ANKARA YILDIRIM BEYAZIT UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

PERFORMANCE EVALUATION OF IOT DATA SECURITY ON CLOUD COMPUTING

M.Sc. Thesis by

Kübra ASLAN

Department of Computer Engineering

September, 2019

ANKARA

PERFORMANCE EVALUATION OF IOT DATA SECURITY ON CLOUD COMPUTING

A Thesis Submitted to

The Graduate School of Natural and Applied Sciences of

Ankara Yıldırım Beyazıt University

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering,

Department of Computer Engineering

by

Kübra ASLAN

September, 2019

ANKARA

M.Sc. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **PERFORMANCE EVALUATION OF IOT DATA SECURITY ON CLOUD COMPUTING"** completed by **KÜBRA ASLAN under the supervision of ASST. PROF. DR. AHMET ERCAN TOPCU** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Ahmet Ercan TOPCU

Supervisor

Asst. Prof. Fahreddin Şükrü TORUN

Jury Member

Asst.Prof. Ali Osman ÇIBIKDİKEN

Jury Member

Prof. Dr. Ergün ERASLAN

Director

Graduate School of Natural and Applied Sciences

I hereby declare that, in this thesis which has been prepared in accordance with the Thesis Writing Manual of Graduate School of Natural and Applied Sciences,

- All data, information and documents are obtained in the framework of academic and ethical rules,
- All information, documents and assessments are presented in accordance with scientific ethics and morals,
- All the materials that have been utilized are fully cited and referenced,
- No change has been made on the utilized materials,
- All the works presented are original,

and in any contrary case of above statements, I accept to renounce all my legal rights.

Date: …./09/2019 **Signature:……..**

Name & Surname: Kübra ASLAN

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor, Assist. Prof. Dr. Ahmet Ercan TOPCU for his tremendous support and motivation during my study. His immense knowledge and precious recommendations constituted the milestones of this study. His guidance assisted me throughout my research and while writing this thesis.

2019, … September **Kübra ASLAN**

PERFORMANCE EVALUATION OF IOT DATA SECURITY ON CLOUD COMPUTING

ABSTRACT

In today's technology, the fact that smart devices are in communication with each other and with the internet environment makes the Internet of Things (IoT) one of the most popular concepts of recent times. With the emergence of this concept, objects can communicate easily with each other, but this flexibility causes data to increase, and that increase results in a space problem. Although no installation is required for the IoT, the cloud platform, which collects powerful requirements, such as software, hardware, and data storage under a single structure on the internet and offers flexibility to work from anywhere, has become very attractive due to these qualities. The system that stores many applications, programs, and data over virtual servers is called cloud computing.

In this study, it was emphasized that the data sent from the smart devices to the cloud and the data sent from the cloud would be sent from a secure place, and accessed in the same way, when necessary. In this sense, the data to be sent is processed through the encryption module and the data is stored in the storage in the cloud through a number of transactions. The encrypted version of the same data is deciphered from the cloud and the original version is provided to the user safely. The advanced encryption standard (AES), data encryption standard (DES) and Rivest-Shamir-Adleman (RSA) algorithms were used for encryption purposes. An application is developed and there is AES, DES and RSA in it for encryption purposes.The mentioned encryption is provided with the choosed algorithms. The algorithm choosen by client is provided the mentioned encryption. A simulation is created for the purpose of taking the IoT data. A performance evaluation was made by taking different dimensions from the Internet of Things captured from a simulator. Although the RSA algorithm could encrypt very small data, the AES and DES were also more performed in terms the large data size. Finally, this encrypted data was stored in the Google cloud platform (GCP). The existing data that was stored was then withdrawn from the cloud and decoded using these algorithms.Consequently, secure transmission for IoT data to cloud is performed.

Keywords: Cloud computing, security, Internet of things, data security, cryptology.

BULUT BİLİŞİMİ İLE NESNELERİN İNTERNETİNİN VERİ GÜVENLİĞİNİN BULUT BİLİŞİMİ ÜZERİNDE PERFORMANS DEĞERLENDİRMESİ

ÖZ

Günümüz teknolojisinde akıllı cihazların hem birbirleriyle hem internet ortamıyla iletişim içerisinde olmaları son zamanların gözde kavramlarından "Nesnelerin İnterneti"ni ön plana çıkarmaktadır. Bu kavramın ortaya çıkması ile birlikte Nesneler birbirleri ile rahat bir şekilde haberleşebilmekte fakat bu esneklik, veri artışına ve veri artışı ise alan sorununa sebep olmaktadır. Nesnelerin İnterneti için herhangi bir kurulum gerektirmemekle birlikte yazılım, donanım ve veri depolama gibi güçlü gereklilikleri tek bir yapı altında internet ortamında toparlayan ve bununla birlikte esnek bir şekilde her yerden çalışma olanağı sunan bulut platformu, bu nitelikleri ile oldukça cazip hale gelmiştir. Sanal sunucular aracılığıyla birçok uygulama, program ve verilerin depolandığı sistem, bulut bilişim olarak adlandırılmaktadır.

Bu çalışmada akıllı cihazlardan buluta gönderilen ve buluttan temin edilen veriler için güvenli bir şekilde gönderilen yerden çıkıp, gerektiğinde yine aynı şekilde ulaşılması konusu üzerine durulmuştur. Bu anlamda gönderilecek veri şifreleme modülünden geçip buluta şifreli halde gönderip saklanmıştır. Aynı verinin şifreli hali ise buluttan çekilerek deşifre edilmiş, orijinal halinin kullanıcı güvenli bir şekilde ulaşması sağlanmıştır. Şifreleme amacıyla bir uygulama geliştirilmiş olup uygulamada AES, DES and RSA algoritmalarına yer verilmiştir. Nesnelerin İnterneti verilerini alabilmek amacıyla bir simülatör tasarlanmış ve yakalanan Nesnelerin İnterneti verilerinden farklı boyutlarda alınarak bir performans değerlendirmesi yapılmıştır. Bu anlamda RSA algoritması oldukça küçük verileri şifreyebilmesine karşın AES and DES daha büyük boyutlu dosyaları şifreleyebilme anlamında oldukça başarılı ve hızlıdır. Son olarak şifrelenmiş olan bu veriler Google Bulut Platform (GCP)'a iletilmiştir. Sonrasında ise saklanan mevcut veriler buluttan çekilerek bahsi geçen algoritma ile bu kez deşifre edilmiştir. Neticede, Nesnelerin İnterneti verilerinin buluta güvenli akışı sağlanmış olur.

Anahtar Kelimeler: Bulut bilişim, güvenlik, nesnelerin interneti, veri güvenliği, kriptoloji.

CONTENTS

NOMENCLATURE

Acronyms

LDAP:	Lightweight Directory Access Protocol
SSO:	Single Sign-On
KVM:	Kernel-Based Virtual Machine
SQS:	Simple Queue Service (Amazon)
SNS:	Simple Notification Service (Amazon)
RESTful:	Representational State Transfer
API:	Application Programming Interface
SaaS:	Software as a Service
PaaS:	Platform as a Service
IaaS:	Infrastructure as a Service
EMC:	Electromagnetic compatibility
HP:	Hewlett-Packard
IBM:	International Business Machines
FS:	File System
QoS:	Quality of Service
IoT:	Internet of Things
GCP:	Google Cloud Platform
AWS:	Amazon Webs Services
AES:	Advanced Encyrption Standart
VM:	Virtual Machine

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1. Overview

As technology continues to advance, the amount of data within the internet environment also continues to increase at an extremely fast rate. Through the usage of smart devices, not only people, but also things, are able to communicate with each other. This brings us to a term called the Internet of Things (IoT), which comprises an extreme amount of data. Data storage for the IoT can be expanded using the cloud environment, through which all of these things can be connected. Cloud platforms are critical for the IoT. Without cloud platforms, the IoT would not be able to send and receive data. Cloud platforms are highly convenient for storing large amounts of data and that data is accessible to anyone from anywhere. Due to the fact that the cloud platform provides a large area for data storage and it is also managed through an application, it is commonly preferred. However, not only are these data reserved visibly in the cloud storage area, but they are also encrypted in a special format that is especially important when considering unexpected situations, such as security. For this special encryption, algorithms such as the data encryption standard (DES), advanced encryption standard (AES), Blowfish, Diffie-Helman (DH), Rivest-Shamir-Adleman (RSA) are used. Keeping data in both the in-view and encrypted versions in the cloud reduces the risks that may arise as the result of security flaws.

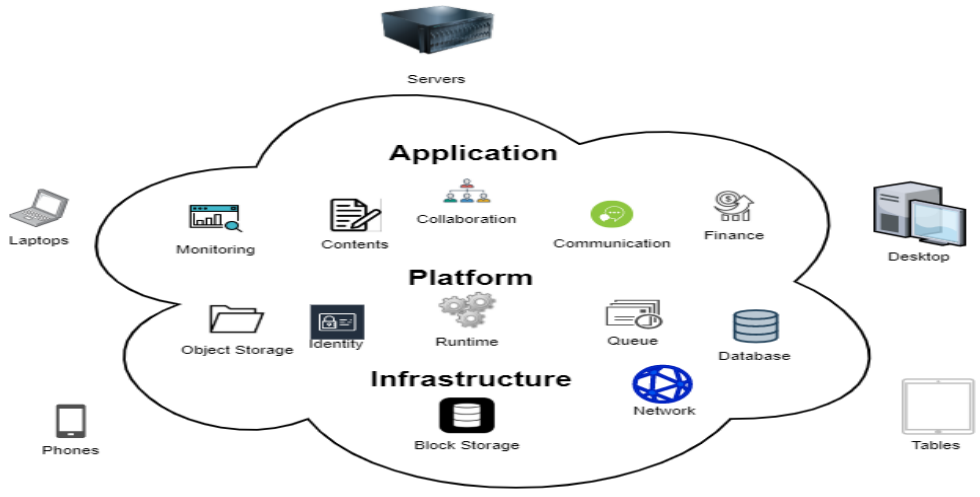


Figure 1.1 Cloud computing environment

A cloud service provider makes source, application, and storage services available over the Internet. The structural parts of it comprise the client, application, platform, and infrastructure. The cloud platform is a floor design in which cloud applications are developed. It includes options that offer cloud software and processing units to users in the cloud architecture. It enables the complexity of each organization's ability to import and manage hardware and software, and the deployment of remote applications. The servers of a cloud platform comprise a layer that includes power units, security equipment, and business processes, such as installation, maintenance, support, and data center management. The infrastructure makes use of virtualization technology. Data is transferred by passing it through different stages on its server, which is located at the bottom of the hierarchy in the cloud computing structure. This study focuses on sending encrypted IoT messages through the virtual machine (VM) on the Google cloud platform (GCP), where it is encrypted and moved to the GCP storage to provide IoT data security, and then receiving the encrypted data from the GCP storage back through the VM after it has been decrypted.

1.2. Research Objectives

We want to provide data security on Cloud Platform for IoT. For that reason, we buid a structure to ensure data transmission securely, between client and server where are all on Cloud Environment. Original IoT data is processed on the client-side by going into security module. With the transaction, the IoT data is reserved on Cloud Platform. When the client wants to use the data, it retrieves from the platform and it is reoperated by the security module. Lastly, performance is evaluated to operation type and size. As a result, we analyse that data can reach, securely to the Cloud Platform and performance to different operations is viewed.

1.3. Structure of The Thesis

The study is composed of five chapters;

The first chapter is the introduction chapter which includes the overview, research objectives, the second chapter is about literature review and Technologies. The third chapter includes data which is used by IoT in the thesis.

The fourth chapter explains encryption algorithms which are symmetric such as Advanced Encryption Standard (AES), Data Encryption Standard (DES), Blowfish, and asymmetric such as Rivest-Shamir-Adleman (RSA) and Diffie-Helman (DH).

The fifth chapter includes methodology and implementation. Lastly, the sixth chapter is about conclusion and future work of the study.

CHAPTER 2

LITERATURE REVIEW AND TECHNOLOGIES

2.1. Theorical Background

Herein, the 3 most important layers of the IoT are analyzed with regards to the security of each one. These layers are: 1) the perception layer, which involves wireless sensor network (WSN) security and robust security network (RSN) security; 2) the transportation layer, which covers access network security, core network security, and local network security, and third generation (3G) access network security, which includes ad-hoc network security, Wi-Fi security, and so on for these sublayers; and 3) the application layer, which contains the application support layer and specific IoT applications. This paper focuses specifically on the security of the perception layer. There are some security issues with the perception layer for WSNs and heterogeneous integration like RSNs. The WSN's solutions to security problems are: 1) cryptographic algorithms, to connect securely to all network services; 2) key management, to provide access securely for transportation and distribution to legitimate users; 3) secure routing protocols, to prevent the network from any attacks related these protocols; and 4) trust management of the nodes, to reduce the negative aspects of the WSN's resources, which include sensor nodes, easy capture of the nodes, and a unique communication mode. Thus, the solution to heterogeneous integration is RSN technology, which contains WSNs to acquire some common attributes, such as storage format, information access format, and application direction [1–2].

The second layer of the IoT, the transportation layer, can be analyzed functionally as the access network and it has some security issues, such as the Wi-Fi security-a center network, ad-hoc security-a noncenter network, and 3G network security. The core network provides access from the perception layer to the core layer, in which the 6LowPAN technology answers the problem when the Internet Protocol for v4 (IPV4) is insufficient for the IP address to correspond, and there are a lot of sensor nodes and local area to prevent data leakage and the server's independent security deficiencies. The last layer is the application layer, which is an advanced version of the transport

layer and it fulfils all requirements related to business services and computational and resource operations. This layer contains the middleware, machine-to-machine, cloud computing platform: a critical part of IoT security, which is of great importance due to the encrypted data and storage of user data in the cloud, and the service support platform: which adapts to different services and can also separate valid, spam, and malicious data, and filter them immediately [3].

Cloud computing is a general definition of internet-based information services that provides computer sources that can be used at anytime and are shared among users on computers and other devices related with the internet. Without users having to set up any programs, they have the chance or store data on the web. One of the services of cloud computing is mobile cloud computing (MCC), which provides an application via mobile phones. Since the cloud model has some properties like on-demand, self-service means that the cloud provides resources whenever they are wanted; hence, broad network access, resource pooling, rapid elasticity, measured services, mobile computing, and cloud computing are combined. According to mobile users, cloud computing provides big advantages [4]. Although it provides convenience to users, there are some restrictions, like wireless communication on applications. Wireless communication is referred to as technology that is used to access network resources and services, and the communication of mobile devices and clouds, while the operations of the applications and resources, such as process, transportation, sharing, and showing, are realized by mobile computing.

It was proposed by Zhong et al. that [5] that there are countermeasures for MCC, such as the integration of mobile computing, cloud computing, and wireless technology, and it has some security issues, especially with regards to security and privacy Nowadays, mobile devices have many capabilities, such as normal mobile phone use, concurrently supplying internet and apps-gaming, social media networks, sending email, sharing files, etc. Although MCC has many advantages, it is not fully competent with respect to safety-related issues, due to the fact that it opens a door to the Internet, services, and clouds. Wireless communication fulfills the needs of MCC by supplying network resources and services to users, and communication between mobile devices and clouds. The same study also listed the 6 essential MCC properties, which included

breaking through hardware limitations, having suitable data access, intelligent load balancing, efficient task processing, cost effective on demand service, and removing the regional boundary. In addition, they also listed the obvious difficulties, such as limited the resources of mobile devices, stability, availability, costs of network access, scarcity of channel bandwidth, heterogeneity, process offloading, mobility management, context-processing, cloud policies for mobile users, elasticity, application services issues, energy efficiency, ensuring QoS, security, trust, privacy challenges, etc.

This paper proposes a new method that provides a secure data exchange process, wherein from various devices to the cloud environment, or vice versa, a third party compute module, which acts as middleware between 2 different endpoints, covers a few wireless and wired connections, which are used to secure data transfer. To enable adaption of the devices, on the client's side of the application, the improved software starts after the operating system has been fully initiated. The symmetric key, which is much faster and secure, was chosen for encryption. The AES encryption algorithm was developed in 2 parts: AES instruction usage and processing parallel computing. There were 2 phases used to provide encryption and decryption with the proposed solution: AES instruction and processing parallel computing. Both were preinstalled in the compute module and setup in the cloud environment, and used in the software solution. They were compatible by means of IoT data connection security without any request [6].

The paper focuses on the secure data connection from an IoT device to the cloud platform. To provide secure communication between them, a device, as middleware, was improved. The AES was chosen to encrypt the files, due to the fact that it is faster than symmetric encryption algorithms. Device resources are mostly used for encryption. Thus, it cannot be used in the encryption period or for solving the problem; hence, the AES was adapted with some improvements. A method called parallel computing was used to provide increased speed for the AES. This method eases processing by dividing computations onto more processors, synchronously, with just a machine. Without changing the pure algorithm, written code separates full text into parts for encrypting and decrypting freely. Moreover, multiple parts can be run at the

same time. Dividing the original text or cipher parts into seperated streams is the first step, and later, encryption and decryption is performed according to the AES algorithm. The device affects the operation speed, which can be increased by up to 3 times faster [6].

Table 2.1 Comparison of the improved algorithm with the original

Data Size	AES	AES Improved
1 MB	0.8–0.9 seconds	0.6–0.7 seconds
10 MB	7–8 seconds	5–5.5 seconds

The web is basically a huge gathering of organized computers. Man has begun to rely upon the web on a consistent premise and it has been consolidated it into our daily lives. Because of our reliance on the web, fear-based oppressors have made the web a potential assault stage. Security, at this point, focuses on the strategies created to safely protect data and the data framework stored on computers. Potential dangers comprise the devastation of personal computer (PC) equipment, programming, robbery, unapproved use, or revelation of information. PCs and the data they contain are regularly considered a secret framework, on the grounds that their utilization is commonly confined to a set number of clients. That secrecy can be vulnerable to risks in an assortment of ways. For instance, information and data can be uncovered by programmers, infections, and worms. In this way, security can be characterized as the resistivity degree to, or security from harm. Security has been one of our essential needs since creation of computers [7].

Osuolale presented research based on cryptography and steganography [7], wherein it was explained that cryptography really guarantees the classification of the message, while steganography incorporates another level and layer of security by keeping secret the way that mystery correspondence takes place. Today, steganography is regularly connected with abnormal state innovation assortment, where information is hidden inside of other information in a document. For instance, a word archive may be hidden inside of a picture document. This is generally done using concealed information bits to supplant the most repetitive or least significant bits of information, which are not

really missed by the human ear or eye, into the first record bits. Figure 2.1 delineates the straightforward stego framework.

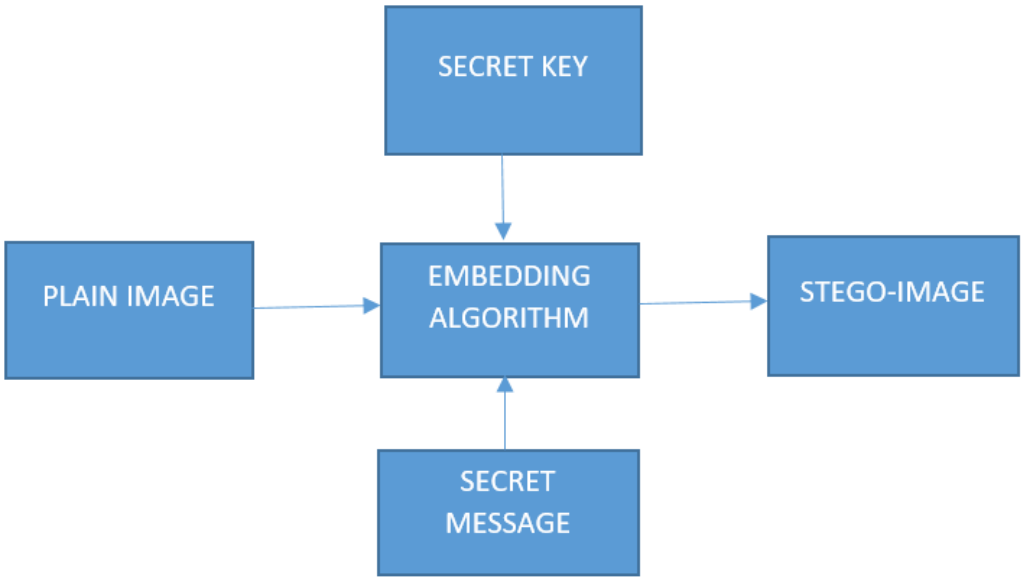


Figure 2.1 Simple stego system

Cryptography and steganography can be utilized to provide information security; however, each of them has an issue. For cryptography, the issue is that the figure content looks pointless, so the assailant will suspend the transmission or make the data more secure from the sender to the party receiving it. For steganography, the issue is that once the concealed data is closer to being known or even suspected, the message becomes closer to being uncovered. Herein, a mixed procedure for information security is proposed that utilizes cryptography and steganography to improve data security. As a starting step, the AES calculation would be remodified and used to scramble a private message. Moreover, the encoded message would be shrouded utilizing a steganographic strategy. Subsequently, 2 degrees of security would be given utilizing the proposed mixture method. In addition, the proposed strategy would provide a high inserting limit and brilliant stego pictures. By consolidating cryptography and steganography, it should be possible to encrypt the information into a framework (Figure 2.2) and afterwards, implant the encoded content into a picture or some other media using a stego key. The blend of these 2 techniques will improve the security of the information installed. This consolidated science will fulfill the

necessities; for example, improve the limit of security and power for secure information transmission over the web [7].

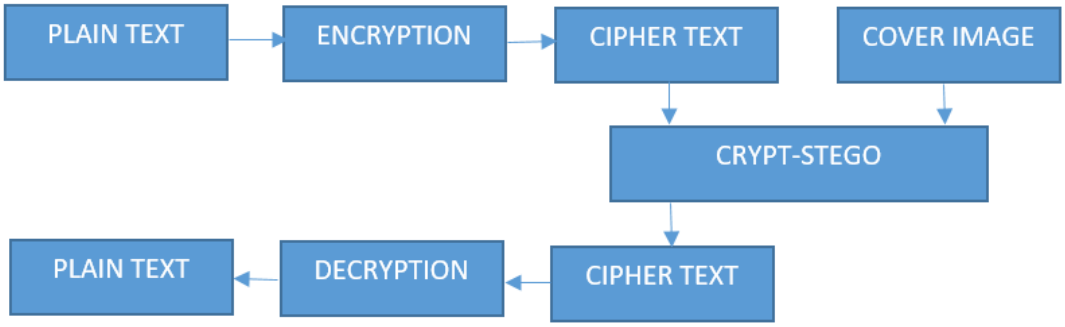


Figure 2.2 Crypto-steganographic system

In this work, 2 methods are combined, cryptography and steganography, which are integrated with the AES and least significant bit algorithms for perfect secrecy. The AES is the most popular symmetric encryption algorithm. It has more efficient performance than the triple DES, owing to the fact that the DES has a very small key size. Clearly, it can be seen that the triple DES is slower than the others and it has a weakness regarding key search attacks. Hence, the AES encryption algorithm is used for encryption and decryption in this paper [7].

Saleh et al. presented a strategy that progressively tied down correspondence by combining cryptography and steganography systems to make it increasingly harder for a steganalyst to recover the plain text of a mystery message from a stego-object [8]. Two sections were serviced within their proposed technique (Figure 2.3). In the initial segment, the AES calculation was adjusted so as to be reasonable for the steganography strategy, and it was named the AES_MPK calculation. In the subsequent part, the AES_MPK calculation was converged with the steganography calculation to shroud the encoded information in a picture, where the message being sent was hidden.

The AES_MPK can undergo various types of transformation, such as substitution as SubBytes, permutation as ShiftRows, MixColumns, and key-adding for security. To arrange the input and output, so as to set it the same as the MPK digits, the AES

algorithm was revised, owing to the fact that the input and output were represented as the hexadecimal version of the digits. Due to the fact that the MPK digits were used to secrete the data by the PVD_MPK and MSLDIP-MPK methods, to enable generation of the input and output in the form of MPK digits, the AES algorithm was modified and renamed the AES_MPK algorithm. The main function of the AES_MPK algorithm is to provide modification to every block of data into MPK digits.

There are 2 phases for every block, which amounts to 16 bytes. The critical point of both states is to change every digit 8 or 9 with 7, to be in an available state for the Galois field (GF) arithmetic operations performed in each part of the encryption. There are also some types of transformations known as substitution, permutation, mixcolumns, and addroundkey [8].

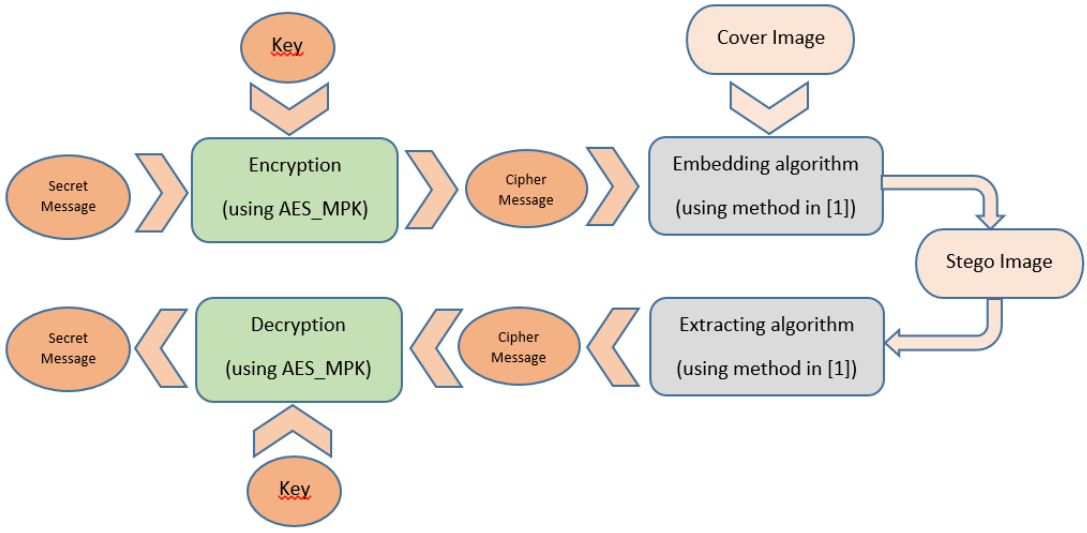


Figure 2.3 Block diagram of the proposed system in [8]

Combining the cryptography and steganography techniques, which are 2 important methodologies for security, results in a secure communication model. Because the secret key is an essential part of the steganography technique, plain text cannot be achieved by the steganalyst for encryption or decryption without it. The AES_MPK provides encryption of the secret key and the PVD-MPK and MSLDIP-MPK methods create the encoded data within the gray image. In addition, the open channel can carry the secret data as encrypted content, and it does not appear useless or suspicious, yet

its quality is hidden in the image utilizing steganography to conceal the figure message. As a result, the proposed strategy is more useful for storing extra information when compared to the other methods and the stego image is advanced with respect to its visual quality, which affects secret data communication [8].

Donno et al. presented their detailed research based on cloud computing security in the advancing era [9]. They did state the security vulnerabilities of the cloud, which are actually a big paradigm; however, no proposed solutions were stated in their work. As the Iot architecture has a core security component, the main purpose of their work was to present the main security issues within the cloud. They used a stepwise approach, wherein they seperated cloud computing security issues into various parts. Specifically, 2 critical aspects, the cloud architectural layer and the implemented security property, acting on the confidentiality, integrity and authenticity (CIA), were the classified issues. The IoT gains a purpose with the cloud. Thus, a secure IoT environment can provide a secure cloud platform. The cloud-specific issues are: confidentiality, which includes issues related to virtualization, application, and data storage; integrity, which includes the same issues as confidentiality; and availability, which includes issues related to virtualization, application, and the network. On the other hand, there are IoT security issues. One of which is the insufficient updating of smart devices. As a solution to the problem, a blockchain-based privacy-preserving software update protocol was proposed by Zhaoa et al. [10] and Yu et al. [11]. Second, default passwords create vulnerability for many IOT devices. As a solution for this, 3 different approaches for framework design and collecting network data were proposed by Giura and Jim [12], each providing different levels of visibility into IoT device behavior. Finally, sharing data with other devices is another security problem. A solution to the problem was proposed by Kong et al. [13], wherein a location privacy-preserving data sharing scheme collected and distributed the data captured by vehicular sensors.

Vaikunth and Aithal [14] focused their research on accessing online software applications, data storage, and processing power of the system in cloud computing, which provides more available capacity without any investment for new infrastructures, IT personnel training, or expensive software. As the use of cloud

computing in companies expands, personal data belonging to the individuals who work in the organizations is saved in the cloud. And as security issues are a major concern, many users are still unsure about their work being saved in the cloud. Due to the fact that the cloud is directly related to the Internet and millions of users make use of the cloud on the Internet daily, sometimes technical impediments develop all of a sudden, which causes a temporary pause in the workflow. During this period, the users are unable to connect to applications and the server, or send or receive information to or from the cloud. To rectify this situation, Vaikunth and Aithal also put forth the development of data centralization and security on resources to provide security in the cloud. Moreover, they mentioned 3 main development models used on the security side of cloud computing: 1) the private cloud model, which is used by an implemented organization and their employees at the authoritative level, and due management by the organization or a third party, model security control is made versatile and more safe with regards to infrastructure and accessibility; 2) the public cloud model, which is less secure because of the fact that the more information that is opened to the public, the fewer malicious attacks are applied to the system; however, the apparent information that belongs to the individuals or organizations does cause some security-related problems; and 3) the hybrid cloud model, which is a combination of both the public and private cloud models. It provides more secure control regarding data and applications from the viewpoint of the organization's requirements, and is available in its own private cloud, while substantial needs for load-balancing are provided from the public cloud.

Herein, a different solution to the cloud computing security problem is proposed. The solution is about cryptographic methods, which cloud-based environments will not generally use in the near future. This method increases the security of applications that are in the cloud-based environments. They have homomorphic encryption schemes that provide qualified security to applications, although they are required to perform more complex calculations. There are specific security rules in traditional architectures, which result in some drawbacks with regards to implementation, whereas cloud deployment models do not have to abide by these rules; hence, control of the data and application is more easily attained. When the cloud environment is used with a reliable third party, by applying cryptographic methods, the CIA are validated.

Moreover, Kumar and Gobi [15] reported there are some properties of data utilization management that provide cloud security, such as policy management, access control, process validations, query validation, encryption, and decryption. All of these conditions are provided on the client's side, the connection's side, and the server's side.

Another trend solution that they presented was securing the cloud with homomorphic encryption, which requires the use of an encryption and decryption key within the cloud to provide data access efficiently and securely, and keep out malicious insiders. Privacy-preserving data mining is also a service in the cloud, wherein a server can actively check defined rules on the data in the cloud, and the company owner's private information can be dissembled, in its transaction database by breaking it into pieces and giving each of them unique numbers, and then assigning different numbers to different pieces to prevent revelation of the secret information. access control in cloud systems, in addition to distributing the access control, accessing control as a value-added-service, providing reliable credential management efficiency, scalability, and fine-grained solutions. Figure 2.4 shows the cloud environment, which includes a variety of data for applications and operations on the data for computing, in addition to its security requirements [4].

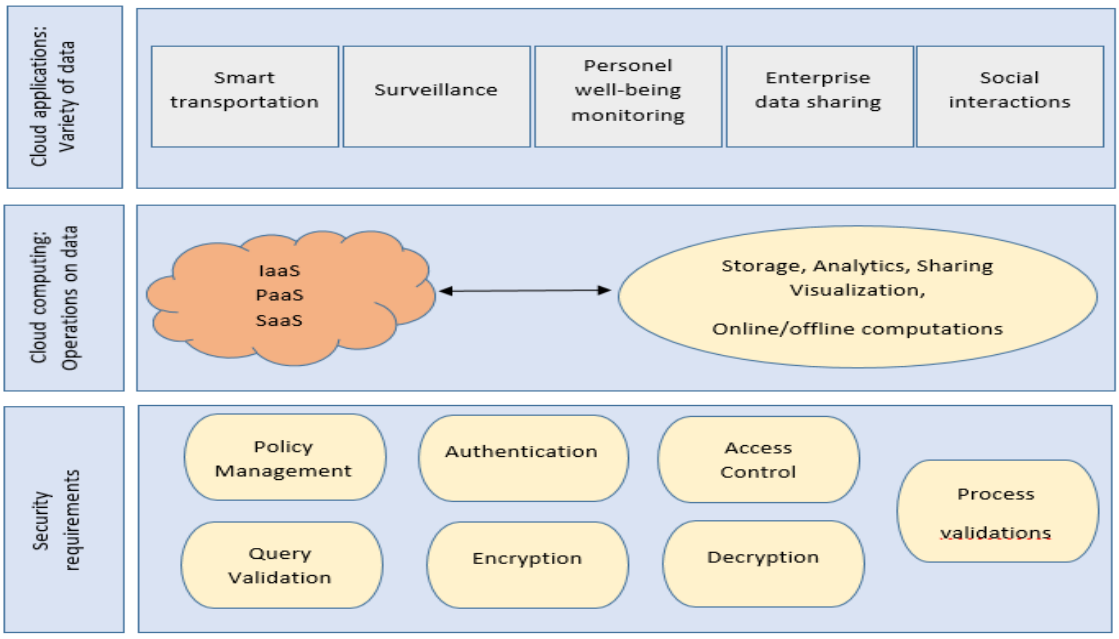


Figure 2.4 Securing the cloud for data utilization management

Day by day, as the social network develops, access controls, data diversity, and data privilege management become more and more difficult to implement in access control models. To solve these problems, Cheng et al. [16] proposed a new social network access control model, which uses a logical authorization language called RuleSN, which is preferred in cloud systems. Their proposal provides more secure communication among user to user (U2U), user to resource (U2R), resource to resource (R2R), and the attributes of its users and resources (Figure 2.5). The side of their research that interests us is their model's authorization specification and verification policies, in addition to the structure of their authorization language.

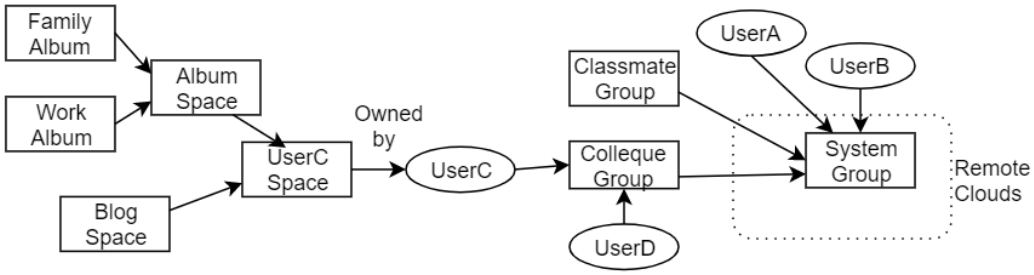


Figure 2.5 Example of a subject hierarchy in the cloud system

There are some difficulties in the traditional model, such as the distributed privilege management and implementation mechanism of the social network, which results in a decentralized structure; different kinds of social network data, such as as documents, images, videos, etc.; management of its own privacy data and the confidential information of users; a big volume; and varied data in the social network. The proposed solutions to these problems presented by Zhong et al. [5] included a merged structure of the social network for cloud systems; effective authorization and comfort in the cloud environment, together with connection between the U2U, U2R, R2R, and attributes of the users and resources to the model; and a preferred authorization language that uses the Datalog logic program, by the restricted logic program (RLP), which basically provides tracking of the authorization.

In their work, Alcaraz et al. [17] researched security integration challenges, while combining them with the security mechanism, services, and data privacy, and specifically, the connectivity of WSNs and the Internet. WSNs for IoT security are

very important. They also mentioned connectivity at the network level when considering the combination of security mechanisms and the users' acceptance. As for data privacy, separation between the data's owner, security, and availability should be selectively implemented only after receiving the user's permission. However, there are some scenarios in which it is necessary to share data as a result of providing a service. Security protocols come into prominence both at the network level and for communication between objects and services. When a WSN connects to the Internet, the connection can be provided by using the stack-based approach, which is based on network stacks of the Internet, and a WSN that is independent from it, as front-end or topology-based classification, which is based on the location of the nodes connected to the Internet. And finally, the transmission control protocol/Internet protocol (TCP/IP) solution can be implemented in the TCP/IP stack by the sensor nodes. When comparing these all of alternatives, the TCP/IP solution is more suitable for integrating the WSN and the Internet. Because the data given by the nodes can be accessed by many kinds of external systems, the nodes that recognize the Internet have the ability for to perform a service query. Finally, the main factors of the TCP/IP solution can be listed as resilience, user authentication and authorization, security of the communication channel, accountability, functionality, hardware, inherent weaknesses, network redundancy, and protocol optimizations.

2.2. TECHNOLOGIES

2.2.1. OPENSTACK TOOL

In 2010, a popular topic, Openstack, which is a cloud-computing platform, was developed by Rackspace cloud and NASA. Cloud computing services became available on standard hardware that was equipped with OpenStack. The first official release code name was Austin. And 4 months following that, the official release launched with the updated version.

The OpenStack cloud-computing platform is preferred because it does not require a licensing fee if a cloud platform is needed for a project. Moreover, a lot of contributions can be provided when known coding and scripting experiences are used. If OpenStack has to be fixed, there are many options as a hypervisor, such as kernel-

based virtual machines (KVMs), Xen, VirtualBox, VMware, or Hyper-V, and OpenStack can be used on those tools. The most interesting aspect of OpenStack is its open-source coding. Due to the fact that the project is available to developers, any desired changes can be made. Moreover, there are many companies like IBM, Cisco, Hewlett-Packard (HP), and DELL that support OpenStack. In addition, a lot of well-known companies have launched their new releases by implementing development on OpenStack. The first preview of the OpenStack release, Essex, was launched by Redhat in 2012. After that, Oracle announced that they were joining OpenStack in 2013 and IceHouse was presented in 2014. The HP Helion product was announced in the same year, and the preview version of the HP Helion OpenStack Community was published. OpenStack became fully workable with VMware vSphere in 2015 [18].

Cloud technology has 3 main services: 1) software as a service (SaaS), which serves software by using hardware infrastructure, and application software in the cloud can be changed or authorized for clients in different groups; 2) platform as a service (PaaS), which serves the platform and includes some different purposeful parts, like application program interfaces (APIs), the operating system, development environments, the programming language execution environment, and web servers, which can be used to design applications on the platform over the Internet; and 3) infrastructure as a service (IaaS), which serves the server infrastructure, VMs, load balancers, raw block storage, and firewalls, and networking services are provided in it, and in addition, housing, running, and maintaining are some of the tasks it can perform [19].

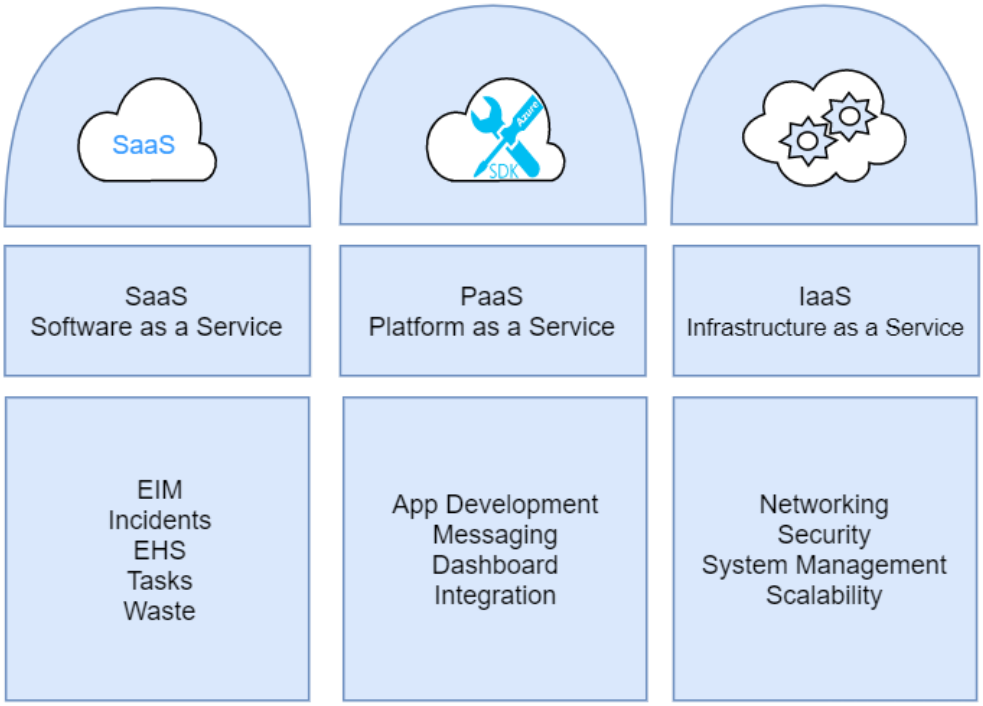


Figure 2.6 Usage, building and moving of the cloud service

There are 4 types of cloud technology that provide usage for different areas (Figure 2.6). The first is a public cloud, called the external cloud, which is a cloud technology setting with servers on the Internet. The second is a private cloud, called the internal cloud, which is preferred by famous associations, as it stores secret information. In the internal cloud, all of the stored information is under the founder's control, and access security and confidentiality are maintained at a high level. As an example of the cloud type, Hyper-V and System Center by Microsoft can be given. The next is a hybrid cloud, called the internal/external cloud, which combines the public and private cloud. These are represented in Figure 2.9. The combination ratio of the total scale can undergo a complete change. The last is the community cloud, which intersects common services in several companies, and all of the members of the community can access the application and data [19].

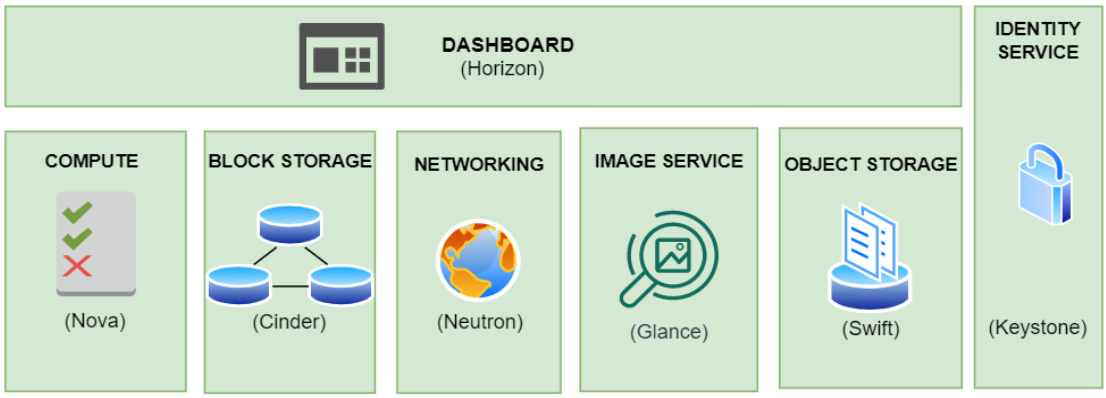


Figure 2.7 OpenStack services

In the OpenStack architecture, there are 3 main services, as shown Figure 2.6. These are: 1) OpenStack identity service, which follows all service installations and has duties like identification, tokens, and cataloging of available services and policies; 2) OpenStack compute service (Nova), which serves the tools of cloud deployment, such as managing the block storage, networking, computing resources, scheduling, authorization, and hypervisors; and 3) OpenStack image service, which includes VM operations, such as the storage, query, and retrieval for the system. Although a little detracted, identity service and compute service can be mentioned here. First, the identity service (Keystone) provides identity management. It resembles the single sign-on (SSO) system in VMware. Keystone can be integrated with the lightweight directory access protocol (LDAP), which provides tracking of both its own specific objects and searching between these objects, or present backend directory services. Next, the OpenStack compute service is a part of IaaS and, moreover, the management of virtualization means that the compute is done. Management is designed for both the server source pool automation and high performance computing, so it can work on the available developed virtualization technology. There are many supporting services, such as KVM, VMware, Xen, Hyper-V, XenServer, etc. One of the most important services working on it is the VM. The OpenStack image service (Glance) provides the management of images on VMs, wherein stored images can be used as a template. Since there is no limit, unlimited images and templates can be kept or used. In Figure 2.7, the OpenStack services are shown with the contents included [20].

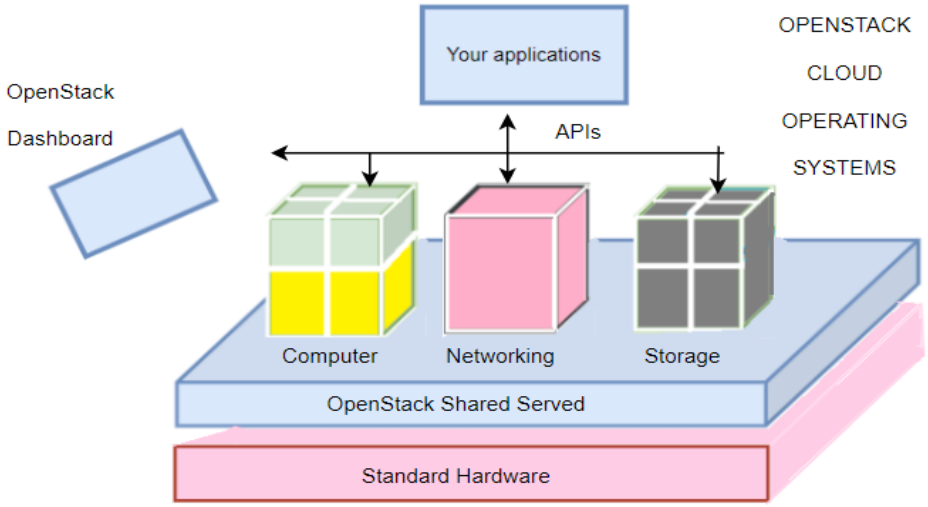


Figure 2.8 OpenStack cloud computing operating system

Moreover, there are some important terms, such as dashboard, multitenant cloud messaging, shared file system, and security API. Dashboard (Horizon) can connect to a graphical interface and then provision operations can be applied. The module includes third-party productions and services (Figure 2.8). This means it has extra management tools like monitoring, billing, etc. There is a convenient structure to customize for branding operations. Multitenant cloud messaging (Zaqar) is a multi-tenant cloud-messaging service for web developers. The service has a representational state transfer-based (RESTful) API structure that allows sending messages between a lot of SaaS components and mobile apps using a kind of communication template for developers, and in addition to that, Amazon has an extra semantic, wherein its simple queue service (SQS) product's pioneering opinions are combined. While the basic API is an efficient messaging motor that is designed with regards to its scalability and reliability, other OpenStack components can be integrated with Zaqar to show events to end users and to connect with visitor agents working within the cloud's upper layer. Zaqar, which was previously known as Marconi, can be used to provide the equality of the SQS and simple notification service (SNS) to customers by cloud operators. The shared file system (Manila) can either be used on its own or in some kind of network environment that allows open an API to manage sharing by the producer within an

unknown scope. There are abilities for sharing the creation, erasure, and access permitting or rejection on a standard basis. Just as some factors, such as electromagnetic compatibility (EMC), NetApp, HP, IBM, Oracle, Quobyte, and Hitachi data systems support commercial storage appliances, the shared file system supports file system technology like Red Hat GlusterFS EMC. And finally, API security (Barbican) is a kind of REST API designed for the reliable storage of secrets, preparation, and management, and it is useful for all environments, including big temporary clouds [21].

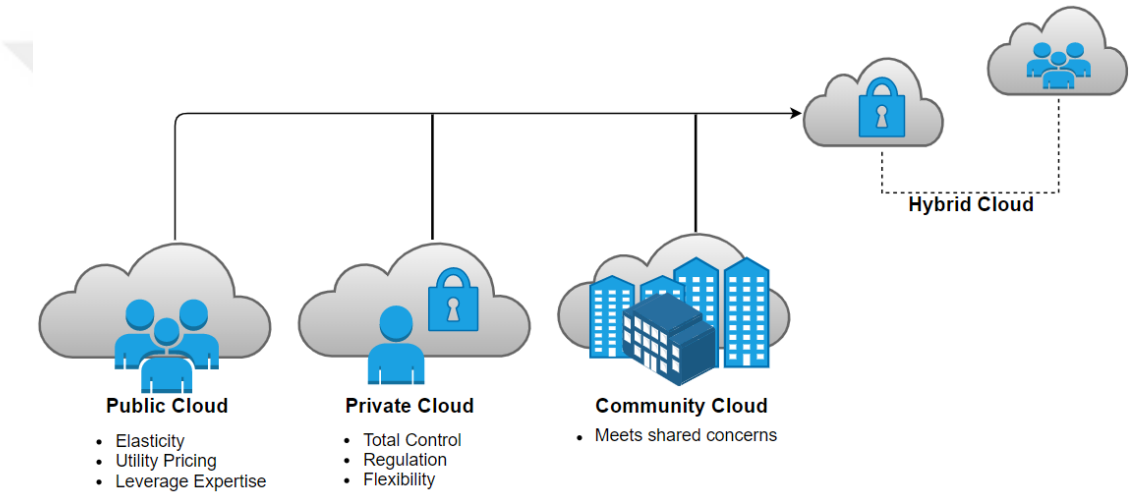


Figure 2.9 OpenStack cloud environment

2.2.2. Amazon Web Services (AWS)

Cloud computing is where the processing power, storage area, application, and other information technology sources are provided for a cost on the Internet. The networked hardware required for these application services is owned and maintained by a cloud service provider, such as Amazon Web Services (AWS), where the resources needed are supplied and used through a web application. The cloud is preferred due to it being agile, flexible, and cost saving.

AWS have the architecture shown in Figure 2.10. The AWS architecture is the most comprehensive and widely used cloud platform in the world and offers over 165 full-

featured services from data centers around the world. Millions of customers, including the fastest growing startups, the largest organizations, and leading government agencies, rely on AWS to strengthen their infrastructure, become more agile, and reduce costs.

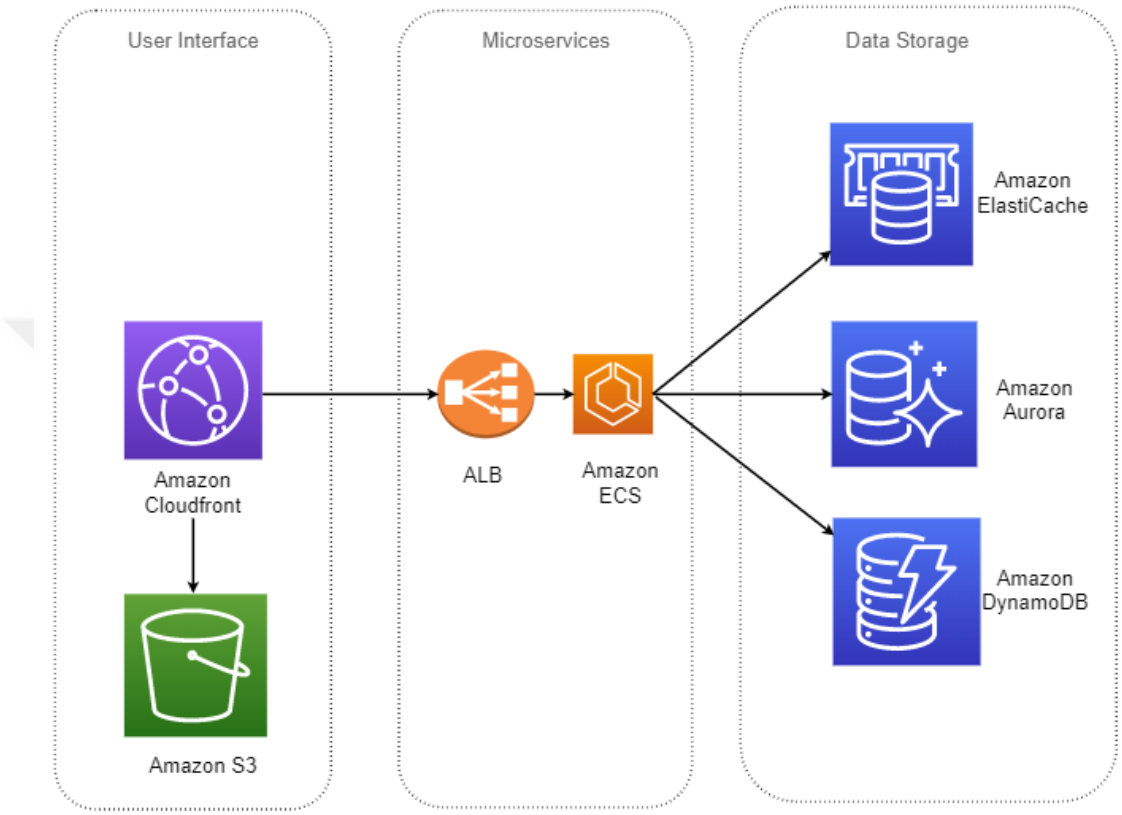


Figure 2.10 The architecture of AWS

AWS is a leader in cloud platforms that provides maximum functionality, and the biggest community consists of customers and solution partners [22].

2.2.3. Google Cloud Platform (GCP)

Google Cloud is a cloud information platform in which Google provides the server infrastructure services used by sites, such as Google search engine and YouTube, to the end user [23]. From simple websites to complex applications, various types of software can be developed through Google Cloud (Figure 2.11). The Google Cloud platform (GCP), which includes servers and computing, cloud storage, data storage,

translation APIs, and prediction APIs, is part of Google's enterprise solution suite, which runs and delivers cloud-based services along with server development tools.

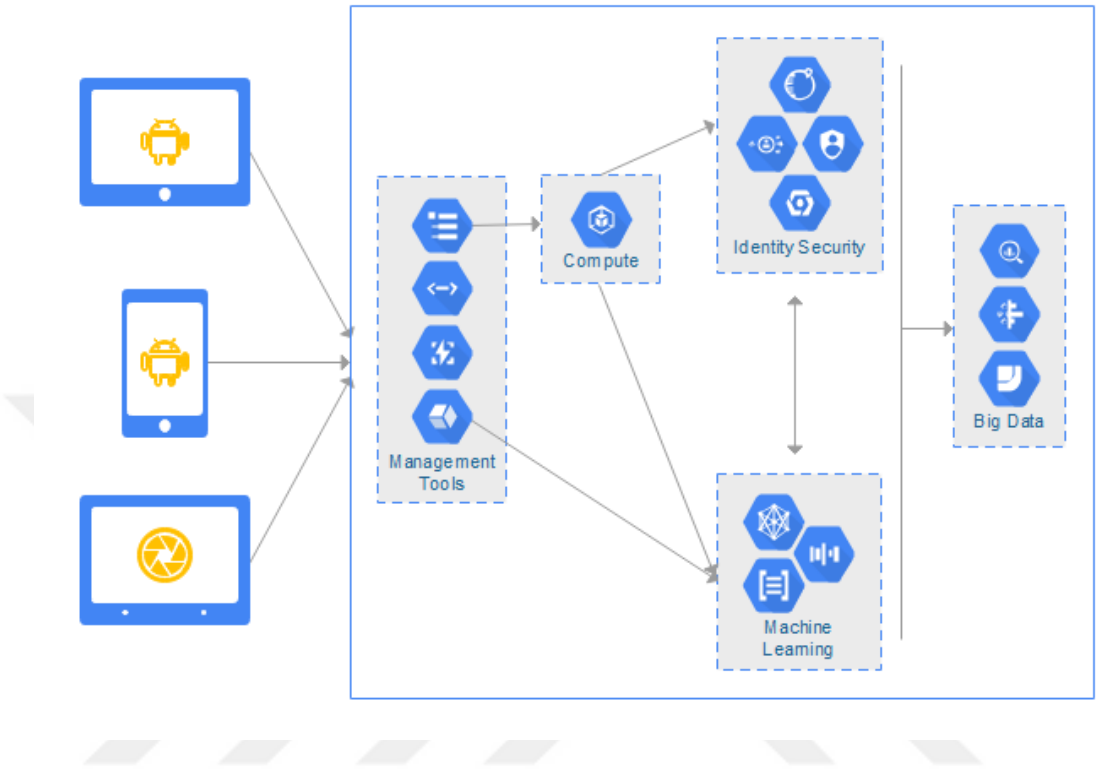


Figure 2.11 The architecture of GCP in [24]

Many useful services are provided on the GCP. One of which is the RESTful API, which is used for storage. Regardless its type, the API provides access to data that are part of the GCP. These data can be photographs, articles, documents, or formatted data taken from the IoT. Within this thesis, the available data used were taken from the IoT simulation by a sensor.

Data can be transferred to the bucket area of the GCP, which is special for every user. Moreover, it can be read from there by functions of the storage API services. The data can be serviced as general or special with some permission. Moreover, the API in the GCP can be used between applications. Hence, when any data are uploaded from an application, they are received by the other application via the Google service. Storage API usage is very easy in terms of understanding data storage logic. All operations can

be performed using the terminal, provided that Google Cloud SDK or Google Shell is installed.

CHAPTER 3

DATA

3.1. DATA

Every IoT object can be thought of as somehow accessing the Internet and communicating with other devices. In other words, IoT is the word ‘smart’ at the beginning of each object. When considered in the history of the IoT, it was first used by Kevin Ashton in 1999. Then, gradually, products with the IoT concept began to develop. It was predicted to be such a growing market that some 26 billion products are expected to be connected to the Internet by 2020 [25].

Smart wristbands, watches, glasses, t-shirts, rackets, home automation systems, cars, and many more can be stated as some examples for the IoT. With investments in objects, all of them will soon be connected to the internet with Wi-Fi and Bluetooth technology. Smart products will also be able to enter and transmit notifications to users via drones, in addition to modems near the Internet.

Developed under the IoT concept, products often work with mobile devices and tablets. Each object has a mobile-accessible app, which allows them to pass notifications to applications. When equipped with objects, sensors, and electronic circuits, they acquire the ability to communicate with people and update status information. With the development of mobile networks and the Internet, these objects have become easier to communicate with, and people have the opportunity to observe and control them from anywhere, anytime.

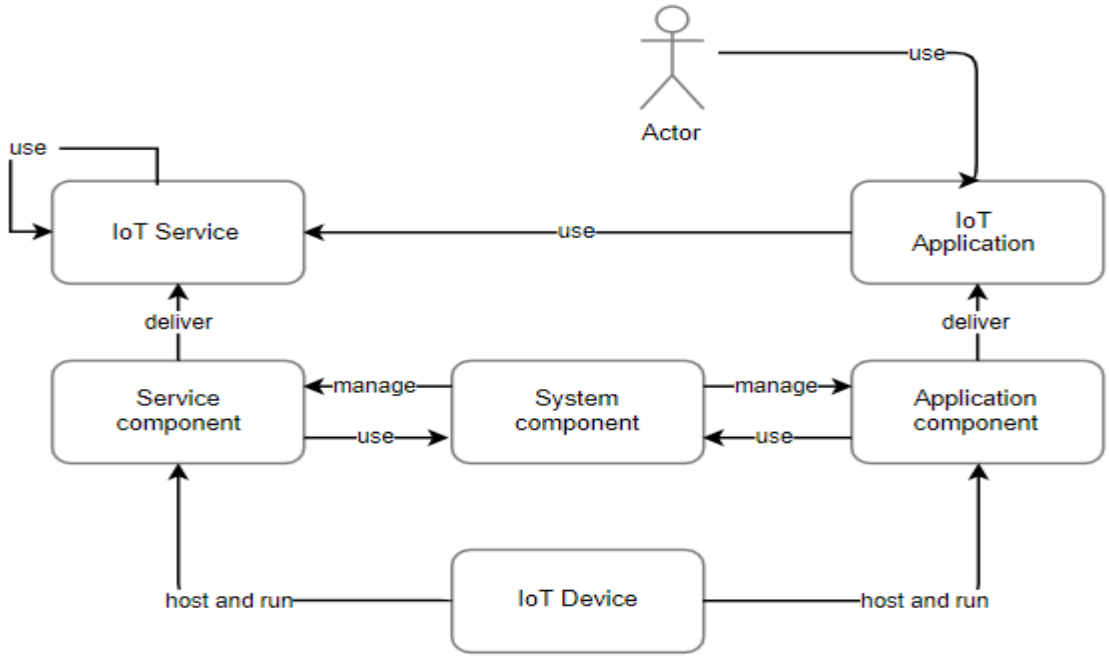


Figure 3.1 User interface with the system through an IOT application

IoT systems harbor the service oriented architecture (SOA). The IoT system comprises 3 major elements: the IoT device, IoT service, and IoT application. Figure 3.1 shows the phases that any of the 3 elements go through, from construction to destruction. It can also be seen that the IoT device has software components that are divided into the system, service, and application partition of it [25].

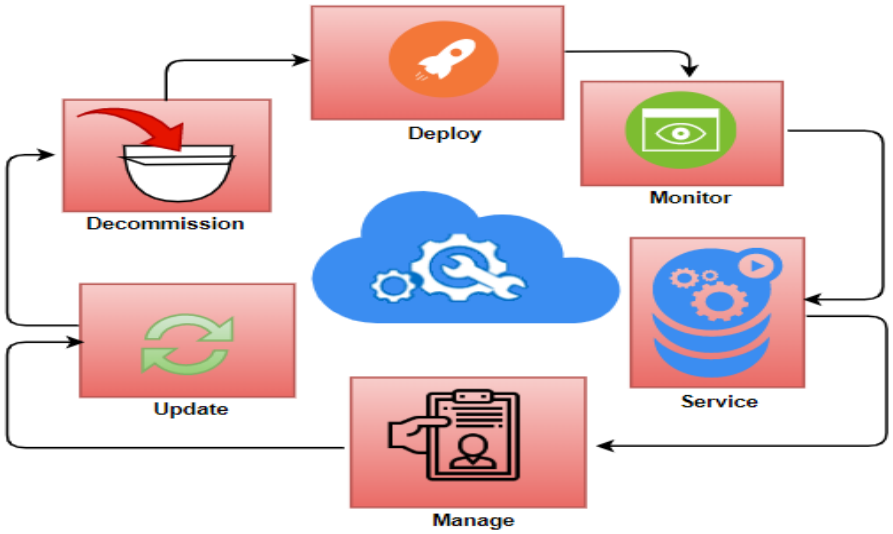


Figure 3.2 Generic life cycle model for IoT devices

The lifecycle of an IoT device is shown in Figure 3.2, where the phases that an IoT device goes through from construction to destruction are summarized. There are 6 main stages in the lifecycle. The deploy stage distributes service components via the network, installs service components of the target devices, and starts the service. The monitor stage runs the system components used by the IoT services and IoT application during their execution, factory-default services, and applications, and mostly, system monitoring. The service stage develops components, such as specification, design, implementation, and testing. The Manage stage enables service activation by publishing services for discovery, delivering the service, and reactivating and disabling services from and for discovery. The update stage provides execution management functions for the deployment, termination, and reconfiguration of the application and service, and system components and firmware update deployment. The decommission stage comprises the factory reset, disconnecting the device from servers, shutting down, physical removal, and the end of the file [25].

Table 3.1 Sample data format of IoT used

sensorID	uniqueID	timecollected	heartrate
heartrate.raspZero	7582c572-6267-4db8-811e-2bce5040a444-heartrate.raspZero	2018-07-07 19:50:33 UTC	7.703.999.057.366.290
heartrate.raspZero	d188341b-81cb-4b8b-8693-5643d8003490-heartrate.raspZero	2018-07-07 20:35:05 UTC	7.386.609.930.906.560
heartrate.raspZero	0decacc6-38ee-4904-bb8c-d035fb6be689-heartrate.raspZero	2018-07-07 20:21:37 UTC	7.309.515.018.448.470
heartrate.raspZero	4f097fe2-15e1-41a1-8102-53c925a13183-heartrate.raspZero	2018-07-07 19:58:55 UTC	8.281.294.568.293.610
heartrate.raspZero	fc56429b-1a15-402b-a553-677c872be3c1-heartrate.raspZero	2018-07-07 20:19:12 UTC	7.668.129.469.742.360
heartrate.raspZero	d78e3b07-09f0-49a8-a7c9-47f59729927c-heartrate.raspZero	2018-07-07 19:44:31 UTC	8.050.770.305.367.730
heartrate.raspZero	6795adc1-2ada-43b6-96f9-848dd02c7f8d-heartrate.raspZero	2018-07-07 20:47:29 UTC	8.364.532.088.472.010
heartrate.raspZero	872a91f1-1156-4944-aed3-a086aa4a3eb9-heartrate.raspZero	2018-07-07 20:51:28 UTC	8.377.509.475.446.920
heartrate.raspZero	0e13cc39-06db-433c-9f0b-e4e824251e66-heartrate.raspZero	2018-07-07 20:15:13 UTC	8.276.083.336.394.490
heartrate.raspZero	081b9ab8-b81e-400c-abb6-9334868ad1fd-heartrate.raspZero	2018-07-07 19:52:57 UTC	7.456.725.813.124.650
heartrate.raspZero	9a2c3d60-4c53-4da6-94af-2706889cae4f-heartrate.raspZero	2018-07-07 20:05:49 UTC	7.437.055.359.402.800
heartrate.raspZero	f811549d-c7dd-41d7-a3d4-ea0d859164fe-heartrate.raspZero	2018-07-07 20:51:37 UTC	7.196.086.052.134.270
heartrate.raspZero	9fb16b49-33d6-4a28-8b0d-3c4ebfc0a697-heartrate.raspZero	2018-07-07 20:17:13 UTC	7.375.927.782.344.650
heartrate.raspZero	4d08c615-6383-47bd-9752-1ef16bcbcc56-heartrate.raspZero	2018-07-07 20:06:59 UTC	765.696.577.754.147
heartrate.raspZero	e5013897-bcb3-4a6c-a51e-1143a72b0bbc-heartrate.raspZero	2018-07-07 20:34:03 UTC	7.900.096.326.850.170
heartrate.raspZero	bc7a08b2-e0e0-4f9b-be4b-bb6eb8e4aae0-heartrate.raspZero	2018-07-07 20:19:47 UTC	7.604.654.325.158.500

heartrate.raspZero	bc449ac4-3114-42a1-ae31-ca16c3f0857f-heartrate.raspZero	2018-07-07 20:11:54 UTC	7.671.596.315.010.760
heartrate.raspZero	43a77970-9944-4ffe-b560-ac0be97b2919-heartrate.raspZero	2018-07-07 20:27:14 UTC	727.335.323.119.662
heartrate.raspZero	0f7750dc-1b55-43f4-8a87-13dc74a06c01-heartrate.raspZero	2018-07-07 20:08:28 UTC	8.711.965.070.750.850
heartrate.raspZero	5cb7b436-89fb-4798-af68-295402f6043c-heartrate.raspZero	2018-07-07 20:25:55 UTC	744.721.586.820.271
heartrate.raspZero	9623a502-ad41-4411-b1ca-df5bc0c641b1-heartrate.raspZero	2018-07-07 20:23:28 UTC	7.804.776.627.411.660
heartrate.raspZero	9fc430e3-bc10-4ff9-a3c0-351ffa99e560-heartrate.raspZero	2018-07-07 19:39:32 UTC	8.056.782.948.826.230
heartrate.raspZero	9175d915-9350-42f2-869b-6b81a0300363-heartrate.raspZero	2018-07-07 20:04:21 UTC	7.167.765.099.364.780
heartrate.raspZero	1fb41eb8-89ed-421c-8937-fb8a8132a052-heartrate.raspZero	2018-07-07 20:39:31 UTC	779.691.518.095.346
heartrate.raspZero	b9272a65-e96a-4356-b743-7961380cbc9a-heartrate.raspZero	2018-07-07 20:53:39 UTC	7.181.430.789.240.080
heartrate.raspZero	bd3c85cf-a701-46d2-8fae-61b0f3c0c8d8-heartrate.raspZero	2018-07-07 19:39:24 UTC	7.757.515.496.335.880
heartrate.raspZero	908fb4ce-2fd6-472c-b61e-d4264d8d830e-heartrate.raspZero	2018-07-07 20:24:12 UTC	7.443.159.518.204.220
heartrate.raspZero	cee5c953-2240-4c20-b465-f0a0a8272b72-heartrate.raspZero	2018-07-07 20:47:21 UTC	8.363.501.048.635.360
heartrate.raspZero	2598cf3c-5df4-482d-9fc0-715f61647f6b-heartrate.raspZero	2018-07-07 20:29:53 UTC	7.637.023.563.505.150

CHAPTER 4

ENCYPTION ALGORITHMS

4.1. ENCYPTION ALGORITHMS

In the computing world, encryption can be converted to a coded format that can be read or processed after being decrypted after an encrypted format. Encryption is the cornerstone of data security. It is the simplest and most important way to get information about a computer from this page, prepare my computer, get information, search, and explain. An encryption method, which takes advantage of both the web and the possibilities, is widely used to protect the privacy of user information on the browser and server. This information is based on your personal data regarding payment data. It uses the encryption method to maintain sensitivity on your website using companies of all sizes. In addition to the obvious advantage of preventing confidential information from being stolen or compromised, encryption also checks that this information is real and comes from the alleged source. This can be used to verify the origin of a message and to confirm that it has not been altered during transmission [26].

Ciphering is attaining a usage area through the evolution of technology in different sectors, such as military, electronics, banking systems, etc. The 2 most important requirements in modern systems are to transfer any data seamlessly and confidentiality. There are some types of enciphering, keying, and analysis algorithms that are improved by cryptography and used for sending data securely and receiving them from the other side. The most common of the cryptography algorithms used is the encryption algorithm.

The encryption algorithm receives the text to be encrypted and the encryption key as input. The decoding algorithm works in the opposite direction to the encryption algorithm (Figure 4.1).

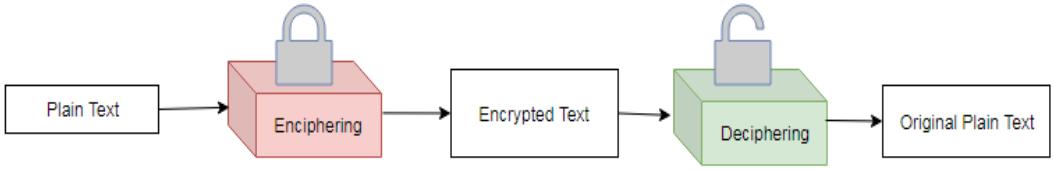


Figure 4.1. Encryption and decryption of a plain text

Basically, there are 2 kinds of encryption algorithms in terms of the type and characteristics of the key used for cryptography.

4.2. Symmetric Encryption Algorithms

In this algorithm, there is just 1 secret key that encrypts a plain text and decrypts the encrypted version of the plain text. There is a critical point in the security of the key. If the key is captured by someone, all encrypted data can be deciphered. The key is a common key that has been agreed on between the encryptor and the person who will decrypt it. Together with the confidential text to be sent, the agreed on secret key is sent to the recipient and the decryption process is performed [27].

Algorithms that can be performed in a short period of time are very useful for encryption and decryption with regards to time management (Figure 4.3). Compared to asymmetric encryption, symmetric algorithms are much faster. The speed of these algorithms are compared with one another in Figure 4.2. However, it is much easier to implement the symmetric algorithm in electronic devices, due to the simple operations involved. Moreover, the length of the key used in symmetric algorithms, and therefore the number of bits, is much smaller. Table 4.1 shows the advantages and disadvantages of symmetric encryption algorithms [28].

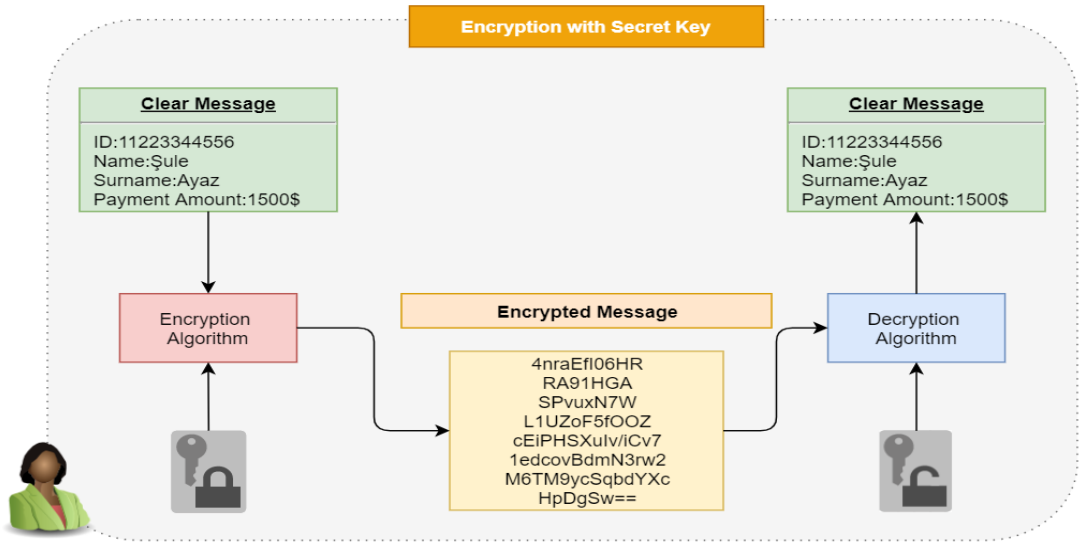


Figure 4.2. Symmetric encryption algorithm sample

Table 4.1 The pros and cons of symmetric encryption algorithm

Strong Aspects	**Weak Aspects**
Algorithms are as fast as possible	Secure key deployment is difficult
Can be used with hardware	Capacity problem
More safer	Authentication and integrity policies are difficult to perform securely

Symmetric algorithms form 2 parts, block encryption and array encryption algorithms. Block encryption algorithms process data in blocks. These algorithms do not have internal memory, so they are known as memoryless encryption. Online block encryption algorithms are preferred to integrate control manageable applications.

Sequence encryption algorithms are the data here as a bit sequence. A generator generates a set of keywords, called a floating key, that you want for your keywords. Floating key generation is time-ready. Sequence encryption algorithms are used to set up voice transmissions in many environments, such as wireless communications.

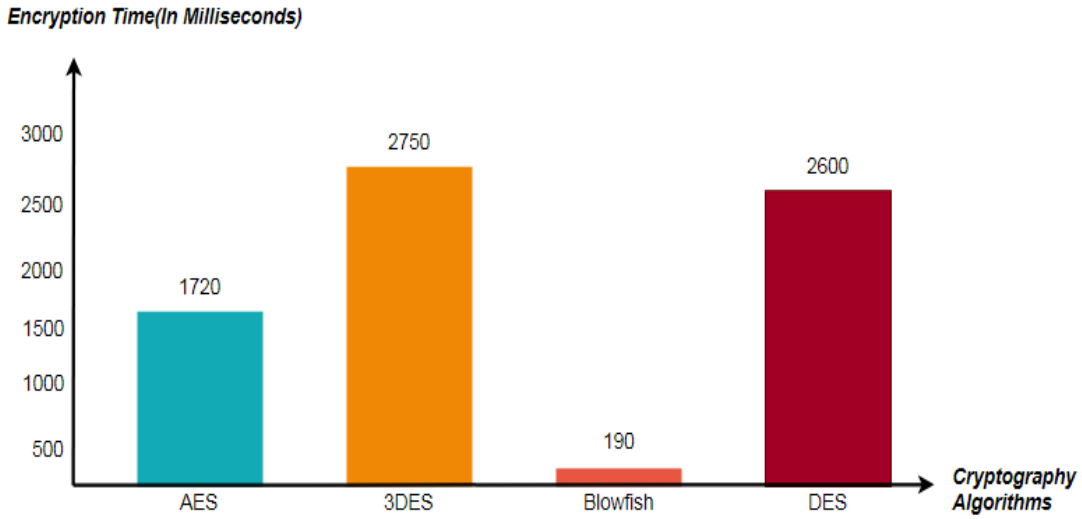


Figure 4.3 Encryption time graph to cryptography algorithms

4.2.1 Advanced Encrytion Standard (AES)

It is a better system than DES. After DES became insufficient, it was improved in 2001 by a few scientists. It was invented by Belgian Vincent Rijmen and Joan Daemen, as a mathematical block encryption algorithm that completely corrected the DES and its inadequacies. There are 3 different key lengths: 128, 192, and 256 bits. AES is said to be faster in hardware and software, easier to implement, and requires much less memory, which is contrary to the DES. It is thought to be resistant to all current academic, practical, and brute force attacks. The symmetric encryption algorithm is mostly used for these advantages [29].

4.2.2 Data Encrytion Standard (DES)

It is a block encryption algorithm. It performs encryption in blocks of a specified length. Although the DES algorithm has a key length of 64 bits, it is a system that uses a 56-bit length symmetric encryption technique. The strength of the DES is that it creates a new key specific for each use, and the weaknesses of the DES are that the algorithm is too slow for today's technology and the 56-bit key length is insufficient. It was found to be inadequate for today's technology, which led to it becoming insufficient in the early 2000s and it lost its reputation. These problems caused by the DES algorithm have been corrected with a new algorithm known as triple DES or

DES-3. Most of the applications used today, such as Secure Shell, use 3DES. The 3DES algorithm works as DES encryption 3 times in a row. Therefore it is 3 times slower than DES. However, a key with a length of 24 bytes is used to perform 3DES encryption. There is 1 parity bit for each byte. Thus, the length of the key is 168 bits. Due to the fact that it runs 6 times slower than the AES encryption method, which has a more advanced algorithm, it has become ineffective with the development of AES [29–30].

4.2.3 Blowfish

Blowfish has a 64-bit heap size and a key length of 32 to 448 bits. Is was designed to replace DES after DES started to become insufficient. The Blowfish algorithm requires at least 4 kb of ram. Therefore, it cannot be used in the smallest systems, such as smart cards. It is considered to be a thriving algorithm, with its effectiveness seen in daily user applications in which high encryption is needed, such as in e-mail. One of the most important features that increased the use of Blowfish is that the encryption algorithms that were used at the time were sold, licensed, and paid for, but Blowfish is completely free. Blowfish is one of the fastest chunk encoders on the market, and its complex key schedule makes it hard to crack [29].

4.3. Asymmetric Encryption Algorithms

The major issue found in symmetric encryption algorithms is key distribution. In a multi-user system using a symmetric algorithm, distributing the same key to all users can be problematic in terms of security. Giving each user a different key can be troublesome, as many different keys in the system are used. For solving these problems, asymmetric encryption algorithms have been improved. The key and the decryption key are different from each other in asymmetric encryption algorithms. The encryption key is the public key and the decryption key is the private key. Public keys are different from the other because it can be distributed to everyone, but it is important to make sure which key belongs to whom. Certificates are used for the stated aim. It is a document of the connection between the public key and the identity of the owner [31]

The private key is only available to the decryption user, and the public key is not confidential (Figure 4.3). Therefore, asymmetric encryption is much more successful in terms of security than symmetry. This prevents key redundancy in multi-user applications that use symmetric encryption using a small number of keys. However, asymmetric cryptography has been left behind in terms of symmetry in terms of speed and hardware compatibility. Enormous prime numbers are used to ensure the security of asymmetric algorithms. This brings enormous problems in terms of time. Systems using an asymmetric algorithm are much slower than systems using symmetric algorithms. In addition, asymmetric encryption algorithms use very large numbers to adapt to the hardware structure, which is very difficult. Table 4.2 compares the advantages and disadvantages of symmetric encryption.

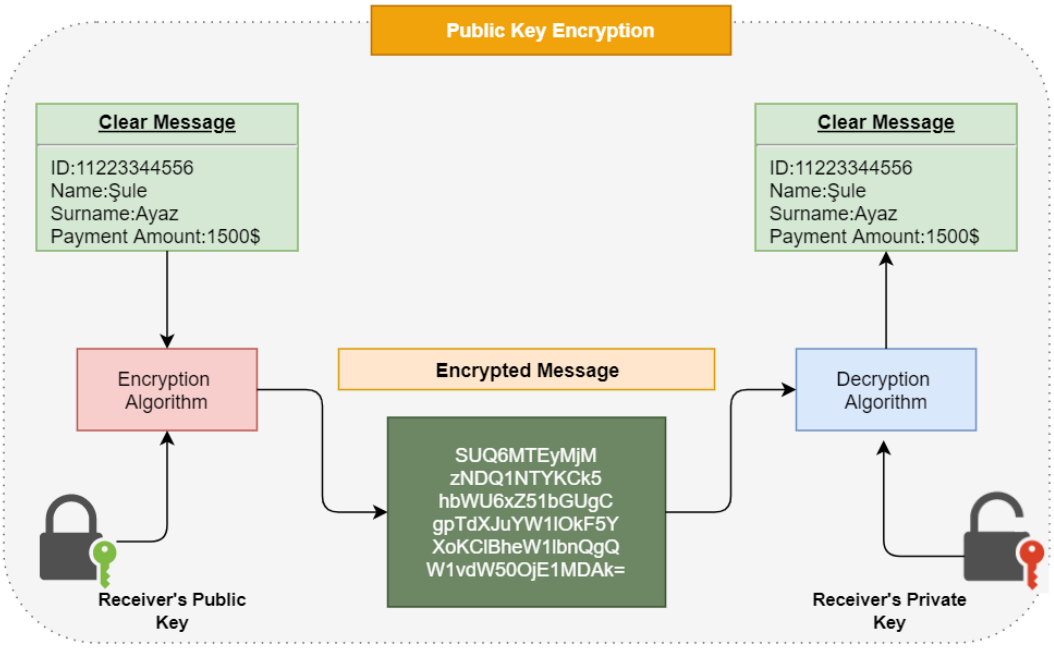


Figure 4.3 Asymmetric encryption algorithms sample

Nowadays, not only high-security, but also high-speed systems, which are called hybrid systems, can be encrypted using symmetric and asymmetric encryption algorithms together. Key encryption, key negotiation, and digital signature operations are generally performed by asymmetric encryption, unsigned data symmetries, and mass data operations if the integrity is maintained.

Table 4.2 The pros and cons of asymmetric encryption algorithm

Strong Aspects	**Weak Aspects**
Considered as the main principles of cryptography; integrity, authentication and privacy can be provided in a secure manner.	Slow operation of algorithms resulting from the length of passwords.
The user can specify the key.	Key lengths can sometimes cause problems.

4.3.1 Rivest-Shamir-Adleman (RSA)

In 1977, a new asymmetric cryptographic algorithm named RSA, which was created by 3 scientists named Rivest, Shamir, and Adleman, performed encryption and decryption operations, as well as key distribution. RSA is an encryption technique whose reliability is based on the challenge of dealing with very great integers. RSA, which is a public key cryptographic technique, was designed for creating very large integers and the challenge of processing these numbers. A more secure structure was created using prime numbers for the key generation. RSA is usually an encryption technology, where the keys are generated from values that are generated by optimizing all of the numbers preferred in the commercial applications used for each of the message encryption and electronic signature purposes. In the RSA algorithm, the numerical size of the key to be used is substantial, both for the reliability and speed of the system. The size required to achieve an adequate degree of reliability is determined using the elliptic curve encryption (ECC) algorithm. With RSA, a 1024-bit key (a number of approximately 300 digits) can now be used as an encryption technique adequate for basic applications. The RSA algorithm uses a basic algorithm for an encryption algorithm. However, it is quite difficult to create a very large prime number. With the establishment of the RSA encryption system, asymmetric encryption algorithms are now more widely used [32].

4.3.2 Diffie-Helman (DH)

Diffie and Helman discovered the first asymmetric encryption algorithm in 1976. DH is a protocol where 2 participants decide on (in a secure way) a common password through an insecure channel without any prior exchange of information. The main purpose of the algorithm key exchange is to enable 2 users to securely transmit a key to each other and then to send ciphered messages to each other using this key. The DH algorithm was created and the protection and distribution of the secret key, which is a major problem for symmetric encryption algorithms, was substantially overcome. However, the specifications of the public secret key are based on the DH algorithm [33]

CHAPTER 5

METHODOLOGY AND IMPLEMENTATION

5.1 METHODOLOGY

This study was conducted to securely provide data transferring between IoT devices and the cloud. The focus of the study was to transfer different kinds of IoT data, such as documents, locations, private tapings, etc., from IoT devices to the cloud, and to store it by ensuring safety. In order to gather the necessary data, a heart rate simulation was created, including IoT data streams via a simulator. The data could be extracted in different formats, such as comma separated value (CSV), text, JavaScript object notation (JSON), a BigQuery table, or Google Sheets. In this study, the text format was chosen for the dataset when simulating the heart rate. Encryption of the IoT data in the cloud platform and keeping them in the cloud storage was the trivet of the survey.

The architecture of the study, shown as Figure 5.1, includes 33 main module groups. The first involves the IoT device simulator, graphical user interface (GUI), command line interface (CLI), and performance evaluation. The IoT device simulator was used to stream IoT data as if received data from a sensor. The GUI provided an interface for the client to choose an encryption algorithm, as the AES, DES, or RSA. The CLI had commands to transmit the IoT data and ease communication between the client and server in the cloud platform. The performance evaluation analyzed the time it took for IoT data encryption and decryption for different sized IoT data in the encryption algorithm. The second group was based on the GCP. Cloud IAM, Compute Engine, Cloud IoT Core, Cloud Dataflow, Cloud Pub/Sub, BigQuery, and Bucket were used to form the architecture. Cloud IAM allows administrators to authorize people who can process specified resources. It provides full control and visibility to centrally manage cloud resources. Compute Engine makes it possible to use VMs running in Google's data centers and fiber network around the world. Thanks to Compute Engine's tool and workflow contribution, it can scale from individual instances to global and load-balanced cloud computing. Moreover, Compute Engine's VMs boot quickly, are

offered with permanent disc storage options, and provide stable performance. Cloud IoT Core and Cloud Dataflow are fully managed services. Cloud IoT Core allows easy connection and management of smart IoT devices all around the world, and Cloud Dataflow is for transforming and enhancing streaming state (real-time) data. Cloud Pub/Sub offers minimally delayed and reliable messaging that helps developers quickly integrate systems that are resolved both on the GCP and externally. Finally, BigQuery stores huge volume datasets and can quickly query the data, and Bucket contains objects that can be accessed in its own way. The last of the 3 main module groups is client-side AI, which allows going through the security module, VM, and data storage.

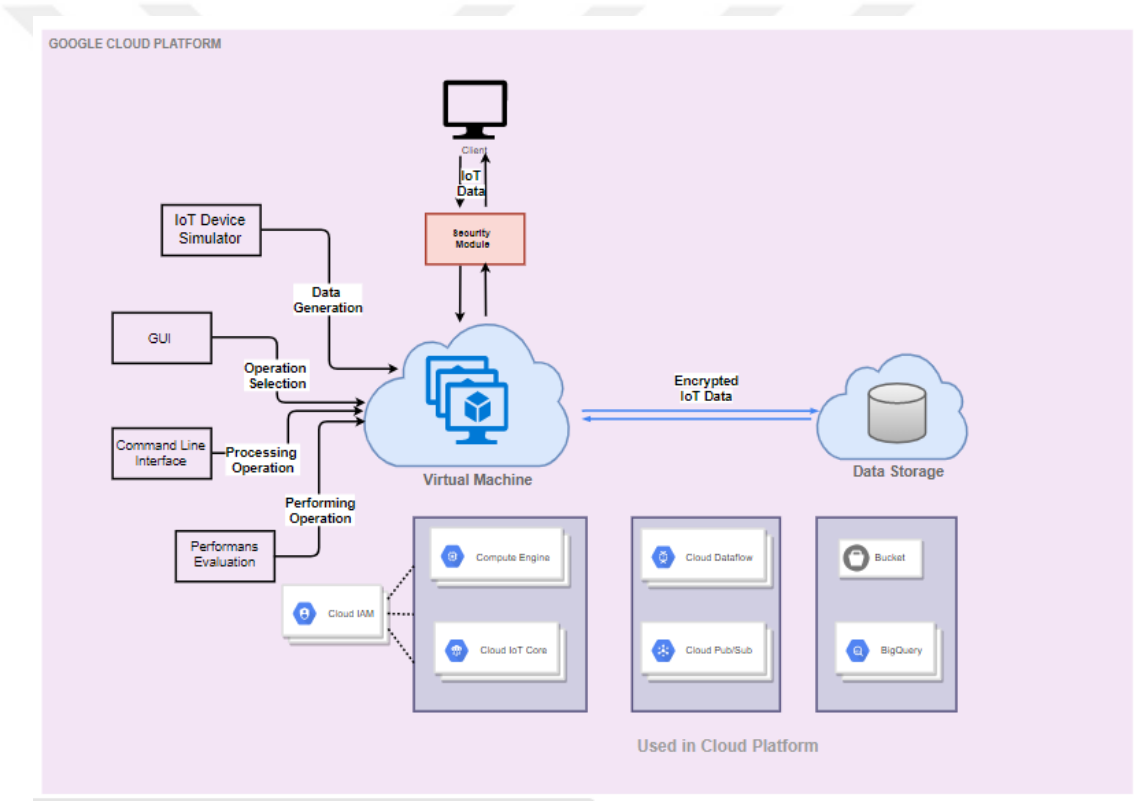


Figure 5.1 The architecture of IoT security with cloud computing

For IoT data, the Raspberry Pi IoT heart rate sensor was used by simulating the device. While the IoT data was extracted from the sensor, at the same time, the different types of documents mentioned above for storing data in the cloud area were used via encryption. There are a lot of encryption methods that provide data security. However, this study was based on 3 algorithm options, AES, DES, and RSA. First, a person

chooses an option between them and the Encrypt All or Decrypt All button is clicked, respectively. Data size is the critical point when an algorithm is chosen. Although there are 3 algorithms for encryption or decryption, the best performance was observed using the AES (Table 5.2).

With respect to data storage, choosing the right cloud provider has become a growing debate that goes beyond scale. We all know the big cloud players, which were mentioned above, but comparing which one of them is the most effective depends on the aim of the study. For this study, there were some strong requirements, such as flexibility, feasibility, suitability, easy integration and configuration, improvable, affluent tutorial, and low cost. The GCP, according to some official references, is better priced than competitors, is a private global fiber network, provides live migration of VMs, improved performance, state of the art security, dedication to continued expansion, and lastly, redundant backups [26].

The GCP has encountered a good deal of them. Moreover, the integration of the compiler, which was Eclipse, especially for Java, was very easy when considering the coding part of the study. For all of the operations, the Java programming language was chosen to make the specified flow as desired, and to operate freely in the cloud platform on behalf of feasibility.

The architecture of the study was placed completely in the cloud platform. Thus, client operations and cloud storage operations were all in the GCP. Moreover, it comprised some different parts, such as management tools, compute, identity security, machine learning, and big data. Stackdriver provides real-time monitoring and logging in the GCP and AWS, as well as useful diagnostic tools. Applications are managed in the GCP with Cloud Shell, which provides instant command-line access from your web-based console, mobile app, or browser. Using these interfaces, all Google APIs, including GCP's Billing API can be accessed. The deployment manager templates make work easier and cloud endpoints can be used to manage APIs.

The method of the study formed 3 steps. First of all, the plain versions of the IoT data were put into the operation, which provided encryption of the data using any of the 3 encryption algorithms on a compiler like Eclipse for Java. The encrypted version of

the data was kept in the VM on the GCP. Storing of the data to the GCP was provided via Google Shell in the second step. In the last step, the encrypted data was decrypted via the decryption method of the chosen algorithm for encryption, by taking from the GCP data storage. Moreover, the IoT data was moved securely between the IoT device that was placed in the VM of the GCP and the data storage of the GCP. Using the proposed strategy, security was provided between the IoT device and the cloud environment.

5.2. IMPLEMENTATION

In this study, the encryption algorithms were presented to the user through an interface. The architecture was based in the cloud platform and the GCP was chosen as the platform for easier management. In the algorithm chosen by the user, the IoT data were obtained from a simulated smart device with the aim of encrypting the IoT data, and preparation was made for the application. When the user selected the encryption algorithm with the application interface located on the VM created in the cloud, the encryption and storage in the cloud occurred after the encryption button was pressed, and the decryption button also obtained the decryption state of the data in the cloud. The mentioned application interface is provided in Figure 5.2.

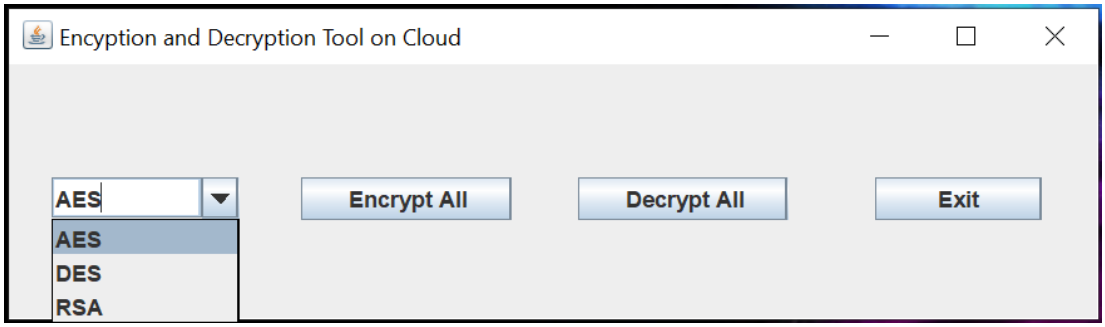


Figure 5.2 The interface of the java application

With regards to the cloud environment, the VM and data storage were created on the GCP. The VM was designed on the client-side in the study, which had the architecture shown in Figure 5.1 in the Methodology section. The application was developed using Eclipse Java, and the IoT data was transferred to the VM in the cloud environment, and it was executed there. The user could transfer some files that included IoT data to

the cloud storage, securely, and whenever they wanted, they could be received from there. While the documents were moved into the VM of the cloud environment, they were encrypted using the chosen algorithm, and the encrypted documents in the data storage of the cloud environment were also decrypted using the decryption method of the same algorithm.

Looking after security, dealing with a huge scale organization, and taking care of the information ingestion prerequisites of many distributed devices are principal requirements as well as critical difficulties. These tasks concentrate on gaining value and understanding from an IoT project. The IoT device that was used herein to capture the heart rate was started so as to provide a fluent data pipeline. The data were securely published to a message line placed within it, which at that point, were shipped into a data warehouse, while the IoT Core was improved. The Raspberry Pi, which collects heart rate data via a sensor, was chosen as the IoT device. In this study, the Raspberry Pi was simulated by a script and the sensor data shown in Figure 5.3 is as it was taken from the simulator.

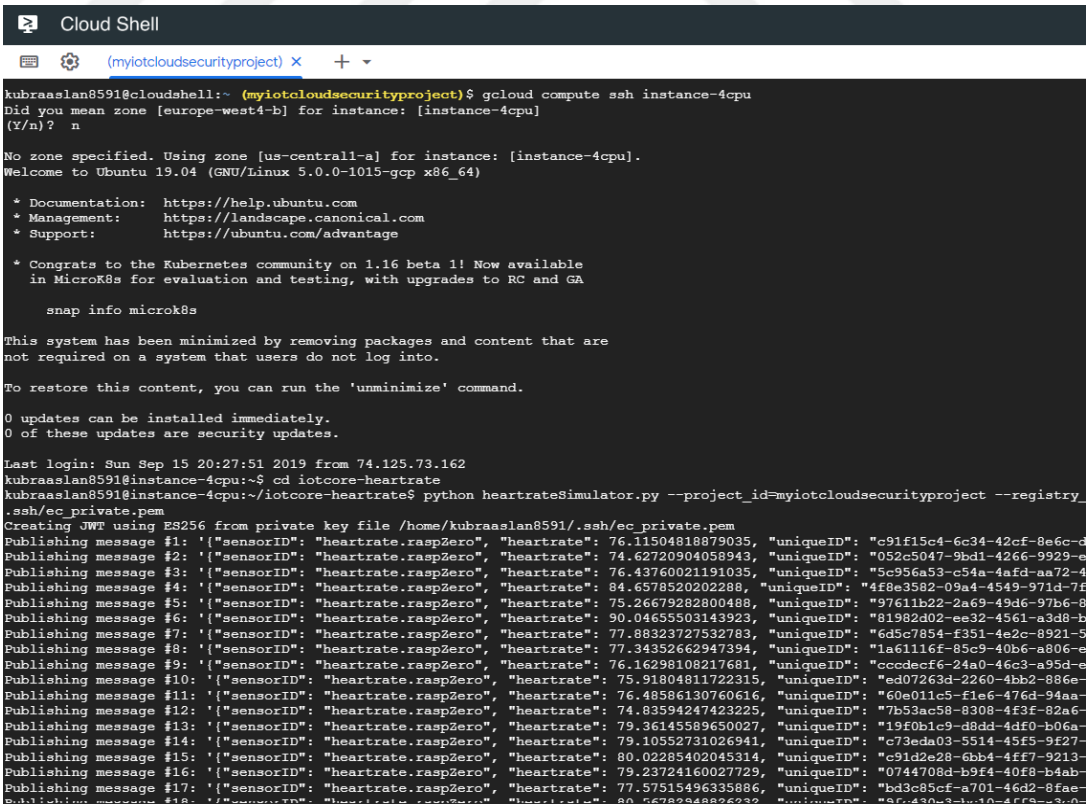


Figure 5.3 Data streaming from heart rate simulator

On the other hand, some components were used for the data-flow, which were respectively, Cloud IoT Core, Cloud Pub/Sub, Cloud Dataflow, and BigQuery (Figure 5.4). First, Pub/Sub, Dataflow, Compute Engine, and IoT Core APIs were used, so they were enabled from the APIs and services placed in the cloud console menu. Second, BigQuery, which is a serverless, extremely adaptable, and low expenditure data warehouse, was used. To maintain the heart rate data, a table was created within a dataset within the BigQuery of the GCP console. A data warehouse was constructed by the created table to save the heart rate data. The next component was Cloud Pub/Sub, which provided streaming data and was focused on the main event. It was a basic, dependable, versatile establishment that significantly and professionally managed the incoming and outgoing IoT messages. From the Cloud Pub/Sub option, a topic was created, and then a subscription was assigned to the topic. The IoT messages were then pulled with the Pub/Sub subscription.

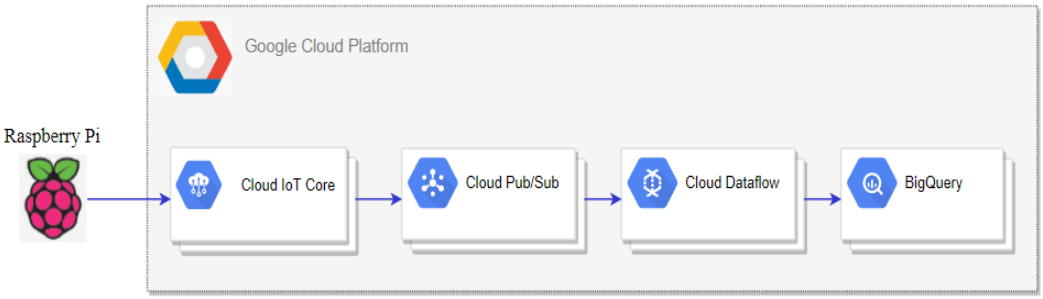


Figure 5.4 Streaming data on cloud

The computing models were carefully managed. For the operation, a process that provided observation of the Pub/Sub topic for the data produced and transferred them to BigQuery was created using a dataflow. For the storage requirement of the dataflow, a location is created in the GCP storage to keep temporary files. Storage was chosen from the GCP console and then the storage was created within the browser option. Next, Cloud IoT Core, as a central service that provides good solutions for collecting, processing, handling, analyzing, and visualizing IoT data in real-time, and is related actively with GCP services, was provided effectively to link, manage, and collapse the data, was distributed to the IoT device. From the IoT Core in the GCP console, a

register was created and then a device was added to the register by creating it. In this step, the necessary software, written in Python, was installed via GCP Shell and a security certificate was produced to authenticate it with GCP, and then it was saved with IoT Core. In this study, the security certificate generated is shown below in Figure 5.5, and its public key was used in the IoT divide in Figure 5.6.

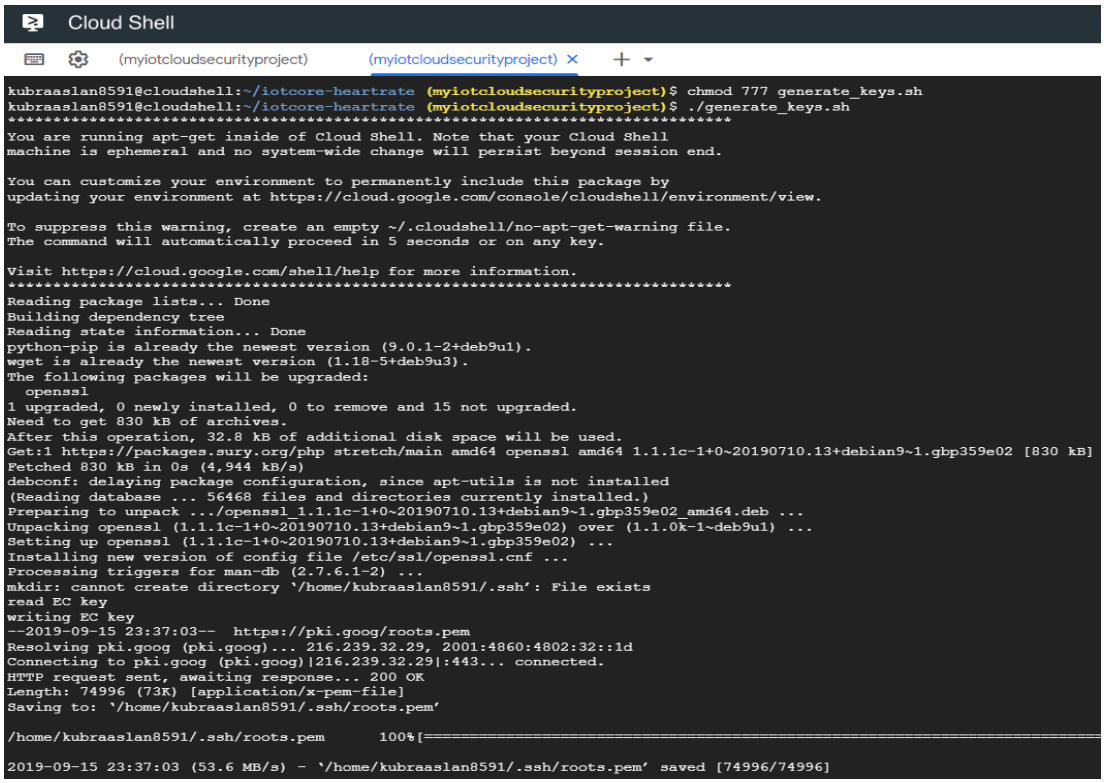


Figure 5.5 Google cloud shell to generate a security certificate

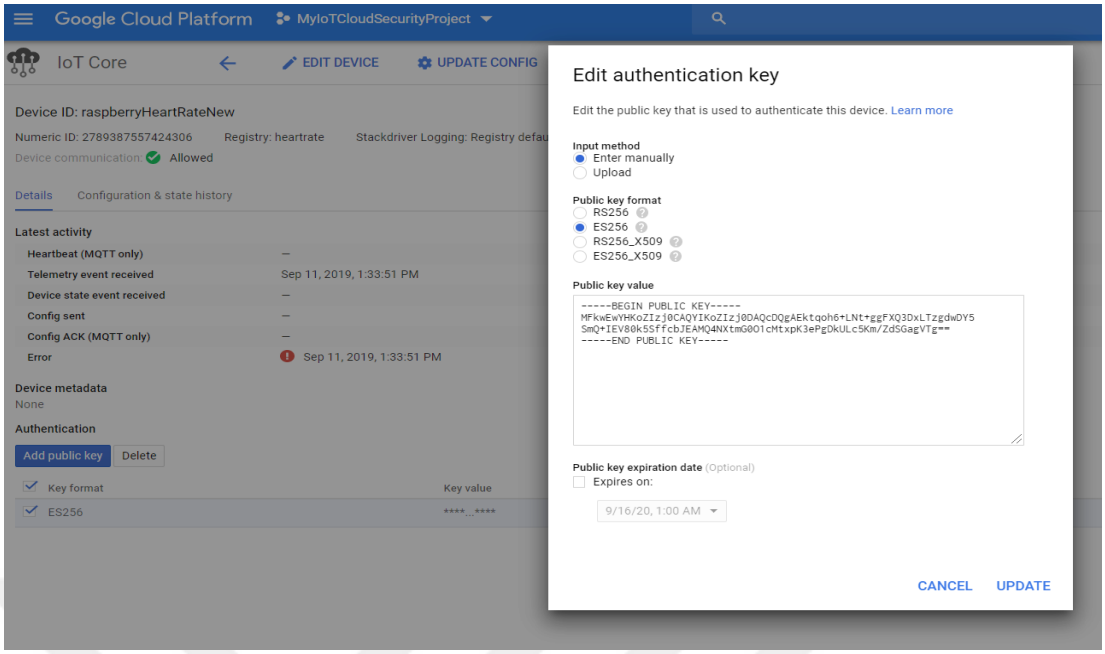


Figure 5.6 Public key ıntegration of google IoT core

Keys generated from the script were saved for use in the device's public key in IoT Core in the GCP. At this point, IoT Core was ready to receive communication from the Raspberry Pi simulation.

A sample dataset from the IoT heart rate sensor was used to stream data and the script written in Python was run on a VM that was created in the Compute Engine of the GCP (Figure 5.7). The script provided input into IoT Core (Figure 5.5). The IoT data can be seen in the BigQuery table (Figure 5.8).

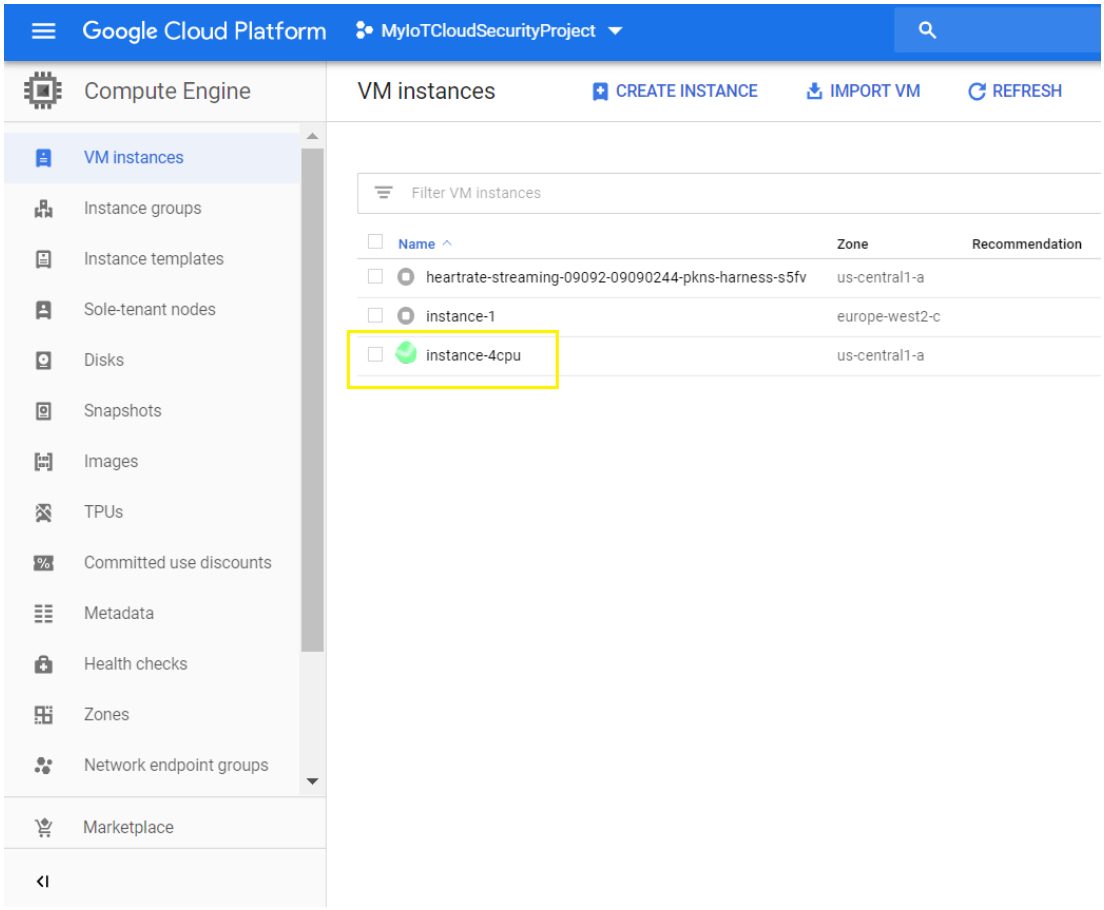


Figure 5.7 The VM of the compute engine in GCP

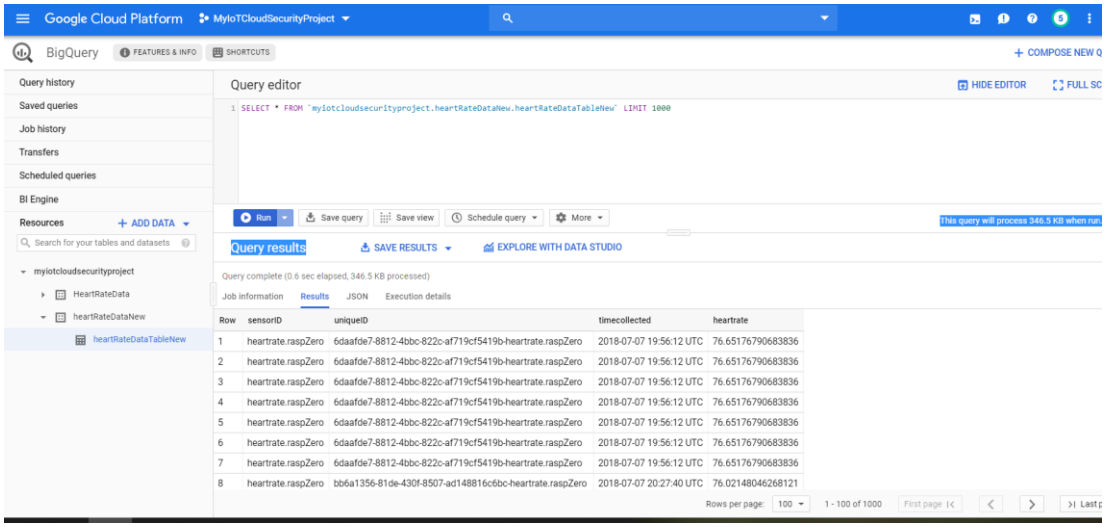


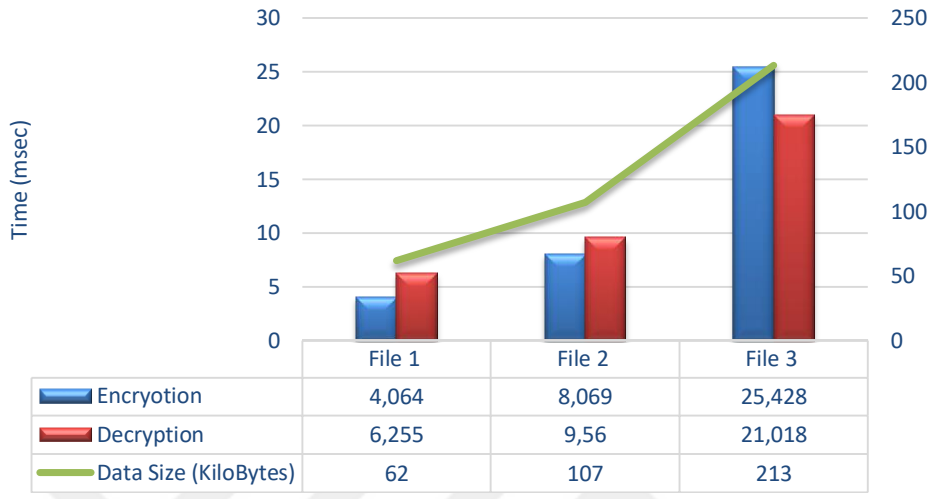
Figure 5.8 IoT data in bigquery of GCP

As a result, the AES, DES, and RSA algorithms were executed in the IoT streaming data on the GCP. These encrypted data were sent securely to Google storage and simultaneously, encrypted into IoT data in the cloud environment, and were then also receiving by decrypting them. Moreover, IoT data security was provided on the GCP, and different sized IoT data files were examined with regards to performance. Tables 5.1, 5.2, and 5.3 show the time spent for each of the encryption algorithms and the file size used to evaluate the performance of the study.

Table 5.1 The encryption and decryption time of sample IoT data to size in AES

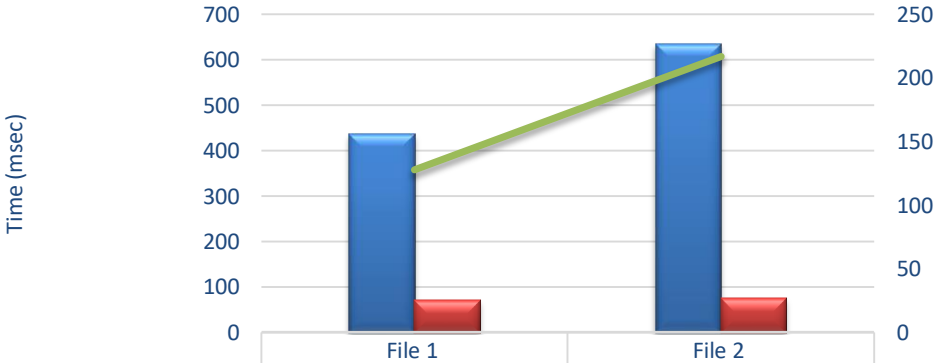
	File 1	File 2	File 3
Encryotion	70,152	73,861	104,291
Decryption	50,161	60,33	64,445
Data Size (KiloBytes)	62	107	213

Table 5.2 The encryption and decryption time of sample IoT data to size in DES



	File 1	File 2	File 3
Encryotion	4,064	8,069	25,428
Decryption	6,255	9,56	21,018
Data Size (KiloBytes)	62	107	213

Table 5.3 The encryption and decryption time of sample IoT data to size in RSA



	File 1	File 2
Encryotion	436,466	635,551
Decryption	71,575	75,991
Data Size (Bytes)	128	217

An clear way to compare the performance of these algorithms are AES, DES and RSA, is to take average of execution times for each algorithm, and analyze them for the reason. The performance can be changed to the larger data size. By the way, three different sized IoT data are choosen for AES and DES, while two different sized IoT data are choosen for RSA which

has limited size as 245 byte. For the big sized IoT data, DES is faster than AES in timing for the same sized data. However, AES is more secure than DES has 64-bit key, based on its algorithm. The algorithm gets slower AES has three keys-128 bit, 192 bit, and 256 bit up to 6 times. Key size for encryption, is also, important in the performance estimating. We study by taking just one sized key. Thus, it is not approached as a criter in the performance evaluation. As for smaller sized IoT data, RSA has 1024-bit key can be preferred by basic application to take more time.

CHAPTER 6

CONCLUSION AND FUTURE WORK

With the communication of smart devices among each other, IoT data explosion has become inevitable. The cloud environment is a good solution for storing the dataset that is acquired from the IoT. Moreover, there are some types of solutions for securely storing the dataset in the cloud. Herein, one that encrypts the IoT data in the cloud platform before it arrives in the cloud storage was preferred. The data was encrypted using a security module and it was transferred to the cloud environment in its encrypted version. When its usage was required, by taking it from the cloud platform, the encrypted data passed through an operation to decrypt it using the decryption function of the same security module. Finally, streaming IoT data transmission between the client and cloud platform service was performed, securely, and its performance evaluation was estimated in the security module using different sized IoT datasets.

Many different adaptations, tests, and experiments remain for future study based on IoT security. Future work concerns deeper analysis of particular mechanisms, new proposals to try different methods, or simply curiosity. There are some ideas that can be tried during the description and the development of IoT data security in the cloud platform. This thesis has mainly focused on the performance evaluation of IoT data security on cloud computing. The following ideas can be tested:

1. A new security module can be developed by combining some existing algorithms or generating a new method based on the existing methods.
2. An IoT device simulation can be created, which generates IoT data encrypted on its own and transmits.
3. Cloud platforms, which are most popular, can be evaluated by comparing them with each other based on IoT data security and their performance evaluation can be analyzed.

REFERENCES

[1] Demir, B., Ayrancıoğlu, G. , Gezer, C. & Gözüaçık, N. A Wireless Sensor Network System Using 6LoWPAN. 2016 National Conference on Electrical, Electronics and Biomedical Engineering (ELECO), 2016.

[2] Miorandi, D., Sicari, S., Pellegrini, F. D., & Chlamtac, I., Internet Of Things: Vision, Applications And Research Challenges, Ad Hoc Networks, *10 (7),* 1497 – 1516. 2012.

[3] Jing, Q., Wan, J., Lu, J., Athanasios, V. & Dechao, Q. Security of the Internet of Things: Perspectives and Challenges. Wireless Networks, *20 (8),* 2481–2501. 2014.

[4] Mollah, M. B., Azad, A. K. & Vasilakos, A. Security and privacy challenges in mobile cloud computing: Survey and way ahead. Journal of Network and Computer Applications, *84*, 38-54. 2017.

[5] Zhong, L., Wang, B. & Wei, H. Cloud computing applied in the mobile internet. Computer Science & Education (ICCSE), 7th International Conference on Computer Science & Education (ICCSE) ,218–221. 2012.

[6] Hovsepyan, V., Securing Data Transfer In Iot Environment, Ukrainian Scientific Journal of Information Security, *22 (2),* 131-134.2016.

[7] Osuolale, A.F. Secure Data Transfer Over the Internet Using Image CryptoSteganography, International Journal of Scientific & Engineering Research, 8 *(12),* 1115-1121.2017

[8] Saleh M.E., Aly A.,A. & Omara F. A., Data Security Using Cryptography and Steganography Techniques, International Journal of Advanced Computer Science and Applications (IJACSA), 7 *(6),* 390-397, 2016.

[9] Donno, M., Giaretta, A., Dragoni, N., Bucchiarone, A. & Mazzara, M., Cyber-Storms Come from Clouds: Security of Cloud Computing in the IoT Era, Future Internet Journal, 11 *(127),* 2019

[10] Zhaoa, Y., Liu, Y., Tian, A., Yu, Y. & Dud, X., Blockchain based privacy-preserving software updates with proof-of-delivery for Internet of Things, Journal of Parallel and Distributed Computing, *Volume 132*, 141-149, 2019.

[11] Yu, Y., Li, Y., Tian, J. & Liu, J., Blockchain-Based Solutions to Security and Privacy Issues in the Internet of Things. IEEE Wireless Communications. *25 (6),* 12–18., 2018,

[12] Giura, P., Jim, T., Sapphire: Using Network Gateways for IoT Security. the 8th International Conference on the Internet of Things, DOI:10.1145/3277593.3277611 2018.

[13] Kong, Q., Lu, R., Ma, M., & Bao, H., A privacy-preserving sensory data sharing scheme in Internet of Vehicles. Future Generation Computer Systems, *Volume 92*, 644-655, 2019.

[14] Vaikunth, T. P. & Aithal, P.S. Cloud Computing Security Issues – Challenges and Opportunities. International Journal of Management, Technology and Social Sciences (IJMTS), 1(1), 33-42. 2017.

[15] Kumar, G. K. & Gobi, M. Role of Cryptography & its Related Techniques in Cloud Computing Security. International Journal for Research in Applied Science & Engineering Technology (IJRASET),5 (8).August 2017.

[16] Cheng, Y., Park, J. & Sandhu, R., A User-to-User Relationship-Based Access Control Modelfor Online Social Networks. 26th Conference on Data and Applications Security and Privacy (DBSec), pp.8-24, 2012.

[17] Alcaraz, C., Najera, P., Lopez, J. & Roman, R. Wireless Sensor Networks and the Internet of Things: Do We Need a Complete Integration?, 1st International Workshop on the Security of the Internet of Things (SecIoT10), 2010.

[18] Urmila R. P., Cloud Computing with Open Source Tool :OpenStack, American Journal of Engineering Research (AJER), 3 (9), 233-240, 2017.

[19] Srınıvas, J., Reddy, K.V.S. & Qyser, A.M., Cloud Computıng Basics, International Journal of Advanced Research in Computer and Communication Engineering, 1 (5), 2012.

[20] Solano, A., Dormido, R., Duro, N., & Sánchez, J. A Self-Provisioning Mechanism in OpenStack for IoT Devices. Sensors, *16 (8)* , DOI: 10.3390/s16081306. 2016.

[21] Kang, M., Kang D., Walters J.P. & Crago, S.P., A Comparison of System Performance on a Private OpenStack Cloud and Amazon EC2, 10th International Conference on Cloud Computing (CLOUD), DOI: 10.1109/CLOUD.2017.47, 2017.

[22] AWS: 10 Dakikalık Öğreticiler. Retrieved July 30, 2019, from https://aws.amazon.com/tr/getting-started/tutorials. 2019.

[23] Neden Google?, Retrieved July 25, 2019, from https://cloud.google.com/why-google-cloud/. 2019.

[24] Google Cloud Platform Diagram Software for Innovative Solutions, Retrieved July 30, 2019, from https://www.edrawsoft.com/google-cloud-platform-software.php.

[25] Rahman, L.F., Ozcelebi,T. &Lukkiena,J., Understanding IoT Systems: A Life Cycle Approach, The 8th International Symposium on Internet of Ubiquitous and Pervasive Things (IUPT 2018), Procedia Computer Science 130 (2018) 1057–1062, 2018.

[26] Yılmaz, M. & BALLI, S., Veri Şifreleme Algoritmalarının Kullanımı İçin Akıllı Bir Seçim Sistemi Geliştirilmesi, Uluslararası Bilgi Güvenliği Mühendisliği Dergisi, 2 (2), 18-28, 2016.

[27] Elminaam, A.D.S, Kader, H.M.A. & Hadhoud, M.M., Performance Evaluation of Symmetric Encryption Algorithms, IJCSNS International Journal of Computer Science and Network Security, 8 (12), 280-286, 2008.

[28] Gharat, M.P. & Motawani, D., Overview on Symmetric Key Encryption Algorithms, Int. Journal of Engineering Research and Applications, 4 (9), 123-126, 2014.

[29] Ural, N. & Örenç, Ö., Şifreleme ve Şifre Çözme Yöntemleri, Pusula Yayıncılık, 2017.

[30] Dhiraj R., Different Encryption Algorithms in Java. Retrieved July 23, 2019, from https://it.toolbox.com/blogs/dhirajray/different-encryption-algorithms-in-java-121917. 2017.

[31] Bala, T. & Kumar, Y., Asymmetric Algorithms and Symmetric Algorithms: A Review, International Journal of Computer Applications, Number 4, 1-4, 2015.

[32] Rachmawati, D., Amalia, A. & Elviwani, E. Combination of Rivest-Shamir-Adleman Algorithm and End of File Method for Data Security, The 2nd International Conference on Science (ICOS) Journal of Physics: Conf. Series, 979 (10), 2018.

[33] Kallam, S., Diffie-Hellman:Key Exchange and Public Key Cryptosystems, Master's Thesis, Indiana State University, United States. 2015.

PERSONAL INFORMATION

Name Surname : Kübra ASLAN

Date of Birth : 06/05/1985

Phone : 05437476781

E-mail : kubraaslan8591@gmail.com

EDUCATION

High School : Prof. Dr. Şevket Raşit Hatipoğlu Lisesi

Bachelor : Atılım Üniversitesi

Master Degree : 3.43/4.00

WORK EXPERIENCE

Software Engineer

TOPICS OF INTEREST

- Cyber Security
- Criptology
- Developing C, C++, C#, Java Application
- Parallel Programing
- Developing Mobile Application