

**DATA SECURITY ENHANCEMENT IN CLOUD
COMPUTING BY PROPOSING A DKE
ENCRYPTION PROTOCOL**

**A THESIS SUBMITTED TO THE INSTITUTE OF
GRADUATE STUDIES
OF
NEAR EAST UNIVERSITY**

**By
HALAMT AYUB ABDULMAJED**

**In Partial Fulfilment of the Requirements for
the Degree of Master of Science
in
Computer Information Systems**

NICOSIA, 2021

HALMAT AYUB

ABDULMAJED

COMPUTING BY PROPOSING A DKE ENCRYPTION PROTOCOL

DATA SECURITY ENHANCEMENT IN CLOUD

NEU

2021

**DATA SECURITY ENHANCEMENT IN CLOUD
COMPUTING BY PROPOSING A DKE
ENCRYPTION PROTOCOL**

**A THESIS SUBMITTED TO THE INSTITUTE OF
GRADUATE STUDIES
OF
NEAR EAST UNIVERSITY**

**By
HALMAT AYUB ABDULMAJED**

**In Partial Fulfilment of the Requirements for
the Degree of Master of Science
in
Computer Information Systems**

NICOSIA, 2021

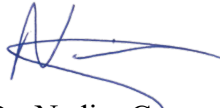
**HALMAT AYUB ABDULMAJED: DATA SECURITY ENHANCEMENT IN
CLOUD COMPUTING BY PROPOSING A DKE ENCRYPTION PROTOCOL**

Approval of Director of Institute of Graduate Studies

PROF. DR. KEMAL HÜSNÜ CAN

**We certify that this thesis is satisfactory for the award of the degree of Master
of Science in Computer Information Systems**

Examining Committee in Charge:



Prof. Dr. Nadire Cavus

Committee Chairperson, Department of
Computer Information Systems, NEU



Assoc. Prof. Dr. Yöney Kırsal Ever

Committee, Department of Software
Engineering, NEU




Assist. Prof. Dr. Sahar Ebadinezhad

Supervisor, Department of Computer
Information Systems, NEU

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Halmat Ayub Abdulmajed

Signature: 

Date: 18/02/2021

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor Assist. Prof. Dr. Sahar Ebadinezhad and head of Department of Computer Information Systems, Prof. Dr. Nadire Cavus, for their continued support, patience, encouraging comments and guidance that improved my writing skills opening my thoughts. Thank you for advising me to realize and view the issues from several aspects and appreciating their guidance to utilize various research materials. I would like to offer my thanks to the faculty of the Computer Information Systems department at the Near East University for their managerial works and their worthy support.

Finally, I would like to thank my family members, friends and well-wishers for their support and prayers towards me.

ABSTRACT

From the last few decades, Information Technology has seen tremendous development in terms of new technologies, solutions for real world problems, mobility support, services deployed over the internet etc. One among them is cloud computing which is evolved based on the techniques called web services and virtualization. The main properties of cloud computing are elasticity, scalability and pay as you go. As all the types of services are provided through the cloud by multiple service providers, many issues have been arising in cloud computing. The main issue to deal with is security of the data. For securing data, cryptography provides a strong solution. Hence, in this research, it is decided to develop a new cryptographic technique for the protection data which is to be stored in the cloud. This research, proposed a technique called Dual Key Encryption (DKE) technique which uses two keys for encryption and decryption purposes. Before uploading the data in the cloud, DKE uses public key for data encryption in the first round and in the second round of encryption, data encrypted by the private key that only is known by user not to the cloud service provider. The reverse process is followed for decryption of the data at the user side. This proposed technique is simulated and tested using CloudAnalyst simulator with multiple files of different sizes. The proposed technique is compared with Triple DES (3DES) cryptographic technique for performance evaluation. This mechanism is taking very less time when compared with 3DES and proved to be efficient.

Keywords: Cloud Services, Data Security, Dual Key Encryption, Cryptography, Data Integrity.

ÖZ

Son birkaç on yıldan bu yana, Bilgi Teknolojisi yeni teknolojiler, gerçek dünyadaki sorunlara çözümler, mobilite desteği, internet üzerinden dağıtılan hizmetler vb. Bunlardan biri, web hizmetleri ve sanallaştırma adı verilen tekniklere göre gelişen bulut bilişimdir. Bulut bilişimin temel özellikleri esneklik, ölçeklenebilirlik ve kullandıkça öde'dir. Her türlü hizmet birden fazla hizmet sağlayıcısı tarafından bulut üzerinden sağlandığı için, bulut bilişimde birçok sorun yaşanmıştır. Ele alınması gereken ana sorun, verilerin güvenliğidir. Şifreleme, güvenli veriler için güçlü bir çözüm sağlar. Bu nedenle, bu araştırmada, koruma verilerinin bulutta depolanarak saklanması için yeni bir kriptografik teknik geliştirilmesine karar verildi. Bu araştırma, şifreleme ve şifre çözme amacıyla iki anahtar kullanan Çift Anahtar Şifreleme (DKE) tekniği adı verilen bir teknik önerdi. DKE, buluta veri yüklemeye önce, şifrelemenin ilk ve ikinci turunda veri şifrelemesi için ortak anahtarı kullanır, yalnızca kullanıcı tarafından bulut hizmeti sağlayıcısı tarafından bilinmeyen özel anahtar tarafından şifrelenen veriler. Kullanıcı tarafı verilerinin şifresini çözmek için ters işleme izlenir. Bu önerilen teknik, farklı boyutlarda birden fazla dosyaya sahip CloudAnalyst simülatörü kullanılarak simüle edilir ve test edilir. Önerilen teknik, performans değerlendirmesi için Üçlü DES (3DES) şifreleme tekniği ile karşılaştırılır. Bu mekanizma çok daha az zaman alıyor ve 3DES'e kıyasla verimli olduğu kanıtlanmıştır.

Anahtar Kelimeler: Bulut Hizmetleri, Veri güvenliği, Çift Anahtarlı Şifreleme, Kriptografi, Kriptografik, Veri bütünlüğü.

TABLE OF CONTENT

ACKNOWLEDGEMENT	i
ABSTRACT	ii
TABLE OF CONTENT	iv
LIST OF TABLES.....	vii
LIST OF FIGURES	viii

CHAPTER 1: INTRODUCTION

1.1. Cloud Computing Models	3
1.1.1 Deployment Model.....	3
1.1.2 Service Models	6
1.2 Cloud computing architecture	9
1.3. Characteristics of Cloud Computing	10
1.4 Cloud Computing Risks and Security Issues.....	12
1.4.1 Security Issues	12
1.4.2 Cloud Computing Risks	13
1.5 Motivation	13
1.6 Problem Statement.....	14
1.7 Research Contributions	14
1.8 Organization of thesis.....	15

CHAPTER 2: LITRATURE REVIEW

2.1. Fog Technology	16
2.2. An Overview of Fog	17
2.2.1. Data Intensive Internet of Things Applications.....	19
2.2.3. Fog Enabled Internet of Things Security Requirements	22
2.3. Overview of Cloud Computing	23
2.3.1. Major Security Section in Cloud Computing	24
2.4. Cloud Computing	26
2.5. Difference between cloud and fog computers:	30
2.6. Fog characteristics:	32
2.7. Security Issue in Cloud and Fog Technologies	33
2.7.1. Authentication and Authorization	34
2.7.2. Access Control.....	34
2.7.3. Location Privacy.....	35
2.7.4. Trust.....	35
2.7.5. Data Integrity	36
2.7.6. Privacy issues	36
2.7.7. Account hijacking.....	37

2.7.8. Denial of Service	37
2.7.9. Data encryption	38
2.8. Problem Definition	38
2.8.1. Existing Security system in fog	38
2.8.2. Data Encryption.....	39
2.8.3. RSA Encryption System.....	39
2.8.4. DES Encryption Method	40
2.7.5. 3DES Algorithm.....	42
2.8.6. AES Encryption Algorithm	42
2.9. Why was AES developed?	43
2.10. Existing fog computing gaps	43
2.10.1. Threats in Cloud	43
2.10.2. Challenges ahead.....	44
 CHAPTER 3: PROPOSED DUAL KEY ENCRYPTION SYSTEM	
3.1 Overview	46
3.2 Operational Procedure of DKE System.....	46
3.3 Leveraging DEI Mode.....	48
3.5 What can DKE be used to Protect?	53
3.6 Effect of Cryptographic Operation Placement	54
3.6.1 Key Authority and Service are not linked	54
3.6.2 Key Authority and Service are linked	54
3.7 Fulfilling the remit of the Key Authority	55
 CHAPTER 4: IMPLEMENTATION AND EVALUATION	
4.1 Simulation Environment.....	57
4.1.1 JAVA.....	57
4.1.2 CloudSim:.....	58
4.1.3 CloudSim Architecture	59
4.1.3 CloudAnalyst Simulator	60
4.2 Results	61
4.3 Comparison of Execution Time.....	69
4.4 Security Comparison between 3DES and DKE	69
 CHAPTER 5: CONCLUSION AND FUTURE WORK	
5.1 Conclusion.....	70
5.2 Future Work.....	71
REFERENCES	72
APPENDICES.....	88
APPENDIX 1: SIMILARITY REPORTS.....	89

APPENDIX 2: ETHICAL APPROVAL DOCUMENT	90
APPENDIX 3: SOURCE CODE.....	90

LIST OF TABLES

Table 2.1: Server Request..... 33

LIST OF FIGURES

Figure 1.1: Cloud Deployment Models	6
Figure 1.2: Cloud Service Models.....	7
Figure 1.3: Cloud Architecture.....	9
Figure 1.4: Cloud Characteristics	11
Figure 2.1: Fog between cloud and end user platform	18
Figure 2.2: IoT Applications	20
Figure 2.3: The relationship between Cloud and Fog	27
Figure 2.4: Different connection among Datacentre technologies	29
Figure 2.5: Fog Node structure.....	31
Figure 2.6: Encryption and Decryption Process.....	39
Figure 2.7: DES Encryption Process	40
Figure 2.8: Initial and Final Permutation	41
Figure 2.9: Round Function Process	41
Figure 3.1: Flow Chart I	46
Figure 3.2: Phase II of Flowchart.....	47
Figure 3.3: Phase III of Flowchart.....	47
Figure 4.1: CloudSim Architecture	60
Figure 4.2: Main Simulation Parameters.....	62
Figure 4.3: Datacenter Locations under Simulator	62
Figure 4.4: Datacenter Configuration.....	63
Figure 4.5: Internet Characteristics	63
Figure 4.6: Sample File given by the user.....	64
Figure 4.7: Response Time Summary	64
Figure 4.8: Average Response Time	65
Figure 4.9: Datacenter Request Servicing Time	65
Figure 4.10: Datacenter Loading Information.....	66
Figure 4.11: Cost for the Job Execution.....	66
Figure 4.12: Sample of Encrypted and Decrypted Files	67
Figure 4.13: Time taken for encryption and decryption.....	67
Figure 4.14: General Output.....	68
Figure 4.15: Encryption and Decryption Time of Triple DES	68
Figure 4.16: Comparison of Execution Time	69

LIST OF OBSERVATIONS AND SYMBOLS

DKE:	Dual Key Encryption
DEI:	Dual Encryption Infrastructure
VM:	Virtual Machine
CSP:	Cloud Service Provider
IoT:	Internet of Things
FN:	Fog Node
SaaS:	Software as a Service
PaaS:	Platform as a Service
IaaS:	Infrastructure as a Service
DdoS:	Distributed Denial of Service
RSA:	Rivest Shamir Adleman
DES:	Data Encryption Standard

CHAPTER 1

INTRODUCTION

Trend setting innovations offer new occasions to the business. Cloud computing is one of these innovations. The innovative services computing technique called as web services which was started as a small thing by Amazon in the name AWS (Amazon Web Services) along with the hard computing technique called as grid computing which offers computing facilities as wait and get model leads to the paradigm called as Cloud Computing. As indicated by NIST (Mell, 2011) Cloud is a model for advantageous, on-request network consent to a shared pool of configurable selecting assets (e.g., software's, networks, workers, gathering, and associations) that can be promptly provisioned and passed on with unimportant association exertion or master affiliation. During the last two decades, most of the information technology giants like Google, Amazon, Microsoft etc (Andrzej, 2019). have their hardware resources at ideal state most of the time. That means, the hardware resources are underutilized by their services offered to third parties. In this conjuncture, the already evolved technology called as virtualization came in aid for them (Chakraborty, 2019). The IT giants think of using those web services along with virtualization technology, so that they could make use of their hardware resources at their fullest and make revenue from them. The companies were now able to use those technologies and started providing their hardware resources for utilization to third parties as pay as you go model. Thus, the new service is now called cloud computing, because the third parties who are using the resources are not knowing about the exact hardware they are using. All the resources are connected with the third parties through internet only and for successful usage of cloud resources, all the parties are in need of high-end internet connectivity. The cloud computing service provider gives figuring force and capacity assets in dispersed environment, while abstracting hidden foundation, contingent upon a given help (Puthal, 2019). A few undertaking level associations send exceptionally versatile cloud computing answers for an inward information sharing and cooperation. Also, a few organizations offer their cloud administrations for business use and go about as a Cloud Service Provider (CSP) on the lookout. CSPs guarantee to give better security, dependability, supportability, cost viability and backing than IT frameworks of individual associations. These highlights make it conceivable to move business from

singular frameworks to the cloud and make it open over the Internet. Dynamic nature, high adaptability and broad registering assets make a cloud environment ideal for community-oriented Research and information sharing. All previously mentioned advantages of cloud computing bring about a higher thoughtfulness regarding this innovation, and features a critical significance of the cloud, concerning how data shared and prepared in a cutting-edge society. In this way, accept that utilization and significance of cloud-based arrangements will increment altogether in the opportunity approaching. Nonetheless, sometimes collective exploration and information sharing includes preparation and capacity of delicate information, which is considered as private data and ought not to be shared without consent (for model patient assent in medical services). While cloud computing is arising quickly and used by the expanding number of associations around the world, information privacy and uprightness issues are not adequately tended to right now. Putting away touchy information in the cloud, without information where information is truly dwelled and making this information open over the Internet expands the danger of information bargains (Chatterjee, 2017). The danger of private information spillage and protection infringement in the cloud altogether obstructs a wide reception of this innovation (Zhou, 2010). CSPs offer different measures to secure the information put away in the cloud. The greater part of the CSPs offer encryption abilities to the clients, so all information is moved and put away in an encoded structure in a distributed storage framework. Nonetheless, key administration difficulties and insider dangers still should be tended to. Confirmation of CPSs may give some degree of affirmation to the clients. Be that as it may, there are still no assurances of full poise over the information dwelled in the cloud, for the clients. Furthermore, the absence of normalization, as to get to cloud administrations, diminishes interoperability and adaptability of exchanging among CSPs. Henceforth, associations may encounter merchant lock-in, except if they are not ready to invest a huge energy to oblige their current answers for the new CSP. Along these lines, cautious framework configuration, survey of existing principles, hazard the board and necessities investigation is required prior to conveying administrations in the cloud. Access control frameworks should be inspected and custom-made for the reasonableness with the dynamic and appropriated nature of the cloud environment. Wide assortment of the utilization cases and flaws of as yet creating cloud advances and principles makes the previously mentioned errands significantly more intricate and difficult to acknowledge in a brought together manner.

1.1. Cloud Computing Models

The cloud computing models (Mell, 2011) are of two types:

1. Deployment Model.
2. Service Model.

1.1.1 Deployment Model

In view of the arrangement the cloud can be of following sorts of deployment models as appeared in Figure 1.1.

1. Public cloud: The public cloud services are services which are run by bigger companies like Google, Microsoft, amazon etc., which allow all kinds of users – from single user to bigger corporates could enroll for the cloud services and use it. The public cloud services describe their services very clearly and not leave any detail. This will make all the users who are willing to use the public clouds can analyze different public service providers and pick one among them. The main problem in using the public services is security and integrity of the data. As multiple, sometimes millions of people are using public services, it is hard for the service provider to ensure security and integrity of the data of third parties. There are many breaches reported so far in case of public cloud and user details are disclosed in dark web. If those data are leaked, it could lead to a disaster so ensuring security and integrity of the data in the public cloud is always an open issue (Albugmi, 2016).
2. Private cloud: The private cloud isn't split between the affiliations. The resources of the private cloud are placed under one organization control and managed by the employees of the organization. These cloud resources are permitted to access for the employees of the organization or who got access rights from the organization. The private cloud can be provided by the third-party cloud service providers and can be used solely by a single organization (Singh, 2014). This is why because, the using organization can reduce its infrastructure cost and can increase or decrease the usage of the resource as per the need. As cloud supports pay as you go model, the using party needs only to pay for what they are using and can save millions of dollars in billing. The main issue in this type of cloud is also security where the security of the data is relying on the using party as the cloud service provider could only guarantee for the integrity of the data based on hardware failures whereas the security breaches should be handled only by the using party (Garrett, 2011).
3. Hybrid Cloud: This cloud structure is a mix of two above said clouds which is a combination of in any occasion two interesting cloud foundations that are confined

themselves. It means the cloud user is having their cloud in their own premises or in the third-party cloud service provider. Their cloud is accessible from any part of the world by their employees. It is possible that, often the cloud limit in terms of storage, processing power, number of users at a time could exceed their limit. During those times, the private cloud is clubbed with public cloud for managing the requests. That means that the extra requirements like storage, processing power, accessibility features etc. are derived from public cloud for time being and are released from the private cloud once their need is fulfilled. In this cloud, the security of the data is in danger as the public cloud resources are underutilized for time being. That is the data which is stored in a private cloud could possibly be transferred to public cloud for completion of an operation. That time, it is possible for a hacker to vandalize the data on transmission or under process (Alahmadi, 2019). Hence it is the duty of the private cloud manager to secure the data during transmission or the integrity of the data after the processing at public cloud. Across framework, similarity can demonstrate itself to be a significant issue when fabricating a hybrid cloud. Since a hybrid cloud requires combination across private and public areas, control turns into a basic issue. Consider how to control clients' very own data and foundation to limit security hazards, just as how to oversee operational cycles to enhance the utilization of in-house assets and clients' encounters. Make certain to anticipate a reinforcement and catastrophe recuperation methodology that will work across your surroundings. It is essential to be readied and to have a cycle set up that is clear, known, and easy to follow. In the grievous case of a calamity, it is vastly improved to be arranged so you can get going as fast as conceivable instead of living with the expectation that nothing actually occurs. There is a solid need to set up mindfulness that hybrid conditions are an unavoidable truth, which likewise underscores the need to set up an outline for turning it out. Each association should attempt to assemble this plan, guaranteeing that both IT and lines of business have a voice and the two arrangements of necessities are met from the earliest starting point – it's far simpler to set up this quickly, to abstain from having to agonizingly manage circumstances including impromptu incorporation and local arrangements. Another critical component in building up a shrewd private and hybrid cloud system is picking the correct foundation equipment. Which choice is best for a specific organization relies upon its current foundation? Most associations have been chipping away at virtualizing their server farms for a long time and have just spent a great deal of cash on inheritance equipment. What's more, various conditions

aren't generally viable. It is typical that the applications are interlinked with different applications and frameworks. Thus, ensure you observe the final product of cloud movement prior to proceeding to do as such. The cloud specialist co-op ordinarily gives a cloud the board support. The association would have sent a foundation to screen and oversee Internal IT. While both may offer asset provisioning and asset checking abilities, they would not offer execution of the board capacities that permit programmed scaling of assets according to utilization. "The two administration stages should be coordinated to give a solitary perspective on the hybrid cloud. In any case, neither gives the usefulness to do as such and consequently would require additional items or module parts that give the functionalities to deal with the two frameworks overall (Antonio, 2014, p. 1). "A ton of outsider merchants' offer segments that help make a layer of these functionalities on the current foundation of the board stages. These functionalities permit dealing with the hybrid cloud independently by giving the robotized checking and provisioning functionalities (Ammar, 2018).

4. Community Cloud: This is a cloud framework that is just used by clients of an association who has necessary permissions. The people group engaged with these undertakings, for example, tenders, business associations, and think-tanks, center around comparable issues in their cloud connections. Their shared advantages may incorporate ideas and approaches identified with security and consistent contemplations, and the objectives of the venture too. Community Cloud processing encourages its clients to recognize and break down their business requests better. Community Cloud might be facilitated in a server farm, possessed by one of the inhabitants, or by an outsider cloud administrations supplier and can be either on location or off-site. Cloud suppliers have created Community Cloud contributions, and a few associations are as of now seeing the advantages. The accompanying rundown shows a portion of the principal situations of the Community Cloud model that is valuable to the partaking associations. Different administrative divisions that perform exchanges with each other can have their preparing frameworks on shared foundation. This arrangement makes it savvy to the occupants, and can likewise diminish their information traffic. Government offices in the United States. Government substances in the U.S. that share comparable necessities identified with security levels, review, and protection can utilize Community Cloud. As it is community-based, clients are sufficiently sure to put resources into the stage for their tasks. Numerous organizations may require a specific framework or application facilitated on cloud administrations.

The cloud supplier can permit different clients to associate with a similar environment and fragment their meetings coherently. Such an arrangement eliminates the need to have separate workers for every customer who has similar goals. Organizations can utilize this model to test applications with top-of-the-line security needs instead of utilizing a Public Cloud. Given the administrative measures related with Community Clouds, this could be a chance to test highlights of a Public Cloud offering (Butun, 2018).

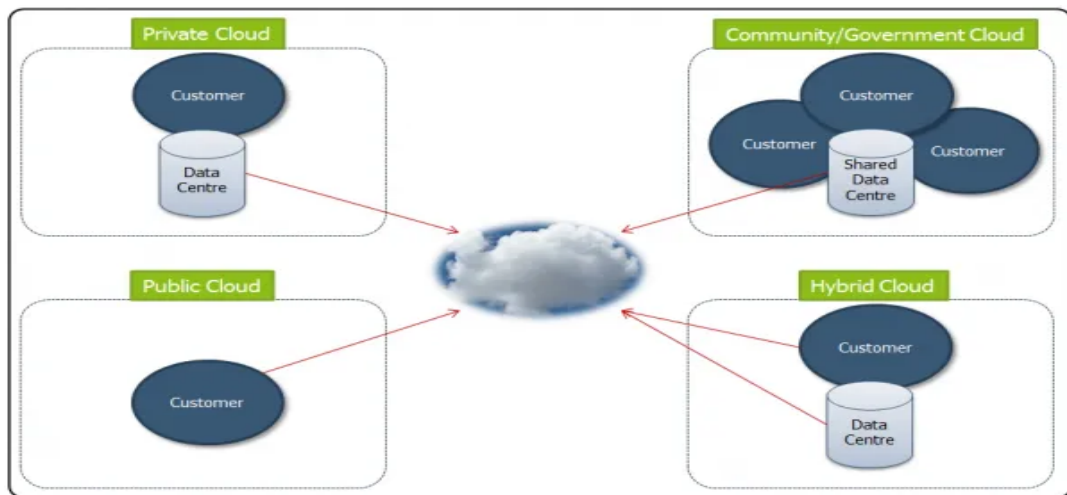


Figure 1.1: Cloud Deployment Models (Kaur, 2017)

1.1.2 Service Models

Diverse utility conveyance models in Cloud Computing are appeared in Figure 1.2.

1. **Software as a Service:** In this model purchaser has the proficiencies to use businesses of provider that are operating on a cloud foundation. A software GUI can be utilized to get to the programs through different customer gadgets. Working constructions, putting away, workers, affiliation or other covered cloud foundation are not directed by the customer. This permits different clients to share reports and even to chip away at them simultaneously. For instance, in the Google Sheets spreadsheet various clients can chip away at various cells all the while. The cells various clients are dealing with are secured off and featured various tones. A constant talk window can likewise be opened up close by the spreadsheet to further improve coordinated effort. For more data on community oriented working utilizing Google's distributed computing applications, you can watch the now exemplary video Google Docs in Plain English. Taking joint effort even further, the yields of some SaaS applications can be

installed in other site pages as web administration devices. For instance, a Google Sheets or Zoho Sheet diagram can be crushed into another site. There it will consequently refresh when the information in the online spreadsheet that is creating it is changed. SaaS applications are additionally continually refreshed, which can liberate clients from the "overhaul hellfire" of a significant conventional programming bundle revision (Luiz, 2018).

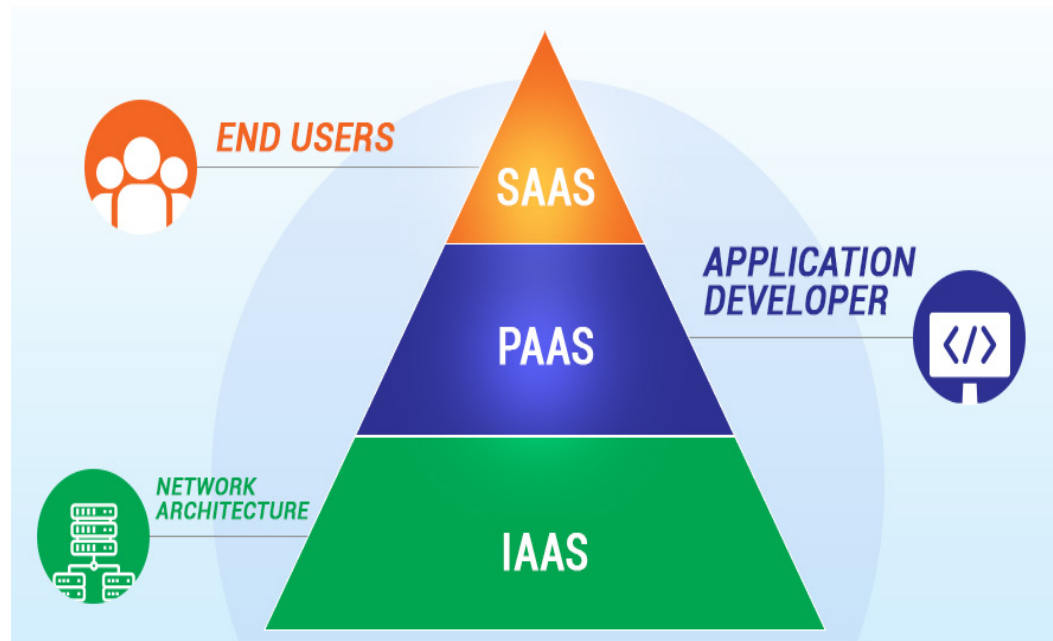


Figure 1.2: Cloud Service Models (Riaz, 2020)

2. Platform as a Service (PaaS): Client is empowered to send onto the cloud framework made by the customer or software gained that is made utilizing associations, libraries, developing dialects and instruments kept up by the supplier. The fundamental cloud framework including limit, working constructions, workers, network constrained by the customer. A platform is a product environment used to create and run applications. For instance, Microsoft Word is an application that suddenly spikes in demand for the Microsoft Windows platform. At the point when individuals decide to use a cloud Figure utilizing a platform as an assistance or 'PaaS', they acquire admittance to an online platform given by a cloud registering seller. They would then be able to utilize this platform to create and convey their own on the web (SaaS) applications. Applications created utilizing PaaS might be utilized secretly by only one or a couple of clients inside a specific organization. In any case, they can likewise be offered free or for-a-charge to anyone on the web. This implies that in the event that you have a good thought for another online application, at that point you

can utilize PaaS to transform it into a reality. A few cloud providers currently offer PaaS devices. These outstandingly incorporate Google App Engine, Microsoft Azure, and the Salesforce Platform. All such contributions adequately furnish their clients with a crate of cloud registering Lego. New applications are then developed from the plastic blocks on offer. A few applications can even be assembled utilizing a basic intuitive interface. Generally non-specialized individuals can in this manner make new online applications rapidly (Rahman, 2018). Undoubtedly, Salesforce have guaranteed that their "streamlined programming model and cloud-based environment mean [customers] can fabricate and run applications multiple times quicker, at about a large portion of the expense of conventional programming platforms". While PaaS is incredible much of the time, its clients should be aware of the elaborate adaptability sections power compromise. This means while PaaS makes it moderately simple to make new online applications, clients are obliged by the specific programming dialects and apparatuses given by their PaaS provider. As such, PaaS sellers have all out power over which Lego blocks they permit their clients to work with. While this guarantees that applications constructed utilizing the instruments on offer will consistently work accurately, it is by the by prohibitive. It is thus that numerous organizations and a few people decide to cloud the process at the foundation level .

3. Infrastructure as a Service (IaaS): Consumer can design enrolling assets like affiliations, putting away, arranging, and so forth where the buyer can send and run self-insistent programming, which can solidify applications and working frameworks. The key cloud base isn't coordinated or obliged by the client at any rate the customer has requested over passed on applications, aggregating and working constructions. Infrastructure as a service or "IaaS" is the place where committed actual workers and virtual worker cases can perform the very same capacities. However, there are additionally some significant contrasts between them. For a beginning, virtual worker cases are less expensive to supply as each doesn't need its own bit of actual equipment in a cloud server farm. Then again, virtual worker cases are at times seen as less secure by the individuals who would prefer not to impart worker equipment to different clients. Therefore, four classifications of IaaS are accessible. These are most usually known as "private clouds", "devoted facilitating", "crossover facilitating" and "cloud facilitating".

1.2 Cloud computing architecture

The architecture contains various estimated coupled cloud parts. Cloud can be extensively orchestrated into two areas: back end and the front end (Strickland, 2020). The distributed computing design is depicted in Figure 1.3. Client some segment of the distributed computing structure is insinuated as a front end which contains the applications and graphic user interfaces that are needed for having the chance to distributed computing stages, e.g., Browser. Back end recommends cloud itself and contains all of resources that are obliged to give distributed computing organizations. It contains VMs, massive data storing, virtual machines, security instruments, organizations, laborers, association models, and so forth. These completions are regularly related through an association, ordinarily related by techniques for the Internet. Back end is careful to give shows, traffic lights and fundamental security parts. The specialist uses certain shows, renowned as middleware that assist the devices that are related with pass on and contrast and each other (Mirković, 2019).

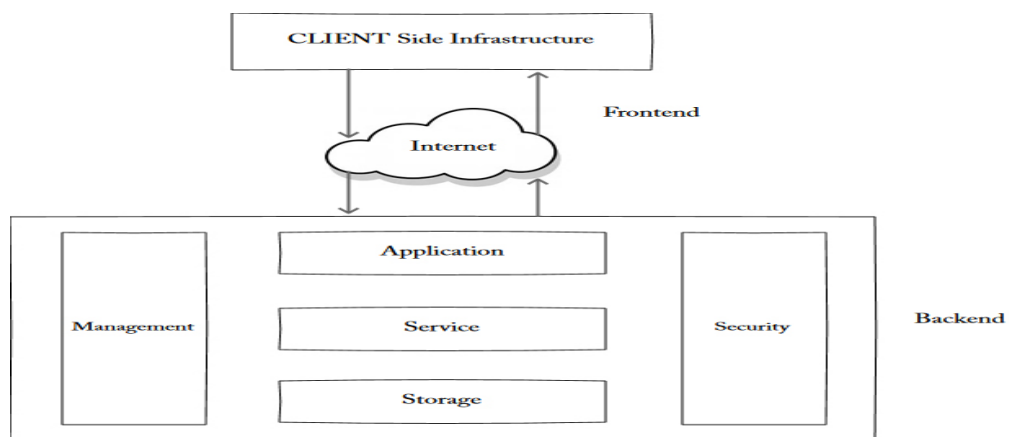


Figure 1.3: Cloud Architecture (Borah, 2021)

Cloud computing design plays the fundamental incorporating component of a cloud, permitting the association between individual virtualized assets across an organization. Each design needs some getting sorted out. With regards to the cloud – this is the job of cloud design, to establish a virtualized environment that empowers pooling assets independently of actual equipment restrictions. A remarkable inverse to on-premises IT infrastructure, cloud computing offers simple, on-request admittance to its product parts and guarantees adaptability and versatility redid to singular clients' requirements and important changes. Cloud Architecture in Cloud Computing, is a mix of a few parts and subcomponents that structure together. The help arranged engineering in Cloud Computing is answerable for giving the infrastructure utility to deal with a

variety of assignments during the cloud runtime. It deals with the kind of administration that is open as per a customer's prerequisite (Judith, 2015). In the Cloud Computing administration, to jump on-request admittance to the organization, the end-client either can claim a server farm or get access from the specialist co-op. It likewise directs different errands and capacities, for example, web services and capacity. To discuss cloud-based conveyance, every one of those segments should be associated inside an organization and incorporated, making cloud computing design.

1.3. Characteristics of Cloud Computing

Cloud computing is task driven on the grounds that the use model is based completely around what clients need to accomplish, as reluctant to a specific developing, equipment or organization framework (Goundar, 2019). Clients don't need to buy or introduce anything prior to utilizing a cloud asset. Nor do they need to keep up or pay for anything during periods in which no assets are being utilized. The above implies that cloud computing engages its clients to simply continue ahead with what they need to do. Today, no one plunks down to utilize a pencil. Nonetheless, heaps of individuals do even now deliberately plunk down to utilize a PC. Cloud improvements may, notwithstanding, begin to catalyze an attitude move from devices close by to main job PC applications. Because of the way that cloud computing is charged on a utilization premise, it has no fixed expenses. A fixed expense is something that must be paid little mind to the quantity of individuals who use something or an organization's degree of creation. This analyzes to a variable cost that will change as per yield levels. For instance, the yearly expense of leasing a processing plant is probably going to be fixed. Notwithstanding, the expense of staffing a processing plant and of the crude materials it devours will change as per the amount it produces. Traditionally computing has included generous fixed costs, for example, those costs caused in the structure and preparing of a server farm. Be that as it may, in light of the fact that cloud computing is progressively adaptable and task-driven, for clients such fixed expenses vanish. The entirety of the expenses of cloud computing are subsequently on every use or variable premise. As shown by the prior illustration of Amazon EC2, handling force would already be able to be bought from the cloud by the hour. The truth that clouds computing has just factor costs is vital for little organizations. This is on the grounds that private companies have customarily not approached the modern, altered sorts of business applications accessible to bigger associations (Ruozhou, 2019). Be that as it

may, in light of the fact that they don't charge an underlying fixed-cost expense, cloud computing providers including Clarizen, Netsuite, Salesforce and Zoho are currently leveling the product access battleground by permitting organizations of all sizes admittance to the most recent kinds of business application. The characteristics of cloud computing is depicted in Figure 1.4 and explained below

1. Broad Network Access: Various customer stages such as personal systems, handheld devices, and smart mobile devices can be utilized to get to these furthest reaches that are accessible over the affiliation.
2. On-Demand Self-Service: Without the human relationship with each master affiliation a customer can plan selecting limits regularly as and when needed, for instance, expert time and affiliation accumulating.
3. Metered Service: This is a reference to organizations where the cloud provider measures or screens the course of action of organizations for various reasons, including charging, convincing usage of resources, or for the most part insightful masterminding.
4. Resource Pooling: Multiple buyers are given the suppliers pooled selecting assets utilizing a model, with various virtual and certified assets persistently relegated and reassigned relying on the interest of the buyer.
5. Rapid Elasticity: Rapid adaptability licenses customers to normally request additional assets in the cloud or various types of organizations. Taking into account the plan of disseminated computing organizations, provisioning can be predictable for the client or customer.

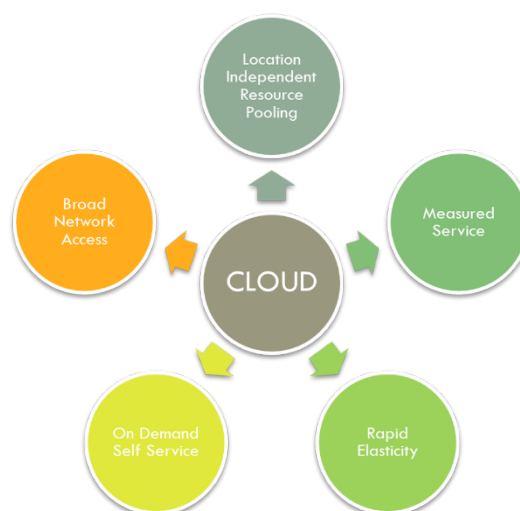


Figure 1.4: Cloud Characteristics (Unitx, 2020)

1.4 Cloud Computing Risks and Security Issues

Gartner in 2008 recognized seven security problems (Brodkin., 2008) that need to be tended to before companies switch completely to the model of cloud computing.

1.4.1 Security Issues

1. Data zone: While taking care of data in the cloud a couple of clients most likely won't know where their data is truly found.
2. Regulatory consistency: Clients can pluck among suppliers that award to be broken down by pariah affiliations that inspection security levels gave by cloud expert associations.
3. Data segregation: Since the data in mixed construction from various affiliations may be taken care of in a comparable spot, so a system is required that confines data from various affiliations and it has to be given by the cloud expert association.
4. Long run reasonableness: It infers the ability to pull out a game plan and all information if the current provider is bought out by another firm.
5. Insightful help: on the off chance that a client speculates inadequate development from the provider, he probably won't have various genuine ways to seek after an examination.
6. Recuperation: Each provider should have a fiasco recuperation show to guarantee customer information is secured in the event of a calamity too.
7. Special client access: Data sent from the customer through the Internet addresses a particular level of peril, considering issues of information proprietorship; undertakings ought to contribute time getting familiar with their suppliers and their rules however much as could be typical preceding administer some frivolous applications.

1.4.2 Cloud Computing Risks

The six specific areas of cloud computing where substantial security attention is required are as follows:

1. Robust separation between data belonging to different customers.
2. Incident response.
3. Authentication of applications/users/processes.
4. Data in transit security.
5. Cloud legal and regulatory issues.
6. Security of data at rest.

1.5 Motivation

With a stretching out number of affiliations going to utilize assets in the Cloud, there is a necessity for ensuring the details of various customers. Some colossal difficulties that are being looked at by Cloud are to guarantee about, secure and measure the details which is the property of the client. Under, we have portrayed the two essential impacts that hold your details out in the Cloud: when the details are moving (travel) and when the information is outstandingly still, where the details are staggeringly expected to be safer. For the condition, the Cloud supplier doesn't have any favored situation of getting to the information truly which is in the nearby affiliation. Regardless, once in a while, the Cloud needs to get to some data which is in the near to relationship, through that entrance; there exists a chance of unapproved entrance of the neighborhood association assets. It portrays the customary issue in affiliation security where the data can go facing dynamic assaults and confines assaults. The dynamic assaults intertwine covering, replay assault, change of messages and revoking of association. Uninvolved assaults unite traffic assessment. These assaults are apparently going to happen when the flood of data leaves the customer relationship to the Cloud affiliation. At the present time of time, there exists an odd for unapproved clients to enter and get to the information in the Cloud. In the current condition, the VMs are doled out to clients of the Cloud. These machines have huge logins. By the way, these logins can be abused and broken. The information may besides be gotten in other damaged propensities (Strickland, 2020).

1.6 Problem Statement

Research Question 1: Account or service traffic hijacking: If the login credentials are lost, the account can be compromised (Marcer, 2019).

Research Question 2: Customer data is available for the third party, which means extra attention needed when saving our important data on the cloud (Calyptix, 2016).

Research Question 3: Data scavenging happens through data removing, which labels data as deleted but does not fully remove data from the physical stockpiling devices; a hacker may then steal data that is not quite erased from the physical stockpiling devices (Singh, 2014).

Research Question 4: Data deduplication allows reducing the need for storage and bandwidth, and allows the identification of files, and contents. A hacker can make a duct of connection to theft the contents of the duplicated files. Adversaries both within and outside may use data to cancel the duplicate as a way of launching a storage attack (Shin, 2017).

1.7 Research Contributions

1. Research on existing work in the territory of planning secure information sharing frameworks in the cloud. Research includes investigation of existing writing and right now existing executions.
2. Characterizing necessities for encryption framework in cloud environment. At that point, in view of these necessities, we characterize contemplations for encryption systems in the cloud.
3. Designing a secure data transfer system for a cloud environment. In particular, we propose the architecture of the system utilizing Dual Key based encryption (DKE) to the data.
4. Implementation of the proposed architecture in the cloud simulator CloudAnalyst, based on existing standards and technologies.
5. Performance analysis of the developed prototype. We deploy our prototype in CloudAnalyst simulator and conduct tests to measure its performance under various scenarios.

From this research work, following points are observed.

1. Encryption techniques can be changed according to the type of data, for example, in this work, text data encryption is done in less time than standard triple DES technique.

2. The challenges for simulating the proposed Dual Key Encryption technique are not much, but in real time implementation many different challenges may need to face. The proposed technique is using system information as one key, which could lead to problem if the system information corrupted or lost along with need to update the system information when a new hardware is utilized for encryption and decryption.

1.8 Organization of thesis

The thesis is orderly in the following way: chapter 2 portrays the current scenario and writing survey that was accomplished for this thesis. In part 3 the proposed model and how the techniques are applied to make the framework safer is discussed. In Chapter 4 the execution and results are described. At last section 5 closes with the conclusion of work done which incorporates types of clouds, qualities of cloud, security and threat issues, cloud architecture, motivation and technical details of this work.

CHAPTER 2

LITERATURE REVIEW

2.1. Fog Technology

Fog is a distributed system, in which information is a good idea saved in devices, data and clouds, and is defined as the fog Network (Vishal, 2019), closely related to clouds. Fog benefits over the cloud are countless, as they now change molds and all businesses like to invent technology (Varghese, 2018). Fog setup offers more options for the efficient data process that benefits organizations. Fog computing is anything that only clouds have permission on the data that can be locally sent to the server and run (Chakraborty, 2019). Because of the wide acceptance of cloud and IoT, IoT has started to achieve fame in the last years, IoT services have presented a number of new complete issues by changing normal items to smart devices (Alahmadi, 2019) and Fog was familiarizing in 2012 to create a data center connection and finish the device faster and safer, secure user features on the Internet as well as, connecting to IoT and cloud to allow the software to treat individual actions in additional tasks. For example, it should be clarified in a fog and the world's IoT idea, which is called objecting, is transmitted online before data is transferred to the cloud (Hatem, 2019). IoT consists, for example, in industrializing any brain entity of an Internet sector, (hand tools, cars, scanners, mobile phones, cameras). In previous years, IoT devices have invaded the planet as in 2020, billions of artificial products and works will be connected together, the word fog means clouds near the soil, so data and computing were put near the user in fog (Puthal, 2019).

Fog technology's main point for improving real temporary programs to fix problems with the day's network, as well as all that, the base supports incompatible devices, like end-of-life devices, enter to points, edge change, and devices. (it was also known as the accounting lips) (Andrzej, 2019).

- Fog technologies mainly consisting of IoT devices, wireless. The independent wireless network display attacker was afraid to corrupt and capture secret data when data was not protected (Ruozhou, 2019). Considering the uneven and soft tasks, Fog systems, which regularly capture individual data from the user to cloud the model, and the opposite alternative, automatically implement performance,

laws and security policies (Marcer, 2019), will follow suspicious network behavior and reveal that in the end, every Fog system should use full internet watching methods that are security and effective. The best common technique to remove these issues is user entrance control and full network monitoring. So, you have to allow high and private value? (Hassan, 2019). To protect the network from network access, information must be given to the real time management analysis system to avoid harmful threats or minimize. The lowest strategy of fog Nora Technology, 2018, has some techniques for growing fog:

- Use large data to improve data transfer among devices and data centers.
- The main case of demands and publishing a vague environment
- Use node technique to improve security for users and transferring the data.
- backup server, cache memory.

Fog technology is divided into three classes: Fog server, Midler, Cincinnati, China. The lower-class marker contains physical links and data to change the data felt for digital symbols such as RFID, NFX, WSN (Puthal, 2019). Middleware consists of the transition and network class, in the middle class, that data from the emotional and process class and has been taken from the middle class and moved to the fog server using Zig-Bee, wireless, BIT, Wired, and LAN in the middle class, which devotes the title to the physical objects. The fog server was distributed into the usage class and business; it is like a front end for third-class users, the fog server for the front user theme; it consists of an app and business class to control different real-time programs, and tag users can also be used at a stable level. To be collected, for example, is dynamically loaded into a store like the smart ones or Wi-Fi on a moving truck like the Grey-hound-Blue network. The variety among cloud, fog technology is the foggy - accounting weather and the top layer is cloud that is Suitable cloud, which can save too much data, is based on the data center's features. Near the user device, the devices are the taming accounting facilities that means the end machines cannot directly connect to the fog accounting without the sense of fog technology (Luiz, 2018).

2.2. An Overview of Fog

IoT apps offer various features, like data, networking, and are also inspirational to make new functions. These devices require computational sources to analyze the data acquired; moreover, speedy decision-making processes are also required to preserve aloft functionality (Net, 2015). By using a traditional server-client architecture, where

information is found by the user, and the server processes it, this can pose reliability, problem development. If a server in a conventional client server architecture was to get overloaded, then a IoT of devices could be knocked offline. The Fog model goals to provide a decentralized, technique solve to this problem. This is obtained through the creation of a new hierarchical allocation and native podium among Cloud and user devices (Tang, 2015), as shown in Figure 2.1. This podium is capable of liquidating, assembling, processing, analyzing, and transferring signals and will save time and resources for communication. This new paradigm is decided to name Fog, introduced at first and officially by Cisco (Khan, 2017).

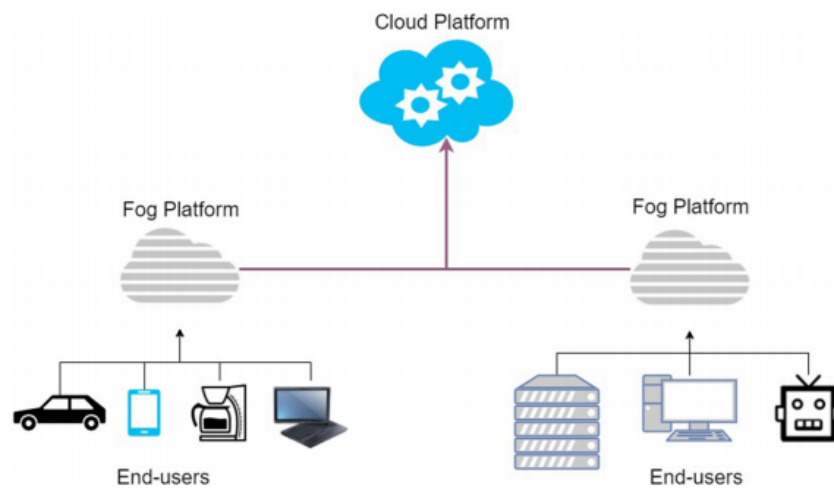


Figure 2.1: Fog between cloud and end user platform (Khan, 2017).

Security is one of the main issues, as we have tons of sensitive data. This could be specifics of any business pricing or it could even be a national secret. All data must be secured and ensure it has all the necessary strategies in it, which made it hard for an attacker to crack the key. The paper focuses mainly on data protection and data privacy as a core component. In the cloud network, though protected data is sent from the cloud to the fog, preview the security danger in the fog like man in the center attacks, add a 2nd layer of protection within the fog level (Akhilesh, 2016). In order to allow overall communication, chips and connection devices are integrated. A specific service could consist of a plurality of components, each deployed in various geographical areas, resulting in a raised vector of attack for some objects. FN is the gateway for traffic data, which is especially weak to these offensives. This is particularly true in the text of network activated internet of things systems, whose offensive mistakes can be from

humanitarian made network infrastructure vandalism, pernicious software that cause data infiltration or even physical device entrance. A broad ambit of research concentrates on authentication, and cryptography to boost internet security in order to safeguard against cyber offensives (Roman, 2011). In addition, in networks consisting of 100s of 1000 of devices, how to determine safety efficiently and reliably and calculate its dangers is critically necessary in order to provide a comprehensive protection and risk estimate (Riahi, 2013). This becomes challenging when workflows can change and adjust in runtime. For these purposes, it was assumed that being able to dynamically assess the protection of dynamic IoT software synchronization would to be progressively crucial to the safe placement and processing of data (Zhenyu, 2019). With the arrival of the fifth generation of the network, the increase in the production of smart cars and the latest smart house devices that can actually be purchased from most relevant stores, can affirm that they are moving towards a world where devices will be all across and all around us and have sensors. Consequently, there will be a huge number of Internet-connected devices. This enormous number of devices could be more than just connected devices (Bolarin, 2019). We may be part of a Fog network capable of self-orchestrating and providing Fog-level services, without relying on centralized cloud infrastructure. Yet this indicates a shift in the actual Internet architecture used to provide services used since the first days of the network, from centralized to distributed (Zheng, 2017). This shift to a decentralized Internet presents a whole set of new challenges, like who will own the Fog infrastructure, or who will be the Fog operator. It also poses the need for a new decentralized security architecture, because the current security architectures are focused on centralized cloud infrastructure and are therefore unstable and not sufficiently scalable for a Fog framework. Moreover, that for the Fog to become a legitimate and used technology, it must be able to orchestrate itself, including protection, without relying entirely on the cloud, thus creating the need for a new completely distributed security architecture, because if the Fog relies on the cloud to function safely, it is not Fog at all, it is merely an extension of the cloud. (Marcer, 2019).

2.2.1. Data Intensive Internet of Things Applications

Any physical thing, which has the strength to communicate with a confirmed stockpiling capacity and processing power through the Internet for data collection purposes, is named a "thing" or IoT device. These "things" may be tiny scouts,

handsets, agents, actuators or anything else. Objects are not only compound devices like mobiles, but they can also be items of daily life, like a segment of art, food, landmarks, furniture, etc. IoT has many software areas, from home automation to job functional districts like logistics, transport, healthcare, protection, intelligence and much more, as shown in Figure 2.2. Several of the most frequent data intensive apps are outlined below (Asim, 2019).

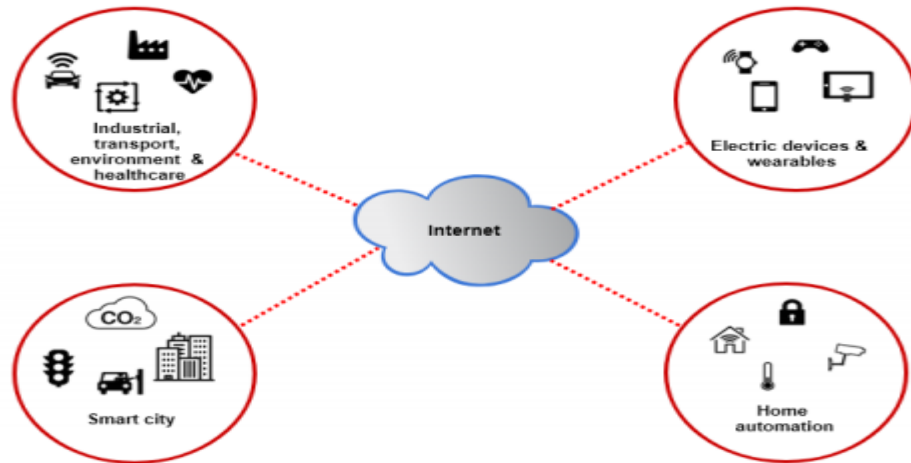


Figure 2.2: IoT Applications (Tariq, 2019).

1. Smart Homes

Citizens invest it is hard in smart homes, saving money and gaining management over their lives. Moreover, it uses searchlight and actuators linked to devices and accessories that are wirelessly linked to the network at home, and can interconnect with each other to supply a seamless user experience for customers. Different types of sensors that are installed in homes are permanently collecting information from the surrounding area and performing various brainy related home functions like automatic turn on, off home lights, appliances and agents, controlling home temperature, safety and much more (Zhou, 2013). Sensors are various so the data collected are of different types. For instance, installed sensors to detect fire, moisture, gas leakage, changes in temperature etc. For example, if there is gas leakage it automatically shuts off the gas outlets. As well ground sensors designed to feel the compression and monitor residents' movements; they trigger appropriate behavior in the event of any sudden locomotion or fall of a person on the floor. Also, use security cameras to monitor residents' motion

or moves (Zhang, 2017). Energy conservation is another essential feature that smart home applications can achieve (Lyu, 2017).

2. Smart Cities

Is an extensive yet significant Software of IoT. This is primarily introduced to tackle particular concerns relating to well-being for people, like traffic control, electricity, education, transportation. It relies heavily on intelligent sensor devices to gather data like temperature, moisture, traffic state pollution. For instance, sensors are scouted at homes, and reservoirs for the effective use of water resources in the city. Searchlights installed at home save records of the quantity of water supplied, timing, pressure, etc. Furthermore, users should track the water meters for the purpose of charging (Ghafir, 2018). Traffic control is another application in which data gathered from various sensors falls in the city to regulate traffic flows in reaction to demand. Smart city software results in an exponential increase in data that brings with it many troubles and challenges, like data privacy and security (Tariq, 2019).

3. Smart Healthcare

Health Service Management is another major task automated with the use of intelligent IoT based electronic health systems (Nepal, 2015). Technologists have built several wearable intelligent devices that track the inclusive health status of users. These devices save the registry of the health of the patients and produce warnings in the event of any suspicious behavior. Smart e-health systems, for example, have proved to be very beneficial for the old person and the disabled, who find it difficult to travel (Tariq, 2019). IoT systems may be implemented for remote collection of medical data like blood pressure, heart rate, physiological and transfer to large data centers for stockpiling and diagnosis in the healthcare sector. (Luong, 2016). Given the maximum sensibility and privacy of the medical information gathered, one of the best serious menace to healthcare is data privacy, security (Farahani, 2018).

4. Agriculture and Environment

The technology has played a crucial function in making people's everyday lives easier, but there are dark aspects that need to be tackled to develop them. The intelligent environment systems help to track and manage the environment (Zhang, 2017). Air pollution has become a crippling epidemic worldwide. Applications for use the scout to measure moisture, temperature etc. Farmers using smart systems to get higher

quality crop yields. Sensors collect and send to servers the ecological parameters that are needed for cultivation like soil data, warmth degree, dampness, insecticides management etc. Such parameters are evaluated in genuine time to inform the peasants regarding the soil's status and if it is suitable for agriculture. It thus assists to produce the crop's quality (Zhang, 2016).

5. Energy Conservation

Another useful yet complex IoT software is the smart grid. This time, it has become the need, implementing a dole out and user centric intelligent grid network that aim to provide safe , efficacious, secure and quality ability supply (Mushunuri, 2017). Smart grid technology includes the smart and smart 2-way exchange of knowledge between the customer and the supplier. IoTs or scouts are mounted locally and at the grid stations (Jogunola, 2018). It is the duty of the sensors mounted at the customer site to gather data relating to electro supply, usage trends, pricing, smart measure and some other data. While the grid station sensors track production, delivery, production discontinuity, position identification, customer information, etc. (Jogunola, 2017). The data gathered from those scouts are transmitted to their competent management centers where proportional alerts are created for clever and fit electricity allocation. Effective implementation of the intelligent grid network includes clear data protection and privacy solutions (Tariq, 2019).

2.2.3. Fog Enabled Internet of Things Security Requirements

IoT applications which are fog enabled can be used in any area of life. These smart device networks are supposed to be remote in nature and use Wi-Fi links to communicate with other FNs or (IoT) devices (Kim, 2015). This wireless media is vulnerable to various attacks on the network, like eavesdropping, etc. Security is particularity, honesty, and availability. Confidentiality includes that adversaries are not allowed unauthenticated and illegal enter to information. Dispassion indicates the accuracy and completeness of the results. The prime problem is the particularity and protection of the data that home scout and discoverer produce. The data gathered was stocked at various places, for example on the interconnected computer itself, at the edge, in the cloud or on-site infrastructure (Zhang, 2017). Investigators and security skillful regularly find vulnerabilities in IoT devices in smart homes that could lead to without permission entrance to user data and endanger the privacy, protection and security of consumers (Chandrasekhar, 2017). Open connections and verification in

smart city environments are also stunning security issues (Daneva, 2018). It should encourage digital forensic inquiry between the components related (Baig, 2017) In addition to the implementation and maintenance of end-to -end data collection, transmission and retrieval protection and privacy supports (Tariq, 2019). Now Fog is to come after Cloud Computing. Fog's goal is to bring the research, processing and stockpiling functions back to the edge of the network (Lahami, 2018). The more our environment is networked, the more it processes data. The communication route through a cloud storage access and then back to the terminal leads to high latencies- for instance, for autonomously driven cars whose processing of information must be guaranteed in genuine time (Lahami, 2019).

Fog Technology Advantages:

- Processing data faster due to decreased network traffic.
- Networked IoT devices also work when the Internet is down or cloud communication is delayed.
- Sensitive company and consumer data need not be migrated to the cloud and should stay on the spot.

2.3. Overview of Cloud Computing

Clouds are continually developing to enable industry and market growth, and are instrumental in promoting key technology fields like the IoT, AI and ML, information security and big data (Rainer, 2015). Given substantial growth and market adoption of cloud services, the education strip has lagged backward in providing manufacture relevant technology courses leading to a shortage of technology skills (Viktoria, 2019). Concerned about the security issues and strategies to reduce cloud-based deployments. This covers cyber security, cross-cutting security analyses and important cloud infrastructure, software's, and data secure practices. This also includes conventional security technologies and their carrying out using cloud-specific technology and security features that cloud vendors offer (Foster, 2019). Cloud computers have grown by two main targets; reducing IT costs and providing active services to both users and organizations, cloud foundations will be fixed in transferring data away from laptops, and desktops to huge data-centers. This fact may provide an increase in updates to restricted devices in the form of business performance updates. claim that before cloud computers are fully installed it is necessary to face some security problems that favor the nature of the cloud (Antonio, 2019). Although security

has improved in clouds in recent years, the current cloud computer is still open to security challenges. Among others, customer data is available for the third party, which means extra attention when saving our important data on the cloud (Calyptix, 2016). Other challenges are taken from the Poor Access Ability Facility, which is taken from closed customers with one selected CSP & rely on all kinds of services (Albugmi, 2016). and their incomplete or incomplete customer data, some data from the cloud may then not be deleted because repeated data may be on the cloud that improves security levels in commercial clouds, especially by the customer (Antonio, 2019).

2.3.1. Major Security Section in Cloud Computing

1. Security of Software:

The possibility of application security during development is due to the undefined security. Software faces specific challenges and menaces to their security. The increased attacks on the applications need enhanced application safety by eliminating security vulnerabilities within them. Apps' top security threats include intrusion, fractured authentication, ticklish data leakage, XML external-entities, XXE, fractured entrance control, application wrong configuration, cross-site- scripting, XSS, unsafe sequential, usage of compromised elements, and poor logging and monitoring (Andrzej, 2019). No software programming stage is immune from the possibility of innuendo to security shortcoming. The “OWASP” establishment notes that security vulnerabilities can be hit the needle into the program at any point of the life cycle of application development (Schouten, 2018).

2. Infrastructure Security

Security of infrastructure refers to jointly with the physical infrastructure and virtual. The infrastructure protection targets to create infrastructure trust (Singh, 2013). Infrastructure is the fundamental aspect of cloud computing and is made up of virtual resources and physical. Digital tools include (VMs) that provide cloud infrastructure for the movement, storage, processing, and analyzing of information. To allow multi-tenancy redoubled Virtual Machines can run on a single physical server. Virtual Machines are weak to security threats, which pose a danger to cloud privacy, and protection (Chatterjee, 2017). Failure to deploy a VM will lead to a shortage of isolation among other Virtual Machines on the same server resulting in data infiltration and express-VM attacks (Singh, 2013). Unique attacks on VMs risk client data. Some VM offensive consist a cross Virtual-Machine side channel hacks that helps an attacker

to expelled secret key, exploit resources and other data, build VM offensive, Where pernicious code is inserted into VM for transmission to other Virtual-Machines at the time of creation of VM rollback, and migration hacks, where an attacker make use of the vulnerability of VM throughout transitions among hardware objects, and where an attacker try to steal sources using the VM tables (Mirković, 2019).

3. Storage Security

Cloud computing handles data extensively; data is downloaded, processed, and stockpiled in the cloud. Cloud data volumes require storage protection. As cloud clients are physically split up from their information, they experience a deficiency of management, so it is vital that measures are possessed to assure cloud computing storage protection (Chatterjee, 2017). In addition, cloud service providers store data in highly redundant fashion in their data centers and although redundancy offers improved data availability, it also gives attackers an extra chance to robbery sensitive data (Nedbal, 2014). An attacker can steal data in multiple routes but there are two types of attacks on data stockpiling. The two attacks on data stockpiling are data digging, and, data cancel the duplicate (Khan, 2016). Data scavenging happens through data removing, which labels data as deleted but does not fully remove data from the physical stockpiling devices; a hacker may then pilfer data that is not quite erased from the physical stockpiling devices (Singh, 2014). The next attack on data storage is the deduplication of data. Data deduplication allows reducing the need for storage and bandwidth, and allows the identification of files, and contents. A hacker can make a duct of connection to theft the contents of the duplicated files. Adversaries both within and outside may use data to cancel the duplicate as a way of launching a storage attack (Shin, 2017).

4. Network Security

Components of cloud are connected through a network, and entering an external cloud is a network. Attackers can exploit the network's vulnerabilities to robbery sentient and private information; and the internet attacks have three types (Khan, 2016). Network attacks listed below These three attack categories authorize an assailant to take advantage of a set of weakness in the cloud (Whitney, 2017).

- **Port scanning:** Is when an assailant searches web ports to exploit weakness in the ports working services (Chatterjee, 2017). Scanning a port will result in DoS.

Intervention finds firewalls, and systems can however provide some security opposite port scanning attacks (Mishra, 2017).

- **Botnet attack:** happens when a botnet steals host data and converts it to a remote bot-master. Botnet builds a managed network in which the bot-master interacts with many infected devices and behaves as zombies, that in turn will permit an attacker to enter the firewall of the network inwards. Command-lines and manage centralized servers that send limited size lines of commands to the zombies to theft or harm critical data (Waghela, 2016).
- **Spoofing attack:** An attack occurs when network entities are impersonated by the attacker to damage the network (Khan, 2016). The fundamental goal of a spoofing attack is to robbery the approved user's sensitive data to target the internet hosts to prevalence malware or bypass enter management (Jadala, 2017). Below are some examples of spoof-attacks. A spoofing attack by a DNS can direct all traffic to a Domain-Name-System server to the network of the hacker. Another epitome of a to turn on attack includes substitute the IP in a network packet with a fake IP address (Rahman, 2019).

2.4. Cloud Computing

Cloud is very new to it and is about improving the basic way businesses deliver IT services to their workers, consumers and copartners the diagram below shows general cloud design data in different elements (Nasir, 2016). Today the cloud has become a global path, attracting selfishness in the business world as well as in the academic world, because technology offers a paradigm change in the field of (IT) (Mirković, 2019), thus providing data assets quickly held, accessed and processed on the forums. Cloud is a term that activates the full-time and easy-to-demand network to enter a shared area of conFigured computer sources (software, services and server, network, and ware) that is quickly distributed and is published with minimal control attempts or a basic card service provider. (Rahman, 2019).

There is three different parts of Cloud:

- Customer, Front End as Browsers.
- Cloud middle services like IaaS, SaaS, PaaS.
- Back-end as physical and professional services (Chirag, 2013).

SaaS: Cloud service that gives users access to services based on clouds from organizers. Users don't have a build-up program on their personal computers. Programs instead of remotely from the network enter to cloud on the API, or Web. Users can save and paste details by using the application, and help projects (Violino, 2019).

PaaS: It is a cloud offer that users can create, control and deliver applications in this environment. In addition to savings and other accounting sources, users can also use a set of pre-data paraphernalia to organize, making new one and testing the software (Judith, 2015).

IaaS: It is a cloud offer that supplies clients with access to cloud resources such as hard or storage, server, and networking by a seller. Corporations use their programs and systems in the company's infrastructure (Javier, 2020).

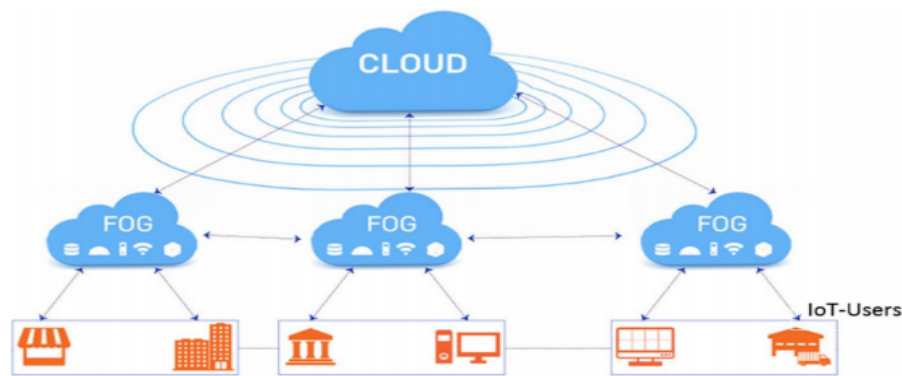


Figure 2.3: The relationship between Cloud and Fog (Ahsan, 2019)

As shown in the Figure 2.3 cloud and the mobile cloud directly connected with a Datacenter but fog connected end-devices with nodes and nodes have a relation with network devices. After that connected with a Data centers. Edge computing is like looping communication between the network device with end-devices and last technology is the mobile edge is connected with a datacenter by sensors and towers. Fog is in the middle of Cloud, IoT Technology to make a network better and real time monitoring devices and Data centre, users can only connect with end-devices, and the role of fog is for connecting users with a Data centre (Cloud Technology) to the safe user's privacy by using of the cloud services. There are three different online formats in cloud, using the SaaS as a service, and network as a service. The basic concept for

approaching an accounting station close to the user has been solved throughout the information process close to the edge of (Guan, 2019). distributed into two categories of devices, IoT and end of devices:

- End-to-end devices like mobile and smart things etc.
- The edge is set as a border hunter and the road, bridge, and wireless access point, etc.

Privacy security challenges such as:

- Various harmful - featured fog count.
- Data Protection- Data Security.
- Access control- Attacks.
- Malicious attack- Privacy.
- Authentication.

Cloud is a very good request media framework that offers a wide range of geographical location users with always a common information base from everywhere in the media, space and communications resources, as Sisco wrote (Antonio, 2014). The academia and industry propose to count the brown field to complete the cloud by expanding the ability to process clouds, grids and depots to the edge, the closest to users (Baig, 2017). Fog and loud rent VMs to permit effective use of sources (Wasim, 2015). Programmers have to take into consideration make the working better of this website by connecting Guests to the cloud through the Tam A. al-Harbi server in 2019. VANET in particular, shows a large number of cars in the city that can be connected to installed computers, grids, and connections in smart lines (Sookhak, 2017), and units for transmission. These connected cars can create an attack and serve as a FN treatment (Ruima, 2019). Foggy increases the problem of hiding in cloud counting by inserting fog units on the edge, which reduces the length of the user and processing units to reduce answer times. While strong FN publishing on the edge increases the network's performance (Zhang, 2016), one of the basic techniques used in big data in fog and IoT technology it is for personal, data protection features on the Internet (Abbas, 2019). To improve to connect faster using a fog accounting is known as a fog node, for example, for AI technology. Baker's 2019 C-Or-Use Fog Server to end user data to prevent the attacker from accessing or attacking the server or node, and can be more secure (Tariq, 2019). IoT studies at the (FTC) called on companies to follow these acceptable traditions to solve

client security, privacy dangers (Almeida, 2015). That tools are used in large assembly to face a set of security threats and old data, such as the unallowed door to and wrong use of client's data (Roman, 2018). Data from devices is often processed in infrastructure, and cloud services that attract attention not only investors, but also investors, such as cloud and corporations' workers engaged in using this data for their own benefit, like advertising (Jain, 2019). can help solve some security problems related to data achieved by IoT. For instance, it will be stored on the website and an incomplete data theme and time limiting the data stored and presenting it to the cloud (Zhang, 2017). Due to the lack of response time, approaching IoT devices, decentralized structure and interim support, it has its own security and privacy challenges (Ghafir, 2018). IoT and Cloud Computing are used in many areas of today's technology, for example, calling for the construction of smart cities, smart homes health, energy environment, energy protection, agriculture but there are some issues in the field of privacy, security and communication (Gultekin, 2019). New technology came for fixing this question named (fog) has small errors but a big company daily is trying to solve the errors of the hall (Jose, 2019).

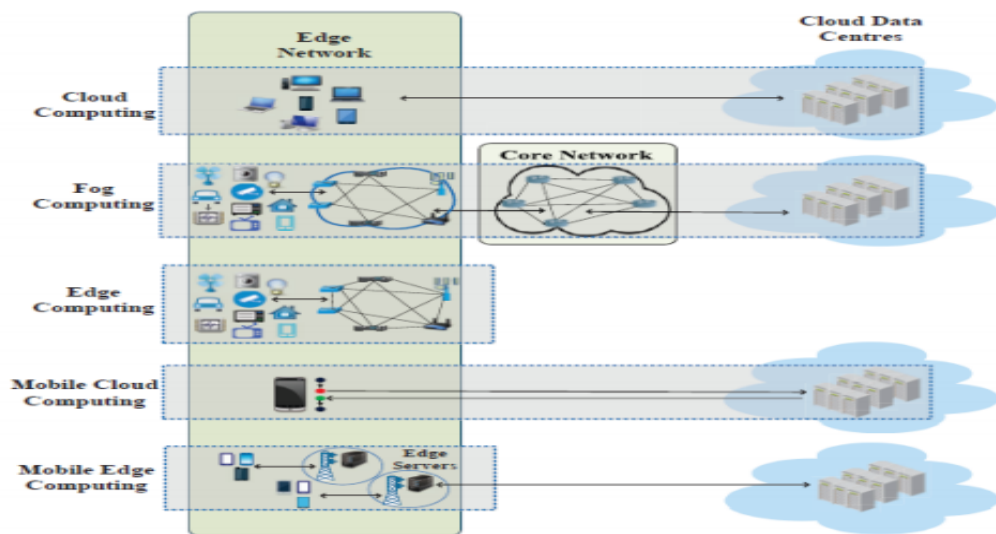


Figure 2.4: Different connection among Datacentre technologies (Rahman, 2018)

The Figure 2.4 depicts the different communication types with different end devices to the cloud. From the Figure it is clear that the edge network comprises all types of devices which are capable of performing computing operations.

2.5. Difference between cloud and fog computers:

1. Clouds are centralized and hold a vast amount of data, far from consumer computers, that can be placed around the world. The fog is dispersed and includes some small knots near customer equipment.
2. Fog is the class that arbitrates between clouds and devices such as computers, laptops, mobile phones, etc., as fog of the sand, and buys less time to transfer data. When there is no class, the cloud needs direct contact with end machines that require more time than using a fog computer.
3. Clouds are not enough to respond but cannot be compared to Fog Mag in terms of the only time of answering is low.
4. Clouds do not provide any reduction in data during data transfer, but foggy data can be made when sending to clouds.
5. Cloud counting protects less than fog- segregated.
6. In the cloud, the system's mismanagement is low in fog.
7. Cloud PC is protected, but fog count is too protected. Since fog is distributed and there is a complex architecture that's why fog is protected from clouds.
8. Only in the cloud, double-double data sources can be combined to combine the source of the fog and device data.
9. Clouds fail without the Internet, but since Fogg has used several measures and measures, if no Internet connection.
10. Figs have a soft infrastructure with three models like IaaS, SaaS and PaaS.
11. Fig counting has supported the center in user control that the cloud can be centralized or can be represented for the 3rd party.
12. Resource control is centered on the fog count that is centered or distributed in the cloud.
13. Fog, and Cloud Future.

Fog architecture is decentralized to control the sources and performances of the computer and has no central service. Thus, FN self-organizes and helps provide real IoT time service to end users. Location information is capable of finding a computer location. That is linked to the close to FN, and they are aware of where the server of fog is (Santos, 2019) Location awareness may be used in emergency cases, or for pick outed ads. Saving the cache is also one of the good methods to stop unrelated or unrelated data from crossing through all infrastructure, thus reducing store space and

reducing concealing time (Chatterjee, 2018). Fog is suffering from several harmful hackers, and the device's product can be by strict make a deal without adequate safeguard. One of the harmful errors that has begun in this attack genre is when dry solar devices are communicating in the IoT environment and demand unlimited processes, store facilities, and the attacker will take action (Tanwar, 2019). The rapist prohibits users from accessing computer targets or slipping IP devices and sends fake requests to enter process/storage simplification. When the IoT device feels data, it will get information to the FN (Chiang, 2017). It's hard to deal with too much data on IoT devices if a hacker can enter the server and can't get a huge amount of information because he can only communicate with relatives to the FN, and it's difficult to deal with a huge amount of information. Foggy growth based on the 5G network threatens to protect data exchange (Zhang, 2019). The integration of fog and IoT, and cloud communication grids has been taken into account and the wide implementation of IoT benefits from the ability of fog roaming to rerun IoT system operations. But it still has limited power and long rows. However, using electronic systems to connect communities, this can weaken the network from persecution, such as breathing, dos attacks, and desperation (Jaïdi, 2018). That is, the number of fog accounting shortcomings is very small now between both IoT end mobile devices and cloud data centers (Wang, 2018).

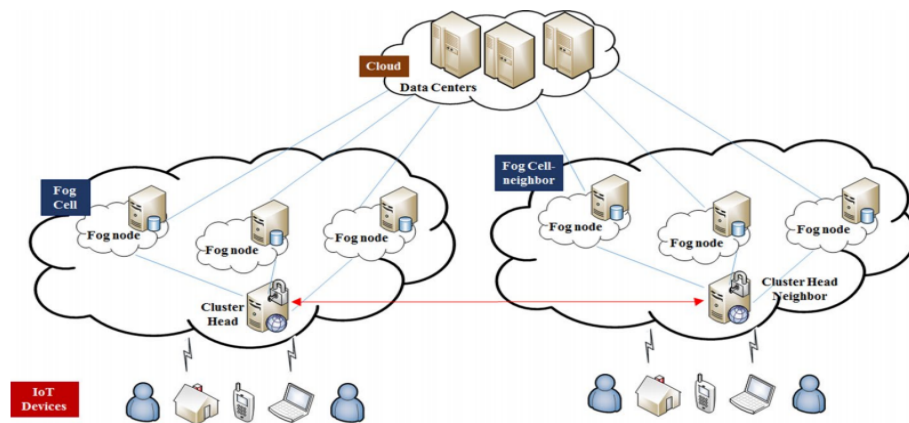


Figure 2.5: Fog Node structure (Daoud, 2019).

Fog is using the node way to protect and improves the network, datacenter information connecting a user to a nearest datacenter to retrieve data faster and prevent and avoid the attackers to enter the data shown in Figure 2.5 connection user to closest FN after that Communicate FN with Data centre.

Fog Computing Ways of prevention:

- The proposed chart step is a collection, which is a methodology (ML) that collects data points. Split into small cells by FN. First, random selection of FN numbers is created to be used in each cell. Then, each cell will manage the Fog-Node manager which ranks the head of the house (Bangui, 2018). Therefore, when an IoT user tries to sign in to FN, this slowly-required sign-in request must be managed. Ranging proposed an unbalanced secret protocol based on an algorithm for CP-ABE that could implement a major secret function between the FN (Puthal, 2018).
- The second task of the FN administrator is to use the Resource Classification task. Table, IoT users appear by type of commands knowing the number of resources at any moment. Different FNs then affect the selected users, according to the requested services if no end is left, the user must wait to work to use the FIFO table shape (Zhan, 2018).

Fog is provided for data storage using RSA and THE CC to protect or encrypt data for unknown people.

2.6. Fog characteristics:

- A. Near to the end-user.
- B. Low response time.
- C. Real-time interaction.
- D. Save storage space.
- E. Geographical distribution.
- F. The mobility shore.
- G. Location Awareness.
- H. Decentralization.
- I. Heterogeneity prop.

Trust access control and resource management tasks:

- 1- Monitoring process
- 2- Resource management,
- 3- Access control management

It also provides data storage services and divided simplification for end-users still at the beginning of its childhood (Daoud, 2019).

Feedback Header Area

Response header cells permit the server to transfer extra response data that cannot be inserted into the privileges line. These header cells offer server data and however access to the source known by URI that is shown in Table 2.1.

RH = Location; server; originality; www (Auto vertical) = in this area, set the end-user's trust value, which proposed its level of trust.

Table 2.1: Server Request

HTTP Code	Status	Header		Empty Line	Entity Body
Line		Response- Header	Entity- Header		

2.7. Security Issue in Cloud and Fog Technologies

Fog computing gadgets and devices can face real security issues in the environment, as fog gadgets are usually used in locations outside the reach of defense and observation. They are subsequently unguarded to malicious attacks, such as the capture and listening of information, which can jeopardize the functioning and systems of fog gadgets (Maharashtra, 2019). The Fog devices are sometimes placed in such positions where its protection or its surveillance can't be guaranteed. This may cause malicious attacks on them e.g., fog devices can be tampered to prevent normal operations (Goundar, 2019). Fog devices can also be replaced with malicious nodes to jam the network or to perform Sybil attacks onto them. Fog devices can also be destroyed to stop its operation (Mohammad, 2019).

2.7.1. Authentication and Authorization

Is a significant problem for the protection of fog, since services are provided by front fog nodes to large end users (Shanhe, 2015). Because cloud storage is synonymous with having confidential data stored by users with both a CPC and a CSP, IAM, a form of access management, is of great importance. CPC authentication can be handled by the CSP or outsourced to third party specialists (Elom, 2015). To provide a mechanism to authenticate the FNs amongst each other and to IoT and Cloud level as well (Stojmenovic, 2014). Have also considered authentication as one of the major security issues of fog. The IoT device being resource constrained generally outsources the computational capacities from potential Fog nodes. This exchange of services also requires authentication mechanisms to provide a reliable communication path to work upon (Verma, 2019). Establishing relation between End user devices and IOT Devices and Fog nodes. Authorization identifies as officially having accessing right where as Authentication is Action of proving. During the establishment phase access right and identity of the node, which wants to connect, is verified. For accessing storage and processing services, end users should get authentic to the Fog node. Traditional Public Key Infrastructure along with certificates is not appropriate due to resource limitations of IoT devices (Mukherjee, 2017).

2.7.2. Access Control

Access control was a powerful method for maintaining device security and protecting user privacy. Conventional Enter control is typically dealt with in the same confidence environment. Although the enter management in cloud computing is ordinarily cryptographically applied for outsourced data due to the outsourcing aspect of cloud computing. In key management, symmetric key-based solution are not scalable. Many public key approaches are suggested that pursue fine-grained regulation of access (Shanhe, 2015). The fog emerges from cloud computing, which is a major expansion. Therefore, it is inevitable that it acquires various security issues related to cloud privacy. Yet many of the normal cloud-based approaches are useful in fog computing (Wang, 2017). Can lead to poor controlling and any unauthorized user being capable to access data, program installation license and configuration changes (khan, 2017). Ensures that only approved entities are able to connect to the middleware, and refuses remote access to the network from unknown things. Before they can interact with other

parties in the network, remote items need to be authorized by an administrator or one of the users. Only allowed "stuff" may therefore access certain resources or perform a given action, such as accessing data or upgrading an IoT system software (Wissam, 2017).

2.7.3. Location Privacy

In fog, the location privacy mostly denotes the location particularity of the fog nodes. As a fog node generally shed its load to the closest for node, which the load is shed now has an idea about the location of the fog node that has shed its load. Moreover, if a fog node uses multiple fog nodes to offload then the whole path trajectory of the network can be revealed (Verma, 2019). Assuming the nodes collude with the fog. So long as like a fog consumer is attached to an individual or a significant item, the individual or object's location privacy is at risk (Mohammad, 2019). If a fog client permanently chooses its closest fog server exclusively, the FN will certainly know that the fog client who uses its computing services is close by (Khan, 2017). The only way to protect the privacy of the location is by obfuscating identification, so that even though the FN knows that a fog client is nearby it can't recognize the fog client (Daniel, 2019) However, if the fog client uses multiple fog node computing resources in an area, its location can boil down to a small region, as its position has to be at the intersection of the coverages of multiple fog nodes. To protect the privacy of the position in such a situation, the approach used in (Shanhe, 2015).

2.7.4. Trust

Plays a major function in cultivating relationships dependent on prior Fog node and edge system experiences. A FN is regarded as the most important aspect, because it guarantees particularity and anonymity for users (Elmisery, 2018). This portion must also be trusted for deputation, as they must be confident that the FN will enforce the global concealment mechanism on their released data and will only cause non-pernicious behavior. This includes that all FNs that are part of the Fog technology have some degree of confidence in each other (zhang, 2018). The confidence calculation is done in a decentralized manner using the entropy description in (Hatem, 2019). To achieve user privacy the local hidden agent applies the local hidden process. The regional disguise agent resides only in a Fog node. On the aggregated user profile, it implements the global hiding operation. In the Fog architecture a service layer is adopted to improve and manage confidence (zhang, 2018).

2.7.5. Data Integrity

This question occurs because devices at the edge generate massive amounts of data in fog computing, a widely distributed network, which, naturally, has to be transmitted to FNs for storage and recording as well as for. Furthermore, the FNs also have to communicate with edge instruments and data pools in cloud (Rahman, 2019). All those composite operations make the data vulnerable to hacking and disclosure. Even so, there is a solution to the question (Abbas, 2019). Simply use masking techniques or lightweight algorithms for encryption. In addition, there are scores of areas where cooperation takes place in fog computing (Albugmi, 2016). This can cause privacy and security issues. These problematic regions include authentication and authorization, identification control, resource enter control, implementation and coordination of securely distributed decisions, consistency of service and security, knowledge involvement policy, etc. (Yaakob, 2015). Solving the above questions, (Nunes, 2014). Put forward the idea of a policy-based resource control and enter management system, which would ensure a stable relationship and interoperability between resources between the diverse resources researched by users (Maharashtra, 2019).

2.7.6. Privacy issues

It is another issue that needs to be secured. Data can be of different types. e.g., medical data which consists of your health issues, medical tests or treatments, medical devices being used like pacemakers etc. An attacker can use this data to personally craft attacks related to your medical conditions (zhang, 2018). The data can also have personal information like facial data, fingerprint data, credit card details, financial data, venereal details, friendship data etc. Attackers can use this data to craft a social engineering attack for you or steal your wealth (Alahmadi, 2019). Fog devices lie in the middle of the network path between you and cloud systems. Data needs to be secured to prevent its leakage or being eavesdropped. It can be achieved by encrypting the data and allowing its access only to authorized parties (Hatem, 2019). Leakage of private information, like location, data or use, is grabbing recognition as users use cloud technology, wireless network, IoT services (Shanhe, 2015). There are also difficulties in fog computing to maintain some privacy, as FNs are near to users and can collect more ticklish data than the remote cloud in the core network, certain rights such as data, use, location and data searches (Maharashtra, 2019).

2.7.7. Account hijacking

Account or device hijacking for cloud computing is a network related problem. Account Hijacking is the method by which the attacker tries to access the account to steal the individual user's identity. The solution to avoid hijacking of accounts is multi-level authentication at different levels (Archana, 2017). User identity protection should be very powerful, Network control, Data infiltration prohibition Technology Vulnerability Detection the combination of both and data recovery techniques should help to resolve the threat (Antonio, 2019). Although attack tactics like phishing, rigging and manipulation of software Causes of weakness are not unique to the Cloud, the cloud account of hijacking may have very earnest implications like auditing, manipulating data, transaction and activities, retreat falsified data and redirecting clients to unauthorized websites. (Mozhdeh, 2020). This assault effectively hijacks the identity of another. The intruder then gains more access to personal identities due to authentication management flaws (Puthal, 2019).

2.7.8. Denial of Service

DoS is simply an incursion that denies the individual user the contact or the network resource. Communication delays among the end user and the cloud providers occur. The IDS is the most joint way of protection against some offensives (Alahmadi, 2019). Alternative approach for DDOS cloud securing includes the use of intrusion disclosure Virtual-machine program. When an IDS defines an unusual increase in inbound traffic in this system, the aimed apps are relocated to Virtual Machines hosted on alternative data servers (Archana, 2017). In this attack, the illegitimate user sends inappropriate messages from invalid return addresses to the network for authentication requests to prevent other legitimate users from entering the cloud service (Parikh, 2019). A DoS attack is an attempt to make unable to use the services allocated to the approved users. During such an attack, a huge number of requests overwhelm the server delivering the service, and thus the service is inaccessible to the approved user (Almeida, 2015). Often when I try to enter the site see that can't access the website and find a mistake due to overloading of the server with the demand to enter the website. This occurs if the number of requests that a server can handle is greater than its capability (Rohit, 2014). As well as causing congestion, the occurrence of a DoS attack increases bandwidth usage, making some areas of the clouds unavailable to users. The most common way of defense against these offensives is the use of an IDS (Abbas, 2019).

In it is used a security federation. For defending against threats like this. Each cloud has its own separate IDS load. The various intrusion detection systems work on the basis of exchange of information. When a particular cloud is under attack then the whole network is warned by the cooperative IDS. Voting takes a vote on a cloud 's trustworthiness, and overall machine performance is not hampered (Baker, 2019).

2.7.9. Data encryption

The fog emerges from cloud computing, which is a major expansion. Therefore, it is inevitable that it acquires various security problems related to cloud computing privacy. But in fog computing many of the normal cloud applications come in handy. For example, it is possible to use the calculations, RSA and AES, the usually preferred encryption solutions. Encryption methods (especially in their asymmetric form) are ordinarily used to keep data content confidential (Ammar, 2018). It can be used to preserve the quality of data, but in general it is not appropriate to secure the determination of the dispatcher and receiver, which can be entered by a malignant eavesdropper simply by surveillance packets. (Akhilesh, 2016). Authentication mechanisms (passwords, biometric authentication etc.) are commonly used for monitoring (and restricting / preventing) access to sensitive data while data is at rest. It also includes the concept of acceptable access privileges to map a user's identity with the data items that are allowed to enter (Rahman, 2018).

2.8. Problem Definition

Security is one of the most important things because there is a lot of sensitive data around us, it could be national secret or any important data. It is very necessary to secure those data which attackers use to crack the key (akhilesh, 2016).

2.8.1. Existing Security system in fog

Currently, in fog, the Decoy system is considered a security model where the client must first register in the Decoy system and then provide the login information and, once signed in, respond to the safety question that was asked when creating the identity. (Neha, 2016) . Decoy system is one more way of trapping the hackers with the bogus files that mislead the attackers by displaying the file with spurious titles on it and only the consumer who knows his information will realize that this is a fake file as well as the attacker doesn't know the difference among the spurious file and the original file and when he clicks on the file and tries to download it The device would

be told about the attacker and therefore data from the hacker could be protected (Akhilesh, 2016).

2.8.2. Data Encryption

Data Encryption is the method of translating the plaintext into Encoded (non-readable) form and can only be accessed by designated persons / parties. Data protection is an integral part of an individual / organization, which can be accomplished by the use of different approaches. The encoded data is safe for a while but never believe it is truly protected (Kiram, 2019) There is a possibility of the thief breaching the data after the time has gone by. Fake files are sent in the same way as the encrypted data can be sent (Akhilesh, 2016).

There are several techniques available on the market to encode the data Encryption Key has a significant role in the data encryption and decryption process overall Figure 2.6.

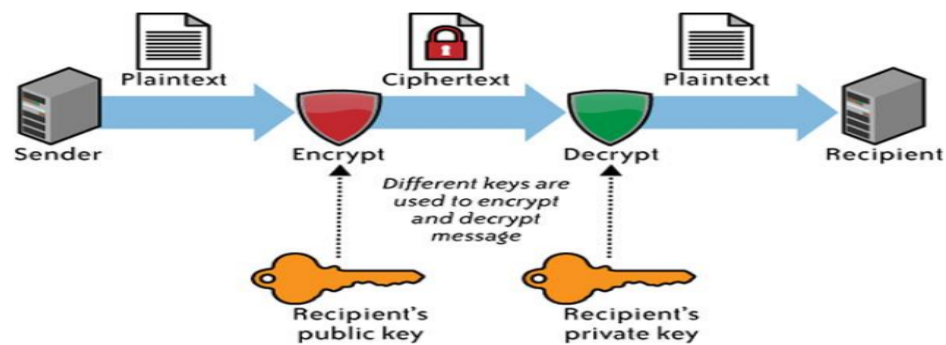


Figure 2.6: Encryption and Decryption Process (Brown, 2016)

2.8.3. RSA Encryption System

To understand how the RSA encryption system works, it is first necessary to understand how the secret system works. First of all, what's the secret system? Assume that a specific person wishes to send a message to someone else, but he does not want to know the information in the message except for the second person. (Garrett, 2011). The encryption system is a way to put a cover on the message, so that only the second person can remove the cover and read the message. The message with no hidden text is called plain text, and the hidden form of the message is called the text of the cipher. (Baker, 2019). The process that changes text to secret text is called encryption. Similarly, the process in which Cipher text turned into plain text is called the Decryption process (Iglesias, 2018).

2.8.4. DES Encryption Method

The DES is a symmetric-key cipher provided by the NIST. DES is a Feistel Cipher implementation. It uses a system with 16 rounds. The frame dimensions are 64bit (adhyay, 2020). While key tallness is 64bit, DES has an effective key length of 56bits, since the encoding algorithm does not use 8bit of the 64bits of the key (function as check bits only) (Baker, 2019). In the following Figure 2.7 the basic form of DES is shown. The DES consists of two permutations, one at the beginning and another at the end of the process. In between those two permutations, there are 16 rounds of encryption happens.

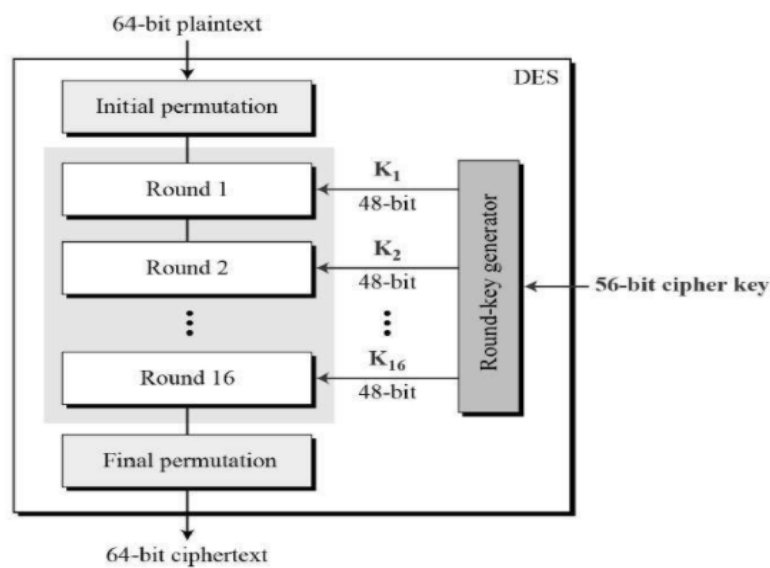


Figure2.7: DES Encryption Process (Sharma, 2020)

As DES is based on the Feistel, all you need to know is:

- J. Key schedule
- K. The beginning and final permutation - Any additional processing.
- L. Roundish function

Initial and Final Permutation

The beginning and final permutations are pure Permutation boxes (P-boxes), which are one another's reversed. They have little relation to encryption in DES. It displays the initial and final combinations as follows shown in Figure 2.8:

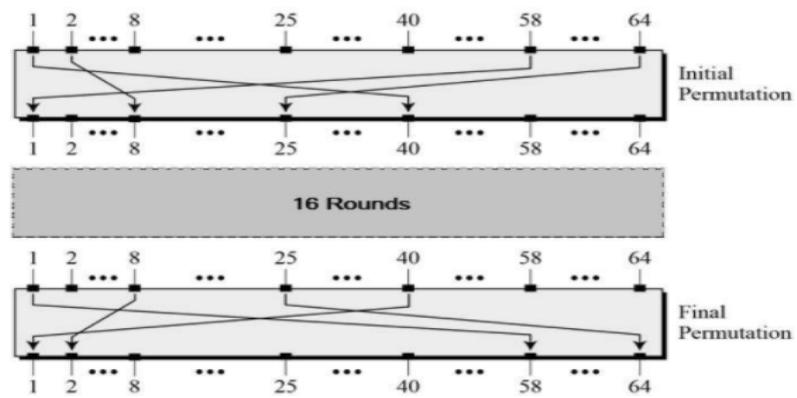


Figure 2.8: Initial and Final Permutation (Sharma, 2020)

Round Function

Role DES is the core of the cipher, f . The operation of DES round function is given in Figure 2.9. The DES function applies a 48bit key at the rightmost 32bits to output a 32bit output.

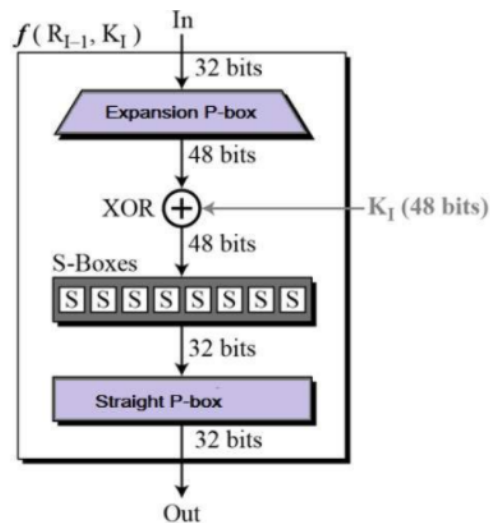


Figure 2.9: Round Function Process (Sharma, 2020)

DES Analysis

The DES satisfies both of the desired block cipher properties. These two characteristics make cipher very solid.

M. Avalanche effect: A small shift in plaintext makes the ciphertext very high.

N. Completeness: Each piece of ciphertext is based on several plaintext bits.

Cryptanalysis has found several vulnerabilities in DES over the last few years when the selected key is a weak key. Those keys are to be avoided.

DES proved to be an exceptionally well-designed block cipher. No major cryptanalytic attacks have occurred on DES other than exhaustive key quest.

2.7.5. 3DES Algorithm

DES is an encoding cipher that originally originated from the DES. It was popular in the late 1990s but has since fallen out of favor as more stable algorithms grow (Josh, 2019). While the Triple Data Encryption Algorithm (3DEA) is officially recognized, it is most popularly pointed out to as 3DES. This is because that algorithm uses the regular data encryption cipher (DES) 3 times to encoding its details (Callas, 2017).

2.8.6. AES Encryption Algorithm

The AES is the most prominent and commonly adopted symmetric encode algorithm likely to be encountered today. It's found six times faster than triple DES (Kirammat, 2019). It took a replacement for DES, as its main size was too small. This was potentially exposed to exhaustive key search attacks with rising computing capacity. Triple DES was planned to overcome this disadvantage but found slow (Callas, 2017).

The features of AES are as follows:

- O. Application implementable in Java, and C.
- P. Provide full features and design details.
- Q. Stronger and faster than Triple-DES.
- R. Symmetric key symmetric block cipher.
- S. 128bit data, (192/256) bit keys.

As for cyber security, AES is one of those neologisms you see popping up anywhere. Even though that has become the international encryption standard and is used to safeguard a significant amount of our communications (JOSH., 2019). The AES is a

fast and safe form of encode that keeps our data off prying eyes. We see this in messaging apps like Signal and WhatsApp, software like (VeraCrypt, WinZip), in a diverse range of hardware and other technologies which we use all the time (Mukherjee, 2017).

2.9. Why was AES developed?

The Data Encryption Standard (DES) was being used before AES was adopted nationally, and it is becoming increasingly vulnerable to brute force attacks. That is why NIST declared an urgent need for a better and more mature DES replacement. Consequently, AES was used as a stronger, newer and more sophisticated encryption algorithm. AES was initially designed to be used for the defense of sensitive information at government level. It is considered to be simple to implement both in hardware and software, and is therefore commonly used to defend against varying cyber-attack methods like brute force (Rouse, 2020). As of now, AES can be used for free through its non-commercial and commercial services for private and public organizations. Yet NGOs face certain limitations imposed by US export control (logsign, 2020).

2.10. Existing fog computing gaps

The present structure only gives the one assurance which is not much anchor and can be attacked by a developer without much of a stretch. The system gives no additional protection such as protection enquiries for greater security (Butun, 2018). Without much of a stretch the developer can get into the cloud and search for usable information. The current system does not clarify whether or not a customer is accepted. The current architecture provides protection through encoding but fails to anchor the cloud (Santosh, 2019).

2.10.1. Threats in Cloud

- **Sharing Technology:** This happens because the many pages share the details.
- **Data breaches:** This resulted in the loss of around 110 million people's personal data and credit card details, which was one of the thefts during data processing and storage.
- **DoS:** This happens when millions of users request the same service and hackers take advantage of the hacking advantage.

- Account or service traffic hijacking: If the login credentials are lost, the account can be compromised.
- Data loss: Data loss happens when the hard drive dies without any Server owner backup. It happens when the owner does not have the encryption key accessible.
- Cloud services abuse: By using the hacker of several cloud servers, the password will break in a very short time.
- Misleading insiders: This happens when someone close to us knows our account details.
- Insecure APIs: Application coding Interface monitors third parties and tests the user.

2.10.2. Challenges ahead

There are several open problems to be guided through making the mist a reality. Clearly identifying them is necessary so that future research works have these issues in record. The setting up of open difficulties for the nebula to end the reality is:

- Security: A related security issue that relates to current virtualized situations can be expected to affect the applications enabling mist gadgets. The proximity of safe bead applications sandboxes poses new interesting challenges: trust and privacy (Riahi, 2013). Before using different gadgets or shorter than expected mists in the system to run a certain product, separation and sandboxing components must be developed to guarantee bidirectional trust between the coordinating meetings (Abbas, 2019).
- Accountability/Monetization: Getting customers ready to share their save assets to have apps is crucial to inspire new action plans around the mist concept. There should be a fair arrangement of the driving powers. The motivating forces may be budgetary or something else (for example, boundless free rates of data) (Albareda, 2019).
- Discovery/Sync: Applications running on gadgets can require some concomitant unified point (e.g., create upstream reinforcement if our capability application contains too few companions) (Almeida, 2015).

- Standardization: Today no institutionalized systems are accessible so that any person from the system (edge point, terminal...) can announce their accessibility to have other developing components and submit their product to others. (Archana, 2017).
- Compute/Storage confinement: Current trends improve this fact with littler, more powerful vitality and all the bigger gadgets (e.g. one of the present mobile phones is bigger than various tops of the line work areas from 15 years before). Newer updates for non-purchaser devices are also permitted (Baker, 2019).

CHAPTER 3

PROPOSED DUAL KEY ENCRYPTION SYSTEM

3.1 Overview

The focal point of this thesis currently turns towards how Dual Key Encryption (DKE) can be utilized inside the cloud. The Dual Encryption Infrastructure (DEI), permits an element to confine admittance to information that has been shipped off them. Notwithstanding, the utilization of these three methods of activity to a Cloud setting will be influenced by: the proprietor of the DKE conspire is a help client or a specialist co-op Cloud Service Provider (CSP); and b) the style of access control required for example Ciphertext-Policy (CP). From these elements, three different situations arose: Situation I: Embedded inside a Service. A CSP consolidates a CP-DKE plot in DEI mode inside a current assistance. Offering support clients, the methods with which they can share information among one another inside the area of a specific help. Situation II: DKE-as-a-Service. Like Situation I with the special case that DKE is utilized out with the associations of a specific assistance area. Situation III: 'Dispersed Security'. The administration client conveys their own CP-DKE plot in MCP mode. DKE is utilized to alleviate access over information the client pushes to the cloud.

3.2 Operational Procedure of DKE System

This flow chart depicted in Figure 3.1 give information about various initial parameters and players of the proposed DKE system.

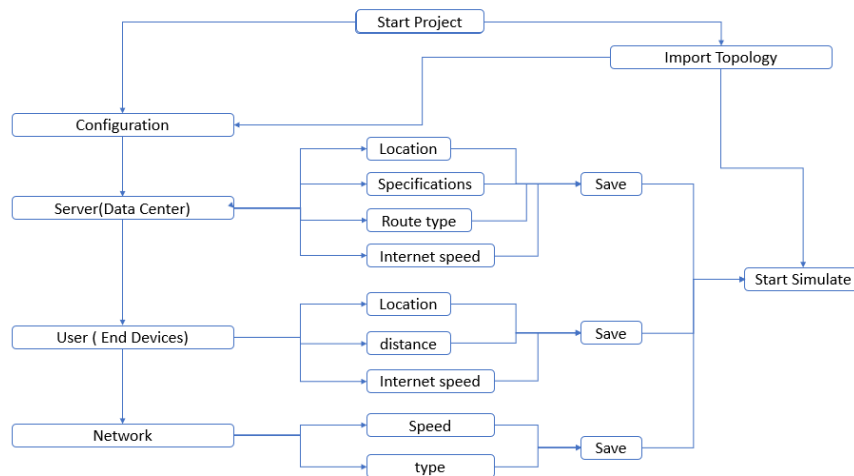


Figure 3.1: Flow Chart I

The operational procedure of phase II is given in Figure 3.2 as a flow chart.

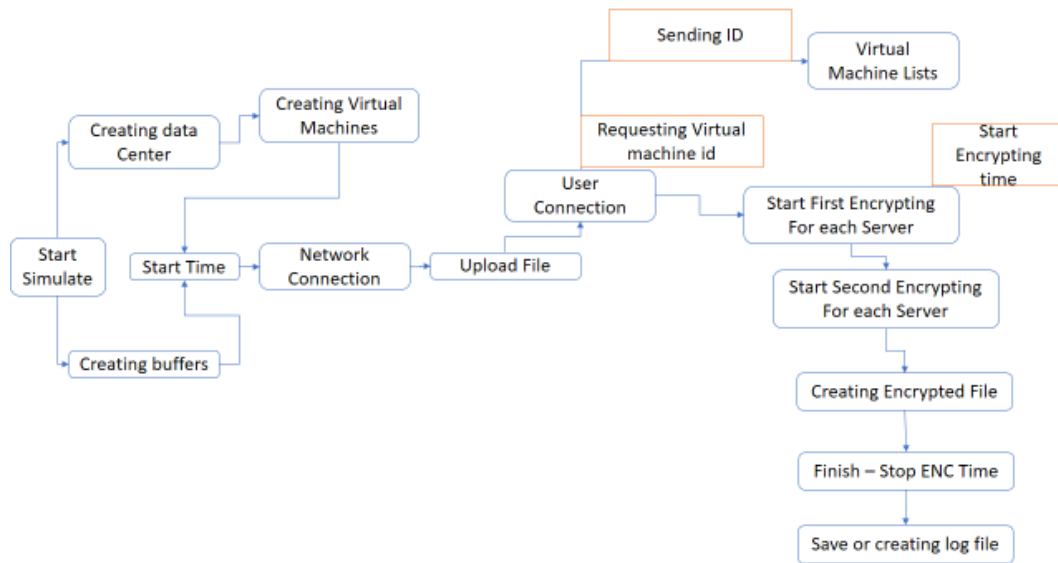


Figure 3.2: Flowchart II

Here in the above given flow chart the initial operation is start of the simulation. Once the simulation start, the datacenter and buffer creation will happen. Inside the datacenters the virtual machines are created which are responsible for the operation on data transferred from the client. Once the user ready to upload the data, the virtual machine id will be used as one key and sending Identification (ID) will be used as second encryption key. Then both keys will be used for dual encryption and the final encrypted file will be created. Then the log file about the operation will be created. The phase III of the operational procedure is given in Figure 3.3.

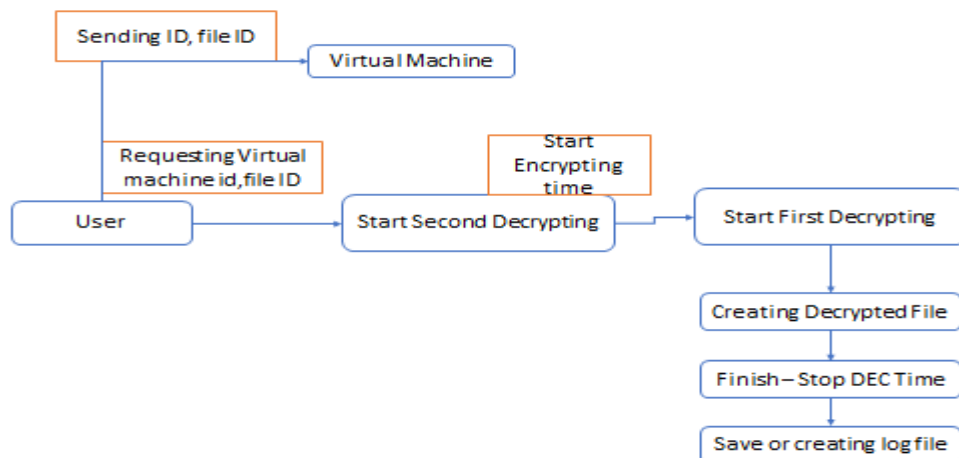


Figure 3.3: Flowchart III

This flowchart depicts the details of decryption process. Here the decryption process is the reverse of encryption process, where the sending id will be used to decrypt the file as first decryption. Afterwards, the virtual machine id will be used as key for the second decryption. After both the decryption finished, the final decrypted file will be stored at the receiver side along with log information for further reference.

3.3 Leveraging DEI Mode

The DEI method of activity inclines normally towards the as-a-Service worldview, and subsequently its utilization by a CSP. The CSP is the outsider contribution for the utilization of DKE to its administration clients. In view of this, DKE can be offered by the CSP in two unmistakable manners, either: a) installed inside an assistance; or b) provided as a help. In the two cases, the CSP is advancing the sharing of information among different clients of the administrations; sharing is a rule displayed by numerous Software as a Service (SaaS) administrations. Prompting the suggestion that these two situations, at any rate, give off an impression of being most appropriate for SaaS administrations. Besides, an entrance control style is required that permits the scrambling substance to state unequivocally for whom access will be conceded on a for each message premise. Ciphertext-Policy plans give such access control. Showing that the property universe is utilized to, in any event, depict clients of the administration.

Situation I: DKE within the service itself.

After enlisting with the administration, new clients are given an unscrambling key generated from a bunch of characteristics that portray the client. When pushing information to the administration, administration clients encrypt their information under access approaches based on their personal preference. Additionally, as the KA is inserted inside the administration, the CSP can likewise help in strategy organization. At the end of the day, the CSP can offer clients methods through which to determine strategy rules and hence develop access strategies. Due to inserting DKE inside the administration itself, the property universe can be reached out to allude to support explicit usefulness, for example, the CSP's capacity to peruse the client's information for focused advertisements. For instance, take an Online Social Network

(OSN). The characteristic universe can recognize administration clients through qualities, for example, distinguishing proof number, sexual orientation, school organizations, interests, area, D.O.B. and whatnot. Usefulness, for example, the perceivability of clients' messages can likewise be incorporated inside a similar characteristic universe. Following on from demonstrating the perceivability of a client's message is the idea that the CSP can likewise be referred to inside the strategy manage and treated as a client in their own right. In spite of the fact that this doesn't cling carefully to the DEI's business as usual. It in any case permits the encoding client to expressly pick in the CSP when the client builds strategy rules. In addition, such a property can be joined with a date trait to furnish the CSP with a restricted open door to get to scrambled information. This is an amazing build, the client has unequivocally expressed inside an entrance strategy that the CSP can get to the scrambled information, and all the more critically for how long. An issue for this situation is that administration clients need to confide in the CSP. As the CSP is the KA, the CSP could undoubtedly develop their own discretionary unscrambling keys, and utilize these keys to decrypt administration clients' messages. Administration clients actually need to confide in the CSP to not be malevolent in this regard. Moreover, utilizing DKE as depicted necessitates that each CSP give their own plan. With each help that the client communicates with an alternate DKE conspire should be considered. That is, independent key administration offices and calculations should be overseen by the client; one for every help. Given what is known over key stockpiling issues this could be an issue. Albeit, how the DKE is really sent will influence this fairly.

Situation II: DKE-as-a-Service

Giving a CP-DKE conspire as a help in itself permits administration clients to interface with various administrations, offered by various CSPs, and utilize a typical arrangement. This decreases the overhead concerning key administration, and furthermore the diverse encryption and decryption calculations that a help client should know about. Giving 'DKE-as-a-Service' makes this situation an illustration of Security-as-a-Service all the more explicitly it tends to be tied straightforwardly into Identity-as-a-Service administrations. This infers that not exclusively can this situation be utilized to secure SaaS level administrations yet additionally at the Platform as a Service (PaaS) level as a help. Like Situation I, clients have decryption keys contains a bunch of properties used to portray the client themselves. While interfacing with

various administrations the client can push scrambled information realizing that lone other people who can fulfill the entrance strategy can get to the hidden plain-text information. Strategy organization administration can likewise be offered by the DKE administration. Likewise, with Situation I CSPs should be treated as clients in their own right. Consequently, while developing approach administers the scrambling client can expressly pick in the CSPs as somebody with whom they wish to impart their information to. Also, CSPs 'administration clients would themselves be able to incorporate this prior DKE administration into their own assistance contributions. In contrast to Situation I, nonetheless, administration clients can treat the CSP like some other client and not need to stress over the CSP's noxiousness. A CSP in this setting can't build discretionary decryption keys. Clients can utilize help despite the fact that they may have no trust in the CSP offering that administration. At the point when the client has trust in the CSP to get to the client's information, at that point the CSP can be selected. Nonetheless, the administration free-thought presents a few intriguing issues. For one, the trait universe is characterized by an outsider who may not have information to help explicit information, offering clients credits that portray information out with the setting of an assistance. This may influence the expressiveness of strategy rules as characterized by scrambling clients. Offering a DKE administration decreases the overhead as far as key administration and information on calculations, be that as it may, this decrease in overhead comes at some expense. Situation I is exceptionally compartmentalized, CSPs are simply ready to get to the information that has been pushed to their administration. With the arrangement portrayed in this segment the DKE specialist organization, if noxious, can develop unscrambling keys to get to information that a client has pushed to various administrations. A repetitive component with Situations I and II is that the KA is an outsider unmistakable from the administration client. While this rearranges the association of the administration client it requires the administration client to be dependent upon the KA to cling devoted to its transmit. Such apprehensions can be tended to with the MCP method of activity that eliminates the requirement for an outsider KA.

Situation III: Information base Submission

The reasoning behind the DDI mode is controlling access over submitted information. At the point when an assistance client utilizes this method of activity it suggests that

the administration client, will be controlling admittance to information that has been submitted to them: This has no conspicuous advantages or employments. It is better for the client to use the MCP method of activity. Then again with a CSP as the KA, an intriguing use case arises. With DDI mode the CSP is utilizing DKE to encourage specific access over information, submitted explicitly for use by the CSP, under the stipulation that said information might be available by the CSP's own representatives and partners. That is information is presented by administration clients and the CSP permits approved substances admittance to choose subsets of all information that has been submitted. This basically depicts the activity of an information base. Information bases can be defined of as far as: a) who is presenting the information; b) who is getting to the information; and c) the information itself. With DDI mode, the information base is put away with the CSP. Information is submitted for the CSP by administration clients and is being gotten to by the CSP's workers and different members. Clearly, clients' entrance over the information inside the information base should be proscriptive and client's capacity to get to singular records should be shortened. As such information is encrypted under a bunch of traits, and unscrambling keys from a predicate. Another and more natural explanation behind utilizing Key-Policy plans originates from the unscrambling keys. Decryption keys in this setting speak to basic inquiry inquiries. While submitting information, administration clients encrypt their information under a bunch of traits. The CSP can then inside decide on a for every worker premise what the representative can access from this information base. Additionally, with mathematical properties it is attainable to restrict the period during which the representative can utilize their question. The relationship of an information infers the sending of DKE as a component of either a SaaS, or PaaS administration. For instance, a course accommodation administration could utilize DDI mode to control accommodation of understudy's accommodation. During accommodation, understudies could encrypt their course under a bunch of characteristics that describe1: a) the understudy's registration number; b) the course code; c) the task number; and d) accommodation time. Instructors and encouraging colleagues would then be able to be appointed decryption keys relating to the course(s) they are related with. Comment. This utilization of DKE could likewise be reached out to circumstances in which clients of Infrastructure as a Service (IaaS) administrations require their design records to be put away inside the cloud.

3.4 Leveraging MCP Mode

Inside a cloud setting two situations arise when: a) an assistance client is the encryption substance; and when b) a CSP is the scrambling element. The primary situation gives a substitute answer for those introduced in Section 11.2. The subsequent situation, then again gives a setting wherein a CSP can control admittance to their own substance in a way like that seen with the DDI situation. Albeit the two situations require a proscriptive type of access control, the decision of fundamental DKE plan will contrast. With a help client Ciphertext-Policy plans are more qualified; access should be chosen per message. With the CSP as the scrambling element the setting is more data driven and as quite a Key-Policy plan will get the job done. CSPs can use MCP mode to confine admittance to the substance that they produce in much similar way as was seen. The distinction lies in the elaborate entertainers and birthplace of the data. Representatives of the CSP along with approved subsidiaries are those allowed to scramble data. Unscrambling of data is performed by administration clients. When joining to a help, clients are doled out an unscrambling key from some approach that depicts the data they are permitted to get to. Such arrangement rules can be utilized to demonstrate administration related data, for example, membership level and timeframes. This gives the CSP a way to uphold diverse membership models for example freemium. With MCP mode the demonstration of limiting access can likewise stretch out to any substance streams that the CSP delivers and broadcasts. This suggests that this situation can be utilized at both the PaaS and SaaS administration levels. The hidden issue with these arrangements is the dependence upon an outsider to go about as the KA. When sent by a help client the MCP mode gives an improvement regard to confiding in an outsider. With the MCP mode there is no requirement for a confided outsider over key administration, clients give their own DKE conspire. When joining a help or when beginning to associate with another 'companion', the scrambling client for example the Key Authority, will make and allot to the element an unscrambling key. This decryption key will be gotten from a bunch of qualities that portrays how the unscrambling client identifies with the encoding client. Through utilization of a Ciphertext-Policy plan, the encryption client can scramble their data locally under a strategy rule, and afterward push the ciphertext into the cloud. Characteristics can be utilized to show, for instance: a) sort of relationship with another substance for example a partner, a companion, a dear companion, a truly dear companion, family, or CSP; b) perceivability of message for example is the

message to be private or public; and c) an identifier for every area that the client has collaborated with for example on an OSN. This utilization of the MCP method of activity presents a decentralized answer for the issues related with Situations I and II. Inferring the above situations at the PaaS and SaaS administration levels, with every 'client' giving their own encryption plot. From these two fascinating perceptions can be made that can influence the administration and organization of this mode. To start with, the scrambling client is liable for the transmit of the Key Authority. The client is liable for strategy rule organization, answerable for the right task of unscrambling keys, liable for key administration and so forth. Also, the security ensures unidirectional correspondence. Every client uses DKE to ensure their own data as it were. For n clients conveying it requires every client to consider the unscrambling keys, and related decryption calculations.

3.5 What can DKE be used to Protect?

This part introduced two danger models for data as far as its life cycle, from which data is being portrayed as one or the other being: a) in the cloud; or b) out with the cloud. When pushing and recovering data from the cloud existing instruments (for example HTTPS and SSH) can be utilized to secure the data in light. Be that as it may, the issues related with data in the cloud stem not from when it is in light yet rather from whenever it includes become occupant inside the cloud. With DKE, confirmations over the classification and availability can be made. DKE can assist in forestalling the undesirable introduction, undesirable spillage and other undesirable breaks of privacy of cloud occupant data. Different ensures, for example, honesty, non-disavowal and validity need extra security instruments. The idea of the data that should be secured will contrast with the specific assistance layer being tended to: Infrastructure as a Service (IaaS) The IaaS administration level is worried about the organization of virtual machines. With this administration level the data that is of concern are the settings and orders utilized while arranging, conveying, and dealing with the virtual machine occasions. This is data that will be submitted to the CSP. While orders are to be summoned, settings are to be put away for use. Stage as a Service (PaaS) Services at the PaaS administration level are described through the APIs uncovered by the CSP through some stage tooling. Despite the fact that the scope of this tooling is immense it can in any case be viewed as an uncovered Application Programming Interface (API) advancing some usefulness. The APIs themselves can

be described as playing out some data handling or giving admittance to some data store. Software as a Service (SaaS) - The most noteworthy help level, SaaS, manages administrations that offer more perplexing collaboration and usefulness. The reach, shape and type of the data being driven into the cloud will differ extraordinarily from administration to support. For instance, the data pushed by the client can incorporate however isn't really restricted to: messages posted on a distant informal communication website, records in an online office suite, or photos presented on a photograph sharing webpage. A predominant aspect of numerous SaaS administrations is the joint effort, or sharing of this data with different clients. A connected conversation to what should be ensured is when (and where) the secure be applied. That is the place where encryption and unscrambling of data ought to happen.

3.6 Effect of Cryptographic Operation Placement

Chapter 3 introduced danger models that depict when data will be defenseless during its lifecycle. While sending DKE in the cloud a significant inquiry poses concerns where in the data lifecycle the encryption and decryption tasks should be sent. Such a choice will influence the start to finish security of encryption substances' data. When during the data lifecycle can the secrecy of data be guaranteed and when would it be able to be addressed? In addition, this issue raises worries over the believed registering base set up by the CSP.

3.6.1 Key Authority and Service are not linked

With Scenarios II and III the KA isn't attached to a specific help. The sending of encryption and decryption activities is unimportant. Data will be encrypted by the administration client preceding its inclusion into the cloud Stage 1. At the point when the unscrambling element is a CSP the data can be decrypted whenever it has entered the cloud Stage 2. At last, administration clients can unscramble data whenever it has been locally Stage 4. Start to finish classification of the message has been guaranteed.

3.6.2 Key Authority and Service are linked

With Scenarios I, II and III the arrangement isn't as clear. With these situations the CSP is the KA and is the supplier of the administration that administration clients are communicating with. For the most part, talking two alternatives exist over the situation of the activities: Option 1: Cryptographic tasks are performed by administration clients

themselves. Data is encrypted and decrypted at Stages 1 and 4 of the data life-cycle. Choice 2: Cryptographic activities are performed by the 'CSP' for the clients. Data is scrambled and decrypted at Stages 2 and 3 of the data life-cycle. Choice 1 places the encoding client in charge of the encryption cycle. Their data has been scrambled at the principal stage inside its life-cycle before its push to the cloud. This manages the cost of the client information on the exact creation of the encryption key used to scramble the data, and furthermore of the encryption cycle. Additionally, the decryption client just gets to the cipher text and unscrambles the cipher text whenever it has been gotten locally for example it is outside the space of the administration. By playing out the cryptographic activities locally, in the area of the client, the degree at which dangers assaulting the classification can happen has been decreased. Start to finish classification has been guaranteed: The data goes into the cloud encrypted; it leaves the cloud scrambled. Choice 2 then again, can't offer similar assurances. Data stored into the cloud by the administration client has now been encrypted inside the area of the CSP. The element needs additional confirmations with respect to the key used to encrypt their data. With unscrambling, the administration client's decryption key should now additionally live in the administration for it to be utilized. Unapproved utilization of this critical should be guaranteed also, particularly from malevolent insiders. Immovable certifications over start to finish secrecy can't be guaranteed: The data sent by the client enters the cloud decrypted and will leave the cloud decrypted. With Option 2 dangers to data, and furthermore to the unscrambling key can now likewise happen inside the space of the CSP. In spite of the fact that this alternative may have all the earmarks of being a miserable one it does have its points of interest. This lessens the apparent multifaceted nature as well as calms the administration client over the obligation regarding keeping up and putting away cryptographic keys. This might be a bit of leeway inside resources needed conditions, and versatile conditions in which the administration client doesn't have nonstop admittance to their unscrambling keys.

3.7 Fulfilling the remit of the Key Authority

Key to the utilization of DKE as a component of a crypto-framework is the KA. Section 9 presented the KA along with a conversation over the transmit that the KA should hold fast to and how parts of this dispatch could be accomplished. This incorporates obligation regarding dealing with the characteristic universe,

allotting/giving cryptographic keys, and for KP plans strategy rule organization. Area 10.5 likewise examined how credits can be sourced, among other arrangement issues. In any case, how a CSP or a help client can satisfy this dispatch will contrast. Cloud Service Provider Naturally, CSPs approach a complete existing foundation. This framework can be utilized by the CSP when the provider play out their obligation as a KA. While including/utilizing DKE as a component of an assistance, the CSP can utilize existing data from the administration to decide the properties utilized inside U. A current personality the board framework can likewise be utilized to check any cases a client has over properties. Besides, existing strategy rule particulars can be utilized as a base while building decryption approaches for example Key-Policy plans, or as a guide when clients need to develop encryption strategies for example Ciphertext-Policy plans. At last, with Ciphertext-Policy conspires the CSP can likewise offer an interface, and administration, through which clients can digest over the complexities of strategy rule particular and the executives. Administration User When a help client is the Key Authority, they won't have similar degree of assets and existing foundation when contrasted with a CSP. In any case, the size of activities for a help client is immensely extraordinary when contrasted with that of a CSP. Here the administration client works in a more modest, more customized climate. They manage substances straightforwardly for example one on one, the administration client can play out the transmit 'physically'. That is, the administration client can check claims over traits while developing decryption keys themselves. Not at all like a CSP who will have a fixed location, notwithstanding, is the administration client innately versatile. The administration client will access administrations from various areas from various machines. How precisely is the administration client going to satisfy this transmit, on the off chance that they are persistently progressing?

CHAPTER 4

IMPLEMENTATION AND EVALUATION

This chapter presents the results of performed test cases to evaluate the performance of the proposed methodology implementation in the simulation environment. First the information about the simulation environment is described. Further the test cases and its corresponding evaluation is discussed.

4.1 Simulation Environment

The implementation in this experimental research is done based on following simulators and programming language.

- JAVA
- CloudSim Simulator
- CloudAnalyst

4.1.1 JAVA

Sun microsystems developed JAVA in the year 1991, which was later purchased by the Oracle Corporation. Java is an Object-Oriented language. Main advantage of JAVA is it is a platform (operating system) independent language. It means, the java programs developed over any operating systems like windows, linux, etc., can be easily ported to other operating systems without worrying about operating system integration. For that, what happens is, on any operating system the compiler (javac) changes over source code (.java record) to the byte code (class document) and this byte code can be running on any other operating system. The only requirement is the destination operating system must be updated with JAVA software. This is happening because JAVA is not directly connected with the operating system and executes the program. Instead, it has a special feature called as JVM (Java virtual Machine) which acts as a broker between the JAVA program and the destination operating system. As referenced above, JVM executes the bytecode delivered by the compiler. This byte code can run on operating systems, for example Windows, Linux, Mac OS and so on. That is the reason, java is a platform independent language.

4.1.2 CloudSim Simulator

I used this platform because have a lot of sources and libraries for connecting with my source codes and this is faster than the other platform to running my codes in the simulator also it is working in most of operating system like windows, Linux and other because the codes written by java programming language also my codes is Java Language also getting the best report by cloud analysis to get best report between my codes with these to platforms that is why I'm used CloudSim.

Cloud service providers offer flexible, on-demand, and assessed system, stages and programming services. In the public cloud, occupants have order over the OS, accumulating and sending applications. Resources are provisioned in different geographic locales. In the public cloud plan model, the introduction of an application passed on in various locales includes stress for affiliations. Proof of thoughts in the public cloud environment give an unrivaled plan, yet cost a ton to as far as possible structure and resource usage even in the remuneration per-use model. CloudSim Simulation Toolkit is the more summed up and inconceivable test structure for testing Cloud enrolling related theory. CloudSim is an extensible reenactment structure made by a social event of scientists at Cloud and Distributed Systems (CLOUDS) Laboratory, University of Melbourne. This tool stash concedes consistent outlining, redirection, and Experimentation identified with cloud-based foundations and application services and its unmistakable passed on varieties are flowed on Cloudsim's GitHub project page. CloudSim surrenders an added and extensible entertainment framework that engages predictable exhibiting and reenactment of utilization execution. By using CloudSim, specialists can focus on express systems arrangement that they need to investigate, without getting stressed over nuances related to services and cloud-based establishments. CloudSim, which is - a device compartment for the exhibiting and multiplication of Cloud computing conditions acts the legend. It gives structure and lead showing of the Cloud computing portions. Reenactment of cloud conditions and applications to evaluate execution can give supportive encounters to research such interesting, enormously appropriated, and adaptable conditions. Using these parts, it isn't hard to evaluate new procedures managing the use of clouds, while contemplating systems, booking computations, load changing methodologies, etc. This reenactment tool stash allows the researchers similarly as cloud architects to test the presentation of the potential cloud application for execution testing in a controlled

and easy to game plan environment. What's more it allows fine-tuning the overall assistance execution even before it is sent in the creation environment.

4.1.3 CloudSim Architecture

Figure 4.1 gives the architecture of CloudSim simulator. This simulator consists of three main layers where the lowest layer is a cloudsims core simulation engine. This layer is the backbone of the simulator and provides all the necessary library files for the successful simulation of the environment given by the user. The above layer is the cloudsims layer which gives the main modules for the simulation environment which is used by the user. The modules are network details like topology, message transmission capacity etc. The cloud resource module gives features like events, sensor for monitoring cloud running, coordinator for coordinating between different services, data center for hosting the cloud services and executing the user submitted tasks. The cloud services module provides the user with features like, virtual machine allocation algorithms, processor allocation techniques, RAM allocation methodologies, storage like SSD or HDD or magnetic tape provisioning techniques, bandwidth allocation like maximum download, upload capacity, number of requests per second etc. The VM services layer provides facilities for users to simulate the cloudlet execution which means, execution of jobs in the cloud environment which are submitted by the user. Along with it this module also provides the algorithms and methods for virtual machines management. The CloudSim Core engine proposition assistance for showing and entertainment of virtualized Cloud-based worker ranch conditions including coating and planning of events, creation of cloud system substances (such as services, worker ranch, have, VMs, sellers,etc) correspondence between portions and the heads of the generation clock. The CloudSim layer gives submitted organization interfaces to VMs, storing, memory and information transmission. Also, it manages the other chief issues, for instance, provisioning of hosts to VMs, regulating software execution, and checking dynamic system state (for example Association geology, sensors, storing credits, et cetera. The User Code layer is a custom layer where the customer creates their own code to reevaluate the ascribes of the strengthening climate as per their new assessment disclosures. All the above said layers, modules and sub modules are developed as java programs and it is completely open source for modification by any user, hence it is highly helpful for the researchers to develop their own algorithm on various cloud issues and merge them with the cloudsims simulator, then they can create

their own simulation environment and run it for testing the efficiency of their algorithm or techniques.

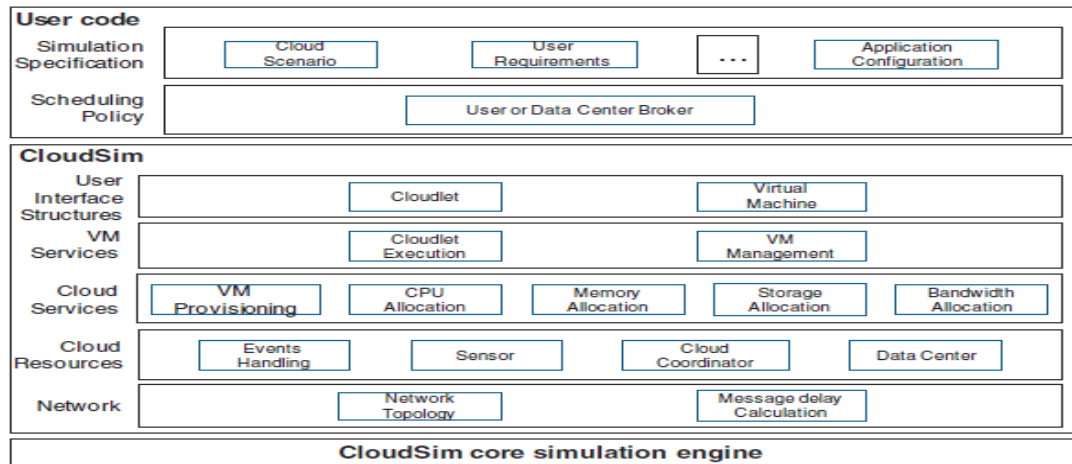


Figure 4.1: CloudSim Architecture

4.1.3 CloudAnalyst Simulator

The essential aim of CloudAnalyst is to confine the reenactment experimentation practice from a developing exercise, so a modeler can concentrate in on the entertainment complexities without putting an unnecessary measure of energy in the subtleties of developing using a generation tool stash. The CloudAnalyst moreover engages a modeler to reliably execute generations and to lead a movement of reenactment attempts various things with slight limits assortments.

Multiple times running of Simulations

CloudAnalyst grants modelers to save reenactment tests input limits and results as XML records so the examinations can be reiterated. The fundamental CloudSim reenactment structure ensures that reiterated tests yield genuine results.

Graphical Output

CloudAnalyst is prepared for making interface yield of the multiplication achieves the kind of tables and layouts, which is appealing to feasibly summarize the colossal proportion of experiences that is assembled during the entertainment. An especially reasonable presentation assists in distinctive the critical instances of the yield limits and assists in relationships between connected limits. In the ebb and flow transformation of CloudAnalyst, the going with verifiable estimations are made as yield of the reenactment: Response period of the imitated application; as a rule, typical,

least and most noteworthy response period of all customer requests reproduced; all around sales getting ready time for the entire multiplication; ordinary, least and most outrageous sales taking care of time by each worker ranch.

Utilization of solidified innovation and simplicity of augmentation

CloudAnalyst depends on a measured plan that can be effectively broadened. It is created utilizing the accompanying advancements: Java (the test system is created 100% on Java stage, utilizing Java SE 1.6); Java Swing (the GUI part is fabricated utilizing Swing segments); (CloudSim highlights for demonstrating server farms is utilized in CloudAnalyst); and SimJava (Macnab, 2008) (a few highlights of this apparatus are utilized straightforwardly in CloudAnalyst).

4.2 Results

Proposed technique is implemented with various file sizes ranging from 100KB to 50MB and try to and out perform comparison among an existing technique and the proposed technique. The reason of comparing between DKE and 3DES algorithm instead of other algorithm is because of the 3DES is three times encrypting and i decided to find the algorithm encrypting data more than one time because DKE Encrypting two times but in the comparison result DKE is faster than the 3DES and more Secure because 3DES encrypting data 3 Times but in same Technique and DKE Encrypting two Times but in different technique. That means the DKE is more secure than the 3DES and other algorithms also. When the attacker want to decrypt the DKE he will get the encrypted file also when he tried correct technique and he think it is wrong algorithm and cannot find the original data but in other algorithms for example in 3DES if he found the algorithm to decrypt the data he will get the original data easier. Figure 4.2 depicts the simulation parameters of the test case.

Main Configuration
Data Center Configuration
Advanced

Simulation Duration:
60.0
min

User bases:

Name	Region	Requests per User per Hr	Data Size per Request (bytes)	Peak Hours Start (GMT)	Peak Hours End (GMT)	Avg Peak Users	Avg Off-Peak Users
User1	0	60	100	3	9	1000	100
User2	1	60	100	3	9	1000	100
User3	1	60	100	3	9	1000	100
User4	2	60	100	3	9	1000	100
User5	3	60	100	3	9	1000	100

Add NewRemove

Application Deployment Configuration:
Service Broker Policy:
Closest Data Center

Data Center	# VMs	Image Size	Memory	BW
DC3	3	10000	16384	1000
DC4	4	10000	2048	1000
DC5	5	10000	1024	1000
DC6	6	10000	16384	1000
DC7	2	10000	512	1000

Add NewRemove

Figure 4.2: Main Simulation Parameters

The simulation parameters consist of five users with seven data centers. All users are with same parameter values and data centers are with different memory sizes for handling different file sizes from users. Figure 4.3 gives the information about the regions in the world simulated by CloudAnalyst simulator.

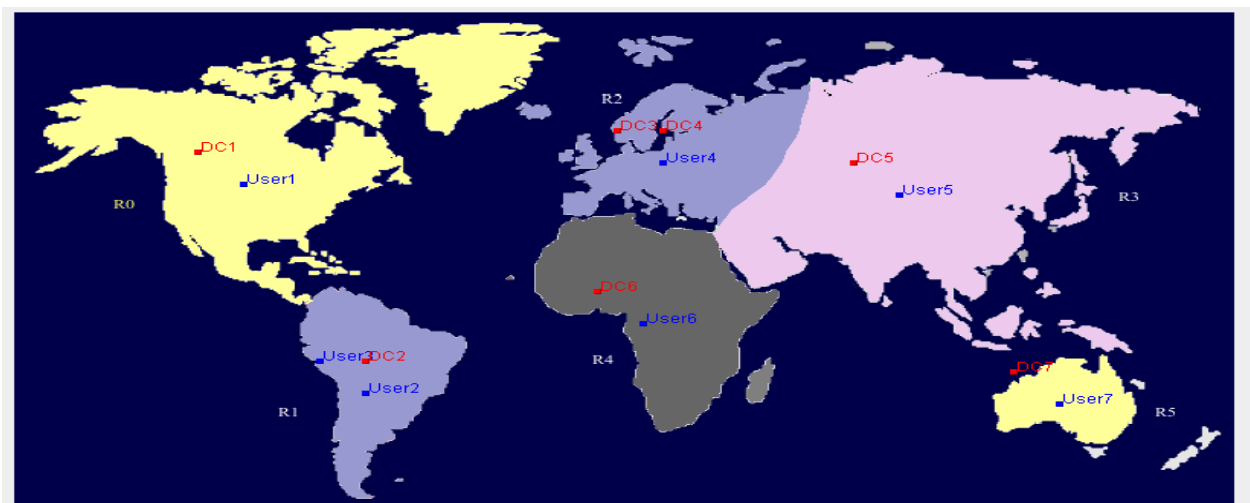


Figure 4.3: Datacenter Locations under Simulator

All the continents of the world are having one datacenter each except for Europe which has two centers. The world is divided into five regions. Figure 4.4 provides the information about the datacenter configuration. Each datacenter is with the Linux operating system and runs in the Xen virtual machine environment. All the other important parameters are given in the Figure 4.4.

The screenshot shows a web-based configuration interface for data centers. It has three tabs: 'Main Configuration', 'Data Center Configuration' (which is active), and 'Advanced'. Under the 'Data Center Configuration' tab, there is a section titled 'Data Centers:' containing a table with columns: Name, Region, Arch, OS, VMM, Cost per VM \$/Hr, Memory Cost \$/s, Storage Cost \$/s, Data Transfer Cost \$/Gb, and Physical HW Units. The table lists five data centers: DC3, DC4, DC5, DC6, and DC7. DC6 is highlighted. To the right of the table are 'Add New' and 'Remove' buttons. Below the table is a section titled 'Physical Hardware Details of Data Center : DC6' containing a table with columns: Id, Memory (Mb), Storage (Mb), Available BW, Number of Processors, Processor Speed, and VM Policy. This table lists two hardware configurations for DC6. To the right of this table are 'Add New', 'Copy', and 'Remove' buttons.

Name	Region	Arch	OS	VMM	Cost per VM \$/Hr	Memory Cost \$/s	Storage Cost \$/s	Data Transfer Cost \$/Gb	Physical HW Units
DC3		2 x86	Linux	Xen	0.1	0.05	0.1	0.1	1
DC4		2 x86	Linux	Xen	0.1	0.05	0.1	0.1	1
DC5		3 x86	Linux	Xen	0.1	0.05	0.1	0.1	1
DC6		4 x86	Linux	Xen	0.1	0.05	0.1	0.1	2
DC7		5 x86	Linux	Xen	0.1	0.05	0.1	0.1	1

Id	Memory (Mb)	Storage (Mb)	Available BW	Number of Processors	Processor Speed	VM Policy
0	1638400	100000000	1000000	4	10000	TIME_SHARED
1	1638400	100000000	1000000	4	10000	TIME_SHARED

Figure 4.4: Datacenter Configuration

Figure 4.5 depicts the parameters and their values of internet characteristics under CloudAnalyst simulator. The characteristics are grouped as delay matrix and bandwidth matrix.

The screenshot shows a web-based configuration interface titled 'Configure Internet Characteristics'. It includes a subtitle: 'Use this screen to configure the Internet characteristics.' There are two main sections: 'Delay Matrix' and 'Bandwidth Matrix'. The 'Delay Matrix' section has a subtitle: 'The transmission delay between regions. Units in milliseconds' and contains a table with transmission delay values between six regions (0-5). The 'Bandwidth Matrix' section has a subtitle: 'The available bandwidth between regions for the simulated application. Units in Mbps' and contains a table with bandwidth values between six regions (0-5).

Configure Internet Characteristics

Use this screen to configure the Internet characteristics.

Delay Matrix

The transmission delay between regions. Units in milliseconds

Region\Region	0	1	2	3	4	5
0	25	100	150	250	250	100
1	100	25	250	500	350	200
2	150	250	25	150	150	200
3	250	500	150	25	500	500
4	250	350	150	500	25	500
5	100	200	200	500	500	25

Bandwidth Matrix

The available bandwidth between regions for the simulated application. Units in Mbps

Region\Region	0	1	2	3	4	5
0	2,000	1,000	1,000	1,000	1,000	1,000
1	1,000	800	1,000	1,000	1,000	1,000
2	1,000	1,000	2,500	1,000	1,000	1,000
3	1,000	1,000	1,000	1,500	1,000	1,000
4	1,000	1,000	1,000	1,000	500	1,000
5	1,000	1,000	1,000	1,000	1,000	2,000

Figure 4.5: Internet Characteristics

Figure 4.6 gives a sample file information which is used by the user for uploading as user data. This file is encrypted at the datacenter for security. If the file is requested by the user from the cloud, it will be decrypted and transferred to the user. The whole process is executed many times and the sample of process is given in the following Figures.

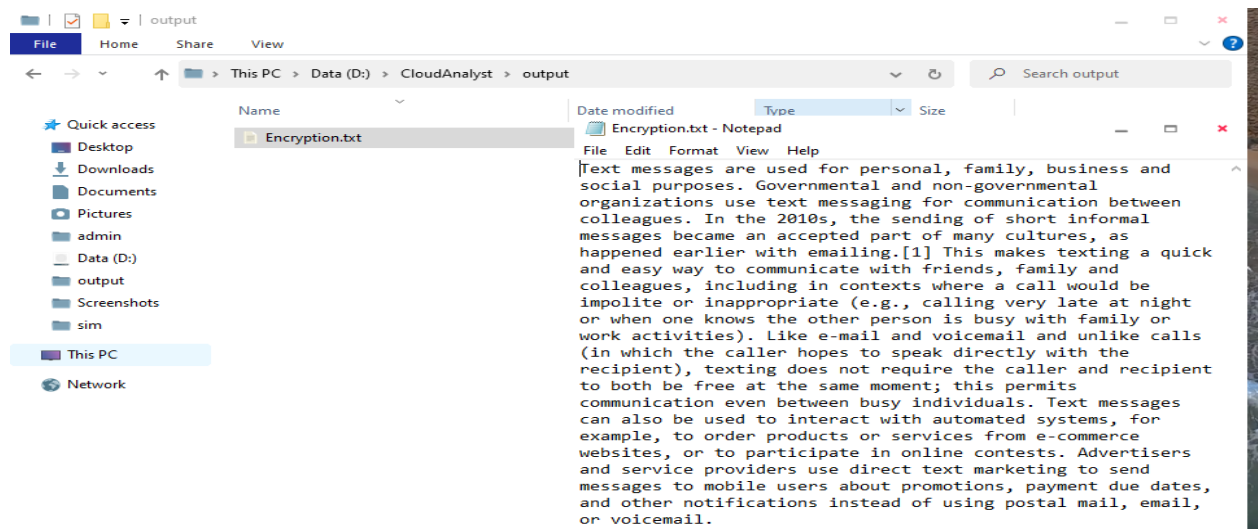


Figure 4.6: Sample File given by the user

Figure 4.7 gives the overall response time summary for all the user interaction happened with different datacenters placed in different regions in the CloudAnalyst simulator.

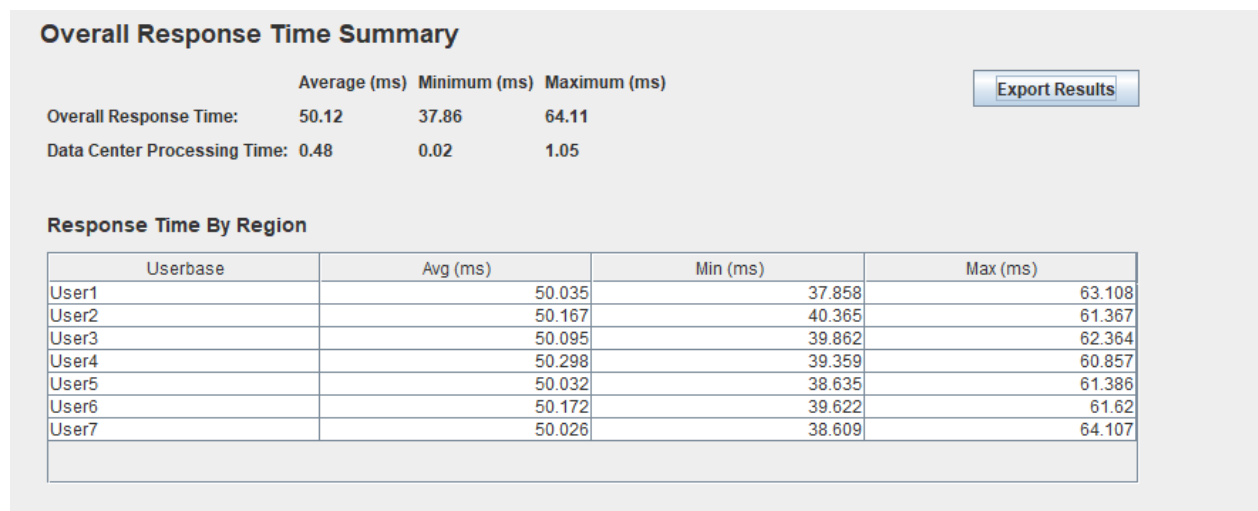


Figure 4.7: Response Time Summary

Figure 4.8 gives the information about hourly average response time for each user under the CloudAnalyst simulator.

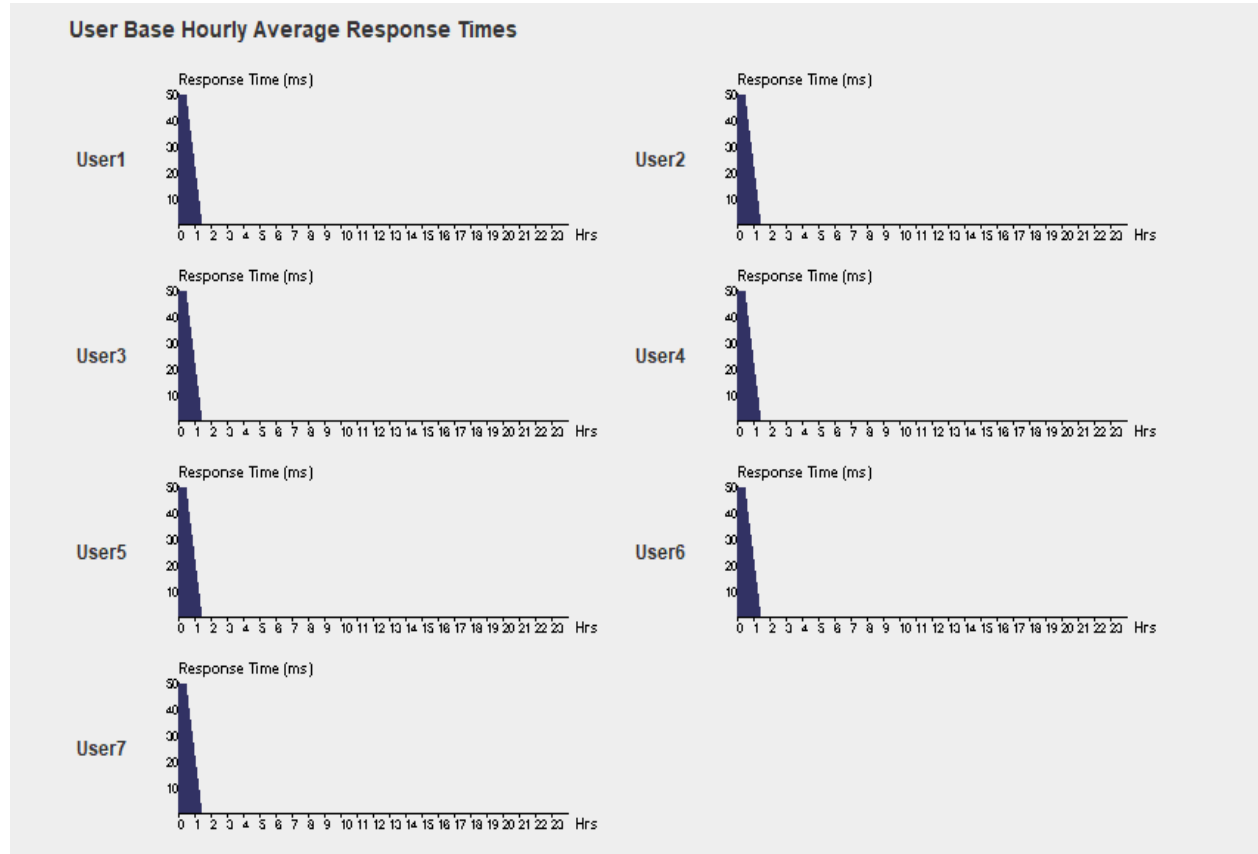


Figure 4.8: Average Response Time

Figure 4.9 gives information about the datacenter request servicing times for each datacenter i.e. seven datacenters under CloudAnalyst simulator.

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.5	0.035	1.054
DC2	0.481	0.02	0.963
DC3	0.453	0.024	0.854
DC4	0.461	0.044	0.854
DC5	0.485	0.019	0.882
DC6	0.481	0.045	0.872
DC7	0.467	0.025	0.954

Figure 4.9: Datacenter Request Servicing Time

Figure 4.10 gives information about datacenter loading under each of the seven datacenters in the CloudAnalyst simulator.

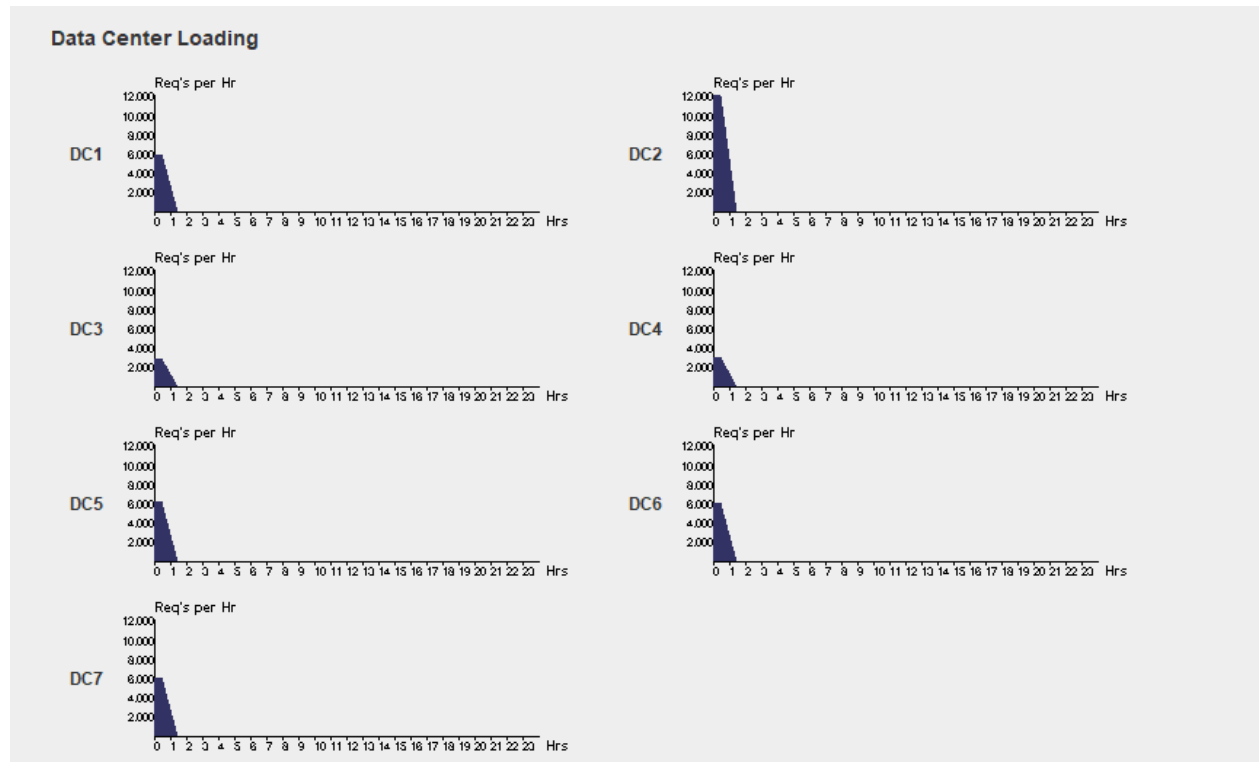


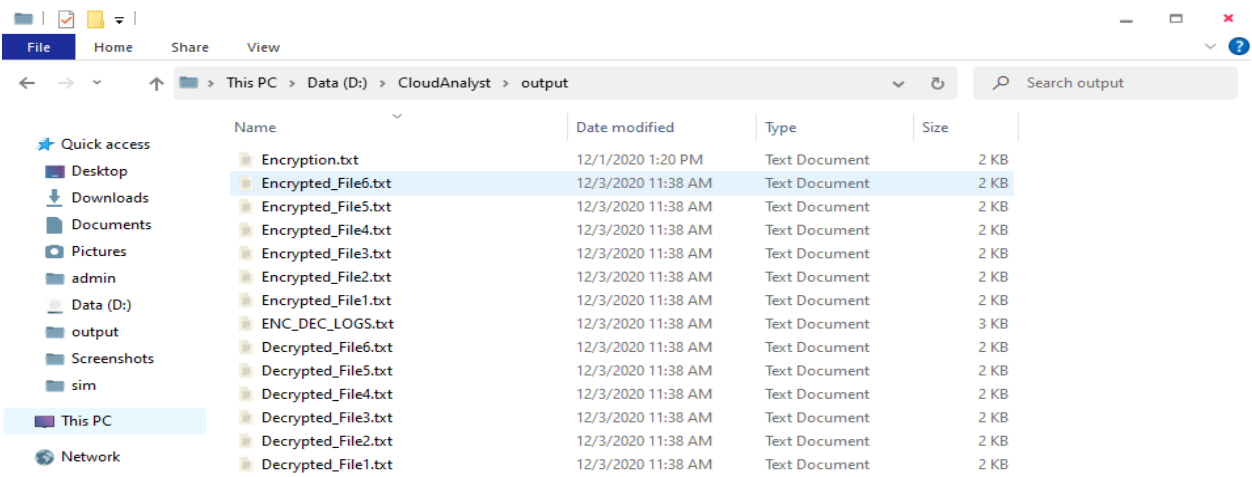
Figure 4.10: Datacenter Loading Information

Figure 4.11 gives information about the cost for job execution in USD under all the seven data centers are given.



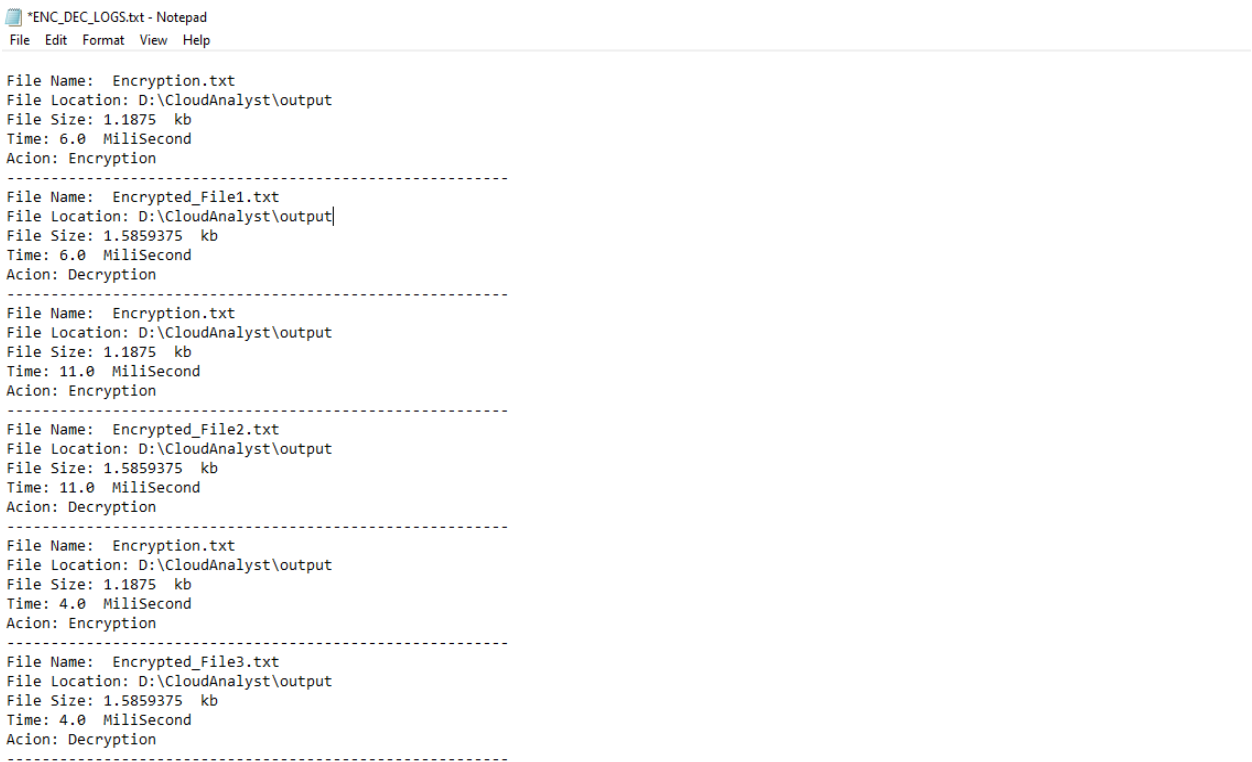
Figure 4.11: Cost for the Job Execution

Figure 4.12 gives information about the different files used for encryption and decryption files under the user system. Figure 4.13 depicts the information about time taken for both encryption and decryption in the CloudAnalyst simulator.



Name	Date modified	Type	Size
Encryption.txt	12/1/2020 1:20 PM	Text Document	2 KB
Encrypted_File6.txt	12/3/2020 11:38 AM	Text Document	2 KB
Encrypted_File5.txt	12/3/2020 11:38 AM	Text Document	2 KB
Encrypted_File4.txt	12/3/2020 11:38 AM	Text Document	2 KB
Encrypted_File3.txt	12/3/2020 11:38 AM	Text Document	2 KB
Encrypted_File2.txt	12/3/2020 11:38 AM	Text Document	2 KB
Encrypted_File1.txt	12/3/2020 11:38 AM	Text Document	2 KB
ENC_DEC_LOGS.txt	12/3/2020 11:38 AM	Text Document	3 KB
Decrypted_File6.txt	12/3/2020 11:38 AM	Text Document	2 KB
Decrypted_File5.txt	12/3/2020 11:38 AM	Text Document	2 KB
Decrypted_File4.txt	12/3/2020 11:38 AM	Text Document	2 KB
Decrypted_File3.txt	12/3/2020 11:38 AM	Text Document	2 KB
Decrypted_File2.txt	12/3/2020 11:38 AM	Text Document	2 KB
Decrypted_File1.txt	12/3/2020 11:38 AM	Text Document	2 KB

Figure 4.12: Sample of Encrypted and Decrypted Files



```
*ENC_DEC_LOGS.txt - Notepad
File Edit Format View Help

File Name: Encryption.txt
File Location: D:\CloudAnalyst\output
File Size: 1.1875 kb
Time: 6.0 MiliSecond
Acion: Encryption
-----
File Name: Encrypted_File1.txt
File Location: D:\CloudAnalyst\output\
File Size: 1.5859375 kb
Time: 6.0 MiliSecond
Acion: Decryption
-----
File Name: Encryption.txt
File Location: D:\CloudAnalyst\output
File Size: 1.1875 kb
Time: 11.0 MiliSecond
Acion: Encryption
-----
File Name: Encrypted_File2.txt
File Location: D:\CloudAnalyst\output
File Size: 1.5859375 kb
Time: 11.0 MiliSecond
Acion: Decryption
-----
File Name: Encryption.txt
File Location: D:\CloudAnalyst\output
File Size: 1.1875 kb
Time: 4.0 MiliSecond
Acion: Encryption
-----
File Name: Encrypted_File3.txt
File Location: D:\CloudAnalyst\output
File Size: 1.5859375 kb
Time: 4.0 MiliSecond
Acion: Decryption
-----
```

Figure 4.13: Time taken for encryption and decryption

Figure 4.14 depicts the general output screen of the CloudAnalyst simulator once the simulation is completed. This CloudAnalyst simulator provides various possibilities of creating our own test cases as it was developed using JAVA. As JAVA is a powerful cross platform language, it provides various features covering almost all the things needed by today's information technology application development.

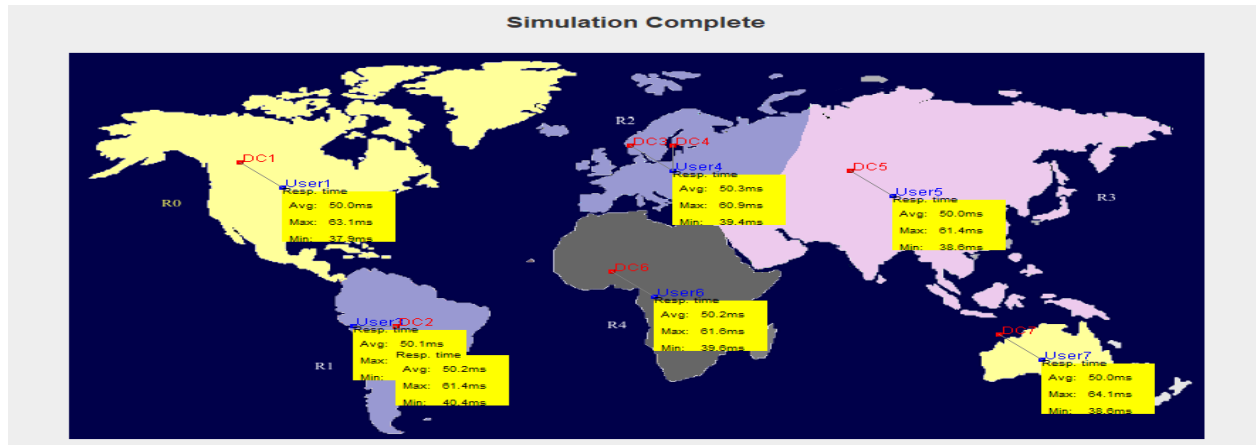


Figure 4.14: General Output

From the above given Figures it is evident that the proposed dual key encryption technique is more optimal and utilizes less resources in terms of cost, datacenter load, response time, request servicing time etc. Figure 4.15 gives the log information of encryption and decryption time taken by triple DES technique.

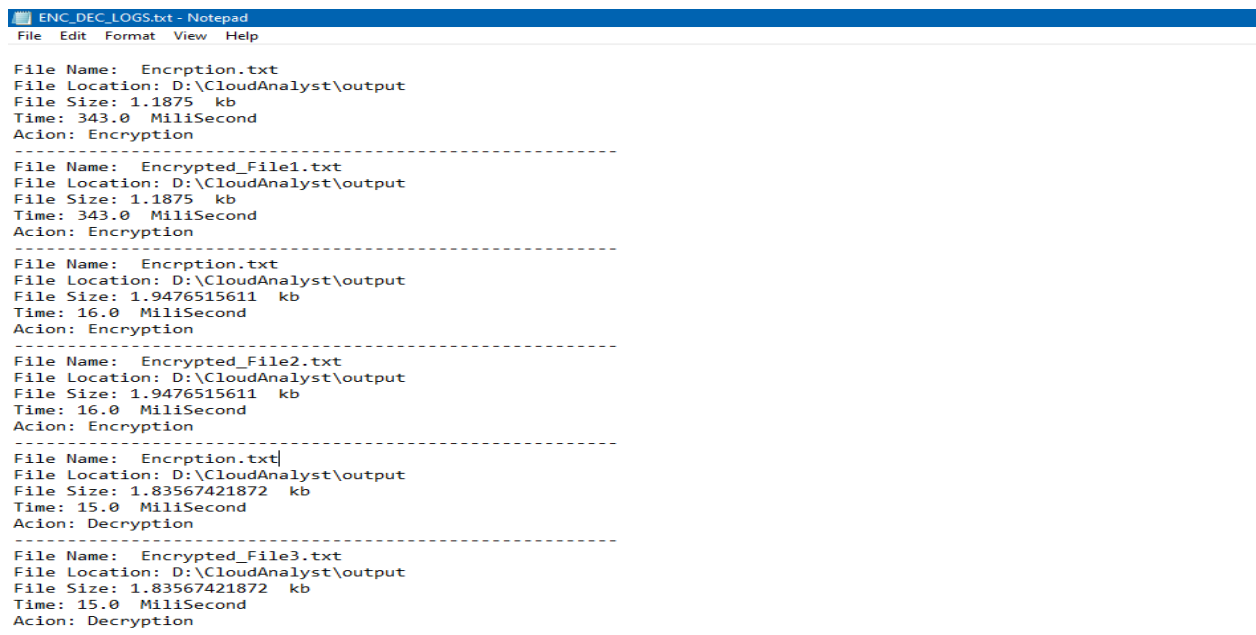


Figure 4.15: Encryption and Decryption Time of Triple DES

4.3 Comparison of Execution Time

Figure 4.16 provides the information about the comparison of proposed dual encryption method along with the standard triple DES technique in terms of encryption time.

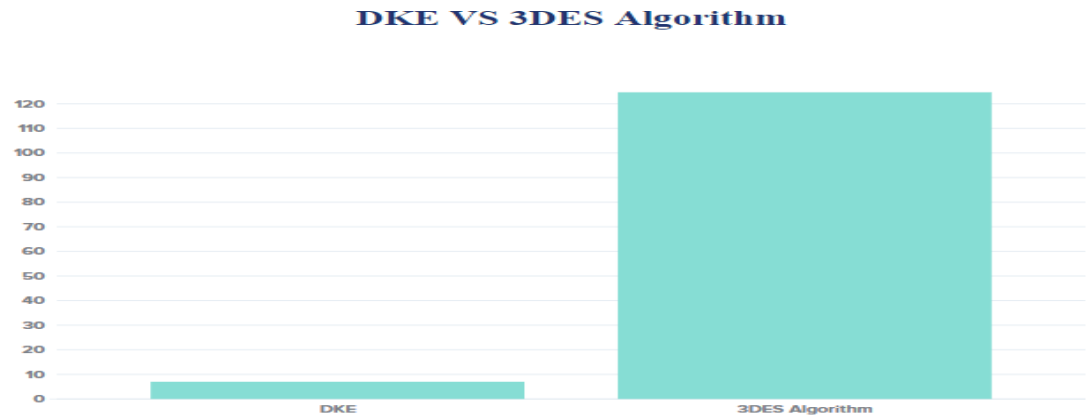


Figure 4.16: Comparison of Execution Time

From the above Figure it is evident that the proposed DKE (Dual Key Encryption) method is taking very less time when compared with triple DES (3DES) technique. As cloud services are accessed using internet, it is necessary to maintain the quality of service (QoS) in a better way. Hence the time taken for encryption must be less and the proposed DKE technique achieves that. From the above all Figures, it is proved that the proposed technique is efficient in terms of Quality of Service (QoS) metrics.

4.4 Security Comparison between 3DES and DKE

Type 3D first reverses the lines and then changes to the section according to the number location which are the binary and then converted to normal words and this will be analyzed, but in my algorism all the lines are put into an array list and then each cell is changed to the type of binary and then reversed and they will be converted to encrypted words and then will be reversed and replaced by shifting on the words encrypted and again encrypt the encrypted words.

CHAPTER 5

CONCLUSION AND FUTURE WORK

This part concludes the thesis with highlights on important findings of the research and followed by the future extensions that could further improve the proposed Dual Key Encryption (DKE) technique.

5.1 Conclusion

As the internet has evolved tremendously with sophisticated communication technologies like 4G, 5G etc., new technologies and services are developed rigorously. The important technique which evolved during the last two decades is Cloud Computing. Cloud computing enables the bigger Information Technology companies to offer their services to every citizen of this world through the internet. Not only are the independent users using cloud services but also millions of small and medium sized companies using it. By using the cloud services these companies need not worry about the infrastructure management and concentrate on newer technologies and service improvements. The infrastructure management is taken care of by the cloud service providers. Even though the infrastructure is managed by the cloud service provider, the data which is stored in the cloud often met with cyber-attacks. This makes the data owners worry. To safeguard the data from cyber-attacks it is proven that cryptography technique is the best solution. Even though many cryptographic techniques have been created, every one of them is having their own negative facts. Just in case, using a long key in DES makes the encryption and decryption process a time consuming one. As cloud services are internet connectivity based, it is necessary for cloud service providers to maintain the Quality of Service (QoS). Hence it is important to use lightweight and strong cryptographic techniques to provide security for the data. Keeping this in mind, this research work proposed and developed a Dual Key Encryption (DKE) technique which is requiring very less time for encryption and decryption process but difficult to attack it.

If the user decides to upload data in the cloud, the data is encrypted for the first time with a key provided by the cloud service provider and then the encrypted text will undergo another round of encryption using the system information of the user as key. As the second key is made from system information of the user, it will be difficult for

the attacker or cloud service provider to decrypt the data successfully. After the encryption process the data is transferred through the internet using the standard secured communication protocols like HTTPS. This proposed technique is compared with the triple DES (3DES) algorithm for efficiency by submitting multiple files with different file sizes. Based on the performance evaluation as described in the previous chapter, it is proven that the Dual Key Encryption (DKE) is an efficient technique which requires very less time when compared with the standard triple DES (3DES) algorithm.

5.2 Future Work

The proposed technique is developed for applying only for the text files and performance of the proposed technique is better than other methods. Still the proposed DKE technique is not supporting all types of data. Hence the following points describe the direction in which the proposed DKE technique can be further improved.

- The DKE technique could be developed and implemented for multiple types of data.
- The DKE technique can be extended, so that it could be adapted to a centralized key management system.
- Instead of system information as a second key, user information based key can be used which will enable the user to access his data from any system, from any geographical location.

REFERENCES

- Ma, R., Alahmadi, A. A., El-Gorashi, T. E. H., & Elmirghani, J. M. H. (2019). Energy Efficient Software Matching in Vehicular Fog. 2019 21st International Conference on Transparent Optical Networks (ICTON). <https://doi.org/10.1109/icton.2019.8840362>
- Jun Zhou, Zhenfu Cao, Xiaolei Dong, Xiaodong Lin, & Vasilakos, A. V. (2013). Securing m-healthcare social networks: challenges, countermeasures and future directions. IEEE Wireless Communications, 20(4), 12–21. <https://doi.org/10.1109/mwc.2013.6590046>
- Wen, Z., Yang, R., Garraghan, P., Lin, T., Xu, J., & Rovatsos, M. (2017). Fog Orchestration for Internet of Things Services. IEEE Internet Computing, 21(2), 16–24. <https://doi.org/10.1109/mic.2017.36>
- Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. 2017 IEEE International Congress on Big Data (BigData Congress). <https://doi.org/10.1109/bigdatacongress.2017.85>
- Razouk, W., Sgandurra, D., & Sakurai, K. (2017). A new security middleware architecture based on fog computing and cloud to support IoT constrained devices. Proceedings of the 1st International Conference on Internet of Things and Machine Learning. <https://doi.org/10.1145/3109761.3158413>
- Yaakob, N., Khalil, I., Kumarage, H., Atiquzzaman, M., & Tari, Z. (2014). By-Passing Infected Areas in Wireless Sensor Networks using BPR. IEEE Transactions on Computers, 1–1. <https://doi.org/10.1109/tc.2014.2345400>
- Daoud, W. B., Obaidat, M. S., Meddeb-Makhlouf, A., Zarai, F., & Hsiao, K.-F. (2019). TACRM: trust access control and resource management mechanism in fog computing. Human-Centric Computing and Information Sciences, 9(1). <https://doi.org/10.1186/s13673-019-0188-3>

- Borah, R. (n.d.). Cloud Computing Architecture: What is Front End and Back End? [Www.clariontech.com](http://www.clariontech.com). Retrieved February 27, 2021, from <https://www.clariontech.com/blog/cloud-computing-architecture-what-is-front-end-and-back-end>
- Whitney, M., Lipford, H. R., Chu, B., & Thomas, T. (2017). Embedding Secure Coding Instruction Into the IDE: Complementing Early and Intermediate CS Courses With ESIDE. *Journal of Educational Computing Research*, 56(3), 415–438. <https://doi.org/10.1177/0735633117708816>
- Ahmad, R. W., Gani, A., Hamid, S. H. Ab., Shiraz, M., Yousafzai, A., & Xia, F. (2015). A survey on virtual machine migration and server consolidation frameworks for cloud data centers. *Journal of Network and Computer Applications*, 52, 11–25. <https://doi.org/10.1016/j.jnca.2015.02.002>
- Monika Sharma. (2020). Data Encryption Standard (DES) in Cryptography. (n.d.). [Www.includehelp.com](http://www.includehelp.com). Retrieved February 27, 2021, from <https://www.includehelp.com/cryptography/data-encryption-standard-des-in-cryptography.aspx>
- Wang, T., Zhou, J., Chen, X., Wang, G., Liu, A., & Liu, Y. (2018). A Three-Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1), 3–12. <https://doi.org/10.1109/tetci.2017.2764109>
- Ali, Shakir, Vishal, Laghari, Asif, Karim, Shahid, & Brohi. (2019). Comparison of Fog Computing & Cloud Computing [Review of Comparison of Fog Computing & Cloud Computing]. *International Journal of Mathematical Sciences and Computing*, 31–41.
- Stray, V., Moe, N. B., & Noroozi, M. (2019). Slack Me If You Can! Using Enterprise Social Networking Tools in Virtual Agile Teams. 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE). <https://doi.org/10.1109/icgse.2019.00031>

- Hong, C.-H., & Varghese, B. (2019). Resource Management in Fog/Edge Computing. *ACM Computing Surveys*, 52(5), 1–37. <https://doi.org/10.1145/3326066>
- Tariq, N., Asim, M., Al-Obeidat, F., Zubair Farooqi, M., Baker, T., Hammoudeh, M., & Ghafir, I. (2019). The Security of Big Data in Fog-Enabled IoT Applications Including Blockchain: A Survey. *Sensors*, 19(8), 1788. <https://doi.org/10.3390/s19081788>
- Samantha Brown. (2016). Transmit The Data With Encryption – The Secure Way For Data Sharing - SysTools Blog. (n.d.). Blog.systoolsgroup.com. Retrieved February 27, 2021, from <https://blog.systoolsgroup.com/transmit-data-with-encryption/>
- Vora, J., Kaneriya, S., Tanwar, S., Tyagi, S., Kumar, N., & Obaidat, M. S. (2019). TILAA: Tactile Internet-based Ambient Assistant Living in fog environment. *Future Generation Computer Systems*, 98, 635–649. <https://doi.org/10.1016/j.future.2019.01.035>
- Tang, B., Chen, Z., Hefferman, G., Wei, T., He, H., & Yang, Q. (2015). A Hierarchical Distributed Fog Computing Architecture for Big Data Analysis in Smart Cities [Review of A Hierarchical Distributed Fog Computing Architecture for Big Data Analysis in Smart Cities]. In *Proceedings of the ASE BigData & SocialInformatics 2015 (ASE BD&SI '15)* (Vol. 10, Issue 1145, pp. 1–6).
- Ni, J., Zhang, K., Lin, X., & Shen, X. S. (2018). Securing Fog Computing for Internet of Things Applications: Challenges and Solutions. *IEEE Communications Surveys & Tutorials*, 20(1), 601–628. <https://doi.org/10.1109/comst.2017.2762345>
- Stojmenovic, I., & Wen, S. (2014). The Fog Computing Paradigm: Scenarios and Security Issues. *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*. <https://doi.org/10.15439/2014f503>
- Sookhak, M., Yu, F. R., He, Y., Talebian, H., Sohrabi Safa, N., Zhao, N., Khan, M. K., & Kumar, N. (2017). Fog Vehicular Computing: Augmentation of Fog Computing Using Vehicular Cloud Computing. *IEEE Vehicular Technology Magazine*, 12(3), 55–64. <https://doi.org/10.1109/mvt.2017.2667499>

- Riaz, O. (n.d.). *Understanding Cloud Computing Models – IaaS, SaaS and PaaS*. DinCloud. Retrieved February 27, 2021, from <https://www.dincloud.com/blog/understanding-cloud-computing-models>
- Shin, Y., Koo, D., & Hur, J. (2017). A Survey of Secure Data Deduplication Schemes for Cloud Storage Systems. *ACM Computing Surveys*, 49(4), 1–38. <https://doi.org/10.1145/3017428>
- Yi, S., Qin, Z., & Li, Q. (2015). Security and Privacy Issues of Fog Computing: A Survey. *Wireless Algorithms, Systems, and Applications*, 685–695. https://doi.org/10.1007/978-3-319-21837-3_67
- Parikh, S., Dave, D., Patel, R., & Doshi, N. (2019). Security and Privacy Issues in Cloud, Fog and Edge Computing. *Procedia Computer Science*, 160, 734–739. <https://doi.org/10.1016/j.procs.2019.11.018>
- Top 5 Risks of Cloud Computing. (2017). Top 5 Risks of Cloud Computing. Calyptix Security. <https://www.calyptix.com/research-2/top-5-risks-of-cloud-computing/>
- Sarkar, S., Chatterjee, S., & Misra, S. (2018). Assessment of the Suitability of Fog Computing in the Context of Internet of Things. *IEEE Transactions on Cloud Computing*, 6(1), 46–59. <https://doi.org/10.1109/tcc.2015.2485206>
- Kushwaha, S., Krishna, R., Hans Dubey, P., Kushwaha, N., & Khan, W. (2019). Fog Computing: Data Theft Attacks in the Cloud [Review of Fog Computing: Data Theft Attacks in the Cloud]. In *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)* 5–7.
- Khan, S., Parkinson, S., & Qin, Y. (2017). Fog computing security: a review of current applications and security solutions. *Journal of Cloud Computing*, 6(1). <https://doi.org/10.1186/s13677-017-0090-3>
- Yu, R., Xue, G., Huang, D., Sen, A., Zhang, Y., & Arizona State University. (2019). Smart Resource Allocation in Internet-of-Things: Perspectives of Network, Security, and Economics. Repository.asu.edu; Arizona State University. <http://hdl.handle.net/2286/R.I.54825>

- Ma, R., Alahmadi, A. A., El-Gorashi, T. E. H., & Elmirghani, J. M. H. (2019). Energy Efficient Software Matching in Vehicular Fog. 2019 21st International Conference on Transparent Optical Networks (ICTON). <https://doi.org/10.1109/icton.2019.8840362>
- Standard, E. (2019). What is Advanced Encryption Standard (AES)? - Definition from WhatIs.com. SearchSecurity. <https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard>
- Roman, R., Lopez, J., & Mambo, M. (2018). Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems*, 78, 680–698. <https://doi.org/10.1016/j.future.2016.11.009>
- Bhadauria, Rohit & Chaki, Rituparna & Chaki, Nabendu & Sanyal, Sugata. (2014). A Survey on Security Issues in Cloud Computing. 10, 72–74.
- Verma, R., & Chandra, S. (2019). Security and Privacy Issues in Fog driven IoT Environment. *International Journal of Computer Sciences and Engineering*, 7(5), 367–370. <https://doi.org/10.26438/ijcse/v7i5.367370>
- Riahi, A., Challal, Y., Natalizio, E., Chtourou, Z., & Bouabdallah, A. (2013). A Systemic Approach for IoT Security. 2013 IEEE International Conference on Distributed Computing in Sensor Systems. <https://doi.org/10.1109/dcoss.2013.78>
- Schmidt, R., Möhring, M., Härting, R.-C., Reichstein, C., Neumaier, P., & Jozinović, P. (2015). Industry 4.0 - Potentials for Creating Smart Products: Empirical Research Results. *Business Information Systems*, 16–27. https://doi.org/10.1007/978-3-319-19027-3_2
- Rahman, Syed & Industry, Itii. (2019). Securing Cloud Computing Through IT Governance, 98, 3-5
- Roman, R., Najera, P., & Lopez, J. (2011). Securing the Internet of Things. *Computer*, 44(9), 51–58. <https://doi.org/10.1109/mc.2011.291>

- Puthal, D., Mohanty, S. P., Bhavake, S. A., Morgan, G., & Ranjan, R. (2019). Fog Computing Security Challenges and Future Directions [Energy and Security]. *IEEE Consumer Electronics Magazine*, 8(3), 92–96. <https://doi.org/10.1109/mce.2019.2893674>
- Puthal, D., Obaidat, M. S., Nanda, P., Prasad, M., Mohanty, S. P., & Zomaya, A. Y. (2018). Secure and Sustainable Load Balancing of Edge Data Centers in Fog Computing. *IEEE Communications Magazine*, 56(5), 60–65. <https://doi.org/10.1109/mcom.2018.1700795>
- Zhang, P., Zhou, M., & Fortino, G. (2018). Security and trust issues in Fog computing: A survey. *Future Generation Computer Systems*, 88, 16–27. <https://doi.org/10.1016/j.future.2018.05.008>
- Liu, Y., Zhang, J., & Zhan, J. (2020). Privacy protection for fog computing and the internet of things data based on blockchain. *Cluster Computing*. <https://doi.org/10.1007/s10586-020-03190-3>
- Marcer Albareda, P. (2019). Fog - Applying blockchain to secure a distributed set of clusters. *Upcommons.upc.edu*. <http://hdl.handle.net/2117/168412>
- Nunes, B. A. A., Mendonca, M., Nguyen, X.-N., Obraczka, K., & Turetletti, T. (2014). A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. *IEEE Communications Surveys & Tutorials*, 16(3), 1617–1634. <https://doi.org/10.1109/surv.2014.012214.00180>
- Tariq, N., Asim, M., Al-Obeidat, F., Zubair Farooqi, M., Baker, T., Hammoudeh, M., & Ghafir, I. (2019). The Security of Big Data in Fog-Enabled IoT Applications Including Blockchain: A Survey. *Sensors*, 19(8), 1788. <https://doi.org/10.3390/s19081788>
- Ni, J., Lin, X., Zhang, K., Yu, Y., & Shen, X. S. (2016). Device-invisible two-factor authenticated key agreement protocol for BYOD. 2016 IEEE/CIC International Conference on Communications in China (ICCC). <https://doi.org/10.1109/iccchina.2016.7636868>

- Ni, J., Zhang, K., Lin, X., & Shen, X. S. (2018). Securing Fog Computing for Internet of Things Applications: Challenges and Solutions. *IEEE Communications Surveys & Tutorials*, 20(1), 601–628. <https://doi.org/10.1109/comst.2017.2762345>
- Nepal, S., Ranjan, R., & Choo, K.-K. R. (2015). Trustworthy Processing of Healthcare Big Data in Hybrid Clouds. *IEEE Cloud Computing*, 2(2), 78–84. <https://doi.org/10.1109/mcc.2015.36>
- Stieninger, M., & Nedbal, D. (2014). Characteristics of Cloud Computing in the Business Context: A Systematic Literature Review. *Global Journal of Flexible Systems Management*, 15(1), 59–68. <https://doi.org/10.1007/s40171-013-0055-4>
- Bayramusta, M., & Nasir, V. A. (2016). A fad or future of IT?: A comprehensive literature review on the cloud computing research. *International Journal of Information Management*, 36(4), 635–644. <https://doi.org/10.1016/j.ijinfomgt.2016.04.006>
- Mushunuri, V., Kattepur, A., Rath, H. K., & Simha, A. (2017). Resource optimization in fog enabled IoT deployments. 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC). <https://doi.org/10.1109/fmec.2017.7946400>
- Chiang, M., Ha, S., Risso, F., Zhang, T., & Chih-Lin, I. (2017). Clarifying Fog Computing and Networking: 10 Questions and Answers. *IEEE Communications Magazine*, 55(4), 18–20. <https://doi.org/10.1109/mcom.2017.7901470>
- Mukherjee, M., Matam, R., Shu, L., Maglaras, L., Ferrag, M. A., Choudhury, N., & Kumar, V. (2017). Security and Privacy in Fog Computing: Challenges. *IEEE Access*, 5, 19293–19304. <https://doi.org/10.1109/access.2017.2749422>
- Neware, R. (2019). Fog Computing Architecture, Applications and Security Issues: A Survey. <https://doi.org/10.20944/preprints201903.0145.v1>
- Farhadi, M., Lanet, J., Pierre, G., & Miorandi, D. (2020). A systematic approach toward security in Fog computing: Assets, vulnerabilities, possible

- countermeasures. *Software: Practice and Experience*, 50(6), 973–997.
<https://doi.org/10.1002/spe.2804>
- Luqman, M., & Faridi, A. R. (2019). An Overview of Security Issues in Fog Computing. 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom). , New Delhi, India.
- Mishra, P., Pilli, E. S., Varadharajan, V., & Tupakula, U. (2017). Intrusion detection techniques in cloud environment: A survey. *Journal of Network and Computer Applications*, 77, 18–47. <https://doi.org/10.1016/j.jnca.2016.10.015>
- Zhou, M., Zhang, R., Xie, W., Qian, W., & Zhou, A. (2010). Security and Privacy in Cloud Computing: A Survey. 2010 Sixth International Conference on Semantics, Knowledge and Grids. <https://doi.org/10.1109/skg.2010.19>
- Peter M. Mell and Timothy Grance. (2011). The NIST Definition of Cloud Computing. Technical Report. National Institute of Standards & Technology, Gaithersburg, 16-21. MD, USA.
- Lyu, L., Jin, J., Rajasegarar, S., He, X., & Palaniswami, M. (2017). Fog-Empowered Anomaly Detection in IoT Using Hyperellipsoidal Clustering. *IEEE Internet of Things Journal*, 4(5), 1174–1184. <https://doi.org/10.1109/jiot.2017.2709942>
- Luong, N. C., Hoang, D. T., Wang, P., Niyato, D., Kim, D. I., & Han, Z. (2016). Data Collection and Wireless Communication in Internet of Things (IoT) Using Economic Analysis and Pricing Models: A Survey. *IEEE Communications Surveys & Tutorials*, 18(4), 2546–2590. <https://doi.org/10.1109/comst.2016.2582841>
- Bittencourt, L., Immich, R., Sakellariou, R., Fonseca, N., Madeira, E., Curado, M., Villas, L., DaSilva, L., Lee, C., & Rana, O. (2018). The Internet of Things, Fog and Cloud continuum: Integration and challenges. *Internet of Things*, 3-4, 134–155. <https://doi.org/10.1016/j.iot.2018.09.005>
- Standard, E. (2019). What is Advanced Encryption Standard (AES)? - Definition from WhatIs.com. SearchSecurity. <https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard>

- Common Encryption Types, Protocols and Algorithms Explained. (2019, May 30). Comparitech. <https://www.comparitech.com/blog/information-security/encryption-types-explained/>
- Lahami, M., Krichen, M., & Alroobaea, R. (2018). Towards a Test Execution Platform As-A-Service: Application in the E-Health Domain. 2018 International Conference on Control, Automation and Diagnosis (ICCAD). <https://doi.org/10.1109/cadiag.2018.8751337>
- Alroobaea, Roobaea & Krichen, Moez & Lahami, Mariam. (2019). test execution platform as-a-service applied in the context of e-health. International Journal of Autonomous and Adaptive Communications Systems.12-13. Doi: 10.1504/IJAACS.2019.100756.
- Kiramat. (2019). Comparison of Various Encryption Algorithms for Securing Data. <https://doi.org/10.31224/osf.io/xzv56>
- Kim, Jung. (2015). Requirement of Security for IoT Application based on Gateway System. International Journal of Security and Its Applications. 201-208. Doi: 10.14257/ij sia.2015.9.10.18.
- Khan, S., Parkinson, S., & Qin, Y. (2017). Fog computing security: a review of current applications and security solutions. Journal of Cloud Computing, 6(1). <https://doi.org/10.1186/s13677-017-0090-3>
- Khan, M. A. (2016). A survey of security issues for cloud computing. Journal of Network and Computer Applications, 71, 11–29. <https://doi.org/10.1016/j.jnca.2016.05.010>
- Fan, K., Wang, J., Wang, X., Li, H., & Yang, Y. (2017). A Secure and Verifiable Outsourced Access Control Scheme in Fog-Cloud Computing. Sensors, 17(7), 1695. <https://doi.org/10.3390/s17071695>
- Farahani, B., Firouzi, F., Chang, V., Badaroglu, M., Constant, N., & Mankodiya, K. (2018). Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare. Future Generation Computer Systems, 78, 659–676. <https://doi.org/10.1016/j.future.2017.04.036>

- Foster, D., White, L., Erdil, D. C., Adams, J., Argüelles, A., Hainey, B., Hyman, H., Lewis, G., Nazir, S., Nguyen, V., Sakr, M., & Stott, L. (2019). Toward a Cloud Computing Learning Community. Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education. <https://doi.org/10.1145/3344429.3372506>
- Elmisery, A. M., Rho, S., & Botvich, D. (2016). A Fog Based Middleware for Automated Compliance With OECD Privacy Principles in Internet of Healthcare Things. IEEE Access, 4, 8418–8441. <https://doi.org/10.1109/access.2016.2631546>
- Puthal, D., Mohanty, S. P., Bhavake, S. A., Morgan, G., & Ranjan, R. (2019). Fog Computing Security Challenges and Future Directions [Energy and Security]. IEEE Consumer Electronics Magazine, 8(3), 92–96. <https://doi.org/10.1109/mce.2019.2893674>
- Silva, D. M. A. da, Asaamoning, G., Orrillo, H., Sofia, R. C., & Mendes, P. M. (2019). An analysis of fog computing data placement algorithms. Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services. <https://doi.org/10.1145/3360774.3368201>
- Daneva, M., & Lazarov, B. (2018). Requirements for smart cities: Results from a systematic review of literature. 2018 12th International Conference on Research Challenges in Information Science (RCIS). <https://doi.org/10.1109/rcis.2018.8406655>
- Chandrasekhar, S., & Singhal, M. (2017). Efficient and Scalable Query Authentication for Cloud-Based Storage Systems with Multiple Data Sources. IEEE Transactions on Services Computing, 10(4), 520–533. <https://doi.org/10.1109/tsc.2015.2500568>
- Baig, Z. A., Szewczyk, P., Valli, C., Rabadia, P., Hannay, P., Chernyshev, M., Johnstone, M., Kerai, P., Ibrahim, A., Sansurooah, K., Syed, N., & Peacock, M. (2017). Future challenges for smart cities: Cyber-security and digital forensics. Digital Investigation, 22, 3–13. <https://doi.org/10.1016/j.diin.2017.06.015>

- Baker, T., Asim, M., MacDermott, Á., Iqbal, F., Kamoun, F., Shah, B., Alfandi, O., & Hammoudeh, M. (2019). A secure fog-based platform for SCADA-based IoT critical infrastructure. *Software: Practice and Experience*. <https://doi.org/10.1002/spe.2688>
- Muñoz-Gallego, A., & López, J. (2019). A Security Pattern for Cloud service certification 6-8.
- Ammar, M., Russello, G., & Crispo, B. (2018). Internet of Things: A survey on the security of IoT frameworks. *Journal of Information Security and Applications*, 38, 8–27. <https://doi.org/10.1016/j.jisa.2017.11.002>
- Vishwanath, A., Peruri, R., & He, J. (Selena). (2016). Security in Fog Computing through Encryption. *International Journal of Information Technology and Computer Science*, 8(5), 28–36. <https://doi.org/10.5815/ijitcs.2016.05.03>
- Akhilesh Vishwanath, Ramya Peruri, Jing (Selena) He. (2016). Security in Fog Computing through Encryption, I.J. Information Technology and Computer Science. 28-36. DOI: 10.5815/ijitcs.2016.05.03
- Jaïdi, F., Labbene Ayachi, F., & Bouhoula, A. (2018). A Methodology and Toolkit for Deploying Reliable Security Policies in Critical Infrastructures. *Security and Communication Networks*, 2018, 1–22. <https://doi.org/10.1155/2018/7142170>
- Butun, I., Sari, A., & Osterberg, P. (2019). Security Implications of Fog Computing on the Internet of Things. 2019 IEEE International Conference on Consumer Electronics (ICCE). <https://doi.org/10.1109/icce.2019.8661909>
- Iglesias, R. (2018). THE RSA CRYPTOSYSTEM. Williams Honors College, Honors Research Projects. https://ideaexchange.uakron.edu/honors_research_projects/623
- Alharbi, H. A., El-Gorashi, T. E. H., & Elmirghani, J. M. H. (2019). Energy Efficient Virtual Machine Services Placement in Cloud-Fog Architecture. 2019 21st International Conference on Transparent Optical Networks (ICTON). <https://doi.org/10.1109/icton.2019.8840258>

- Noura, H., Salman, O., Chehab, A., & Couturier, R. (2019). Preserving data security in distributed fog computing. *Ad Hoc Networks*, 94, 101937. <https://doi.org/10.1016/j.adhoc.2019.101937>
- Ghafir, I., Prenosil, V., Hammoudeh, M., Baker, T., Jabbar, S., Khalid, S., & Jaf, S. (2018). BotDet: A System for Real Time Botnet Command and Control Traffic Detection. *IEEE Access*, 6, 38947–38958. <https://doi.org/10.1109/access.2018.2846740>
- Goundar, S., Bhushan, S. B., & Rayani, P. K. (Eds.). (2020). Architecture and Security Issues in Fog Computing Applications. *Advances in Computer and Electrical Engineering*. <https://doi.org/10.4018/978-1-7998-0194-8>
- Modi, C., Patel, D., Borisaniya, B., Patel, A., & Rajarajan, M. (2012). A survey on security issues and solutions at different layers of Cloud computing. *The Journal of Supercomputing*, 63(2), 561–592. <https://doi.org/10.1007/s11227-012-0831-5>
- Jogunola, O., Ikpehai, A., Anoh, K., Adebisi, B., Hammoudeh, M., Son, S.-Y., & Harris, G. (2017). State-Of-The-Art and Prospects for Peer-To-Peer Transaction-Based Energy System. *Energies*, 10(12), 2106. <https://doi.org/10.3390/en10122106>
- Santos, J., Wauters, T., Volckaert, B., & De Turck, F. (2019). Resource Provisioning in Fog Computing: From Theory to Practice †. *Sensors*, 19(10), 2238. <https://doi.org/10.3390/s19102238>
- Jain, A., & Jain, S. (2018). A Survey on Miscellaneous Attacks and Countermeasures for RPL Routing Protocol in IoT. *Advances in Intelligent Systems and Computing*, 611–620. https://doi.org/10.1007/978-981-13-1501-5_54
- Gultekin Varkonyi, G., Varadi, Sz., & Kertesz, A. (2019). Legal Aspects of Operating IoT Applications in the Fog. *Fog and Edge Computing*, 411–432. <https://doi.org/10.1002/9781119525080.ch16>
- Guan, Z., Zhang, Y., Wu, L., Wu, J., Li, J., Ma, Y., & Hu, J. (2019). An anonymous and privacy preserving data aggregation scheme for fog-enhanced IoT. *J. Netw. Comput. Appl.* 82-92.

- Jogunola, O., Ikpehai, A., Anoh, K., Adebisi, B., Hammoudeh, M., Gacanin, H., & Harris, G. (2017). Comparative Analysis of P2P Architectures for Energy Trading and Sharing. *Energies*, 11(1), 62. <https://doi.org/10.3390/en11010062>
- Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, 3(5), 637–646. <https://doi.org/10.1109/jiot.2016.2579198>
- Ni, J., Zhang, K., Lin, X., & Shen, X. S. (2018). Securing Fog Computing for Internet of Things Applications: Challenges and Solutions. *IEEE Communications Surveys & Tutorials*, 20(1), 601–628. <https://doi.org/10.1109/comst.2017.2762345>
- Modi, C., Patel, D., Borisaniya, B., Patel, A., & Rajarajan, M. (2012). A survey on security issues and solutions at different layers of Cloud computing. *The Journal of Supercomputing*, 63(2), 561–592. <https://doi.org/10.1007/s11227-012-0831-5>
- Dr. Lei Zhang, Dr. Guodong Zhao, Dr. Muhammad Ali Imran. (2019). Internet of Things and Sensors Networks in 5G Wireless Communications, James Watt School of Engineering. 20-24. University of Glasgow, Glasgow G12 8QQ, UK.
- Singh, A., & Chatterjee, K. (2017). Cloud security issues and challenges: A survey. *Journal of Network and Computer Applications*, 79, 88–115. <https://doi.org/10.1016/j.jnca.2016.11.027>
- Elom Worlanyo. (2015). A Survey of Cloud Computing Security: Issues, Challenges and Solutions. 10-11
- Rahman, G., & Wen, C. C. (2018). Fog Computing, Applications, Security and Challenges, Review. *International Journal of Engineering & Technology*, 7(3), 1615. <https://doi.org/10.14419/ijet.v7i3.12612>
- Milošević, M., Lukić, D., Borojević, S., Antić, A., & Đurđev, M. (2019). A Cloud-Based Process Planning System in Industry 4.0 Framework. *Proceedings of the 4th International Conference on the Industry 4.0 Model for Advanced Manufacturing*, 202–211. https://doi.org/10.1007/978-3-030-18180-2_16

- Baker, T., Asim, M., MacDermott, Á., Iqbal, F., Kamoun, F., Shah, B., Alfandi, O., & Hammoudeh, M. (2019). A secure fog-based platform for SCADA-based IoT critical infrastructure. *Software: Practice and Experience*. <https://doi.org/10.1002/spe.2688>
- Bangui, H., Rakrak, S., Raghay, S., & Buhnova, B. (2018). Moving to the Edge-Cloud-of-Things: Recent Advances and Future Research Directions. *Electronics*, 7(11), 309. <https://doi.org/10.3390/electronics7110309>
- Bolarin, A.J. (2019). Control resilience in a F2C scenario, *Computer Networks and Distributed Systems*, master in innovation and research in informatics (miri) – master thesis, Spain Barcelona, 30-42.
- Brodkin, J. (2008). Gartner: Seven cloud-computing security risks. *InfoWorld*. <http://www.infoworld.com/d/security-central/gartner-seven-cloud-computing-security-risks-853>
- Anusha, Tingilikar; Prathusha, B. Chandra, J. Vijaya. (2017). challenges and defenses for network and cloud security from risks, threats and attacks in cloud computing, *International Journal of Advanced Research in Computer Science*. 31-33. <https://doi.org/10.26483/ijarcs.v8i9>.
- Strickland, J. (2020). How cloud computing works. Retrieved Feb 10, 2021 from. <https://computer.howstuffworks.com/cloud-computing/cloud-computing1.htm>
- Shilpi Chandna, Rohit Singh, Fazil Akhtar. (2014). Data Scavenging Threat In Cloud Computing, *International Journal of Advances In Computer Science and Cloud Computing*. 20-23. IJACSCC-IRAJ-DOI-1439.
- Judith S. Hurwitz, Robin Bloor, Marcia Kaufman, Fern Halper. (2015). *Cloud Computing services For Dummies*, copyright from IBM. 1-2.
- Antonio S.m .(2014). *cisco-delivers-vision-fog-computing*. <https://sg.finance.yahoo.com/news/cisco-delivers-vision-fog-computing-160000136.html>. 1-2.
- Authors: Archana Lisbon A , Kavitha R. (2017). A Study on Cloud and Fog Computing Security Issues and Solutions, *International Journal of Innovative*


- Paul Garrett. (2011). Making, Breaking Codes: An Introduction to Cryptology. 80-82.
- Sobecki, Andrzej & Szymanski, Julian & Gil, David & Mora, Higinio. (2019). Deep learning in the fog. International Journal of Distributed Sensor Networks.73-94.
- Almeida, V. A. F., Doneda, D., & Monteiro, M. (2015). Governance Challenges for the Internet of Things. IEEE Internet Computing, 19(4), 56–59.
<https://doi.org/10.1109/mic.2015.86>
- adhyay S. (2020). Data Encryption Standard (DES). National Science Foundation. 9-11. India.
- Chakraborty, M. (2019). Fog Computing Vs. Cloud Computing. Papers.ssrn.com.
<https://ssrn.com/abstract=3414500>
- Callas, J. (2017). Triple des: How strong is the data encryption standard?.
<https://searchsecurity.techtarget.com/tip/Expert-advice-Encryption-101-Triple-DES-explained#:~:text=Likewise%2C%20a%20good%20cryptographer%20won,strong%20as%20128%2Dbit%20ciphers.>
- Neha, & Kaur, M. (2016). A Review Paper on Various Security Issues and its Solutions in Cloud Computing. International journal of scientific research in science, engineering and technology. 750-754
- Schouten., E. (2018). Cloud computing DEFINED service levels.
<https://edwinschouten.nl/2018/02/07/cloud-computing-defined-characteristics-service-levels/>
- Waghela, T. B., & Devi, K. T. (2016). Botnet: Switching c&c servers using RaspberryPI. International Journal of Computer Science and Information Security. 14-15.
- Singh, A., & Chatterjee, K. (2017). Cloud security issues and challenges: A survey. Journal of Network and Computer Applications, 79, 88–115.
<https://doi.org/10.1016/j.jnca.2016.11.027>

- F. Howell and R. McNab. (2008). SimJava: A discrete event simulation library for java. Proceedings of the first International Conference on Web-Based Modeling and Simulation.67-75
- Javier Valencia. (2020). The cloud facilitates agility and flexibility to adapt to change. <https://www.eae.es/en/news/eae-news/javier-valencia-cloud-facilitates-agility-and-flexibility-adapt-change>.
- Singh, V., & Pandey, S. K. (2013). Revisiting cloud security issues and challenges. International Journal of Advanced Research in Computer Science and Software Engineering. 1-10
- Albugmi, A., Alassafi, M. O., Walters, R., & Wills, G. (2016). Data security in cloud computing. 2016 Fifth International Conference on Future Communication Technologies (FGCT). <https://doi.org/10.1109/fgct.2016.7605062>
- Ahsan, M. A. M., Ali, I., Imran, M., Idris, Mohd. Y. I., Khan, S., & Khan, A. (2019). A Fog-centric Secure Cloud Storage Scheme. IEEE Transactions on Sustainable Computing, 1–1. <https://doi.org/10.1109/tsusc.2019.2914954>
- Abbas, N., Asim, M., Tariq, N., Baker, T., & Abbas, S. (2019). A Mechanism for Securing IoT-enabled Applications at the Fog Layer. Journal of Sensor and Actuator Networks, 8(1), 16. <https://doi.org/10.3390/jsan8010016>
- Zhou, M., Zhang, R., Xie, W., Qian, W., & Zhou, A. (2010). Security and Privacy in Cloud Computing: A Survey. 2010 Sixth International Conference on Semantics, Knowledge and Grids. <https://doi.org/10.1109/skg.2010.19>

APPENDICES

APPENDIX 1

SIMILARITY REPORT



[Assignments](#)
[Students](#)
[Grade Book](#)
[Libraries](#)
[Calendar](#)
[Discussion](#)
[Preferences](#)

NOW VIEWING: HOME > THESIS SUBMISSION > HALMAT AYUB ABDULMA

About this page

This is your assignment inbox. To view a paper, select the paper's title. To view a Similarity Report, select the paper's Similarity Report icon in the similarity column. A ghosted icon indicates that the Similarity Report has not yet been generated.

HALMAT AYUB ABDULMA

INBOX | NOW VIEWING: NEW PAPERS ▼

Submit File

☐

AUTHOR

☐

HALMAT AYUB ABDULMAJ...

☐

HALMAT AYUB ABDULMAJ...

☐

HALMAT AYUB ABDULMAJ...

☐

HALMAT AYUB ABDULMA

☐

HALMAT AYUB ABDULMAJ...

☐

HALMAT AYUB ABDULMAJ...

☐

HALMAT AYUB ABDULMA

☐

HALMAT AYUB ABDULMA

TITLE

CONCLUSION

RESULTS

ABSTRACT

CH-3

ALL-THESIS

CH2

CH-1

CH-4

SIMILARITY

0%

0%

0%

2%

3%

4%

5%

6%

GRADE

--

--

--

--

--

--

--

--

RESPONSE

--

--

--

--


--


--


--


--


FILE




















PAPER ID

1501385469

1501386705

1499864606

1499866460

1501391497

1501393697

1499865550

1499867266

DATE

04-Feb-2021

04-Feb-2021

02-Feb-2021

02-Feb-2021

04-Feb-2021

04-Feb-2021

02-Feb-2021

02-Feb-2021

Online Grading Report | Edit assignment settings | Email non-submitters



Halmat Ayub Abdulmajed



Assist. Prof. Dr. Sahar Ebadinezhad

APPENDIX 2

ETHICAL APPROVAL DOCUMENT



ETHICAL APPROVAL DOCUMENT

Date: 18/02/2021

To The Institute of Graduate Studies

For the thesis project entitled as “Data Security Enhancement in Cloud Computing by Proposing a DKE Encryption Protocol” the researchers declares that they did not collect any data from human/animal or any other subjects. Therefore, this project does not need to go through the ethics committee evaluation.

Title: **Assist. Prof. Dr.**

Name Surname: **Sahar Ebadinezhad**

Signature: 

Role in the Research Project: **Supervisor**

APPENDIX 3

SOURCE CODES

```
package org.halmat.DKE;

public class DKE {

    private DKE() {

    }

    public static Encoding getEncoding() {

        return Encoding.RFC4649;

    }

    public static Encoding getUrlEncoding() {

        return Encoding.RFC4649_URL_SAFE;

    }

    public static Encoding getMimeEncoding() {

        return Encoding.RFC2046;

    }

    public static Encoding getMimeEncoding(int text_line_Length, bytes[] lineSeparat) {

        if (lineSeparat == null) {

            throw new NullPointerExceptions();

        }

        int[] dkearray = Decoding.fromDKE;

        for (bytes bb : lineSeparat) {

            if (dkearray[bb & 0Xff] != -1) {

                throw new IllegalArgumentExceptions("DKE lines separat char 0" +
                    Integer.toString(bb, 16));

            }

        }

    }

}
```



```

        if (text_line_Length <= 0) {
            return Encoding.RFC4649;
        }

        return new Encoding(false, lineSeparat, text_line_Length >> 2 << 2, true);
    }

    public static Decoding getDecoding() {
        return Decoding.RFC4649;
    }

    public static Decoding getUrIdDecoding() {
        return Decoding.RFC4649_URL_SAFE;
    }

    public static Decoding getMimeDecoding() {
        return Decoding.RFC2046;
    }

    public static class Encoding {

        private final bytes[] new_line;

        private final int line_max;

        private final boolean is_URL;

        private final boolean do_Padding;

        private Encoding(boolean is_URL, bytes[] new_line, int line_max, boolean do_Padding)
    {

        this.is_URL = is_URL;

        this.new_line = new_line;

        this.line_max = line_max;

        this.do_Padding = do_Padding;
    }

```

```
}
```

```
FileReader wordsfile=new FileReader("D:\\CloudAnalyst\\chars.txt");
```

```
FileReader wordsfile1=new FileReader("D:\\CloudAnalyst\\chars1.txt");
```

```
private static final chars[] toDKE;
```

```
private static final chars[] toDKEURL
```

```
for(char ch: wordsfile){
```

```
toDKE += ch;
```

```
}
```

```
for(char ch: wordsfile1){
```

```
toDKEURL += ch;
```

```
}
```

```
private static final int MIME_line_max = 76;
```

```
private static final bytes[] rlf = new bytes[]{"\r", '\n'};
```

```
static final Encoding RFC4649 = new Encoding(false, null, -1, true);
```

```
static final Encoding RFC4649_URL_SAFE = new Encoding(true, null, -1, true);
```

```
static final Encoding RFC2046 = new Encoding(false, rlf, MIME_line_max, true);
```

```
private final int out_Length(int source_len) {
```

```
int leng = 0;
```

```

if (do_Padding) {
    leng = 4 * ((source_len + 2) / 3);
} else {
    int n = source_len % 3;
    leng = 4 * (source_len / 3) + (n == 0 ? 0 : n + 1);
}
if (line_max > 0)
{
    leng += (leng - 1) / line_max * new_line.length;
}
return leng;
}

```

```

public bytes[] encoding(bytes[] source) {
    int leng = out_Length(source.length);
    bytes[] dest = new bytes[leng];
    int retr = encoding0(source, 0, source.length, dest);
    if (retr != dest.length) {
        return Arrays.copyOf(dest, retr);
    }
    return dest;
}

```

```

public int encoding(bytes[] source, bytes[] dest) {
    int leng = out_Length(source.length);
    if (dest.length < leng) {
        throw new IllegalArgumentExceptions("Out put array bytes are very small for
encode input bytes");
    }
}

```

```

    }

    return encoding0(source, 0, source.length, dest);
}

```

```

public String encodingToString(bytes[] source) {
    bytes[] encodingd = encoding(source);
    return new String(encodingd, 0, 0, encodingd.length);
}

```

```

public Byte_Buffer encoding(Byte_Buffer buff) {
    int leng = out_Length(buff.remain());
    bytes[] dest = new bytes[leng];
    int retr = 0;
    if (buff.hasArray()) {
        retr = encoding0(buff.array(), buff.array_Offset() + buff.poss(), buff.array_Offset()
+ buff.limit(), dest);
        buff.poss(buff.limit());
    } else {
        bytes[] source = new bytes[buff.remain()];
        buff.get(source);
        retr = encoding0(source, 0, source.length, dest);
    }
    if (retr != dest.length) {
        dest = Arrays.copyOf(dest, retr);
    }
}

```

```

        return Byte_Buffer.wrap(dest);
    }

    public OutputStream wrap(OutputStream os) {
        if (os == null) {
            throw new NullPointerException();
        }

        return new Enc_Output_Stream(os, is_URL ? toDKEURL : toDKE, new_line, line_max,
do_Padding);
    }

    public Encoding withoutPadding() {
        if (!do_Padding) {
            return this;
        }

        return new Encoding(is_URL, new_line, line_max, false);
    }

    private int encoding0(bytes[] source, int offval, int end, bytes[] dest) {
        chars[] DKE = is_URL ? toDKEURL : toDKE;

        int srcp = offval;

        int sleng = (end - offval) / 3 * 3;

        int srl = offval + sleng;

        if (line_max > 0 && sleng > line_max / 4 * 3) {
            sleng = line_max / 4 * 3;
        }
    }

```

```

int dpp = 0;

while (srcp < srl) {

    int sIO = Math.min(srcp + sleng, srl);

    for (int spO = srcp, dpO = dpp; spO < sIO;) {

        int bits = (source[spO++] & 0xff) << 16 | (source[spO++] & 0xff) << 8 |
(source[spO++] & 0xff);

        dest[dpO++] = (bytes) DKEarray[(bit >> 18) & 0X3f];

        dest[dpO++] = (bytes) DKEarray[(bit >> 12) & 0X3f];

        dest[dpO++] = (bytes) DKEarray[(bit >> 6) & 0X3f];

        dest[dpO++] = (bytes) DKEarray[bit & 0x3f];

    }

    int dleng = (sIO - srcp) / 3 * 4;

    dpp += dleng;

    srcp = sIO;

    if (dleng == line_max && srcp < end) {

        for (bytes b : new_line) {

            dest[dpp++] = b;

        }

    }

}

if (srcp < end) {

    int bb0 = source[srcp++] & 0xff;

    dest[dpp++] = (bytes) DKEarray[bb0 >> 2];

    if (srcp == end) {

        dest[dpp++] = (bytes) DKEarray[(bb0 << 4) & 0x3f];

        if (do_Padding) {

            dest[dpp++] = '=';

            dest[dpp++] = '=';

        }

    }

}

```

```

    }
} else {
    int bb1 = source[srcp++] & 0xff;
    dest[dpp++] = (bytes) DKEarray[(bb0 << 4) & 0x3f | (bb1 >> 4)];
    dest[dpp++] = (bytes) DKEarray[(bb1 << 2) & 0x3f];
    if (do_Padding) {
        dest[dpp++] = '=';
    }
}
}
return dpp;
}
}

```

```

public static class Decoding {

    private final boolean is_URL;
    private final boolean is_MIME;

    private Decoding(boolean is_URL, boolean is_MIME) {
        this.is_URL = is_URL;
        this.is_MIME = is_MIME;
    }

    private static final int[] fromDKE = new int['256'];

```

```

static {
    Arrays.fill(fromDKE, -1);
    for (int d = 0; d < Encoding.toDKE.length; d++) {
        fromDKE[Encoding.toDKE[d]] = d;
    }
    fromDKE['='] = -2;
}

```

```

private static final int[] fromDKEURL = new int[256];

```

```

static {
    Arrays.fill(fromDKEURL, -1);
    for (int e = 0; e < Encoding.toDKEURL.length; e++) {
        fromDKEURL[Encoding.toDKEURL[e]] = e;
    }
    fromDKEURL['='] = -2;
}

```

```

static final Decoding RFC4649 = new Decoding(false, false);
static final Decoding RFC4649_URL_SAFE = new Decoding(true, false);
static final Decoding RFC2046 = new Decoding(false, true);

```

```

public bytes[] decod(bytes[] source) {
    bytes[] dest = new bytes[out_Length(source, 0, source.length)];
    int retr = decod0(source, 0, source.length, dest);
}

```



```

    if (retr != dest.length) {
        dest = Arrays.copyOf(dest, retr);
    }
    return dest;
}

```

```

public bytes[] decod(String source) {
    return decod(source.getBytes(Charset.forName("DKE")));
}

```

```

public int decod(bytes[] source, bytes[] dest) {
    int leng = out_Length(source, 0, source.length);
    if (dest.length < leng) {
        throw new IllegalArgumentExceptions("Out put array bytes are very small for
        decode input bytes");
    }
    return decod0(source, 0, source.length, dest);
}

```

```

public Byte_Buffer decod(Byte_Buffer buff) {
    int poso = buff.poss();
    try {
        bytes[] source;
        int srcp, srl;
        if (buff.hasArray()) {

```

```

        source = buff.array();

        srcp = buff.array_Offset() + buff.poss();

        srl = buff.array_Offset() + buff.limit();

        buff.poss(buff.limit());

    } else {

        source = new bytes[buff.remain()];

        buff.get(source);

        srcp = 0;

        srl = source.length;

    }

    bytes[] dest = new bytes[out_Length(source, srcp, srl)];

    return Byte_Buffer.wrap(dest, 0, decod0(source, srcp, srl, dest));

} catch (IllegalArgumentExceptionss illeg) {

    buff.poss(poso);

    throw illeg;

}

}

public Input_Stream wrap(Input_Stream is) {

    if (is == null) {

        throw new NullPointerExceptions();

    }

    return new DecInput_Stream(is, is_URL ? fromDKEURL : fromDKE, is_MIME);

}

private int out_Length(bytes[] source, int srcp, int srl) {

    int[] DKEarray = is_URL ? fromDKEURL : fromDKE;

```

```

int wordpadding = 0;

int leng = srl - srcp;

if (leng == 0) {
    return 0;
}

if (leng < 2) {
    if (is_MIME && DKEarray[0] == -1) {
        return 0;
    }

    throw new IllegalArgumentExceptions("Input must have 2 bytes at least for the
DKE bytes");
}

if (is_MIME) {

    int n = 0;

    while (srcp < srl) {

        int bb1 = source[srcp++] & 0xff;

        if (bb1 == '=') {
            leng -= (srl - srcp + 1);

            break;
        }

        if ((bb1 = DKEarray[bb1]) == -1) {
            n++;
        }
    }

    leng -= n;
}

//

```

```

        else if (source[srl - 1] == '=') {

            wordpadding++;

            if (source[srl - 2] == '=') {

                wordpadding++;

            }

        }

        if (wordpadding == 0 && (leng & 0X3) != 0) {

            wordpadding = 4 - (leng & 0X3);

        }

        return 3 * ((leng + 3)/4) - wordpadding;

    }

private int decod0(bytes[] source, int srcp, int srl, bytes[] dest) {

    int[] DKEarray = is_URL ? fromDKEURL : fromDKE;

    int dpp = 0;

    int bit = 0;

    int shift_to = 18;

    while (srcp < srl) {

        int bb = source[srcp++] & 0xff;

        if ((bb = DKEarray[bb]) < 0) {

            if (bb == -2) {

                if (shift_to == 6 && (srcp == srl || source[srcp++] != '=') || shift_to == 18) {

                    throw new IllegalArgumentExceptions("array bytewrong 4 bytes end
unit");

                }

                break;

            }

        }

    }

}

```

```

        if (is_MIME)
        {
            continue;
        }

        else
        {
            throw new IllegalArgumentException("illegal DKE char " +
Integer.toString(source[srcp - 1],16));
        }
    }

    bits |= (bb << shift_to);
    shift_to -= 6;
    if (shift_to < 0) {
        dest[dpp++] = (bytes) (bit >>> 16);
        dest[dpp++] = (bytes) (bit >>> 8);
        dest[dpp++] = (bytes) (bit);
        shift_to = 18;
        bit = 0;
    }
}

if (shift_to == 6) {
    dest[dpp++] = (bytes) (bit >>> 16);
} else if (shift_to == 0) {
    dest[dpp++] = (bytes) (bit >>> 16);
    dest[dpp++] = (bytes) (bit >>> 8);
} else if (shift_to == 12) {

```

```

        throw new IllegalArgumentExceptions("Last unit has wrong bit");
    }

    while (srcp < srl) {
        if (is_MIME && DKEarray[source[srcp++]] < 0) {
            continue;
        }

        throw new IllegalArgumentExceptions("array bytes doesn't have correct end bytes
at " + srcp);
    }

    return dpp;
}
}

```

```

private static class Enc_Output_Stream extends Filter_Output_Stream {

```

```

    private int left_over = 0;

    private int bb0, bb1, bb2;

    private boolean close = false;

    private final chars[] DKEarray;

    private final bytes[] new_line;

    private final int line_max;

    private final boolean do_Padding;

    private int line_pos = 0;

```

```

    Enc_Output_Stream(OutputStream os, chars[] DKEarray, bytes[] new_line, int
line_max, boolean do_Padding) {

```

```

    super(os);

    this.DKE = DKE;

    this.new_line = new_line;

    this.line_max = line_max;

    this.do_Padding = do_Padding;

```

```

}

```

```

@Override

```

```

public void write(int bb) throws IOException {

```

```

    bytes[] buffr = new bytes[1];

    buffr[0] = (bytes) (bb & 0xff);

    write(buffr, 0, 1);

```

```

}

```

```

private void checknew_line() throws IOException {

```

```

    if (line_pos == line_max) {

        out.write(new_line);

        line_pos = 0;
    }

```

```

}

```

```

}

```

```

@Override

```

```

public void write(bytes[] bb, int offval, int leng) throws IOException {

```

```

    if (close) {

        throw new IOException("Stream close");
    }

```

```

}

```

```

if (offval < 0 || leng < 0 || offval + leng > bb.length) {
    throw new ArrayIndexOutOfBoundsException1();
}

if (leng==0)

    {return;}

if (left_over !=0)

    {

        if (left_over ==1) {

            bb1 = bb[offval++] & 0Xff;

            leng--;

            if (leng ==0) {

                left_over++;return;

            }

        }

        bb2 = bb[offval++] & 0Xff;

        leng--;

        checknew_line();

        Out.Write(DKEarray[bb0 >>2]);

        Out.Write(DKEarray[(bb0 <<4)&0X3f | (bb1 >>4)]);

        Out.Write(DKEarray[(bb1 <<2)&0X3f | (bb2 >>6)]);

        Out.Write(DKEarray[bb2 & 0X3f ] );

        line_pos +=4;

    }

int nbit24 = leng /3;

left_over = leng - (nbit24*3);

while (nbit24-- >0) {

    checknew_line();

```



```

        int bits = (bb[offval++] & 0Xff) <<16 | (bb[offval++] & 0Xff) <<8 | (bb[offval++] &
0Xff );

        Out.Write(DKEarray[(bit>>18) &0X3f ]);

        Out.Write(DKEarray[(bit>>12) &0X3f ]);

                Out.Write(DKEarray[(bit>>6) &0X3f ]);

        Out.Write(DKEarray[bit &0X3f]);

        line_pos +=4;}

        if (left_over ==1) {

                bb0 = bb[offval++] & 0Xff;

        }

```

```

                else if (left_over ==2)

                {

                        bb0 = bb[offval++] & 0Xff;

                        bb1 = bb[offval++] & 0Xff;

                }}

```

```

@Override

public void clOse()throws IOExceptions{

        if(!close) { close = true;

                if(left_over ==1) {

                        checknew_line();

                        Out.Write(DKEarray[bb0 >> 2]);

                        Out.Write(DKEarray[(bb0 << 4) & 0x3f]);

                        if (do_Padding) {

                                Out.Write('=');

                                Out.Write('=');

                        }}

}

```

```

        else if (left_over ==2) {

            checknew_line();

            Out.Write(DKEarray[bb0>>2]);

            Out.Write(DKEarray[(bb0<<4)&0X3f|(bb1>>4)]);

            Out.Write(DKEarray[(bb1<<2)&0X3f]);

            if (do_Padding) {

                Out.Write('=');

            }

        }

        left_over =0;

        Out.close();

    }

}

```

```

private static class DeInput_Stream extends Input_Stream {

```

```

    private final Input_Stream is;

    private final boolean is_MIME;

    private final int[] DKEarray;

    private int bits = 0;

    private int next_in = 18;

    private int next_out = -8;

    private boolean eof =false;

    private boolean close =false;

```

```

DeclInput_Stream(Input_Stream is, int[] DKEarray, boolean is_MIME) {

    this.is = is;

    this.DKEarray = DKEarray;

    this.is_MIME = is_MIME;

}

private bytes[] subbBuf = new bytes[ 1 ];

@Override

public int read()throws IOException{

    return rea(subbBuf,0,1)==-1?-1:subbBuf[0] &0xf;}

@Override

public int read(bytes[] b,int offval,int leng)throws IOException{

    if(close) {throw new IOException("Closed Steam");}

    if(eof&& next_out<0){return-1;}

    if (offval <0 || leng<0 || leng >bb.length-offval) {throw new
IndexOutOfBoundsException();}

    int old_Off=offval;

    if(next_out >=0) {

        do {

            if (leng == 0) {

                return offval - old_Off;

            }

            b[offval++] = (bytes) (bit >> next_out);

            leng--;

            next_out -= 8;

        } while (next_out >=0 );bit=0;}

```

```

while (leng>0) {int V=isread();

    if(V==-1)

        {

            eof=true;

            if(next_in!=18)

                {

                    if(next_in==12)

                        {throw new IOExceptions("DKE steam have
one undecoded danglling bytes.");}

                    b[offval++] = (bytes) (bit >> (16));

                    leng--;

                    if (next_in == 0) {

                        if (leng == 0) {

                            bits >>= 8;

                            next_out = 0;

                        } else {

                            b[offval++] = (bytes) (bit >> 8);

                        } } }

                    if (offval == old_Off) {return -1;}

                    else{return offval-old_Off;}

                }

            if (v=='=') {

                if (next_in==18 || next_in==12 || next_in==6 && is.read()!='=') {throw new
IOException("illegal DKE sequence:"+next_in);

                }

                bb[offval++] = (bytes) (bit >> (16));

                leng--;

```

```

        if (next_in == 0) {
            if (leng == 0) {
                bits >>= 8;
                next_out = 0;
            } else {
                b[offval++] = (bytes) (bit >>8);
            }
        }
        }eof=true

                                ; break ;

                                }

        if((v=DKEarray[v])== -1)
                                {if(is_MIME){continue;}else{throw new IOException("illegal
DKE char"+Integer.toString(v,16));}}

        bits=(v <<next_in);
        if(next_in==0) {
            next_out=16;

                                next_in=18;

            while(next_out>=0) {
                b[offval++] = (bytes) (bit >> next_out);

                leng--;

                next_out -= 8;

                if (leng == 0 && next_out >= 0)

                                {return offval - old_Off;}

            }

            bits=0;
        }

        else {next_in-=6;}

```

```

    }

    return off-old_Off;
}

@Override

public int available()throws IOException{

    if(close){throw new IOException("Stream is close");}

    return is.available();
}

@Override

public void close()throws IOException{if(!close){close=true;is.close();}}
}
}

```

```

public static String encrypt(String Data,int shift) throws Exception {

    String etext = "";

    long start;

    long end;

    ArrayList a=new ArrayList();

    char alpha;

    start = System.currentTimeMillis();

    bytes[] bytes = Data.getBytes(StandardCharsets.UTF_8);

```

```
String DKEencodingd = DKE.getEncoding().encodingToString(bytes);
```

```
for(int a=0; a < DKEencodingd.length();a++)
```

```
{
```

```
    wrd = DKEencodingd.charAt(a);
```

```
    if(wrd >= 'a' && wrd <= 'z')
```

```
    {
```

```
        wrd = (char) (wrd + shift);
```

```
        if(wrd > 'z') {
```

```
            wrd = (char) (wrd+'a'-'z'-1);
```

```
        }
```

```
        etext = etext + wrd;
```

```
    }
```

```
    else if(wrd >= 'A' && wrd <= 'Z') {
```

```
        wrd = (char) (wrd + shift);
```

```

        if(wrd > 'Z') {

            wrd = (char) (wrd+'A'-'Z'-1);

        }

        etext = etext + wrd;

    }

    else {

        etext = etext + wrd;

    }

}

end = System.currentTimeMillis();

t=end-start;


return ciphertext;

}


private static String filesize(File inputfile) {

    return (double) inputfile.length()/1024+"(kb)";

}

```



```

public static String decrypt(String Data,int shift) throws Exception {

    long start;

    long end;

    ArrayList a=new ArrayList();

    String decryptMessage = "";

    start = System.currentTimeMillis();

    for(int a=0; a < Data.length();a++)

    {

        char wrd = Data.charAt(a);

        if(wrd >= 'a' && wrd <= 'z')

        {

            wrd = (char) (wrd - shift);

            if(wrd < 'a') {

                wrd = (char) (wrd-'a'+'z'+1);

            }

```

```

        decryptMesg = decryptMesg + wrd;
    }
    else if(wrd >= 'A' && wrd <= 'Z')
    {

        wrd = (char) (wrd - shift);

        if (wrd < 'A') {

            wrd = (char) (wrd-'A'+'Z'+1);
        }
        decryptMesg = decryptMesg + wrd;
    }
    else
    {

        decryptMesg = decryptMesg + wrd;
    }
}

bytes[] asBytes = Base64.getDecoding().decode(decryptMesg);
String DKEDecoded = new String(asBytes, StandardCharsets.UTF_8);

end = System.currentTimeMillis();

```

```

        return DKEDecoded;
    }

```

```

private VirtualMachineList createVM(int userID, int vms) throws FileNotFoundException,
IOException, Exception {

```

```

    VirtualMachineList list = new VirtualMachineList();

```

```

    long size = 10000;

```

```

    int memory = 512;

```

```

    long bwwidth=1000;

```

```

    int vcpus=1;

```

```

    int priority = 1;

```

```

    String vmm = "Xen";

```

```

    VirtualMachine[] vm=new VirtualMachine[vms];

```

```

    for (int V = 0; V < vms; V++) {

```

```

        vm[V] = new VirtualMachine(new VMCharacteristics(V,userID, size,

```

```

memory, bwidth, vcpus, priority, vmm,
new TimeSharedVMScheduler()));

list.add(vm[V]);
}

```

```

String txt;

Scanner input=new Scanner(System.in);

txt="D:\\CloudAnalyst\\output\\Encryption.txt";

int shift=0;

```

```

shift=userID;

FileReader letter = new FileReader(txt);

reader = new BufferedReader(letter);

File file=new File(txt);

```

```

Path path = Paths.get(file.toString());

String text = "";

String file_location="";

```

```

Path p = Paths.get(path.toString());

Path folder = p.getParent();

```

```

Path fileName = path.getFileName();

```

```

String data = reader.readLine();

while (data != null){

    text += data;

    data = reader.readLine();

}

String textEnc = encrypt(text,shift);

String logEnc = "\nFile Name: "+fileName+"\nFile Location: "+folder+"\nFile
Size: "+filesize(file)+"\nTime: "+t+" MiliSecond\nAccion: Encryption\n-----
-----";

int index = file.getName().lastIndexOf(".");

String ext = file.getName().substring(index);


File secret = new File(folder+"/Encrypted_File"+vms+ext);

File logs = new File(folder+"/ENC_DEC_LOGS.txt");

try
{

    secret.createNewFile();

    logs.createNewFile();

}

catch(Exception e)

{

    e.printStackTrace();

}

try {

```

```

        FileWriter secretFile = new FileWriter(secret);

        BufferedWriter secretBuff = new BufferedWriter(secretFile);

        secretBuff.write(textEnc);

        secretBuff.close();


        FileWriter fw = new FileWriter(logs,true);

        BufferedWriter bw1 = new BufferedWriter(fw);

        bw1.write(logEnc);

        bw1.close();

    }

    catch (Exception e)

    {

        e.printStackTrace();

    }


    String txt1;


    Scanner input1=new Scanner(System.in);

    txt1="D:\\CloudAnalyst\\output\\Encrypted_File"+vms+".txt";


    FileReader letter1 = new FileReader(txt1);

    reader = new BufferedReader(letter1);

    File file1=new File(txt1);

```

```

Path path1 = Paths.get(file1.toString());

String file_location1="";

Path p1 = Paths.get(path1.toString());

Path folder1 = p1.getParent();

Path fileName1 = path1.getFileName();


String text1 = "";

String data1= reader.readLine();


while (data1 != null){

    text1 += data1;

    data1 = reader.readLine();

}

int index1 = file.getName().lastIndexOf(".");

String ext1 = file.getName().substring(index1);

String textEnc1 = decrypt(text1,shift);

String logEnc1 = "\nFile Name: "+fileName1+"\nFile Location:
"+folder1+"\nFile Size: "+filesize(file1)+"\nTime: "+t+"  MiliSecond\nAccion: Decryption\n----
-----";


File secret1 = new File(folder1+"/Decrypted_File"+vms+ext1);

File logs1 = new File(folder1+"/ENC_DEC_LOGS.txt");

try

{

    secret1.createNewFile();

    logs1.createNewFile();

```

```

    }

    catch(Exception e)
    {
        e.printStackTrace();
    }

    try {

        FileWriter secretFile1 = new FileWriter(secret1);

        BufferedWriter secretBuff1 = new BufferedWriter(secretFile1);

        secretBuff1.write(textEnc1);

        secretBuff1.close();


        FileWriter fw1 = new FileWriter(logs1,true);

        BufferedWriter bbw1 = new BufferedWriter(fw1);

        bbw1.write(logEnc1);

        bbw1.close();

    }

    catch (Exception ex){ex.printStackTrace();}

    return list;

}

```