key and the encrypted key with time notice respectively.

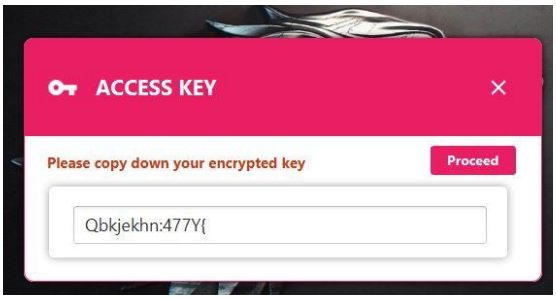
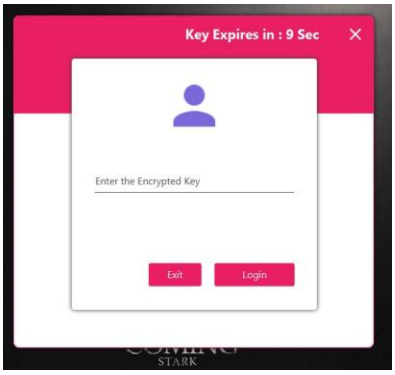


Figure 9 (a) – Encrypted Key



(b) Encrypted key with Notice

After successful authentication and authorization, the user will now gain access to the file directory in the file server for upload and download accordingly. Figure 10 shows the file upload and download page. Once a file is uploaded into the file system, it cannot be deleted.

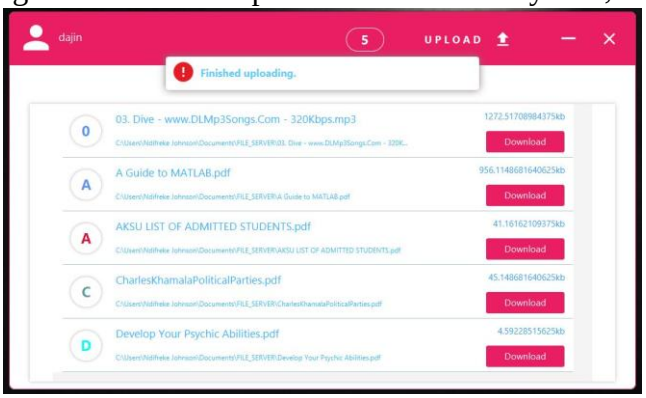


Figure 10 – File Directory in the File Server

5. CONCLUSION

The application of authentication, authorization and AES with 192 bits key size has actually beefed-up the security framework within a distributed system. At implementation, several attempts were made to hack a distributed system with a spoofing strategy, but those attempts proved abortive. It is on this premise we conclude that a distributed environment will be more secured with the deployment of many security approaches within a given distributed environment.

REFERENCES

1. Buyya, R. and Venugopal, S. (2010). Market Oriented Computing and Global Grids: An Introduction. In *Market Oriented Grid and Utility Computing*, Rajkumar Buyya and Kris Bubendorfer, Eds. Hoboken, NJ, USA: John Wiley & Sons, 3-27.
2. Coulouris, G., Dollimore, J., and Kindberg, T. (2012). Distributed systems – Concepts and design (5th ed.). Addison – Wesley, London

3. Dabas, P., and Arya, A. (2013). Grid computing: An introduction. International Journal of Advanced Research in Computer Science and Software Engineering, 3(3), 467–470.
4. Firdhous, M,. (2011): Implementation of Security in Distributed Systems – A Comparative Study. International Journal of Computer Information Systems, 2(2), 1 – 6.
5. Kaur, K., and Rai, A. (2014). A comparative analysis: Grid, cluster and cloud computing. International Journal of Advanced Research in Computer and Communication Engineering, 3(3), 5730–5734.
6. Li, W. and Vaughn, R. (2006). Cluster Security Research Involving the Modeling of Network Exploitations Using Exploitation Graphs. In *Sixth IEEE International Symposium on Cluster Computing and the Grid Workshops (CCGRIDW'06)*, Singapore, 26-36.
7. Luntovskyy, A. and Spillner, J. (2017). Security in Distributed Systems. In Architectural Transformations in Network Services and Distributed Systems. Springer Vieweg, Wiesbaden, 247 – 308.
8. Shahabi, R. N. (2015): Security Techniques in Distributed Systems. Computer Science and Information Technology, 3(2): 49 - 53
9. Shen, Z., and Wu, X. (2010). The Protection for Private Keys in Distributed Computing System Enabled by Trusted Computing Platform. In *International Conference on Computer Design and Appliations (ICCDA 2010)*, Qinhuangdao, Hebei, China, 576-580
10. Tanenbaum, A., and Steen, M. (2007). *Distributed systems: Principles and Paradigms* (2nd ed.). Pearson Higher Education Inc. Press, Upper Saddle River, NJ, USA:
11. Xi, T. and Qin, T. (2008). Security-Aware Resource Allocation for Real-Time Parallel Jobs on Homogeneous and Heterogeneous Clusters. *IEEE Transactions on parallel and distributed Systems*, 19(5), 682-697.
12. Xing, Q., Xue, S. and Liu, F. (2010). Research of Grid Security Authentication Model, in *International Conference on Computer Application and System Modeling (ICCASM)*, Taiyuan, Shanxi, China, 78-80.