# Security and Privacy Analyzed Cloud Framework Design and Implementation for Ethiopian Universities

By: Samuel Gudeta

A Thesis Submitted to the

Department of Computing

School of Electrical Engineering and Computing

Presented in Partial Fulfillment for the Degree of Masters of Science in Software Engineering

Office of Graduate Studies

Adama Science and Technology University

February 2019

Adama, Ethiopia

# Security and Privacy Analyzed Cloud Framework Design and Implementation for Ethiopian Universities

By: Samuel Gudeta

Name of Advisor: Dr. N. Satheesh Kumar

A Thesis Submitted to the

Department of Computing

School of Electrical Engineering and Computing

Presented in Partial Fulfillment for the Degree of Masters of Science in Software Engineering

Office of Graduate Studies

Adama Science and Technology University

February 2019

Adama, Ethiopia

# DECLARATION

I hereby declare that this MSc thesis is my original work and has not been presented as a partial degree requirement for a degree in any other university, and that all sources of materials used for the thesis have been dully acknowledged.

Name: Samuel Gudeta

Signature:_____

This thesis has been submitted for examination with my approval as thesis advisor.

Name: Dr. N. Satheesh Kumar

Signature:_____

Date of Submition:_____

## Approval of Board of Examiners

We, the undersigned, members of the Board of Examiners of the final open defense by _____have read and evaluated his/her thesis entitled "_____" and examined the candidate. This is, therefore, to certify that the thesis has been accepted in partial fulfillment of the requirement of the Degree of ……

| | | |
|---|---|---|
| _____ | _____ | _____ |
| Supervisor/Advisor | Signature | Date |
| _____ | _____ | _____ |
| Chairperson | Signature | Date |
| _____ | _____ | _____ |
| Internal Examiner | Signature | Date |
| _____ | _____ | _____ |
| External Examiner | Signature | Date |

# Acknowledgment

First and for most I would like to thank God for his everlasting love, mercy and support throughout my life and during the course of this thesis work. Without his blessing I can't imagine where I might end up.

My special thanks go to Dr. N. Satheesh Kumar, who is the staff member of the School of Electrical Engineering and Computing in Adama Science and Technology University for his advice and valuable comments in doing this thesis work. And also, I am grateful to the department of computing and Dr. Mesfin Abebe that allowed me to proceed unfailingly on the course of my thesis work. I wouldn't have finished this thesis without their help.

I would also like to thank the peoples who provided a valuable data during the information gathering, the EthERNet security team leader and DU ICT staffs.

Finally, I would like to express my profound gratitude to my family. Mother, father, brothers and sisters thank you for being by my side throughout the ups and downs of life and encouraging, supporting and following up throughout my journey and this thesis work. And also, I would like to thank my friends for supporting and encouraging me in the most difficult and seemingly hopeless situations. Without your encouragements, prayer and support I wouldn't have finished this thesis work.

# Table of Contents

# List of Tables

# List of figures

# Acronyms and Abbreviations

| | |
|---|---|
| AES | Advanced Encryption Standard |
| AWS | Amazon Web Services |
| BaaS | Backup as a Service |
| CRT | Chinese Remainder Theorem |
| DCS | Data Centric Security |
| EthERNet | Ethiopian Education and Research Network |
| GCD | Greatest Common Divisor |
| HEI | Higher Education Institutions |
| IAM | AWS Identity and Access Management |
| IT | Information Technology |
| MD5 | Message Digest algorithm |
| NIST | National Institute of standards and technology |
| RSA Algorithm | Rivest Shamir Adleman Algorithm |
| STaaS | Storage as a Service |
| TEaaS | Test Environment as a Service |
| SaaS | Software as a Service |
| SHA | Secure Hash Algorithm |
| PaaS | Platform as a Service |
| IaaS | Infrastructure as a Service |

# Abstract

In now days educational institutions throughout the world and in Ethiopia are highly depending on IT to deliver different services that enhance the teaching-learning process, researching as well as the different business requirements. Cloud computing is a promising sector in the IT industry to support HEIs because of its performance, accessibility, low cost and many other essential characteristics. Also, the cloud computing approach can maximize the capacity of these institutions without investing on new infrastructure or buying licensed and expensive software. Even though a wide range of luxuries are provided by cloud computing there are still several issues that act as obstacle to use cloud computing in the HEIs. Privacy and security of users' data is the major issue in cloud computing. The objective of this paper is to find a solution that can enhance some technical requirements with relate to user's data security and privacy in the cloud. The research paper discusses the proposed solution which is based on the DCS approach. This approach allows the data owner to fully control the data security from within the data itself and throughout the data life cycle on the cloud. To achieve this a symmetric encryption algorithm, that is AES-256 encryption algorithm is used to encrypt each user's data and then using the CRT combined with the RSA 1024-bit encryption algorithm the access control policies and secure key sharing are enforced to the data. And here the privacy is enhanced since the number of authorized users is first defined and the key sharing or the CRT solution is calculated based on these users and the process is even hidden from the cloud provider. Finally, the integrity proof is computed by using SHA 256-bit hashing algorithm and attached to the data. All these security parameters are set and managed by the data owner and a secure self-protective and self- descriptive file created and this file can be out sourced and shared on the cloud environment to be accessed by only authorized users based on the attached security policies. The research findings are evaluated in terms of addressing the research questions, meeting the objective of the research and in terms of the computation and storage overhead caused during the implementation. The prototype implementation and the evaluation result show the proposed solution is simple and can be used practically.



**Key words:** Cloud computing, DCS approach, HEIs, CRT, Data Confidentiality, Data Availability, Data Integrity, Data access control, Secure key sharing, Data Leakage, Privacy

# Chapter 1 Introduction

## 1.1. Background

Cloud computing is one of the most popular subject areas of software engineering to conduct research. And its impact is real in peoples social, cultural and economic activity with business, enterprises, industries and even institutions continue to undergo considerable changes. Cloud Computing offers better computing through improved utilization of infrastructure costs, flexibility and delivery platform for providing business or consumer IT services over the Internet for not only higher education but also for different work domains. This behavior of cloud provides a great opportunity to provide cost effective, more available and flexible service in higher education institutions.

Cloud computing is defined by the National Institute for Standards and Technology (NIST) as follows “cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”. In addition to the above definition NIST also stated that a cloud computing model is composed of five essential characteristics three service delivery models and four deployment models. On-demand self-service, Broad network access, Resource pooling, Rapid elasticity and Measured service are the essential characteristics and the three basic service delivery models that allows organizations use cloud service are Software as a service (SaaS), Platform as a service (PaaS) and Infrastructure as a service (IaaS). Finally, the four deployment models are Private cloud, Community cloud, Public cloud and Hybrid cloud [9].

Although cloud technology has become popular the concerns about data security and privacy is the biggest cloud security concern. As Holger Schulze descried in his research project general security risk is holding back cloud adoption in many of the organizations. Therefor ensuring cloud data security and privacy is important for cloud adoption and this can be achieved by deep studying on common security and privacy threats for data on the cloud as well as the relevant state-of-the-art solutions for those threats.

In Ethiopia the number of universities are increasing and the number of students involved in those universities are ironically increasing this means that there is a high level of need to use IT resources

so that the quality of education given in these institutions gets improved. And the government of Ethiopia has initiated a program called The Ethiopian Education and Research Network in 2001 with the aim of building and delivering high performance networking that can connect these universities and other research institutes with each other and similar institutions in the world, and by doing this to enable them to share educational resources and collaborate both within Ethiopia and globally. And in this project different security mechanisms are implemented from traditional security methods to trusted computing technologies and the entire security control is in the hand of the cloud service provider due to this the current system does not give the data owner control of the security, privacy and integrity of his/her own data that is outsourced to the cloud.

So, the intent of this research is on improving the data security and privacy problems of the data centers and users by suggesting a cloud security framework that will be designed by considering the ICT resource of the universities and network infrastructure of Ethiopian Education and Research Network. And implementing an APIs that will help to provide data security and improve privacy of users, so that information and resource sharing among Ethiopian universities will get better secured.

## 1.2. Motivation

The first thing that motivates me is the findings of a research report conducted by Holger Schulze in 2018. It indicates top three cloud security concerns by cyber security professionals. These are protecting against data loss and leakage, threats to data privacy and breaches of confidentiality all of which related to user's data.


**Figure 1.1** Top Cloud Security Concerns [1]

The second thing that motivates me to undertake this title is the fact that cloud computing is a very blossoming and advantageous computing choice for businesses, organizations, institutions and etc. and most proposed frameworks especially for educational institutions try to address problems from organizational and technical perspective meaning in organizational view the distinction focuses on the extent to which the users and providers/organizational units are separated from each other whereas technical view is oriented towards functional features and still the security challenges for this technology are not addressed adequately. That is why I want to undertake my research under this title and contribute something.

## 1.3. Statement of the Problem

According to [2] Security in particular is one of the most argued-about issue in cloud computing and the foundation of cloud adoption. Several enterprises and businesses look at cloud computing warily due to projected security risks and security issues have prevented businesses, health, education and many sectors from fully accepting cloud platforms. Therefore, the need to research regarding the integration of security in cloud computing is very necessary.

The Public Higher Educational Institutions in Ethiopia suffer from common problems like lower levels of IT efficiency, unequal distribution of information resources and huge costs involved in managing the entire Information and Communication Technology (ICT) infrastructure. If the concept of cloud computing in higher education is applied all the universities in the country will get a chance to access similar resources with reduced infrastructure cost, improved flexibility and reliability. But issues related to security and privacy were the major factors since sensitive and critical information is outsourced in a cloud to assure an appropriate level of information sharing among universities. The data centers built around in Ethiopian universities may encounter different types of data security problems. So the essence of this research will be challenging the data security and privacy problems the datacenters and users may encounter during adopting cloud computing as well as coming up with a robust mechanism to with stand attack and unauthorized access. Keeping in mind the above problems we need to analyze and design a model that may be used to reduce the occurrence of those problems. As per my knowledge several cloud frameworks were proposed to address infrastructure cost reduction, improved reliability or improved availability for Ethiopian higher education institutions but the essence of this research will be in developing a framework that

may be used to prevent the occurrence of the above discussed problems and enhance the data security and privacy of Ethiopian higher education institutions' cloud.

Therefore, this research will try to address the following research questions:

1. What are the data security and privacy problems that might occur with related to adopting cloud computing technology?
2. How can existing security techniques be used to enhance the security and privacy of users' data?
3. How to design a cloud framework that is applicable for Ethiopian universities that can withstand the occurrence of data security and privacy attack?

## 1.4. Objectives

### 1.4.1. General objective

The general objective of this research is to propose a framework that considers the security and privacy enhancement of users' data stored in Ethiopian higher education Institutions cloud system.

### 1.4.2. Specific objectives

The specific objectives of this research are:

1. To analyze the various cloud data security and privacy management issues during adopting cloud computing in universities.

2. To identify the best data securities approach with the available security techniques and algorithms to achieve the main objective.

3. To demonstrate a secure framework that may enable universities to share information with better level of security compared to the current ICT infrastructure model.

4. To audit or evaluate the proposed framework level of security in terms of providing critical and sensitive information among Ethiopian higher education institutions cloud.

## 1.5. Research Methodology

The different types of methodologies used to collect data, analyze data, to design and implement prototype are discussed as follows. Based on the assumption that all universities require similar services and can have similar standards since they all have same objective; a descriptive approach is planned to be used for the purpose of the empirical data gathering. So, a semi-structured interview was used to understand the current data security model, effectiveness of the current security model and their recommendation on the enhancement of the users' data security, privacy and integrity to give a better service. In addition to the interview observation method is also used to collect primary data such as the activity and role played in each ICT offices of the selected universities, equipment and tools in the data centers and the interaction of systems to secure the environment.

A document analysis was used to understand the current network and infrastructure of the current system and also a literature study was used to understand the various data security and privacy management issues in higher education institutions based on cloud computing technology.

In order to successfully design the framework, we used Edraw Max 9.1 modeling tool, Microsoft office to prepare the documentation and snipping tool to take the snapshot image of the execution result. In addition to this AES Crypt tool combined with Eclipse IDE (eclipse neon.3) is used to implement and audit the proposed solution.

## 1.6. Scope and Limitations

This study focuses on providing a data security framework for Ethiopian universities by analyzing existing reference architecture models and understanding the security and privacy gap of those models at all layers of the cloud computing that means it maybe at physical layer, resource abstraction or service layer. Since security is a very vast topic this research will be limited to implementing the security and privacy at the data level by using some cryptographic techniques and other methods to enhance the data security as well as users' privacy. In addition to this the scope of this research is limited to unstructured data resources such as images, videos and documents of any type and focuses on encrypting these data, securely sharing and accessing these data and enforcing the file integrity for these data. So, the research focuses on securing the users' data confidentiality, integrity, availability and privacy in a universities cloud environment.

## 1.7. Significance of the study

The significance of this research is to implement a data security and privacy analyzed framework for higher education cloud. A conceptual data security model is provided to be integrated with universities cloud model. The essential characteristics, benefits and technical challenges with relate to this framework are discussed.

The proposed solution provides important features for its users such as:

- It uses a combination of symmetric encryption, asymmetric encryption, hashing algorithm as well as Chinese reminder theorem to enhance users' data security and privacy.
- The security perimeters such as access control policy, integrity proof capability is embedded on each data itself and this make the data self-descriptive and self-protective therefore the security does not depend on cloud service provider security measures.
- The data owners are capable of enforcing and managing the security measures for their own data whereas the authorized users are capable of verifying the integrity and securely accessing the data.
- This solution allows the data owners to implement unique encryption key for each of their files and attaches the key securely to the data itself so that authorized users can retrieve the key and securely access the file.
- The content of the data as well as the identity of the users allowed accessing the data cannot be read if the data is in the hand of any potential unauthorized user and the confidentiality, integrity, availability and privacy of the file and users identity is protected when the data is outsourced to the cloud.

## 1.8. Beneficiary of the Research

The most beneficiaries from the research result are universities and the EthERNet. And to be specific this includes researchers, teachers and students of this institutions. The proposed solution incorporates different types of cryptographic techniques to enhance the data security and privacy of these users.

The EthERNet project, which is initiated to provide the country with educational cloud, is provided with additional security mechanism to enhance the data security and privacy of its clients with out

farter need for infrastructure or new software cost. And this is because the proposed solution is defined, implemented and managed by the data owner from the data owner side.

In addition to this the proposed and implemented solution could be used as a baseline for the realization of data security and privacy issues for Ethiopian higher education institutions cloud. Additionally, the study could be a baseline for further studies on data security and privacy issues in ICT utilization using cloud computing and could be considered as a strategy to different Ethiopian researchers.

## 1.9. Organization of the study

The remaining chapters of this research are organized as follows:

Chapter two of this thesis discussed the definition of cloud computing, general and technical overview of cloud data security and privacy challenges, the possible reference architectures to tackle the security issues, the current state of cloud technology in Ethiopian HEIs and the state-of-the-art solutions in cloud computing data security and privacy which is discussed in the subtopic related work of this chapter.

In chapter three the methodology and tools used in doing this research from beginning to end are discussed. It discussed the research design processes and design principles implemented in this research, the data collection methods, the development environment and tools as well as the evaluation methods are discussed.

Chapter four starts by describing the result of data analysis which shows the existing scenario then the selected security approach which is data centric approach that is used to enhance the security of users' data is discussed followed with its characteristics and security challenges then the conceptual security framework was designed and discussed.

Chapter five describes the simulation tool and experiment environment then the implementation and discussion of the proposed solution is discussed. It includes all the operations required to create the secured file from the data owners' perspective and operations required to securely access from the authorized user perspective.

Chapter six presents the Evaluation, Result and Discussion based on the data collected from the execution result in chapter five and the result is analyzed in terms of computation and storage efficiency.

And finally, chapter seven contains conclusion and recommendation on security and privacy of data for cloud users as well as the future work that would be taken as further research focus from this research topic perspective.

# Chapter 2 Literature review

## 2.1. Introduction

In this chapter we will try to discuss the research area on data security, users' privacy and protection as well as cloud computing with its security and protection concerns, cloud data security and users' privacy solutions, different security and privacy reference architectures for higher education institutions, simulation tools that are used to design and implement the cloud environment. Data security and user privacy are one of the core challenges on cloud computing that result in limitation of integrity, availability and confidentiality of the data in cloud services. Definition of the term cloud computing from different experts and organizations, the major types of cloud deployment and service delivery models, aspects considered in cloud data security and users' privacy as well as the existing state of the art on cloud security and privacy architecture are discussed as follows.

## 2.2. Cloud computing

Cloud computing appears in its modern sense in early 1996 [3] and now it is one of the fastest growing computing paradigms in computing and technology and this is because of its property that includes advantages of high computing power, cheap cost of services, high performance, scalability, and accessibility as well as availability.

Different cloud professionals and experts define cloud computing on their own way since there is no universally accepted definition. To define cloud computing in simple terms it is an information technology model that provides services like computing or processing power, storage power and any type of technological services over the network for users. The NIST definition characterizes important aspects of cloud computing as well as cloud services and deployment strategies. "Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." [4]

### 2.2.1. Service delivery Models of cloud computing

The NIST definition of cloud computing describes three cloud service delivery models. These are Software as a service (SaaS), Platform as a service (PaaS) and Infrastructure as a service (IaaS).

Software as a service (SaaS): here customers can access applications running on cloud infrastructure by using their devices like computer or smart phone that is connected to internet through thin client interfaces such as web browser or a program interface [4]. Since customers cannot control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities this service delivery model is more secure. Good example could be Microsoft office365, Dropbox, Google App, SalesForce, Netflix and many more.

Platform as a service (PaaS): in PaaS the vendor provides a platform and customer can deploy their own created or acquired application created with programming languages, libraries, services, and tools supported by the vender. The consumer can use hosting environment and also can control the deployed application as well as the configuration setting of the hosting environment. Examples include Google App Engine which allows anyone to run applications on Google's infrastructure as well as Microsoft Azuri that allow customers to create and deploy application in .NET, Node.js, PHP, Python, Java and Ruby.

Infrastructure as a Service (IaaS): in this model the customer uses computing resources such as processing, storage, network or middle ware, and other fundamental computer resources. [4] But it doesn't mean the customer can control the entire infrastructure, they only can control the applications and operating system they deployed, network components such as host firewall and load balancing and also storage settings to fully utilize a specific solution made available for customers.

### 2.2.2. Deployment Models of cloud computing

The National Institute of standards and technology (NIST) outlined four major types of cloud deployment models [4] and they are discussed as follow.

**Public Cloud:** In this model the cloud infrastructure is provisioned on shared environment for open use by customers. It is usually used for operations without sensitive data and main advantage of this type of cloud is that it is much larger than a private cloud company could have with the ability to scale up or down based on customers' demand. Mostly this type of cloud may be owned and managed by a business, academic, or government organization or some combination.

**Community Cloud:** in this model a community which is organizations with shared concerns like security requirements, mission, and policy and compliance consideration get service from cloud. [4] The management might be controlled by one or more organizations found in the community and members of the community share access to the data and applications.

**Hybrid Cloud:** this model consists of a mixed deployment of a public and private cloud infrastructure that allows users to keep and process non-critical data and applications in the public cloud and critical data and applications in the private cloud. This model offers flexibility with security and more scalable and accessible when compared with other models.

**Private Cloud:** in this model the cloud is built for a single company or organization comprised of multiple consumers to exclusively use and controls it. This model provides a greater control of the cloud infrastructure and security because user access and network used are restricted and selected. A private cloud may be owned by customer but built and managed by a third party at the customer's premises or off premises [4].

### 2.2.3. Essential characteristics of cloud computing

There are five essential characteristics [4] that make cloud computing unique than other computing technologies and they are discussed as follows.

**On-demand self-service:** this aspect states that in cloud computing customers can unilaterally access computing resources such as server time, storage or network band width as needed without any human interaction from the provider side.

**Measured service:** every service accessed by the customer is controlled and monitored automatically by the cloud system and reported for the consumer as well as for the provider for the sake of transparency.

**Rapid elasticity:** meaning quick scale up or down of computing resources provisioning and realize according to the need of the customer. Therefor customers can purchase as much or as little as computing resources they need.

**Broad network access:** meaning cloud services are available through a network and can be accessed through standard mechanisms by both thin and thick clients such as laptop, mobiles, workstations and etc.

**Resource pooling:** the cloud provider can serve multiple customers using a multi-tenant model with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources, but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter) [4].

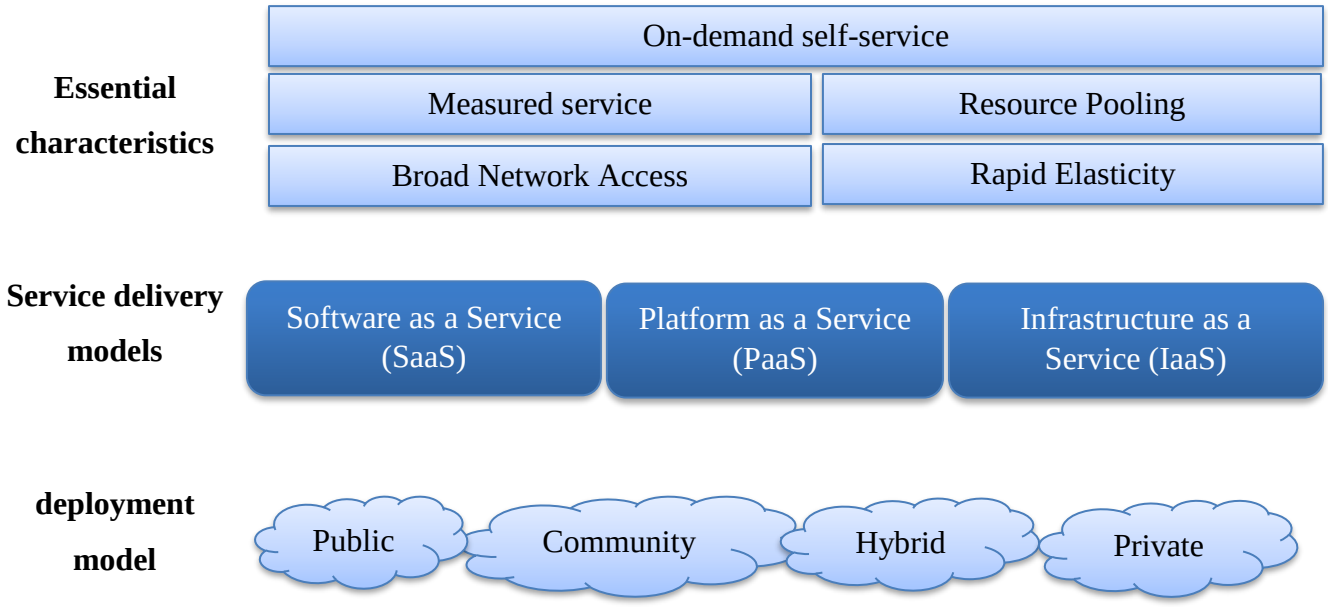



**Figure 2.1** Cloud computing Definition based on the NIST visual model

## 2.3. Cloud Data Security and Privacy Aspects

The cloud environment is more exposed to challenges and risks when compared with traditional IT environment for the following reasons. The first reason is cloud models have dynamic scalability, service abstraction and location transparency features that means applications and data in the cloud does not have fixed infrastructure and security boundaries because of this during security breach it is difficult to isolate the exact physical resource that has a threat or compromised. Secondly resources provided by the cloud provider may be owned by multiple providers and in most cases those providers don't have a common security measures due to several reasons like computation and conflict of interest among them. Thirdly due to openness and multitenant property of cloud, users' data may be accessed by unauthorized user. Finally, there is a massive information storage and fast information access in cloud environment therefore the security measures must meet this requirement [5].

The term data security according to [6] is defined as providing the user confidentiality, availability and integrity by protecting users' data and information system from unauthorized access, use, disclosure, disruption, modification, or destruction. The term privacy is defined as "the rights and obligations of individuals and organizations with respect to the collection, use, retention, disclosure, and disposal of personal information" [7]. From the above definition of data security, the major pillars for a better security are confidentiality, availability and integrity.

## 2.4. Cloud computing in higher education

Education is one of the foundations for the development of a country in a way that it contributes to poverty reduction, increase economic growth and leads to an improved standard of living. Because of this in Ethiopia both the government and private players are investing in the education sector to leverage the opportunities.

The government of Ethiopia has taken several initiatives in order to promote the level of education in Ethiopia. These initiatives include restructuring teachers training program, focusing on skill development of students (Macro Considerations), increased budget for expansion of higher education facilities as well as for researches and improving the curriculum as well as the assessment method used in the higher education are the major points. Besides the government initiated a national capacity program called Ethiopian Education and Research Network (EthERNet) which is owned by MOE. In this project all the public universities are interconnected with 10GBit/sec network speed. Therefore, this is a clear indication that the education sector in Ethiopia is shifting their ICT paradigm from traditional data centers to cloud based services. So many researchers have proposed a central hybrid cloud model that can be used by all HEIs of Ethiopia.

Higher education institutions such as University of California, Washington State University's School of Electrical Engineering and Computer Science and other educational institutions from UK, Africa and around the globe have been using the cloud in the education sector and their affirmative feedback made the cloud very promising for this sector [8].

Cloud computing can benefit universities with advantages such as academic staff can focus on education and research activities instead of dealing with the complexity of high performance computing systems, Ethiopian universities can reduce IT capital expenditures and decrease ongoing operating expenses paying only for the services they use or sometimes the service may be free, can be easily implemented or deployed by any university, resource allocation can get bigger or smaller

depending on demand and cloud customers can provision cloud services without going through a lengthy process in addition to this Cloud services have standardized APIs, which provide instructions on how two application or data sources can communicate with each other and this lets the universities more easily link cloud services together [9]. Therefor to summarize, a cloud computing model will improve knowledge sharing reduces cost and time wastage as well as services can be easily accessed through electronic devices.

Despite of the above stated advantages, there are still some challenges that need to be overcome to leverage the full potential of cloud computing in this sector. These challenges include privacy and risk of identity theft when we say privacy in most cases sensitive data is stored out of the boundary of higher educations and this fact makes researchers and educators to hesitate to use cloud to store confidential data and also, they are very concerned about the protection of their intellectual property. Besides, admission process and online assessment services provided by cloud have concern with relate to validation, information, and identity theft.

The other problem is with related to unexpected denial of service so there has to be a strategy deployed to handle this situation. Besides users' data stored on cloud provider's infrastructure may be of value for onward selling to third parties anonymously.

## 2.5. Data security and privacy issues

Cloud computing comes with lots of advantages like increased collaboration, cost reduction flexibility and many more but there are still some issues that hinder cloud adoption in many organizations and cloud data security and privacy is the number one issue in hindering cloud computing acceptance since customers' data is outsourced and it is never known what is going on with the data.

When we say data in cloud computing it means any of the three types [10] which can be data in transit or transmission data, data at rest or stored data and data in processing. And cloud providers can secure their users data by using different encryption technique for the different types of data such that data at rest can be encrypted by using symmetric encryption, processing data can be encrypted using homomorphic encryption and data at transmission can be encrypted by using SSL encryption.

Data security issues are mainly caused because users do not know the exact place where their data is stored and how their data is handled. According to [11] some of data security challenges include the following.

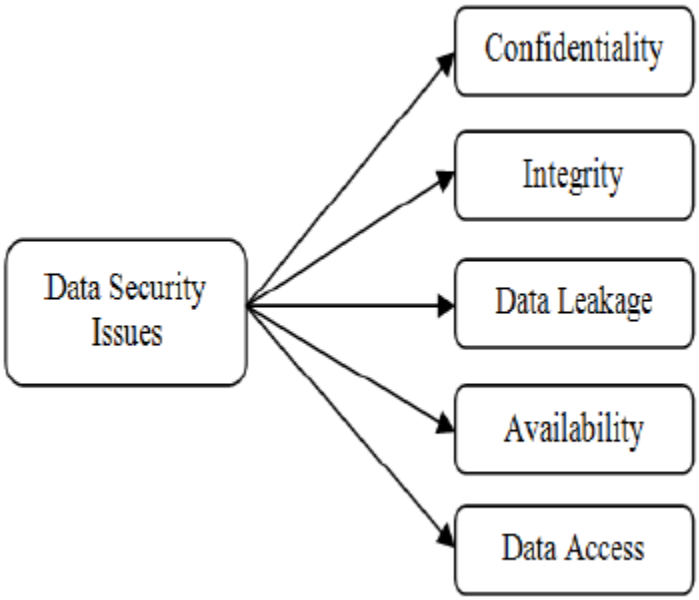



**Figure 2.2** Cloud computing Data Security Issues [11]

### 2.5.1. Confidentiality

Confidentiality refers to the prevention of the information from the unauthorized access by preserving authorized restrictions on access and disclosure. The users can get any service from the cloud and their records as well as different data that may be stored with single service provider or with multiple service providers. Whenever the information shared over the cloud is viewed or read by unauthorized individual then the question of confidentiality arises.

### 2.5.2. Availability

Availability refers to ensuring timely and reliable access to and use of information to the users. Availability is ensured when the resources and data are reliable and timely accessed by the users [12]. Redundancy, fail-over, Redundant Array of Independent Disks (RAID) even high-availability clusters can mitigate availability problems. Fast and adaptive disaster recovery is essential for the worst-case scenarios.

### 2.5.3. Integrity

Integrity refers to the prevention of intentional or unintentional alteration of the information by ensuring information non-repudiation and authenticity. Integrity should be maintained when authorized users can get the data access [11]. The cause of non- integrity could be human (unauthorized user) or non-human such as server crash or electromagnetic pulses. The integrity of the data can be checked by comparing the data present on cloud with the backup of original data, so the data is properly maintained or backup by the data owner or by third party for the auditing of the cloud data.

### 2.5.4. Data Leakage

Data Leakage refers to uncontrolled or unauthorized transmission of classified information from a Data Centre or computer system to the outside. To prevent data leakage techniques such as encryption, implementing strong API security measures, analyzing data protection and etc can be implemented.

### 2.5.5. Data Access

According to [12] data access is the other main issue related to security policies provided to the users. Here all cloud service users should not have equal privilege to access a data and also employees of the cloud provider are not given access to certain amount of data due to security policy consideration. And the other thing is a single cloud environment is shared for multiple organizations in this case there must be a clear organizational boundary with in that cloud model. [12] Indicated forward and backward access control can be used to control that which user can get access and which not so that authorized users can obtain the data access.

According to [11] if user data privacy is assured in cloud then customers trusts to use the cloud service will increase in other words insuring data privacy is significant in getting users trust. So, the question is how can we assure privacy? To answer this question, we must address the following questions that can help in determining the risks to information privacy in clouds.

1. Who are the stakeholders involved in a given operation on data?
2. What are their roles and responsibilities?
3. Where are data kept?
4. How are data replicated?

So, the possible privacy problems include data breach, data protection problem, data location problem, broken authentication and compromised credentials, data ownership and content disclosure problems [12].

### 2.5.6. Broken authentication and compromised credentials

Broken authentication refers the situation in which there is insufficient mechanism to validate users profile in a given system. Therefore, privacy on cloud services users may occur when the providers cannot confirm that access to data in the cloud is performed by authorized users. Therefore, absence of security measures may lead to compromised credential and cloud service providers' must assure an efficient authentication system to assure authenticity of customers' credentials.

### 2.5.7. Data breach

A data breach is the situation when sensitive or protected user data in the cloud is viewed, stolen or used by an unauthorized individual [12]. Data breach may be caused intentionally by stealing users' data or due to negligence and accident as well as improper use of Internet or improper disposal of data. Therefore, cloud providers should build an effective confidentiality measures that can help in monitoring the systems as well as quickly identifying the possible breaches and block them.

### 2.5.8. Data location problem

When data of organizations is stored in cloud data redundancy occurs in multiple physical locations and the organizations have no detailed information about the location of their data [12]. Because of this it is difficult to make sure whether adequate measures have been implemented to protect the customers' data. Besides, the customers cannot be sure whether legal or regulatory requirements have been met by the providers.

### 2.5.9. Data ownership and content disclosure problem

The other crucial privacy matter is data ownership and content disclosure this problem is caused when the cloud provider contractually becomes the data owners as well as data custodian. Such practice causes privacy issue in a way that [12] despite the lawful ownership, along with the right of disclosure being at the disposal of the original data owner, owner's right might be violated in a way that some cloud providers, as data custodians, retain the right of disclosure, while others do not.

There are different methods implemented to address the above data security and privacy issues. From those methods the major one is cryptographic concept.

## 2.6. Reference architecture for Higher Educations

The main objective of this research is to enhance the data security and privacy model of Ethiopian higher education institutions. Therefor to propose a new model we need to study as well as be based on architectures proposed and implemented in terms of different organizations worldwide.

The first one is the AWS shared responsibility model. AWS delivers service to different types of businesses including enterprises, educational institutions, and government agencies around the world. The customers include financial services providers, healthcare providers, government agencies, and educational agencies [13]. Here in AWS shared responsibility models both the customers and the service provider (AWS) work together to provide a secure application environment.

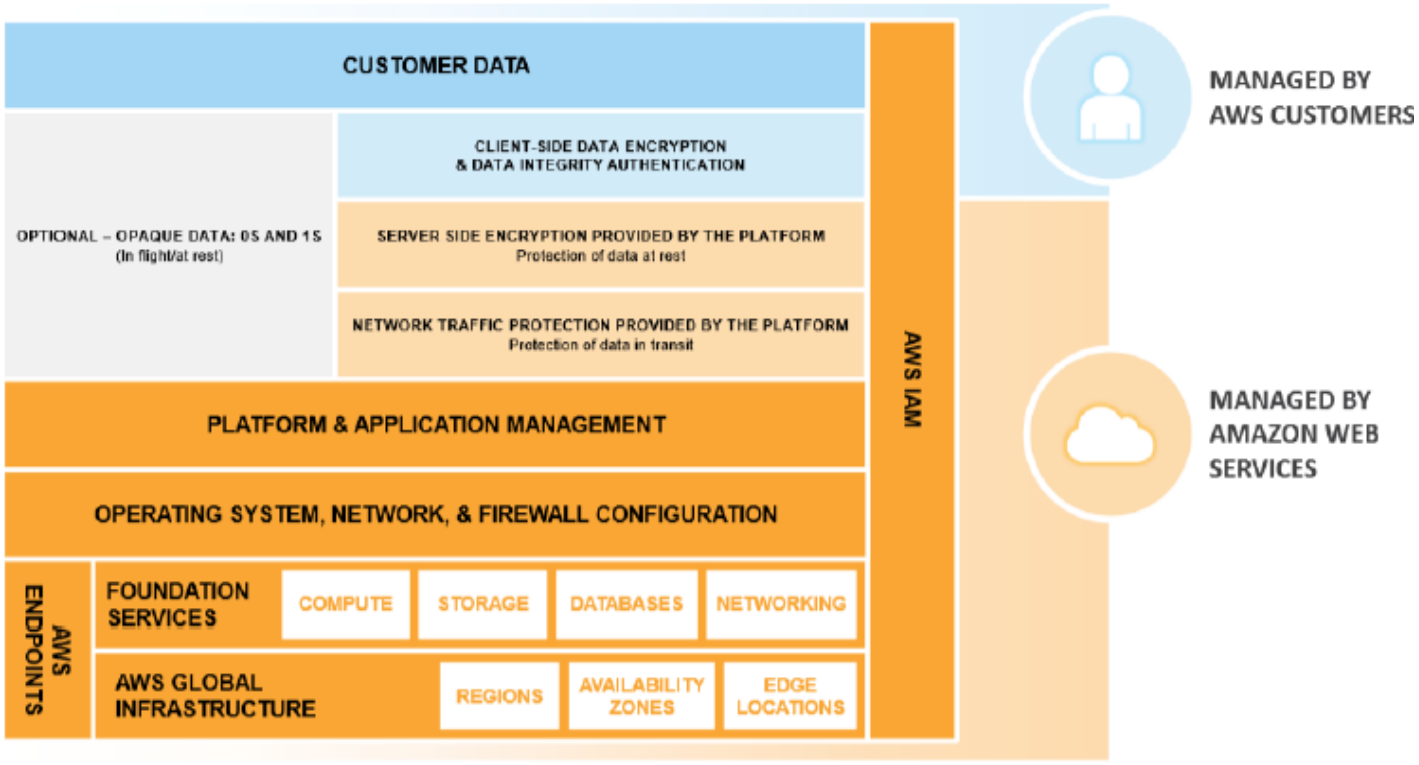



**Figure 2.3** Shared Responsibility Model for AWS abstracted services [13]

According to [13] AWS manages the underlying service components or the operating system on which they reside whereas the customer shares the underlying infrastructure, and abstracted services provide a multi-tenant platform, which isolates customer's data in a secure fashion and provides for

powerful integration with IAM. Therefor the AWS shared responsibility model gives a clear insight of the different security controls for different features so that it will be easy to design solutions that can meet data privacy and data security requirements and ensure protection of education data.

The other one is the NIST cloud computing security reference architecture this architecture identifies the major actors, their activities and functions in cloud computing. The actors are cloud consumer, cloud provider, cloud carrier, cloud auditor and cloud broker [14]. Each actor has their own role in transaction, processing or performing tasks in the cloud.

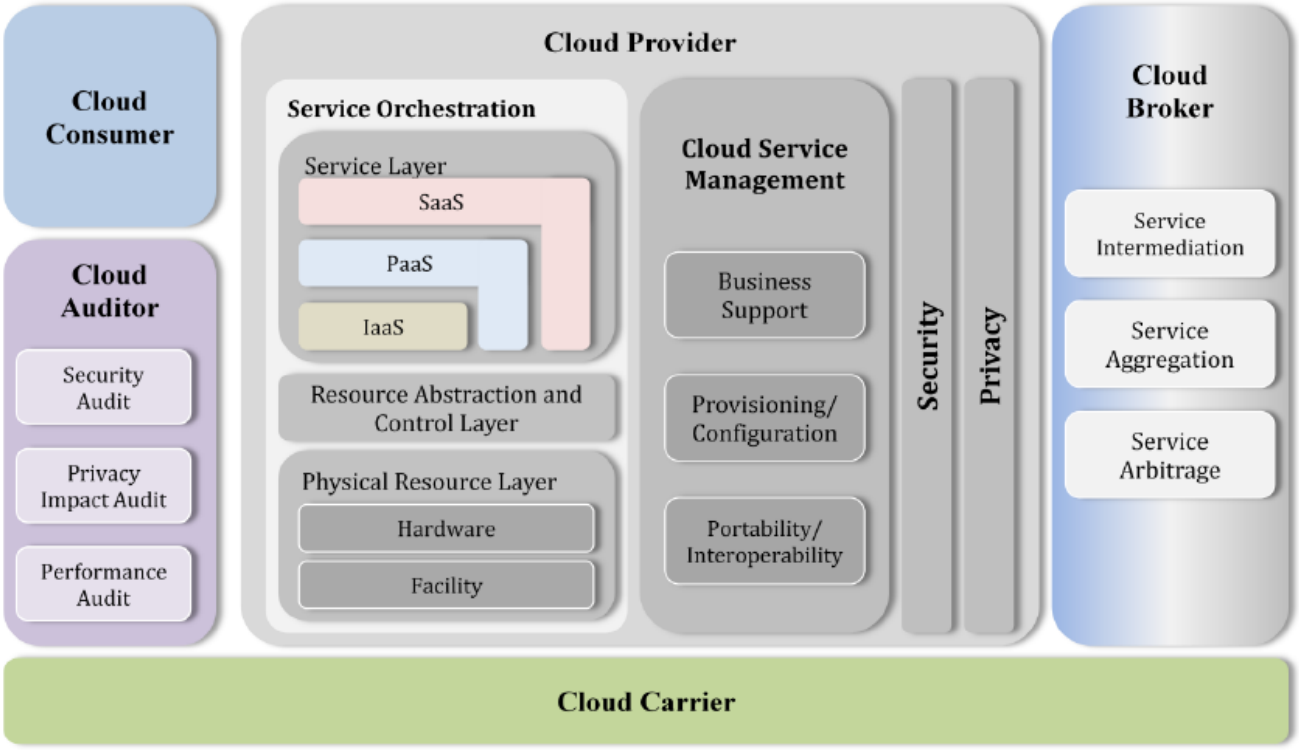



**Figure 2.4** The NIST cloud computing reference architecture [14]

The other most important reference architecture is developed by Cisco. This model focuses on the data center aspect of cloud computing and contains five layers [15]. The first one is data center technology architecture which contains all the services that are delivered for the cloud consumer such as network, compute and storage, secondly the security layer, thirdly the Service Orchestration layer which is an important layer that maps the technology components to the service components and serves as a reference point during service provisioning, fourthly service delivery and management layer and finally the consumer-facing layer.

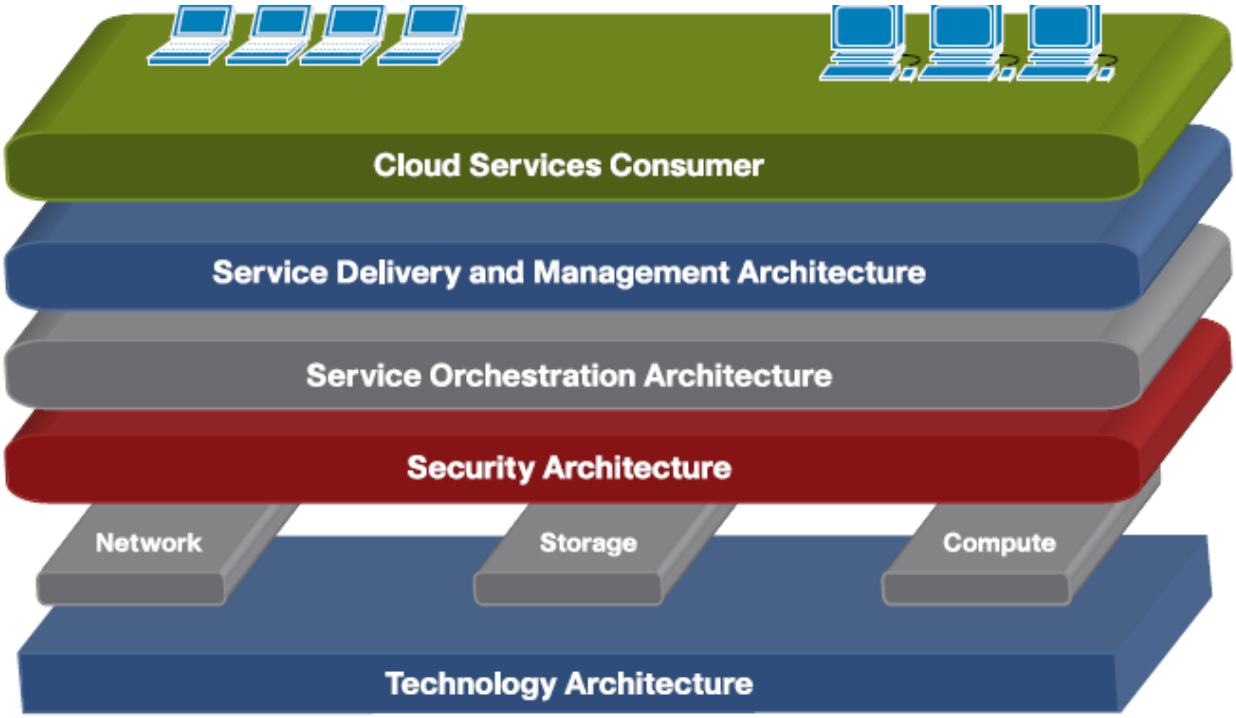



**Figure 2.5** Cisco Cloud Reference Architecture Framework [15]

So, the above three cloud computing reference models can be used as a base line for designing a security framework for Ethiopian higher education institutions cloud.

## 2.7. Related work

This chapter presents a review of some most related works with relate to cloud data security and privacy framework for the HEIs. We will try to analyze and identify gaps that exist in previous works and then we summarize the works reviewed.

Several models have proposed to improve the data security and privacy of a cloud environment. [16] Proposed a new security model that has implemented a combination of RSA encryption and digital signature techniques which can provide security for the entire cloud computing features like: PaaS, SaaS and IaaS so that secure communication and hiding of information from unauthorized users is achieved. The problem that this article addresses includes intruders attack on original data that may be caused because of the open accessibility of the data stored in the cloud and the most common problem that occurs during processing of individuals' data caused because users cannot possess the control over the place of data storage and poor resource allocation and scheduling facilities provided by the Cloud service provider at the time of processing.

The proposed framework result shows while RSA algorithm is used for both encrypting and securing communication MD5 hashing is used for digital signature and hiding key information to secure the entire cloud computing environment. The strength of this paper is problem such as slow downing the system that may be caused due to the RSA algorithm is overcome by executing each algorithm in different servers. The limitation of this paper is fact that this paper chooses RSA encryption algorithm to encrypt user's data. According to different research results this algorithm is not as storage efficient or processing-efficient as other algorithms studied and requires the use of longer key lengths for equivalent security [17]. Therefore, there is a gap in choosing and implementing the secure algorithm for data security and the other is there is no clear indication on who manages the encrypted keys.

The second reviewed solution is proposed by [18]. The main objective of this article is to propose a strong solution by combining hybrid encryption with digital signature so that cloud computing security gets improved. The encryption and decryption process of the proposed solution is shown as follows.

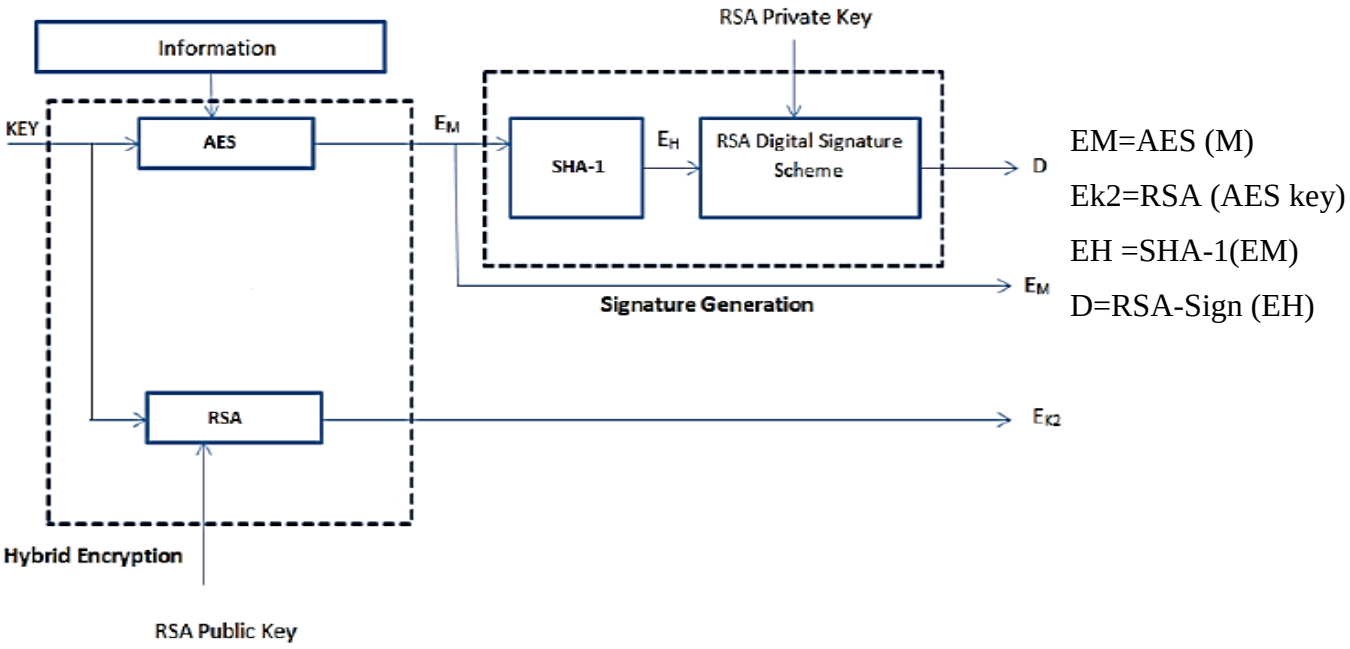



**Figure 2.6** Encryption of the proposed hybrid method [18]

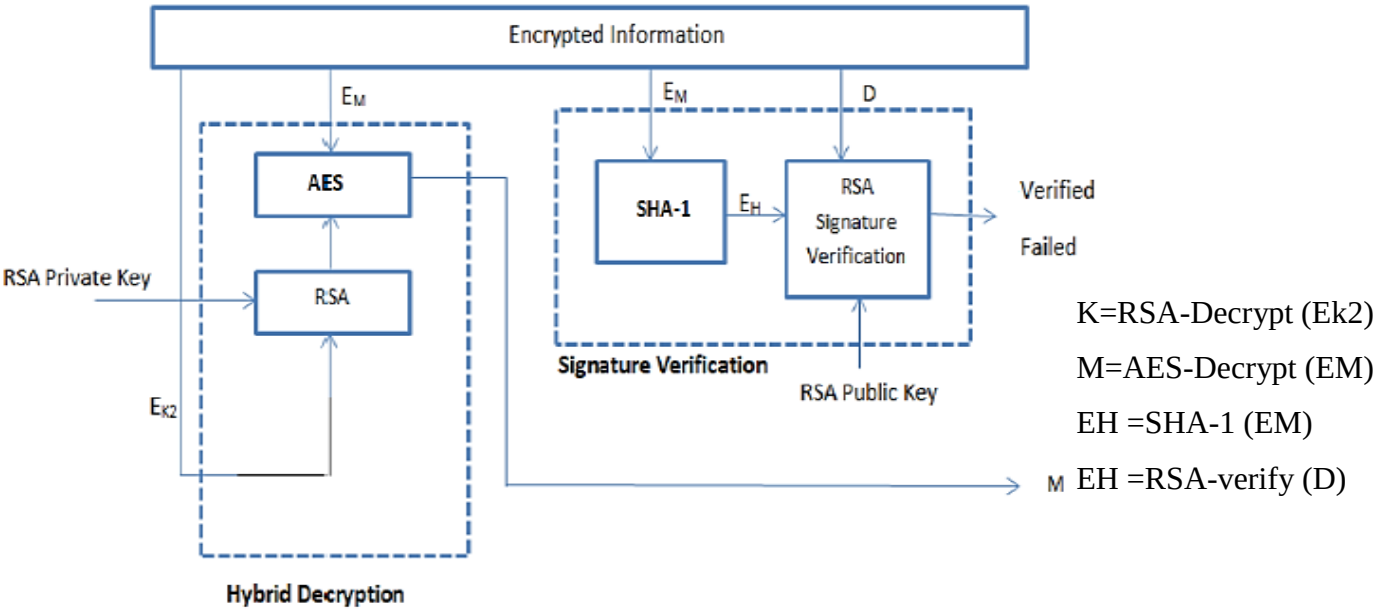



**Figure 2.7** Decryption of the proposed hybrid model [18]

The strength of this article is it uses both symmetric and asymmetric algorithms in the encryption process. AES algorithm is used to encrypt user data and provide privacy, RSA algorithm which is an asymmetric algorithm is used to verify user's identity and SHA-1algorithm is used to achieve integrity by producing the brief message [18]. Therefore, in this method data protection, authenticity and data integrity are achieved.

The limitations of this article is there is no clear indication on how the access policies are enforced for the encrypted file and also there is a repetition of RSA algorithm in this solution and RSA is not processing efficient by its nature and it takes high computation overhead during the encryption and decryption process.

In [19] the authors discussed a generic cloud computing security and privacy model that helps to satisfy security and privacy requirements and protect the cloud from various vulnerabilities and malicious behaviors. This work started by studying the most important aspects of secure cloud such as security and privacy requirements, different types of attack and threats cloud vulnerable to and risks and concerns about cloud security and end up by proposing a generic cloud computing security model.

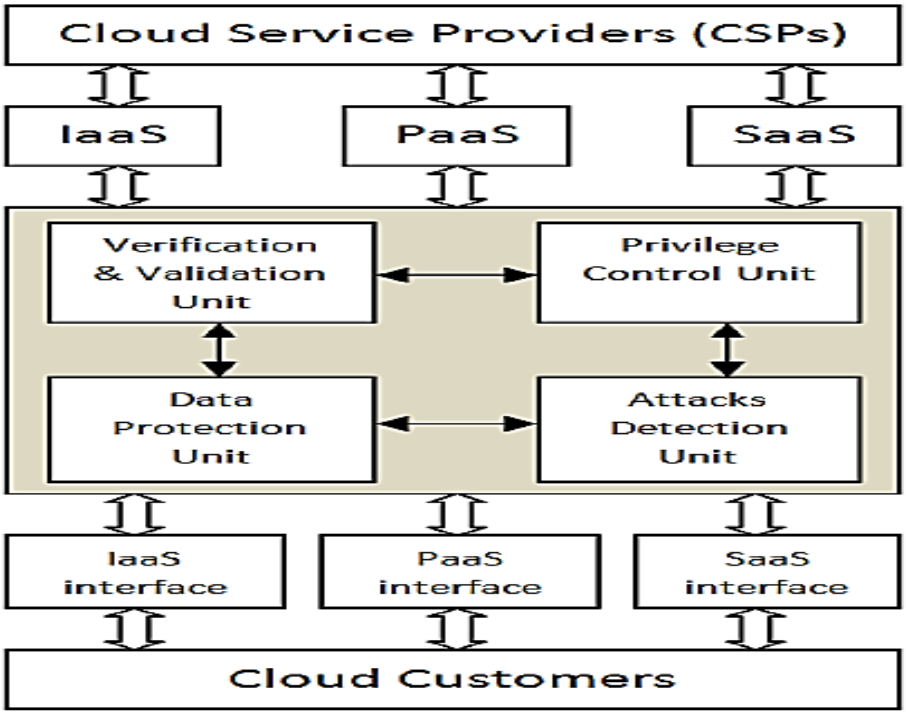



**Figure 2.8** A generic security model for cloud computing [19]

The above security model consists of four security units and each unit has its own importance. The first one is Verification and Validation (V&V) Unit. This unit is responsible in user authentication as well as data and service correctness on the cloud. The next unit is Privilege Control Unit which is responsible for users' privacy and ensures data integrity and confidentiality by applying rules and policies that controls who has the authority to do what on the cloud. The other unit is Data Protection Unit in this unit very sensitive and critical data such as patent information like patients' private information and their health history and banking information such as client account numbers, balances and Transactions must be protected by providing secure storage server.

# Chapter 3 Research Methodology

## 3.1. Overview

This research studies the possible technical data security and privacy issues that Ethiopian HEIs cloud users might face during storing, sharing or accessing of their data in the educational cloud. After studying several state-of-the-art solutions and data security frameworks a highly secured data security and privacy framework is proposed. In this chapter the research methodologies such as methods, techniques and tools used to complete the study are discussed. Data collection techniques such as semi structured interview questions, direct observation and document analysis were used to collect data. Tools that can be used to design and implement the proposed frame work are selected and discussed.

## 3.2. Research Design

A descriptive research design which is the type of quantitative designs is the best approach since Descriptive research measures the results at a moment in time and simply describes the sample's demography. Although this is not seen as a statistically robust or difficult exercise, a good description of the variables helps the researcher evaluate the statistical output in the proper context [20]. According to [21] quantitative research design uses a standardized measure, numerical values, have large sample size, and analyze data using statistical program. It is applicable to phenomena that can be expressed in terms of quantity or number. Therefore, a quantitative design is used to manipulate the data and to evaluate the prototype results on this study.

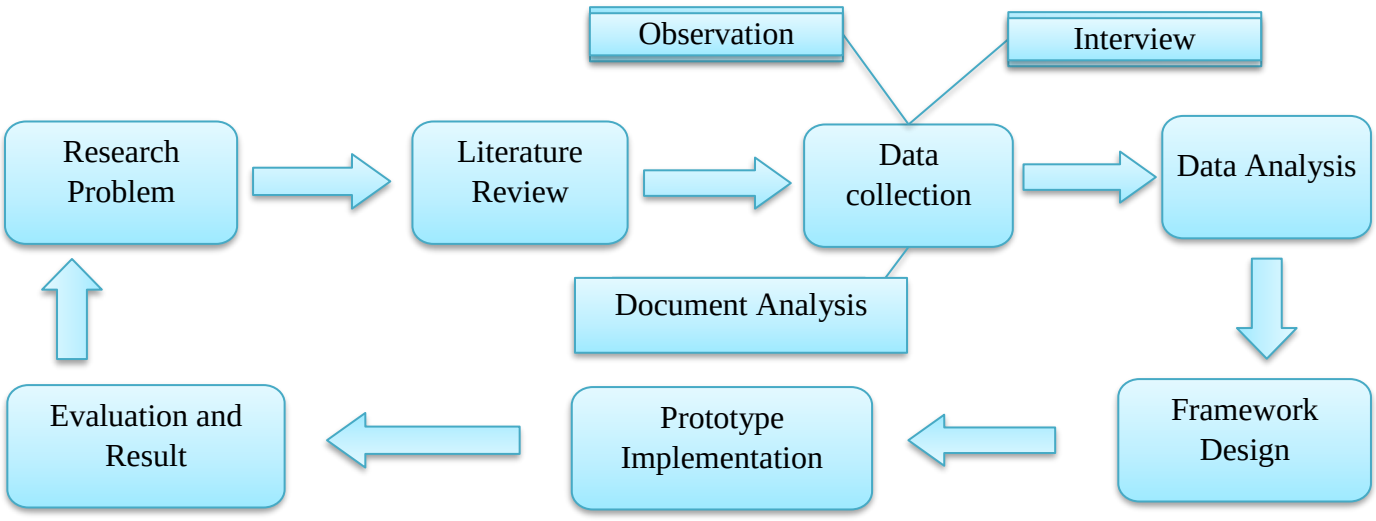



**Figure 3.1** Research Design process

The cloud computing architecture or framework is the structure of a cloud solution that defines cloud services and different components such as security, monitoring, operations and management. Any cloud architecture in general comprises of hardware resources such as the networks, storage and etc. as well as software components, services and the relationship between these entities.

During designing the proposed framework, we needed to keep-in-mind on how to maintain the essential characteristics of cloud computing. Therefore, researchers need to deeply visualize the possible solutions and decide for the optimum solutions by focusing on some design principles that are going to be discussed as follows.

- **Flexibility: -** the design needs to be flexible since as the security of a cloud increases the flexibility decreases and cloud is moderately about flexibility.
- **Performance: -** the performance of a system should be considered in a way that the method that we implement in order to secure the cloud data shall not have a negative effect on the performance of the education cloud.
- **Data protection: -** data in the cloud can be categorized as data in transit or data at rest. It is essential to protect data in every step by considering any state-of-the-art method and this principle will help as on achieving confidentiality of data.
- **Secure development: -** throughout the service development security must be strictly considered and we need to identify and mitigate threats to their security.

In addition to the above discussed principles the design also considers other principles such as Access control, Backup and recovery, Vulnerability and incident management, Security monitoring as well as Risk and compliance.

## 3.3. Data collection technique

In this research the primary data is obtained by using a semi-structured interview questions with predefined questions and observation. And secondary data were collected by using document analysis from different sources like conference articles, reports, thesis, journal and white paper from research links such as Google scholar, SpringerLink and etc…. the primary data collection enables to collect a written document such as organization's rule and regulations as well as how a data is managed in HEI data centers and the EthERNet to safeguard data security and privacy requirements of the cloud users and data owners and also how the current security solution can be enhanced. And the secondary data collection allows identifying methods and techniques to include in the proposed

framework for the higher education institutions to address the identified requirements. The data were collected from selected Ethiopian higher education institutions and EthERNet which is the countries cloud service provider.

## 3.4. Development tool

The different types of development tools used in this research are discussed as follows in this subtopic. This includes the tools that are used for formatting and documenting this thesis, tools used to design and draw diagrams and also tool that are used to develop and simulate the prototypes. The selection of the tools is based on the features they support, their simplicity to use, their compatibility and robustness as well as being the latest and advanced one.

**Edraw max 9.1:** is the easiest and fastest diagram software to create diagrams of any kind and support more than 200 types of diagrams.

**Eclipse neon. 3:** is the most widely used java IDE that can be used to develop Java, Web, PHP, Python, C++, IoT and many more computer programs. The main operations in this thesis are implemented by using this tool.

**AES Crypt:** is a file encryption/ decryption tool that uses the industry standard advanced encryption standard algorithm to easily and securely encrypt/decrypt files.

**Snipping tool:** is a screen shoot utility that allows taking the selection of results or any object on the computer screen. And in this thesis this tool is used to capture execution results and save it on the document.

## 3.5. Evaluation approach

The evaluation criteria for the proposed framework were prepared based on, the data security and privacy requirements identified are achieved or not, wither proposed framework achieves the required objective or not and by examining the required storage and computation overheads during implementing the prototype. Finally, the evaluation was performed by the developer during the implementation of the prototype and results are presented in chapter 6.

# Chapter 4 the Proposed framework and Discussion

## 4.1. Overview

This research studies the causes that refrained, the wider adoption of cloud computing in higher educations which is security and privacy of user's data and provides a framework to enhance the security and privacy of users data that is shared and stored on the cloud by allowing the data owner to control his/her data as well as enforce the security measures. So in this chapter we tried to describe the existing scenario of the Ethiopian universities cloud ecosystems with relate to the security and privacy of data on the cloud and the proposed solution to enhance the security and privacy of data stored and shared on the cloud. Data gathering conducted through a structured interview questions and observations were analyzed in this chapter. In addition to this existing framework of the Ethiopian universities cloud were reviewed for suggestions on the proposed cloud data security and privacy for this ecosystem here in this chapter. The analysis of the existing framework was based on the interview of ICT directors from selected universities in Ethiopia and team leaders of the EthERNet which is the countries education cloud provider.

## 4.2. Data analysis

### 4.2.1. Interview Analysis

In this thesis a semi-structured interview questions were prepared for EthERNet team leaders and ICT directors from selected higher education institution of Ethiopia. The interview question has five sections, it starts by general and introductory questions and then it will go to the type of services given then the next section focuses on the possible security issues the current system faces and then about the security strategy being implemented and finally conclusive questions on how to improve the security for the future. The interview questions are attached at appendix I.

The interview findings are organized based on the research focus area and results are analyzed as follows.

The EthERNet was initiated in 2001 E.C. with the goal of building capacity of higher education institutions to share educational resources and research among member institutions locally and globally. This institution spends more than 1.2 billion ETB to construct its infrastructure project. And currently thirty-six universities and public institutes were interconnected with highly reliable fiber optic medium with a backbone of 40G connection speed and 10G accesses as it is seen at annex II.

EthERNet gives all types of cloud services for its clients such as Datacenter, DNS hosting for universities websites, emails, e-library and video conferencing system and again research collaboration platforms, students and stuffs developed social networking and other useful application such as SaaS which includes Version controlling system (Gitlab), monitoring system (ICing), deployed configuration management system (PVPPET), Automatic backup system (Bacula), domain controller (samba 4) etc… and also STaaS, BaaS, TEaaS are some of the services. Besides to the above listed EthERNet gives technical support such as trainings, ICT academies, project management and etc. in all areas of ICT.

From the interview analysis we tried to investigate on how the current system deals with different security threats and problems.

In general security issues are related to the three core technologies in which cloud computing relies. These are: -

**Application and Services**: - are technological means to access cloud computing services.

**Virtualization**: - is the other main technology in cloud computing that allows giving SaaS and PaaS by virtualizing the infrastructure provided at IaaS level.

**Cryptography**: - is the common technique to achieve an acceptable level of security in cloud computing.

Therefore, different security techniques were implemented in the current system which includes both logical and physical security. Logical security includes securing network firewalls, running antivirus and anti-spyware programs, authentication, authorization and password are some to list and physical security such as restricted access and lock on servers. In addition to this a backup and restore system is implemented to be able to restore the data after an incident occurrence.

### 4.2.2. Observation analysis

During data collection we tried to observe what the infrastructure layout of the selected universities datacenters looks likes and it is verified by the concerned body and discussed as follows.

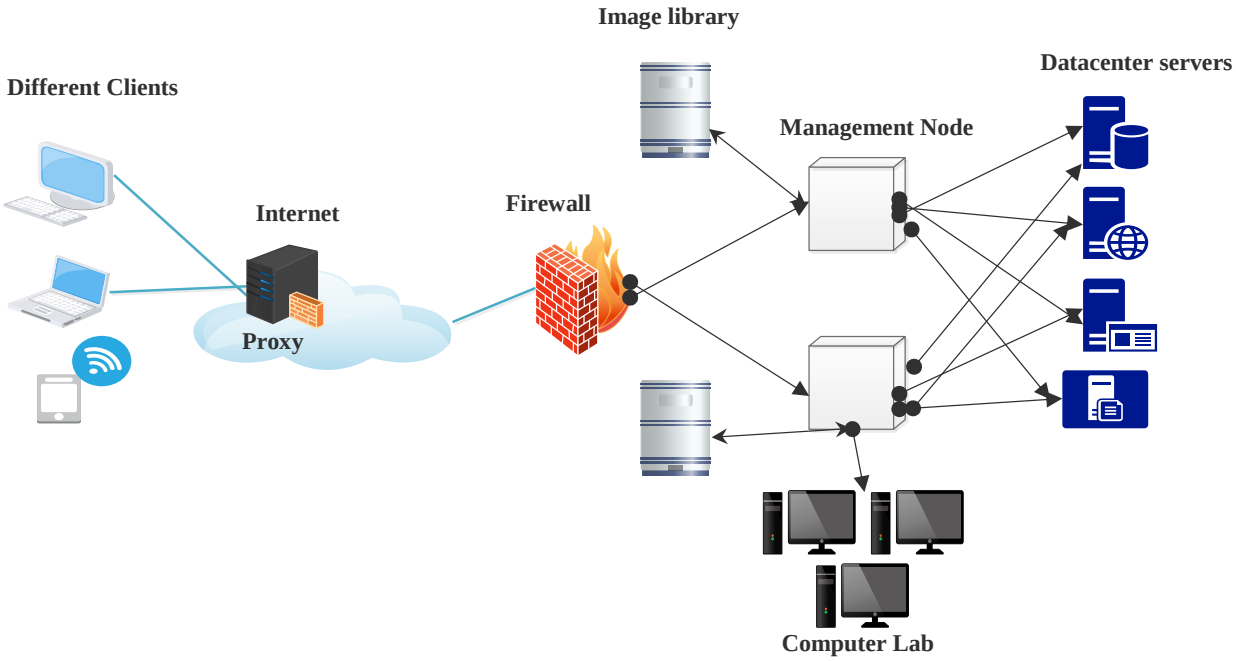



**Figure 4.1** The Current Infrastructures Architecture of a Datacenter

As we can see from the above Figure 4.1 the infrastructure layout of the selected universities is composed of three major components. These are management nodes, datacenter servers and computer labs.

**The management node:** this component processes requests, load nodes, reclaim nodes and easily adds provisioning engines such as physical deployer like XCAt and virtual deployer like ESXI and Xen. Each management node has image repository to keep image files, image metadata and profile of database.

**Datacenter server:** also known as a server farm which includes networked servers of file servers, database servers, application servers, DNS servers and etc... that are located in same place to accomplish the server needs. In addition to the above servers there is also a backup server which functions as a primary server during the event of primary server failure due to several reasons.

**Computer labs:** are also parts of the current cloud framework and are used to support large number of users and efficient desktop computers to store and compute.

So, in general the security in the current system is provided by the server which stores the data and the method used to protect the data as well as the entire management is controlled by the administrators of the server. This approach has its own limitations in a way that the security approach being used is system centric and if the core component is compromised the entire solution may be affected. In addition to this the current security approach being used couldn't provide the cloud users enough control of their data especially in terms of privacy. There are also some specific vulnerabilities issues such as unauthorized access of resources, data leakage problems caused due to virtual machine template images and this image might be used by attackers to get access of other customers resources and luck of efficiency in key management and random numbers used in cryptographic operations. Therefore, there is a clear limitation in the protection of client data.

Therefore, the proposed framework improves the data security and privacy problem of Ethiopian universities cloud users based on the security concept called data centric security approaches that can allow the cloud data owner to manage its data based on security services. A detail discussion on DCS approach as well as the proposed framework and different modules in this solution are discussed as follows in the next subtopics.

## 4.3. The Data Centric Security approach

### 4.3.1. Overview on DCS approach

In cloud computing the client's data are stored on virtual storage of cloud service providers infrastructure and weather it is the SaaS, PaaS or IaaS model the client uses the hardware related to the storage and processing of their data that is owned by the cloud service provider. Therefore from clients' perspective their data is the most valuable asset in the cloud. Cloud service may be prone to security attacks like any other service on the internet and we need to protect the clients' data from privacy comprise since the effect might be lifetime and can be hard to recover. And the clients data not only faces threats from outside the cloud in which the data is stored but it also faces an incident the cloud (internal) threat such as malicious users using same cloud or the file can be moved, processed or even altered without the knowledge of the data owner. In addition to this the clients experience limited control over their data whereas the provider has excessive control over the data and this may cause a privacy concern on the data owner. Therefore, a security solution based on

DCS approach is suggested in this thesis to address the security issues of the cloud and this approach focuses on securing user's data to the data itself.

In general, the data security solution for a cloud computing model can be classified based on two criteria. The first one is at which level the security is provided and this means the security solution may focus on particular level such as hardware or virtual machine level or operating system level or on the data level. Therefore, when data security solution focuses giving security from the data itself it is considered as the DCS. And the second one is based on who is the responsible to provide the security. In this case the responsible body for the data security might be either the service provider or another trusted third party or like in DCS approach the data owner is responsible for the security and the entire security management of its own data.

Several researchers describe the concept of DCS approach diversely and some of them are discussed as follows. According to [22] the DCS is "the enabling and enforcement of data specific security policies" and relating to this approach three aspects are discussed and these are primary authorization which means only authorized users and programs get access to data, secondly dissemination meaning authorized users are not allowed to violate the data's confidentiality policy i.e. by sending the data to unauthorized user and lastly life time policy enforcement irrespective of the user or application accessing it.

The other description of DCS is given by [23] and states that "The active data-centric framework facilitates data to defend external attack even though it is located in an untrusted environment. Comparing to computing-centric or application-centric data protection models, some verification and data operation functions in data-centric framework is executed by data rather than a third party or service provider."

In [24] they identify the desirable features for ensuring a secure and privacy-preserving data service in the cloud by focusing on the critical features of ensuring data confidentiality and access privacy, as they usually conflict with the normal functioning and performance of data services in the cloud, and thus give rise to numerous research challenges.

Therefore, by referring the above research works the DCS approach can be defined as a security approach in which the data owner is responsible for the security requirement of his own data and all the information required for the protection are attached to the data.

### 4.3.2. Characteristics of the DCS approach

In this subtopic the characteristics of the DCS approach is discussed and this leads to a clear view of this approach and its conceptual requirement.

**Data set:** -a data set can be defined from security perspective as the collection of information that can be protected and securely threated as in dependent of the data. And the data stored in the cloud can be generally categorized as either unstructured data which means a data that do not follow a predefined model when it is stored such as a document, image and video files or structured data in which the data have a predefined model and represented strictly. Example can be a database which stores data in a table format or semi-structured data in which they use tags or other markers to separate semantic elements but not records to enforce hierarchy. Xml, HTML and email are Good example for semi-structured data. So the term data set is used to refer to any of the above forms of data and the DCS approach needs to specify to which data set format the approach will be implemented.

**The data life cycle: -** refers to the entire stage of any data set starting from its creation stage to the stage it becomes useless and destroyed. The different stages are discussed in the following figure.

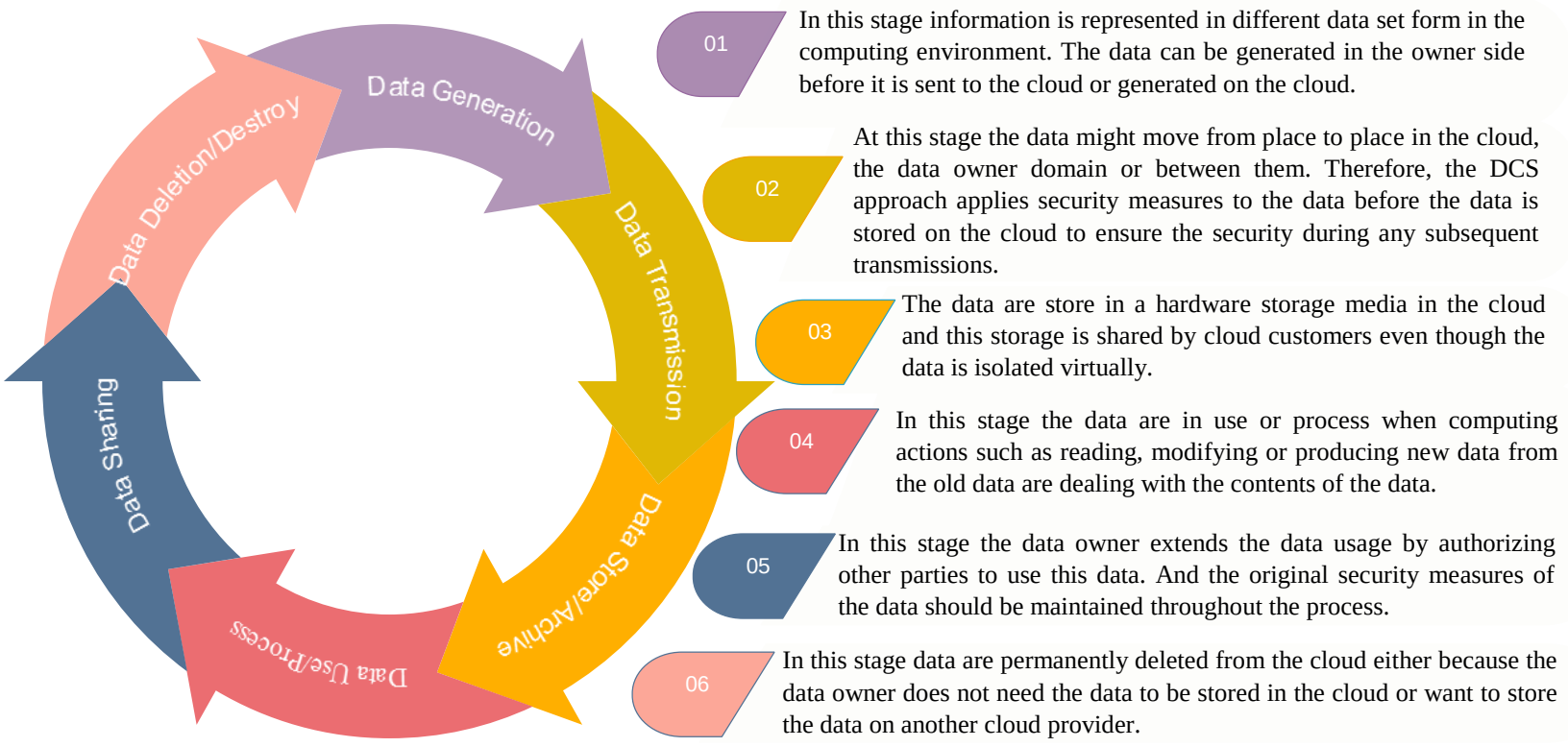



**Figure 4.2** Data Lifecycle Stages

**Data classification: -** in DCS approach the data classification is based on the data owners' security requirement. For instance, the data can be simply classified as top secret, confidential or secret and based on this the access control policies and security properties to handle the data are provided.

**Data history or tracing: -** the data history covers the entire life cycle of a data set starting from the generation stage to dissemination and every modification, usage actions or every transmission that happened at the middle, so that the data owner will be able to trace their data in the cloud at any time.

**Access policies and enforcement: -** the concept here is controlling who or what can access the data with what granted rights. In the DCS approach the data owner provides the access control policies and mechanisms so that the cloud service providers will comply with that.

**Self-protection: -** self-protection refers to the protection of the data from any an authorized entity or even from the cloud provider. To achieve this data are encrypted by a sufficient encryption technique and remain encrypted until they are transferred to a totally trusted domain and decrypted by authorized users. The encrypted data is packaged with protected secret key and the detail usage policy.

**Data integrity verification: -** integrity of the data in the cloud should be verified because there might be an intentional or accidental modification of data during the data life cycle and in the DCS approach verification of the integrity of the data is embedded in the data.

**Privacy and confidentiality: -** the risk of privacy and confidentiality increases in the cloud model due to several reasons such as an increased number of users involved in same cloud, the cloud data control is delegated to the cloud service provider and lack of hardware separation are some to list and protection of privacy is more than protecting the actual data from unauthorized access it also need to prevent leakage of any information during handling of the data. Therefore, the DCS approach takes in to account the confidentiality of users' data and privacy of the cloud consumers.

**Availability: -** refers to the availability and accessibility of cloud services for authorized users based on the request or requirement. In addition to this authorized user should be able to obtain data from cloud at any time. The cloud service provider should have an effective backup and archive system to protect data availability.

## 4.4. The proposed conceptual secured framework for the universities

In this study a more enhanced cloud data security framework for Ethiopian HEIs cloud is proposed. This framework design is composed of three functional units or components. The framework design with its three components and the workflow and the integrity of these components to provide security and privacy of users' data is clearly discussed as follows.

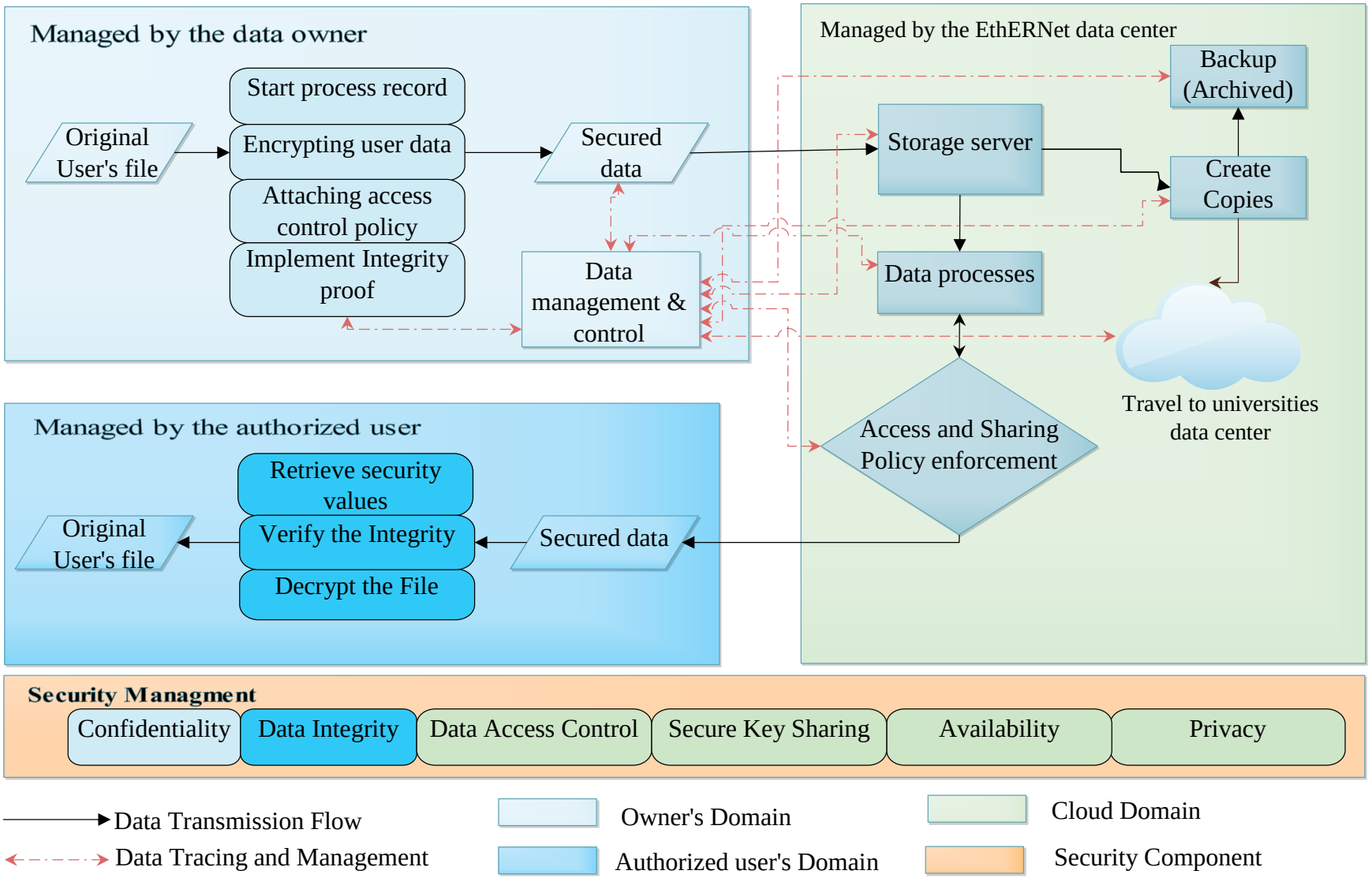



**Figure 4.3** Overview of the proposed data security and privacy Framework

### 4.4.1. The Data owner Module

This is the first module of the proposed framework and this module is fully managed by the data owner. The major activities in this module includes first classifying the owners' file based on their security requirements and creating the secured file by attaching the required security parameters such as the access control policy as well as the key sharing, the integrity proof and etc. in this module the final output is to produce the secured file that is going to be outsourced and shared on the cloud. The data management and controller enable the owner to control the location, usage and who accessed the data in the cloud environment throughout the data life time. The major activities performed in this module are discussed in the following sub topics.

#### *4.4.1.1. Data encryption*

In this proposed solution data encryption is the first activity performed by the data owner. In this stage the data is encrypted by using the well-known symmetric encryption algorithm called AES 256. The users' data which can either be a text or document file, video file, audio file or image file will be used as a resource or input file and these users' data will be encrypted by using AES algorithm and a secret key ($S_k$). In this way the original data will be encrypted and prepared to pass to the next stage. The secret key ($S_k$) passed to the next stage to be encrypted by the asymmetric method with the authorized users public key so that it can only be accessed by an authorized user.

The AES encryption algorithm is selected due to the fact that this encryption algorithm is more efficient in handling and encrypting large amount of data without causing system overhead and this encryption algorithm can be implemented in different key length in our cause we use the 256-bit key length which is the strongest of this algorithm.

#### *4.4.1.2. Access control and key sharing module*

Access control and key sharing stage is the second core activity in the data owners' module of the framework that allows the secure sharing of users' data stored in the cloud environment. To achieve this Chinese reminder theorem (CRT) is used. The Chinese reminder theorem (CRT) was first discovered in ancient Chinese mathematics manuscript for analyzing arithmetic problems [25]. This theorem is also used to create calendars during the first century.

**Theorem a:** Chinese Remainder Theorem

For any given integers, $a_1$, $a_2$, …, $a_k$, such that $0 \leq X < N_i$ for every i then , there is one and unique solution X, such that $0 \leq X < N$ and the remainder of the Euclidean division of X by $N_i$ is $a_i$ for every i. this can be written in short as follows.

$$\begin{aligned} X &\equiv a_1 \ (mod\ n_1) \\ X &\equiv a_2 \ (mod\ n_2) \\ &\vdots \\ X &\equiv a_k \ (mod\ n_k) \end{aligned} \tag{4.1}$$

Where the $n_i$ are pairwise coprime and $N=n_1n_2….n_k$.

The unique solution X can be computed by the following equation

$$X = \sum_{i=1}^{k} a_i M_i \, M_i^{-1} (mod\ M) \tag{4.2}$$

Where

$M= n_1 n_2 …. n_k$

$M_i=M/n_i$

$M_i^{-1}$ is the multiplicative inverse of $M_i$ mod $n_i$, such that $M_iM_i^{-1} \equiv 1$ (mod $n_i$)

$M_i$ is relatively prime to $n_i$ and there is a unique multiplicative inverse in mod $a_i$. Hence, calculating the multiplicative inverse in modular is an essential part of determining the CRT solution. Therefore, in this proposed framework we used the Extended Euclidean Algorithm (EEA) which is the essential component of Garner's algorithm to compute the multiplicative inverse and the Garner's algorithm which is generally used to find the CRT solution.

The Garner's algorithm used in the proposed solution to calculate the CRT is discussed as follows [26].

Input: Positive Integers $n = \prod_{i=1}^{k} n_i > 1$, where gcd($n_i$,$n_j$)=1 for all i=j and a modular representation $a_i=a_1,a_2,…,a_k$ can be any integers.

Output: an integer X in radix b representation

1. For i=2 to k do:
   1.1. $c_i \leftarrow 1$
   1.2. For j=1 to (i-1) do
      1.2.1. $U \leftarrow n_j^{-1} \bmod n_i$
      1.2.2. $c_i = u \cdot c_i \bmod n_i$
2. $u \leftarrow a_1, X \leftarrow u$
3. for i=2 to k do:
   3.1. $u \leftarrow (a_i - X) \cdot c_i \bmod n_i$, $X \leftarrow X + u \cdot \prod_{j=1}^{i-1} n_j$
4. return (X)

where $n_j^{-1}$ is the multiplicative inverse of $n_j$ in modular $n_i$.

The best feature about the CRT is that during adding or removing a module from an existing congruence the new solution requires a simple calculation. For example, if a new module is added on the solution X of Equation (4.1) then the new solution X′ can be found by computing the following two congruencies.

$$X' \equiv X \bmod n_1 n_2 \ldots n_k \text{ and}$$
$$X' \equiv a_{k+1} \bmod n_{k+1}$$

In the same way if a module is removed from the solution X of Equation (4.1) the new solution X″ can be calculated by the following operation. [27]

$$X'' \equiv X \bmod n_1 n_2 \ldots n_{k-1}$$

This feature can be used in the proposed solution during granting a new user an access right to a certain resources.

CRT is used because of its arithmetic nature that doesn't consume high computational resources. There for CRT is suitable with limited computing resource environment. The other reason is because CRT is best known for its secret sharing in cryptography, [25] states that CRT allows to share a secret by deriving it from a set of shares that can only be recovered with a certain prearranged set of values. In addition to this CRT can be implemented in different areas for authentication and access control schemas such as wireless sensor network authentication and in mobile ad-hoc networks to provide security without causing overhead on the available resources.

In this stage the secret key used to encrypt the users data is encrypted using RSA algorithm with the users' public key. The encryption also takes another value called Cr (secret value) among authorized users. So let us see how the access control and key sharing of each user's confidential file is enhanced.

$$a_i=(\mathbf{E_{pubi}}(C_r||K_s)) \text{ for } i=1,2,\ldots, k \quad (4.3)$$

Where k = is the number of authorized users.

$a_i$ = the resulting cipher text

$\mathbf{E_{pubi}}$ = RSA algorithm with the public key of each user

$C_r$ = secret value used to enforce access control

$K_s$ = secret key used to encrypt the file

Then the resulting cipher text ($a_i$) is used as an input to the CRT algorithm to calculate the shared value Xr and $a_i$ is substituted in eq. (4.1) to produce the following congruence.

$$\begin{aligned} X_r &\equiv (E_{pub\,1}(C_r||K_s))\ mod\ n_1 \\ X_r &\equiv (E_{pub\,2}(C_r||K_s))\ mod\ n_2 \\ &\vdots \\ X_r &\equiv (E_{pub\,k}(C_r||K_s))\ mod\ n_k \end{aligned} \quad (4.4)$$

Then the result value $X_r$ (the shared value of the resource r) is attached to the protected data and the protected data will be stored in the cloud. When authorized users tries to access the secured resource, the cloud server sends the shared value ($X_r$) to the user and the user decrypts the shared value to produce $C_r||K_s$ by using the corresponding private key.

$$C_r||K_s = \mathbf{D_{prv\,i}}(X_r \bmod n_i) \quad (4.5)$$

Where $\mathbf{D_{prv\,i}}$ = decryption using the private key of user $u_i$.

$n_i$ = the relative prime number specific to the user $u_i$.

The user sends back the value $C_r$ to the server to prove that the user is authenticated to access the secured resource. Then the server sends the user the secured resource and the users decrypt that resource using the key Ks and allows to see the content.

#### *4.4.1.3. The Integrity checking stage*

In this stage the integrity protections for the encrypted data is created. The encrypted file that comes from data encryption module is hashed using SHA256 algorithm and then the resulting value is digitally signed with the data owner private key. In this way the authenticity and integrity of users' data is preserved without depending on the security provided by the cloud service provider. In addition to this the data contains all the necessary integrity and authenticity information after passing through this module and this allows authorized users to prove the integrity of the data that they want to access is right easily.

#### *4.4.1.4. Creating the Secured file*

This stage is the last activity in the data owner module of the proposed framework. It creates a secured file by combining the security measures taken in the previous modules at each user's data. This stage indicates the solution that enhances the security and privacy of a cloud computing environment.

### 4.4.2. The cloud module

The cloud module is the second module of the proposed framework. The major activities performed in this module includes storing the secured file created in the data owners environment and keeping multiple copies as well as a backup of these secured files in order to increase the availability during sharing and accessing of these files as well as during unexpected incident occurrences. The other main activity performed here is during when the authorized user request for accessing the secured file. The access control is enforced in the cloud domain even though it is defined by the data owner and the security parameters are attached to the data itself in the owners' module. So the role of the cloud provider is to enforce the access control policy without revealing the security details and the original data. This module is managed by the EthERNet.

### 4.4.3. The Authorized user module

This module is the third module of the proposed framework. To download and access the secured file stored and shared on the cloud, the authorized user must first request the cloud server for that file then the cloud server verifies the access right of the user based on the access policies attached on the secured file and decides whether to grant or revoke access permission. Authorized users are able to

download the secured file and then verify the integrity based on the integrity proof attached on the file to check the originality of that file and decrypt back to the original data.

As we can see from the figure 4.3 the security measures taken in each module are attached to the data so that the proposed solution is able to preserve the security and privacy of users data from not only unauthorized users but also from the cloud service provider itself. The proposed data security and privacy framework is expected to satisfy the security requirements described in the security management module as follows.

- **Data confidentiality:** the data confidentiality is going to be preserved in a way that the data is encrypted before it is outsourced to the cloud and the protection/encryption is implemented by the data owner on the owners' environment. And the encrypted data cannot be undone in the cloud or any unauthorized environment. And this allows the data to remain secure even on times when the cloud server is compromised from inside or outside.

- **Data Integrity:** the data integrity is going to be preserved by implementing strong cryptographic methods such as hashing the file with SHA 256 algorithm and then encrypting the resulting digest by RSA algorithm. In this way the integrity proof is expected to get implemented and authorized users can verify the integrity of each file individually.

- **Data availability:** the proposed framework improves data availability by allowing the data owner and the cloud provider which is the EthERNet to store the secured file in multiple copies in different HEIs data centers located in different geographical locations. Therefore the data is available even if the server in one location fails.

- **Access control policy, key sharing and users as well as data privacy:** are going to be preserved as the policies are hidden inside the data and even the cloud provider won't be able to compromise it since the security parameters are defined and managed by the data owner. Data access is performed without the necessity of exposing sensitive information to the cloud provider. Low key management and sharing overhead is expected since the keys are securely attached to the data and authorized users extract these keys during secure access enforcement. CRT algorithm combined with RSA encryption algorithm allows to implements the access control and key sharing in one mechanism.

- **Privacy:** is preserved in a way that data can be accessed by authorized users without revealing any users' credentials, which can be used to decrypt the file, to the cloud provider.

- **Data leakage:** this problem is resolved since the data outsourced to the cloud is secured by multiple security perimeters. Although the secured data and its security parameters stored in the cloud may be leaked but its content cannot be accessed without breaking these multiple security parameters attached on the data.

# Chapter 5 Implementation and Result Discussion

## 5.1. Introduction

The proposed data security and privacy framework for Ethiopian universities is based on a data centric security approach. It insures the confidentiality of the data by encrypting each data using symmetric encryption algorithm. In our case we used AES encryption/decryption algorithm. In addition to this the secret key sharing as well as access control of the data are combined and implemented by a mechanism called CRT which results low computation and management overhead during accessing the resources. And also, in this proposed solution an integrity verification mechanism is implemented for each data owners file. Finally, all this security parameters are attached to the data file and used to produce the secured file. Therefore, the proposed solution enhances the privacy and security of Ethiopian higher education cloud users' data by preserving the integrity authenticity and privacy of users' data out sourced to the cloud.

In this chapter prototype for the security modules of the proposed framework along with the executions results are discussed. The implementation has two major parts. The first one is from the data owners' side and this includes all the operations required to create a secured file. The second one is from the authorized users' side which includes the operations required to securely access and share available resources among authorized users. Finally, a summary for this chapter is provided.

## 5.2. Experiment Environment to Implement the Prototype

The simulation tool used to implement the proposed solution is a java language with eclipse IDE for java developers combined with AES crypt encryption tool. In the implementation of the encryption decryption of the data owners file AES 256-bit key length which is the strongest of this algorithm is used. In this way the confidentiality of sensitive users' data is enhanced. The detailed step on how to use as well as the result from this tool is discussed later in this chapter.

The experiment environment used to test the solution from both the data owner side and from authorized user side is a laptop computer with the following specification.

- Windows 10, 64-bit operating system
- Intel core i5 CPU @ 2.6 GHz processor
- 8.00 GB RAM installed memory

The implementation for the data owner side includes the operations for creating a secured file before it is outsourced to the cloud and AES crypt is used for encrypting users' data in this side. Whereas for the authorized users side the operation includes calculating the symmetric key and secret value from the shared value using equation (4.5) and decrypting the data from the secured file that contains the encrypted data and in this side AES crypt is used for decrypting back the data file. Finally, the computation and storage overhead during creating and accessing the secured file based on the above experiment environment is discussed.

## 5.3. Prototype implementation

The implementation includes many operations such as symmetric encryption of the data owners' file, computing the hashing function of a file, finding the CRT solution and asymmetric encryptions of keywords. These operations are performed based on two sides the data owner side and the authorized user side. The implementation is discussed as follows.

### 5.3.1. Creating the secured file from the data owners' side

Creating the secured file passes through major operations such as AES encryption of the original file, calculating the CRT solution then digesting the file using HMAC-SHA256 algorithm and finally RSA encryption and signing the encrypted message digest of the input data. Each of these operations is discussed in the following sub topics.

#### *5.3.1.1. Encryption of the original file*

Encryption operation is the most important operation to enhance the security of a data. Therefore, after classifying the data as confidential we choose to implement the symmetric encryption algorithm over the asymmetric encryption algorithm due to the efficiency of handling large amount of data. AES crypt tool is used to implement the file encryption and decryption process. This tool uses advanced encryption algorithm with 256-bit key length. AES is used because it is an adequate encryption algorithm with strong key length compared to other symmetric encryption algorithms.

AES crypt is an industry standard very easy tool to implement AES algorithm. Now let us see how a single input data is encrypted by using this tool.

- First the data owner right clicks on his sensitive data
- From the list of options that come click on AES Encrypt
- Enter and confirm the password and then click ok
- Finally, an encrypted file with an AES file extension will be created

In this way the data owner enhances the security of his data and what we noticed from the result is that the original file and the encrypted file has an exact same size when measured in MB which shows storage efficiency property and the process took in fraction of seconds which shows computation efficiency. Then the encrypted file created will pass to the next phase in which the access control and sharing protocols are defined for the file by using the CRT.

#### *5.3.1.2. Implementation of the CRT*

The CRT guarantees that under certain condition any system of linear congruence equations has a unique solution. The CRT states the existence of a solution but doesn't show how to drive the unique solution. Therefore, to implement this theorem Garners' algorithms that are extracted from the constructive proof of the CRT is used and the Extended Euclidean algorithm, is used to compute the modular multiplicative inverse.

The CRT allows us to share the two secret values, the first one is symmetric key which is used to encrypt the data owners' file and the second one is secret value which is used to enforce the access control of that given data owners' file. But before sharing these two values the values are encrypted by using RSA encryption algorithm with the public key of each authorized users as it can be seen in equation (4.3) then the resulting cipher text of each authorized user will be the input to the CRT as represented in equation (4.4) so that the shared value gets calculated. In addition to the above cipher text that is created from the RSA encryption the CRT requires relative prime numbers. $n_i = n_1, n_2, \ldots, n_k$ where k is the number of authorized users. The relative prime numbers can be generated once for all authorized users and then each authorized user gets assigned with a unique relative prime number.

The source code to implement Garner's and then the Extended Euclidean Algorithm (EEA) is available in [28]. However, the original code supports only 64-bit integer length and in our case the minimum length of the cipher text resulted from RSA encryption is 1024-bits therefore the original code needed to be modified to support a minimum of 1024-bits length. Therefore the code is

modified based on the class library java.math.BigInteger so that the code could support integers with length more than 64-bit.

The source code to find the CRT solution is discussed in appendix III, labeled Code for finding the CRT solution and the function CRTforBigInteger.findSolution(BigInteger[ ] a, BigInteger[ ] n) in this source code is used to find the CRT solution where a and n are the input BigInteger arrays of the $a_i$ and $n_i$ values. And in appendix III, labeled code to calculate the modular multiplicative reverse the entire code to calculate the modular multiplicative inverse is discussed. And the function cryptoBig.inverse(BigInteger p, BigInteger g) is used to calculate the result and the returned value of the inverse function is the $g^{-1}$ (mod p). The GCD of two integers can be calculated by using the function BigInteger p.gcd (BigInteger g) included in the Class java.math.BigInteger library. So by using this two source code the CRT solution and the modular multiplicative inverse are calculated.

#### *5.3.1.3. Implementation of the RSA Encryption*

The RSA encryption method is used to implement the encryption of the value Cr_Ks. According to [29] the java.security package provides several class libraries that include functions used for generating RSA key pairs and RSA encryption/decryption operations. From these two java class files were coded, the first one is GenRSAk.java which is used for generating RSA pair keys and storing them in a file and the other class is RSAforBytes.java which contains two functions the first one is rsaEncrypt(byte[] data, String PubKeyFile) for encrypting an array of bytes and the second one is rsaDecrypt(byte[] data, String PrivKeyFile) for decrypting this array of bytes. The source to generating RSA pair keys and store those keys in a file as well as the code to RSA encryption/decryption operations are discussed in appendix III, labeled code for generating Public and Private key Pairs and store them in a file and code for RSA encrypting/decrypting an array of byte respectively.

The above two codes allow us to generate the RSA key pairs and then to encrypt/decrypt the combination of symmetric key and secret key values (Cr_Ks) using the generated public keys.

Now let us consider the value of the combination of symmetric key and secret key values (Cr_Ks) to be “104133327533874159885472334501 20” then the first task is to convert the above big integer value to byte array. Since the rsaEncrypt function from RSAforBytes class programmed to compute data in the form of bytes array format. After converting the value to byte array the value is encrypted

by the public key of each authorized user. In our case we have generated the pair keys for five users and the below figure shows the implementation execution and result.

```
41              // The start time of RSA encryption
42              long starRSAenc = System.currentTimeMillis();
43
44              // the Cr_Ks value
45              BigInteger C_K = new BigInteger("104133327533874159885472334501 20");
46              long Cr = 1041333275; // the Cr value
47              System.out.println("The Cr value: "+Cr);
48              byte [] data = C_K.toByteArray(); //transforming BigInteger to Byte array
49
50              /* RSA encryption of the byte array Cr_Ks with each authorized users public key */
51              byte[] encData1 = RSAforBytes.rsaEncrypt(data, "public.key");
52              byte[] encData2 = RSAforBytes.rsaEncrypt(data, "public2.key");
53              byte[] encData3 = RSAforBytes.rsaEncrypt(data, "public3.key");
54              byte[] encData4 = RSAforBytes.rsaEncrypt(data, "public4.key");
55              byte[] encData5 = RSAforBytes.rsaEncrypt(data, "public5.key");
56              long endRSA = System.currentTimeMillis();; //End time of RSA
57              System.out.println("RSA encrytption took:"+(endRSA - starRSAenc) + " ms");
58              System.out.println("Length of Cr_Ks:"+data.length + " byte");
59              System.out.println("Length of encrypted Cr_Ks:"+encData1.length + " byte");
60  //end of encryption
61                          // The transforming the encrypted byte array of Cr_Ks (ai) to bigInteger
62              BigInteger a1 = new BigInteger(encData1);
63              BigInteger a2 = new BigInteger(encData2);
64              BigInteger a3 = new BigInteger(encData3);
65              BigInteger a4 = new BigInteger(encData4);
66              BigInteger a5 = new BigInteger(encData5);
```

Problems @ Javadoc Declaration Console

```
CreateSecureFile [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 12, 2018, 10:19:39 AM)
start encyption
Closed reading the Key file.
encrypted data Done
RSA encrytption took:928 ms
Length of Cr_Ks:14 byte
Length of encrypted Cr_Ks:128 byte
```

**Figure 5.1** code for RSA encryption of the Cr_Ks value by the public key of each five users

From the figure 5.1 we can see that the length of the Cr_Ks value is 14 bytes and the RSA encryption of this value by each user public key has a length of 128 bytes and the time it took to finish the encryption process is 928 ms. Then each encrypted value has to be converted back to big integer form so that it will be used as an input with a second input that is the relative prime number for each user ($n_i$) to calculate the CRT solution.

### *5.3.1.4. Implementing the integrity proof*

The secured file created from the above processes encapsulates the original data to keep it from unauthorized access when stored on a cloud server. The major operation here is to hash the encrypted file and then to digitally sign the resulting file digest by the file owners' private key. The SHA-256 hashing algorithm is implemented to calculate the file digest and the code is based on the

java.security package. The code implementation for integrity check is discussed on Appendix III, labeled code for integrity proof.

The function createFileHash inputs two values the file name and the name of the hash algorithm which is in our case the SHA-256 and returns the resulted file digest in the form of byte array format. Then the resulting file digest is encrypted by RSA encryption algorithm using the private key of the data owner to form the digital signature.

#### *5.3.1.5. Implementation to create the secured file*

The creation of the secured file by the data owner is the result of the above four major operations which are first encrypting the input data by AES encryption algorithm, then calculating the CRT solution and then RSA encryption and digitally signing the encrypted message digest. Therefore, creation of the secured file is the result of the above operations. To encrypt the owners file, we used a tool called AES crypt and the resulting encrypted file is used as input to create the secured file by the SecuredData.java class which is designed and implemented to create the secured file. SecuredData.java class code starts by initializing a value for some parameters and the code is discussed as follows.

```
public static void main(String[] args) throws Exception
{
    String Owner = "Sami"; //Data owner ID
    //add keywords
    String stringToConvert1 = "Cloud";
    String stringToConvert2 = "Network";
    String stringToConvert3 = "Privacy";
    String stringToConvert4 = "Integrity";
    String stringToConvert5 = "Security";

    int keyword_no = 5; //Number of encrypted keywords
    long starthmac = System.currentTimeMillis();// Calculate the start time

    //Encrypting the keywords with HMAC-SHA256 algorithm
    byte[] theByteArray = HMACSHA256.encode("keyforKeywords",stringToConvert1);
    byte[] theByteArray2 = HMACSHA256.encode("keyforKeywords",stringToConvert2);
    byte[] theByteArray3 = HMACSHA256.encode("keyforKeywords",stringToConvert3);
    byte[] theByteArray4 = HMACSHA256.encode("keyforKeywords",stringToConvert4);
    byte[] theByteArray5 = HMACSHA256.encode("keyforKeywords",stringToConvert5);

    //End time of encrypting keywords
        long endhmac = System.currentTimeMillis();
        System.out.println("Calculating HMAC_SHA of keywords took in ms:"+(endhmac - starthmac) + " ms");
        //End of encrypting keyword

        // The start time of RSA encryption
        long starRSAenc = System.currentTimeMillis();

        // the Cr_Ks value
        BigInteger C_K = new BigInteger("104133327533874159885472334501 20");
        long Cr = 1041333275; // the Cr value
        System.out.println("The Cr value: "+Cr);
        byte [] data = C_K.toByteArray(); //transforming BigInteger to Byte array
```

**Figure 5.2** code for creating the secured file

First parameter is to define the data owners ID and it's in the form of string and then the search keyword strings defined. In our case five keyword defined and then number of encrypted keywords defined then the above five keywords are encrypted by using HMAC-SHA256 algorithm. Then the symmetric key and secret key combination (Cr_Ks) as well as the secret value (Cr) is initialized and it has a type of bigInteger. After transforming the Cr_Ks big integer value to byte array we encrypted by using RSA encryption algorithm and public key of each authorized user then the files containing the public key for each of the authorized users is defined. For example, the Cr_Ks value can be encrypted by the public key of the fifth user and file containing the public key can be defined as,

```
byte[] encData5 = RSAforBytes.rsaEncrypt(data, "public5.key");
```

The result then transformed from byte array to biginteger value by the following code and used as an input to compute the CRT value by creating the elements of BigInteger a[ ] array.

```
BigInteger a5 = new BigInteger(encData5);
```

Then the relative prime numbers that is BigInteger n[ ] array elements are defined for the authorized users i.e. the relative prime number for n1(user 1) can defined as `BigInteger n1 = new BigInteger ("215143111331331113151515157719135111111");` and in this way the relative prime values are defined for all authorized users. After defining the above two bigInteger array values then we can calculate the CRT solution.

The other parameter defined was the file that contains the data owners' private key and this allows calculating the file digest and data owners' signature. The code is discussed as follows.

```
String PrivKeyFile = "private.key";
```

Finally the hashing algorithm used to calculate the file digest is defined and SHA-256 hashing method is implemented. The code to define the hashing parameter is `byte[] FileDigest = Hash.createFileHash(InputFile, "SHA-256");`

After defining the above parameters the code is executed. The first operation is to encrypting the Cr_Ks value by using the RSA encryption algorithm and with each user public key. Here the time taken to compute the RSA encryption is calculated and also the length of the Cr_Ks value after and before encryption is calculated. Then the length of relative prime integer, which is the other input to calculate the shared value, is calculated. Then it passes to the calculation of the CRT solution which produces the shared value Xr. The length of this value is calculated in bits. After the above

operations are executed successfully the program prompts the user to enter the AES encrypted file with its extension.

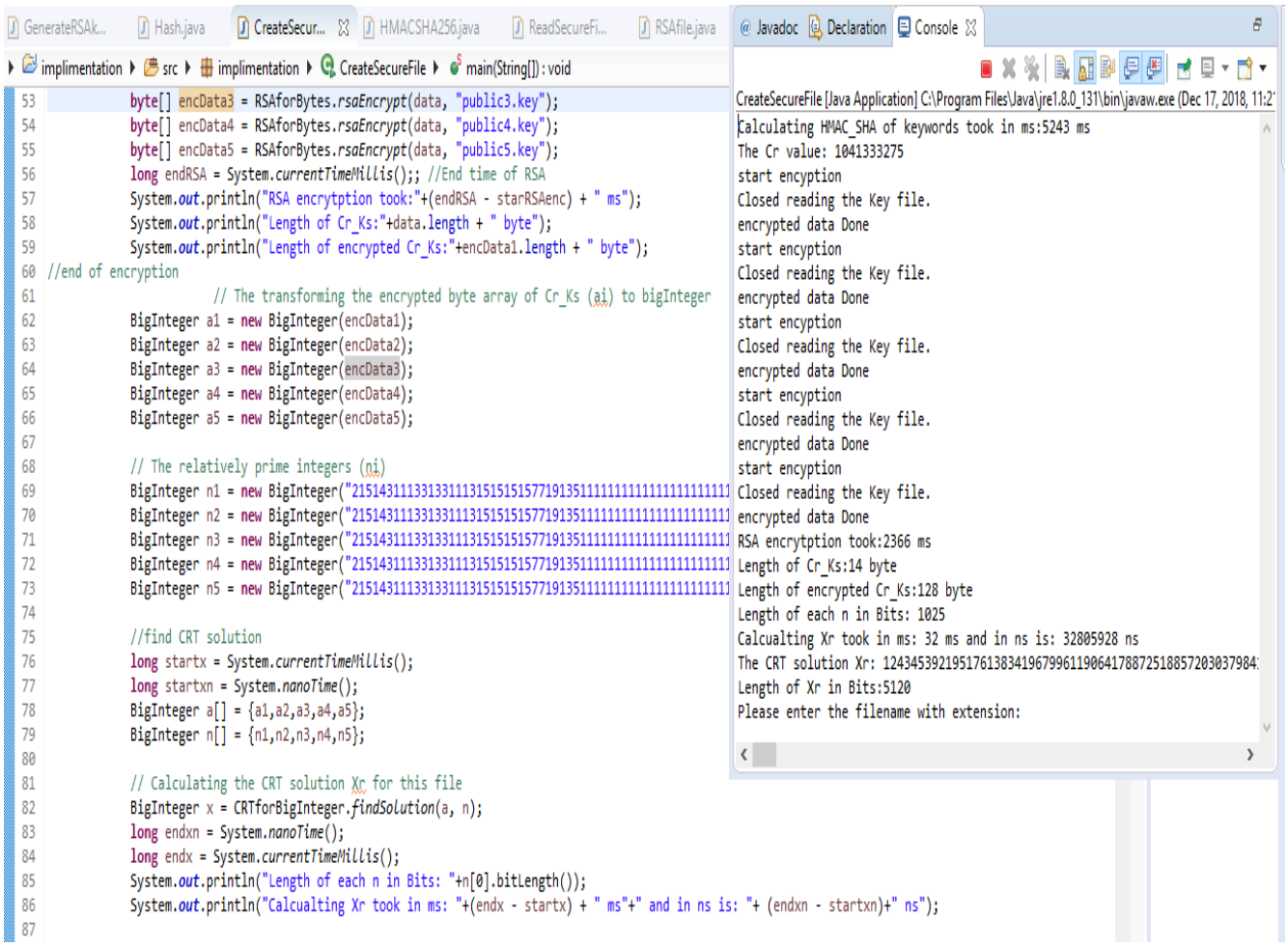



**Figure 5.3** execution result showing RSA encryption of Cr_Ks and the CRT solution to create secure file

Now let us see the result of the program when an encrypted video file of 38.6 MB is inserted. After specifying the path and file name successfully the program calculates the file digest and the resulted file digest is signed by the data owners' private key. The file digest is calculated using the SHA-256 hashing method and 1024-bit RSA encryption algorithm is used to encrypt or digitally sign the given file. The time taken to calculate the file digest and digital signature for the above video file is 5301ms and file signature length is 128 bytes. After calculating the above perimeters the result is appended on the inputted file and the program request the name, path and file extension which is ".dcs" by default of the processed secured data and produce an output based on the given name, path and extension.

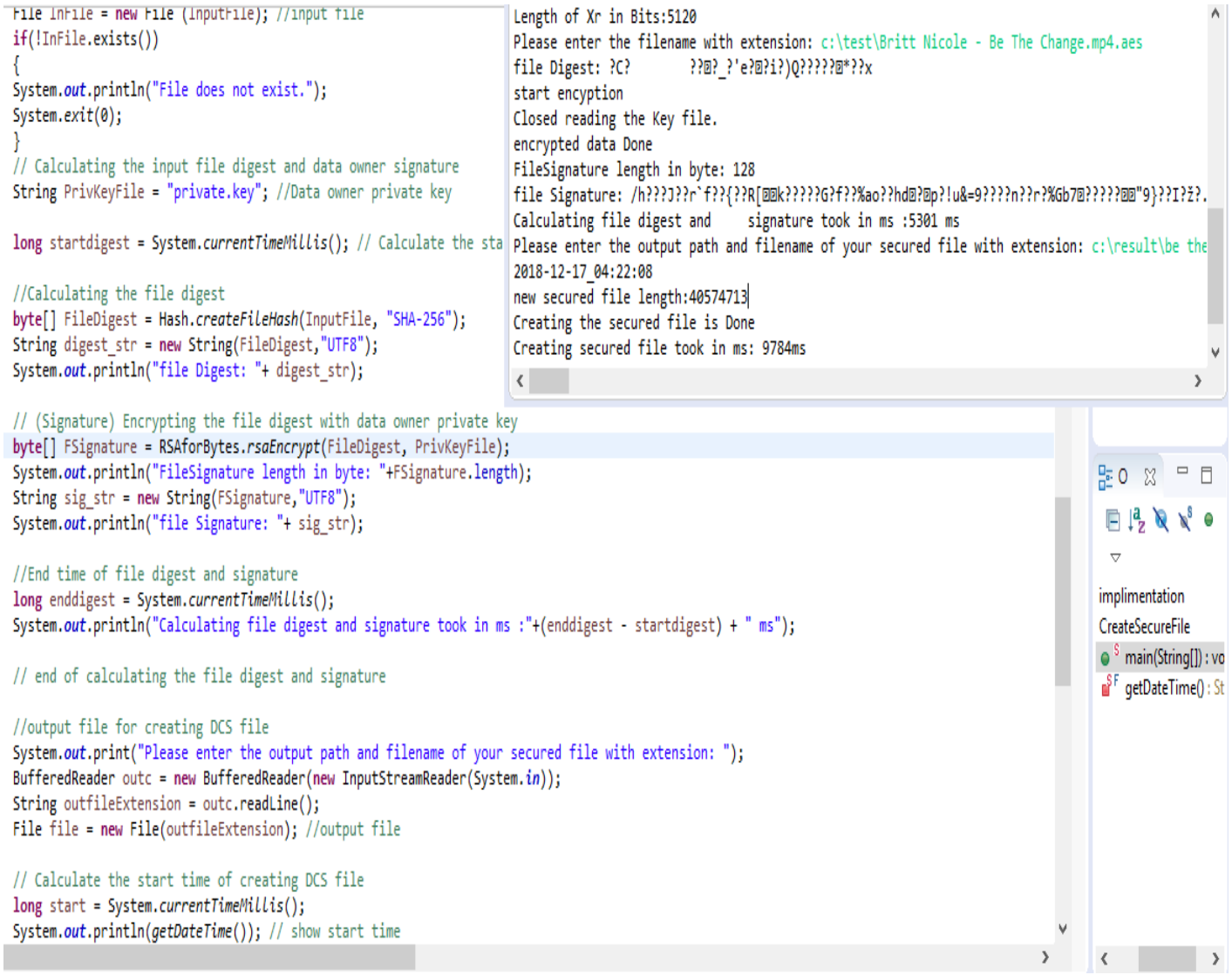



**Figure 5.4** execution result showing the file digest and signature of the inputted file

**Description of code to create the secured file**

`RandomAccessFile f = new RandomAccessFile(file, "rw");` allows to create a random access file stream to write to the file specified by the File argument (i.e. file) which is the output secured file.

`f.writeInt(keyword_no);` Writes the number of keywords attached (this integer number occupies 4 bytes). This number is used to calculate the range of searching on the secured file and to move the pointer, when reading the secured file after the encrypted keywords.

`f.write(theByteArray);` Writes the encrypted keywords resulting from the HMAC-SHA256. The total size occupied by the keywords = 32 byte × number of key words.

`f.writeUTF(Owner);` Writes the data owner name as UTF-8 encoding string.

`f.writeLong(Cr);` Writes the value of the secret value( Cr) and this long integer number occupies 8 bytes.

`f.writeInt(xlength);` Writes the number of bytes representing the share value (Xr) as an integer to the output file occupies 4 bytes.

`f.write(xr);` Writes the Xr value as byte array. The number of its bytes is specified in the previous step.

`f.write(FSignature);` Writes the 32 bytes array resulting from signing the digest of the input encrypted file or during the integrity check.

`RandomAccessFile in = new RandomAccessFile(InFile, "r");` allows to creates a random access file stream to read from the file specified by the File argument (i.e. InFile) which is the input encrypted file. The complete code to create the secured file is listed in appendix …

### 5.3.2. Reading the secured file from the data owner side

Reading the secured file and retrieving the original file from the secured file passes through four major operations such as computing the value of Cr_Ks which is the combination of the symmetric key and secret value from the shared value (Xr), after finding the Cr_Ks value then retrieving the encrypted file from the secured file is computed then the integrity of the file is verified and finally the last operation is to decrypt back the file for use. Each of these operations is discussed in the following sub topics.

#### *5.3.2.1. Calculating the secret value from the shared value*

The shared value is the result of CRT solution so the main operations here are first to read the shared value from the secured file and compute the encrypted Cr_Ks value from shared value (Xr) whereas the second operation is to decrypt the encrypted Cr_Ks value by using the private key of user ui. The first operation can be performed by first converting the shared value from byte array to big integer value and then implementing a simple modular function such that Xr mod ni to calculate the encrypted Cr_Ks value. Then decrypt the encrypted result formed from the modular function by using RSA decryption algorithm with the private key of the user ui. Now let us see some description of the codes to implements the above operations.

`BigInteger x = new BigInteger(xr);` allows to convert the value of Xr from byte array to big intiger format.

`BigInteger encKey = x.mod(N);` allows to calculate the encrypted Cr_Ks from Xr, where N is the relative prime number provided for user ui.

`byte[] encData = encKey.toByteArray();` allows to convert the format of the encrypted Cr_Ks value from big integer value to byte array so that it can be processed by the RSA decryption algorithm.

`byte[] dec = RSAforBytes.rsaDecrypt(encData, PrivKeyFile);` allows to decrypt the encrypted Cr_Ks value by using private key of user ui.

`BigInteger key = new BigInteger(dec);` allows to convert the format of the resulting Cr_Ks value from byte array to big integer format.

#### *5.3.2.2. Retrieving the encrypted file and attached security information from the secured file*

The program to retrieve the encrypted file as well as the security information attached to the secured file is discussed as follows in this subtopic. The security information attached on the secured data includes the owners' name, shared value (Xr), the secret value and the digital sign digest of the encrypted data. During the execution of the program, first it request the user to insert the path and file name of the secured file (reads the secured file) request the user to input the path, file name and default extension which is ".aes" then processes and produce the encrypted file. The code starting from reading the secured file to retrieving the encrypted file is discussed as follows.

`RandomAccessFile f = new RandomAccessFile(file, "r");` allows to create a random access file stream to read the inputted argument which is the secured file.

`RandomAccessFile out = new RandomAccessFile(Dest, "rw");` allows to create a random access file stream to write the argument specified as Dest which is the encrypted file extracted from the secured file.

`String File_owner = f.readUTF();` reads the data owners name in the form of string.
`long Cr = f.readLong();` reads the value of Cr which is 8 bytes and represent in the form of long integer.

`int x1 = f.readInt();` reads the number of bytes representing the shared value Xr and stores it in an integer value.

`byte[] xr = new byte[x1];` specifies the length of the byte array based on the previous code and reads the shared value based on the length.

```
byte[] FSignature = new byte[128];
f.read(FSignature);
```

The above pair of codes allows representing the length of byte array to hold the file signature value and reading the file signature bytes based on the specified length.

```
byte[] buf = new byte[1024];
int len;
while ((len = f.read(buf)) > 0) {
out.write(buf, 0, len);}
```

The above set of codes extracts the AES encrypted file from the secured file, reads the content of the secured file that represents the encrypted file and write it to output file 'out'.

#### *5.3.2.3. Verifying the file Integrity*

The above two operations produced the encrypted file. So here the integrity of the file is verified by comparing the file digest attached on the secured file with the new calculated file digest value. If they are equal that means file integrity is achieved otherwise there is integrity problem. The code for verifying the file integrity is discussed as follows.

`String PublicKeyFile = "public.key";` allows to specify the file containing the data owners' public key.

```
byte[] OldDigest = RSAforBytes.rsaDecrypt(FSignature, PublicKeyFile);
String unsig_str = new String(OldDigest, "UTF8");
```

The above two codes allows to decrypt the attached file digest by using the data owners public key and the result represents the original encrypted file digest.

```
byte[] FileDigest = Hash.createFileHash(outfile, "SHA-256");
String digest_str = new String(FileDigest, "UTF8");
```

0The above two codes allows to compute the extracted encrypted file digest by hashing it with SHA-256 algorithm and the result represents the digest of the new extracted file in a byte array.

```
if   (Arrays.equals(OldDigest,FileDigest))   System.out.println("OK,   the
integrity is verified");
else System.out.println("Attention, the integrity is NOT maintained");
```

Compares the attached digest with the new calculated digest and if they are equal the encrypted file integrity is maintained and originated from its data owner. Otherwise, there is a concern about the file integrity.

Therefore the above source codes are combined and implemented in a java class called ReadSecurefile.java, let us see the execution result.

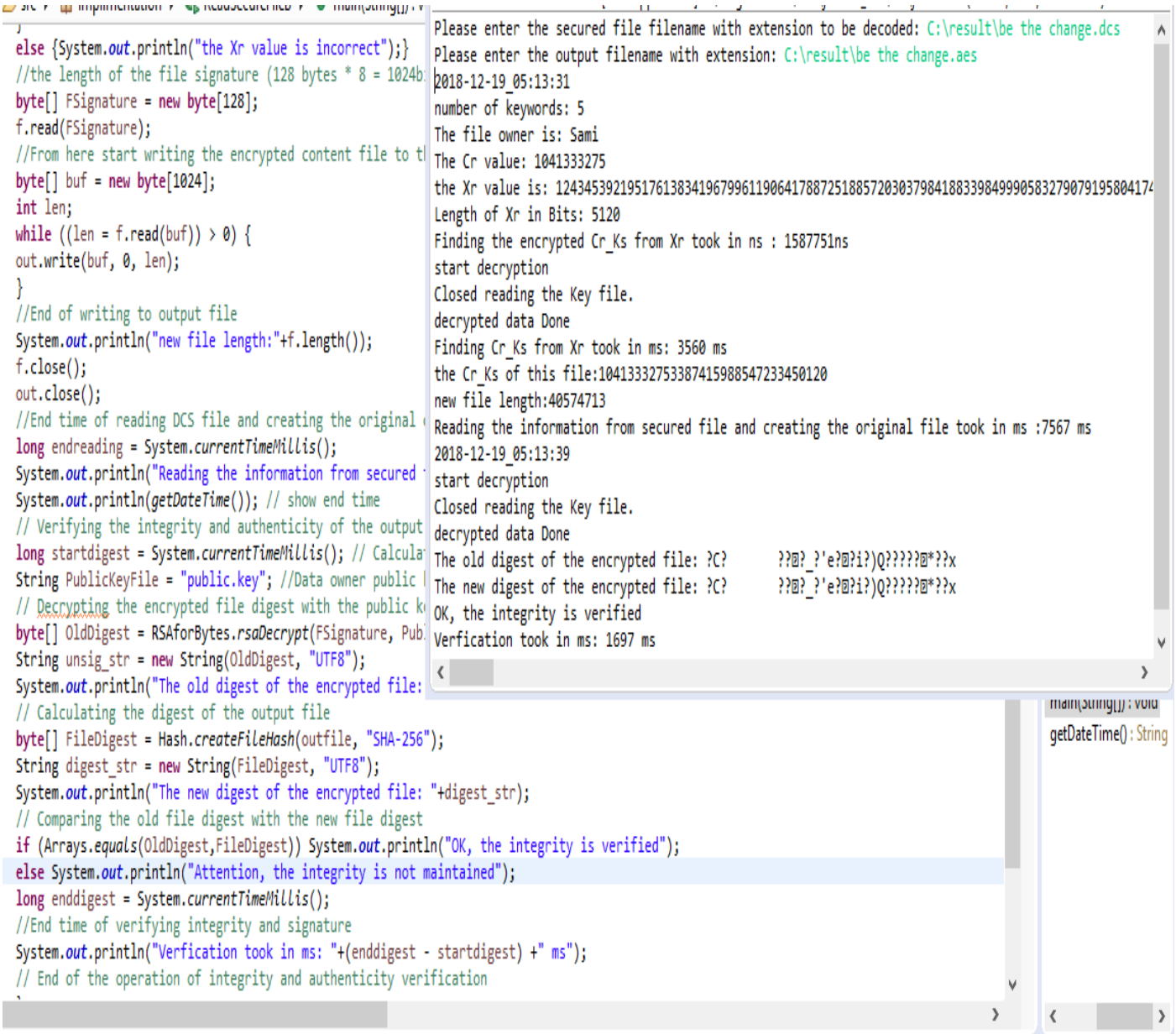



**Figure 5.5** The execution result of read secured file class

The above figure (figure 5.5) shows the steps and execution result of the operation to extract the encrypted result from the secured file (all the codes discussed in section 5.3.2.1, 5.3.2.2 and 5.3.2.3). And the access control procedure for the authorized user is discussed and the computation overhead is measured and represented. So to retrieve the Cr_Ks value from the shared value (Xr) the code uses a simple modular function which is Xr mod ni and the length of the Xr value is 5120 bits and the length of ni( which is the relative prime number for user i) is 1024 bits, to compute the encrypted Cr_Ks from shared value Xr it took 1587751ns (1.59 ms). And then the resulted encrypted value is decrypted by using RSA decryption algorithm and this operation took 3560ms. The next operation is to read the extracted Cr_Ks value from above operations so that the encrypted file is accessed from secured file. And to perform this operation it took 7567ms. Then the next operation is to compute the file integrity and it took 1697ms. Basically this result is for 38.6mb of secured video file.

#### *5.3.2.4. Decryption of the secured file*

After successful implementation of the above operations, know the data can be decrypted back by the authorized user for use. To implement the decryption process we used a tool called AES crypt.

Now let us see the steps to decrypt the file by using this tool.

- First locate the file that is produced from the above operations and it has an ".aes" extension.
- Right click on the file and select AES decrypt
- The tool prompts to enter the password which was set during the file encryption, enter the password and press ok.
- Finally the tool will decrypt the file and place it to same directory as the encrypted file.

## 5.4. Summary

The current trend of security for Ethiopian higher education cloud doesn't give the data owner a full control of the security and privacy management for his/her own data and the entire security not only for users' data but also the applications, the infrastructures, the identity and access managements are controlled by the universities and EthERNet data center admins. Due to this the service of the educational cloud are limited to services such as web hosting, E-mail and collaboration, DNS and identity management services. The proposed cloud security framework and the implemented prototype allow to secure the data owners' file by the data owner him/herself and allows a secure

access and sharing when the data is outsourced to a cloud environment and this creates a way to share education and research resources and files on the cloud under the control of the data owner.

This chapter discusses the experiment environments, the implementation and execution of the code. The tools selected for experimenting are an open source technologies which allows features developments with less effort, affordable and cost-effective for our country economy. Eclipse neon.3 platform was used to implement the main operations of the proposed solution and an open source AES encryption tool called AES crypt were used to encrypt the data owners file and then to decrypt the data from authorized user side.

The implementation from data owners' side has many java classes but the main code to create the secured file is CSecuredData.java and other classes are created to support this code. The implementation starts by encrypting the owners' file by using AES encryption algorithm and AES crypt which is an open source tool is used to perform this operation then the second operation is on enforcing the access control policy for the encrypted file. To implement this CRT combined with asymmetric encryption algorithm is used. The symmetric key which was used to encrypt the owners file and the secret value (Cr) are combined and encrypted by using the RSA encryption algorithm. Then the result is taken as an input to calculate the CRT with the second input which is the relative prime number of the user. Then the shared value (Xr) is computed from this two values by calling the function findCRT() which is coded in CRTforBiginteger.java class then the other security parameter which is about the file integrity check is implemented. Then the results of all these operations are attached to the data so that the data becomes self-protective and self-describing.

The other important code is RSecuredData.java which is implemented from the authorized user side. And here the major operation is to retrieve the original file from the secured file. This includes first, reading the shared value from the secured file and computing the RSA encrypted secret value from the shared value as discussed in section 5.3.2.1 then based on the retrieved secret value all the security parameters attached to the secured file such as the file owner name and digital sign digest of the encrypted file are retrieved in section 5.3.2.2 and then the file integrity is checked. If the above all parameters are meet and the file integrity is correct then the file gets decrypted back and can be accessed by the authorized user.

# Chapter 6 Result and Discussions

## 6.1. Overview

In this chapter the result of the proposed solution is evaluated from different perspective in terms of whether the research questions are answered or not, the research objectives are meet or not and by evaluating the computation as well as the storage overhead during implementing the solution both from the data owner side and the authorized user side.

## 6.2. Good data security and privacy handling principle in cloud computing

In this subtopic we deal on how the proposed framework could support Ethiopian higher education institutions on addressing the security and privacy issues discussed on chapter two. Addressing these issues benefits the higher education institutions by enhancing the security and privacy of their users' data. The way how this framework could support the higher education institutions to achieve these principles would be discussed as follows.

- Data Confidentiality: the users' data is kept encrypted and self-protected. The data can be accessed by authorized user only and if there is a security breach in the cloud either from the inside or outside the data stays protected and secured.
- Data Availability: the proposed solution improves data availability from both the cloud side as well as from the data owner side. Data availability from cloud side is provided by implementing effective backup and archive strategy that allows keeping data in multiple geographic locations which might be on another institution data center or the EthERNet data center. Data availability from owners' side is provided by allowing them to control the security of their data by themselves.
- Data Integrity: the proposed solution provides Integrity for each file such that when the authorized user tries to access the data it provides integrity proof based on the necessary information the data contain.
- Data access control and key sharing: the proposed solution provides access control and key sharing by implementing a mechanism called CRT and CRT is efficient in computation and management overhead. Besides, this method allows protecting the access control from unauthorized user and the cloud service provider itself and allows the data owner to set and manage the data access policies throughout the data life time.

- Data Leakage: the proposed solution reduces data leakage problem in a way that the secured data can only encrypted or decrypted on the data owners' or authorized users' environment and the secured data stored on the cloud with its security parameter might be leaked but to access the original content it needs to pass the different security parameters and break the encryption.
- Privacy: the proposed solution preserves the privacy access to the data using DCS approach where the data is self-protective. This approach provides privacy in a way that the identity of authorized users is not revealed during data access, access policies are hidden from unauthorized user and even from the cloud provider during access enforcement, the security parameters are implemented within the data itself for each data and this allows the data to be secured regardless of the data location in the cloud and if the security perimeter of a single data is compromised only a single data is affected, the data owner is responsible to set and manage the security requirements so that the data ownership and content disclosure within the cloud environment is no more privacy issue and security provided doesn't depend on the service provider or trusted third party security.

## 6.3. Evaluation of result from the file owners' side

### 6.3.1. AES Encryption storage and computation overhead

The first operation on the proposed solution is to encrypt the file by using 256-bit key length AES algorithm and the resulting encrypted file is used to attach the different security parameters and create the secured file. In table 6.1 the computation and storage overhead to encrypt different types of user's files using 256-bit key length AES encryption algorithm is evaluated. And from the result we can conclude that the execution time needed to encrypt the file increased as the size of the file increases. The encryption process took the most computation overhead in the proposed solution. The encryption process is the core operation of this solution and implementing a strong encryption algorithm with a longer key length will allow achieving a higher security. However to reduce the computation overhead and data owners can use weaker encryption algorithms with shorter key length based on their security need and this makes the proposed solution more flexible.

| Owners' file name | Original File Size(in bytes) | File size after Encryption(in bytes) | Storage overhead(in bytes) | Execution time |
|---|---|---|---|---|
| Chapterone.doc | 96,768 | 97,074 | 306 | 0.3s |
| save_1419452365563.jpg | 100,900 | 101,218 | 318 | 0.5s |
| 01 Track 1.mp3 | 4,445,113 | 4,445,426 | 313 | 1s |
| BasicComputerSkillsModule(Degree).pdf | 4,656,046 | 4,656,354 | 308 | 1.4s |
| cloud-complete.ppt | 5,923,328 | 5,923,634 | 306 | 1.7s |
| Britt Nicole - Be TheChange.mp4 | 40,573,454 | 40,573,762 | 308 | 4s |
| Data Centric Security- 2018.mp4 | 222,447,817 | 222,448,130 | 313 | 30.5s |

**Table 6.1** Storage and computation overhead to encrypt seven different types of files

### 6.3.2. Storage and Computation overhead to calculate the CRT solution

The second discussion is on the evaluation of the computation and storage overhead caused during the calculation of the CRT solution which results the shared value. The operation to calculate the shared value was discussed in detail on section 5.3.1.2 and section 5.3.1.3. So the main operation here was first to encrypt the Cr_Ks value by using 1024 –bit key length RSA encryption algorithm and the public key of each authorized users then the resulting cipher-text for each user authorized to access file combined with the relative prime number of each user is used to compute the CRT solution and shared value (Xr) produced. The result of the computation and storage overhead to calculate the shared value is discussed as follows.

From the result we can see that the execution time for seven users to encrypt the Cr_Ks value by using RSA encryption took 721ms and to calculate the CRT solution (Xr value) it took 37ms so the total storage overhead recorded is 758ms which is a very low computation overhead. And the storage overhead caused during calculating the CRT solution for seven users is 7180bits. So let's see the storage and computation overhead caused during calculating the CRT solution for different number of users.

| Authorized users | RSA encryption of Cr_Ks value in ms | Claculatin the shared value(Xr) in ms | Total time in ms | Storage overhead caused due to Xr in bits |
|---|---|---|---|---|
| **4** | 602 | 12 | 614 | 4095 |
| **5** | 648 | 22 | 670 | 5121 |
| **6** | 682 | 25 | 707 | 6151 |
| **7** | 721 | 37 | 758 | 7180 |

**Table 6.2** Storage and Computation overhead to calculate the CRT solution

From the table 6.2 we can conclude that the computation and storage overhead caused is not related with the file size but it is related with the encryption algorithm used to encrypt the Cr_Ks value and the number of authorized users. The total time to calculate the CRT solution increased by approximately 50ms for each extra user and the storage overhead increased by 1024bits for each extra user and this result is expected since the RSA encryption used is 1024-bit key length. So as a conclusion the overhead recorded for the security and privacy provided is in satisfactory level.

### 6.3.3. Storage and Computation overhead during implementing data Integrity

The third discussion is on the computation and storage overhead caused during implementing the data integrity and the operation to implement the data integrity is discussed in section 5.3.1.4. The main operation to implement data integrity includes first to hash the encrypted file by using SHA-256 hashing algorithm and then the resulting file digest is encrypted by 1024-bit key length RSA encryption algorithm using the private key of the data owner to form the digital signature. So from the result what we recognize is the computation overhead increased as the file size increases and the storage overhead is 128bytes for all file sizes which is the result of encrypting the 256-bit digest of the file by 1024-bit key length RSA algorithm

.

| AES encrypted file | File size (in bytes) | Storage overhead(in bytes) | Execution time(in ms) |
|---|---|---|---|
| Chapterone.doc.aes | 97,074 | 128 | 158 |
| save_1419452365563.jpg.aes | 101,218 | 128 | 166 |
| 01 Track 1.mp3.aes | 4,445,426 | 128 | 403 |
| BasicComputerSkillsModule(Degree).pdf.aes | 4,656,354 | 128 | 542 |
| cloud-complete.ppt.aes | 5,923,634 | 128 | 665 |
| Britt Nicole - Be TheChange.mp4.aes | 40,573,762 | 128 | 2138 |
| Data Centric Security- 2018.mp4.aes | 222,448,130 | 128 | 7150 |

**Table 6.3** Storage and Computation overhead to implement the data integrity

### 6.3.4. Total computation and storage overhead caused to create the secured file

In this section the total computation time and the total storage overhead caused to create the secured file is discussed. The total computation overhead caused during creating the secured file is discussed in table 6.4 and the overhead is due to the summation of all execution time required to perform operations such as time to encrypt the file by using AES encryption, to calculating the CRT solution for the secure access and sharing purpose, to compute the file digest and encrypt it so that the file integrity implemented and finally the time to appending all this results to the file so that the secured file is created. Table 6.4 shows the result of total computation overhead for seven different types of users' files and the time to calculate the shared value for seven authorized users is also showed.

| Original File Size(in bytes) | Execution time of AES encryption (s) | Execution time to find Xr (s) | Execution time of Integrity (s) | Total computation overhead (s) |
|---|---|---|---|---|
| **96,768** | 0.3 | 0.758 | 0.158 | 1.216 |
| **100,900** | 0.5 | 0.758 | 0.166 | 1.424 |
| **4,445,113** | 1 | 0.758 | 0.403 | 2.161 |

| | | | | |
|---|---|---|---|---|
| **4,656,046** | 1.4 | 0.758 | 0.542 | 2.7 |
| **5,923,328** | 1.7 | 0.758 | 0.665 | 3.123 |
| **40,573,454** | 4 | 0.758 | 2.138 | 6.896 |
| **222,447,817** | 30.5 | 0.758 | 7.150 | 38.408 |

**Table 6.4** Total computation overhead to create the secured file

The total storage overhead caused during creating the secured file is discussed in table 6.5 for seven different types of files both in size and format. The total storage overhead depends on the encryption and hashing algorithms used which is 256-bit key length AES algorithm, 1024-bit key length RSA algorithm and 256-bit key length SHA algorithm. And also the storage overhead depends on the number of users authorized to access the files.

| **original file name** | **Original File size (in bytes)** | **Secured file size (in bytes)** | **Storage overhead (in bytes)** |
|---|---|---|---|
| Chapterone.doc | 96,768 | 98,282 | 1514 |
| save_1419452365563.jpg | 100,900 | 102,425 | 1525 |
| 01 Track 1.mp3.aes | 4,445,113 | 4,446,634 | 1521 |
| BasicComputerSkillsModule(Degree).pdf | 4,656,046 | 4,657,562 | 1516 |
| cloud-complete.ppt | 5,923,328 | 5,924,842 | 1514 |
| Britt Nicole - Be TheChange.mp4 | 40,573,454 | 40,574,970 | 1516 |
| Data Centric Security- 2018.mp4 | 222,447,817 | 222,449,338 | 1521 |

**Table 6.5** The total storage overhead to create the secured file

Therefore from the result we can conclude that the execution time as well as the storage overhead caused is due to the implemented security perimeters and it is considered very low to cause a system overhead especially when we see the storage overhead it is around 1520 bytes and this value is almost unrecognizable as the size of file gets larger and larger.

## 6.4. Evaluation of result from the authorized user side

The computation overhead from the authorized user perspective is evaluated and discussed as follows in this subtopic. The major operations performed on the authorized user side are discussed in section 5.3.2 in detail and the operations include calculating the secret value from the shared value, then retrieving the encrypted file from the secured file, verifying the integrity and finally decrypting the file. So here the computation overhead of each operation caused during retrieving the original file from the secured file is evaluated as follows.

Table 6.6 discusses the total computation overhead caused during performing the above listed operations in order to retrieve the original file from the secure file. What the result shows is the AES decryption causes the most computation overhead and computing the secret value from the shared value causes the list computation overhead when relatively compared.

| Secured File Size(in bytes) | Execution time to find Cr_Ks from Xr (s) | Integrity proof took (s) | Creating the encrypted file | Execution time of AES decryption (s) | Total computation overhead (s) |
|---|---|---|---|---|---|
| **98,282** | 2.421 | 0.019 | 2.443 | 0.3 | 5.183 |
| **102,425** | 2.486 | 0.021 | 2.610 | 0.5 | 5.701 |
| **4,446,634** | 2.570 | 0.171 | 2.628 | 1.1 | 6.469 |
| **4,657,562** | 2.523 | 0.203 | 2.696 | 1.3 | 6.722 |
| **5,924,842** | 2.487 | 0.375 | 2.708 | 1.4 | 6.97 |
| **40,574,970** | 2.609 | 1.750 | 4.099 | 5.1 | 13.558 |
| **222,449,338** | 2.523 | 9.061 | 10.356 | 23.3 | 45.24 |

**Table 6.6** The total computation overhead from authorized user side

## 6.5. Research problem resolution

The essential security and privacy problems derived from the research question in chapter one addressed properly as shown in the following table.

| Research question | Answer |
|---|---|
| What are the data security and privacy problems that might occur with related to adopting cloud computing technology? | There are several security and privacy issues of a cloud environment studied and surveyed by different researchers. From these studies data/ information security is the main category of study in cloud security and in this thesis section 2.5 the main data security and privacy problems are discussed. These includes Data Confidentiality, Data Availability, Data Integrity, Data access control and key sharing as well as Data Leakage are identified as data security problem in cloud environment and Broken authentication and compromised credentials, Data breach, Data location problem as well as Data ownership and content disclosure problem are identified as data privacy problems. The proposed solution is designed to address these problems. |
| How can existing security techniques be used to enhance the security and privacy of users' data? | The other focus of this research was on reviewing the available security techniques on addressing the above security requirements. A combination of DCS approach with cryptographic techniques used. Symmetric key algorithm is used to encrypt the data, then CRT combined with asymmetric key encryption is used for privacy preserving of shared data and access control and finally hashing algorithm to compute the digest of the data and encrypting the result by using data owner private key for data integration verification. |
| How to design a cloud framework that is applicable for Ethiopian universities that can withstand the occurrence of data security and privacy attack? | A framework that can withstand occurrence of data security and privacy attack is designed. The framework considers the different types of domains or users, how the data gets secured before leaving the owners domain and how security parameters attached to create the secured file. From the evaluation result the framework achieved the required goals. |

**Table 6.7** Addressing research questions

## 6.6. Summary

In this chapter, the implementation result of the proposed data security and privacy framework for higher education cloud users is illustrated. In the proposed solution several security requirements are achieved to list some data confidentiality, data integrity, data availability, secure data access and key sharing, data leakage as well as users and data privacy are some to list. Besides the storage and computation overhead to create the secured file as well as to retrieve the secured file and access the original file is illustrated and the result shows the proposed framework provides data security and users privacy with minimal computation and storage overhead.

# Chapter 7 Conclusion, Recommendation and Future work

## 7.1. Conclusion

In this research a cloud data security and privacy framework for Ethiopian higher education institutions is designed and implemented. To achieve this scientific step were followed. These includes first information gathering and analysis about the current scenario, then literature study on cloud computing data security and privacy issues as well as assessment of the state of the art solutions to address the identified data security and privacy requirements. In addition to these different types of reference architectures, cloud data security and privacy frameworks and related issues were assessed. Based on the assessment to address the research problems a cloud data security and privacy framework were designed. This solution was based on a DCS approach in which, this approach provides security in cloud environment based on the data itself and without fully depending on the security provided by the cloud service provider or trusted third party.

The proposed security framework considered several data security and privacy requirements. These includes data security requirements such as data confidentiality, data availability, data integrity, data access control and key sharing as well as data leakage and privacy requirements such as broken authentication and compromised credentials, data breach, data location problem and data ownership and content disclosure problems.

The proposed data security framework enhances the security of users' file of any type which can be either video, audio, picture or document of any type. To achieve this proposed solution incorporates different security parameters to create the secured file. These parameters include first to encrypt the original file by using 256-bit key length encryption algorithm and secondly, enforced the secured data access policy. This policy is implemented by using the CRT. CRT is used to calculate the shared value. And this value contains two important values. These are the secret key value, a value that is used to decrypt the file and secret value, a value that is used to authenticate and authorized users based on the access procedure. As a result this solution allows enforcing the access control and secret key sharing in one mechanism which made the solution efficient in computation and key management overheads. And then the shared value which is resulted from the CRT calculation is attached to the encrypted file so that authorized users don't need to exchange or keep these keys. The access control policy is set in this way and any authorized user can extract the secret key and secret value from the attached shared value. Since each file has its own unique shared value and incase that value gets compromised only that file will be affected and the other files stays safe. The

implementation shows the computation and storage overhead to calculate the shared value is significantly low. The other security parameter incorporated is the integrity proof which is attached to the secured file. This allows the file owner to be confident of his/her file regardless of the file location in the cloud. The authorized users can access and verify the integrity since the digested value is attached in the secured file. The implementation result for integrity proof shows that it doesn't require high storage and computation overhead and can be implemented in simple operations.

Finally after incorporating each security parameters in the encrypted file the secured file created and can be outsourced to the cloud to be stored, shared and copied. The copied secured files in the cloud server are used to maintain the security parameters of the original secured file in case of improvement of access control to support multiple users or any relevant other reason. The other important feature of the proposed solution is allowing the data owner to select the encryption and hashing algorithm based on the required security strength and overhead considerations.

## 7.2. Recommendations

This research provides an opportunity to enhance the security and privacy of users' data stored on the cloud environment which can be accessed and shared by authorized users. So in this thesis a security solution that is fully set and managed by the data owner is designed and implemented. This security solution has a strong data security and privacy mechanism incorporated. So this solution is recommended when sensitive users' data such as course materials, modules, research results, laboratory experiment results and in general any type of data related to education and research are shared, stored and accessed among educational institutions and the EthERNet which is the countries cloud service provider. Consequently, the users learn and share security experiences, on how to handle and manage their files before outsourcing it to the cloud.

## 7.3. Future work

The proposed data security and privacy solution needs further enhancement on the following issues.

- The proposed framework and prototype should include a secured search capability that satisfies requirements such as like the other security parameters the search capability needs to be integrated to the secured file, the search capability shouldn't depend on the database that

stores the data on the cloud and the search capability needs to be encrypted by the data owner and stays hidden from unauthorized users as well as from the cloud provider.

- In now a day there is a tremendous technological advancement on handheld devices and most people choose thin clients like tablets and smart phones. A possible enhancement needs to be made on the encryption/decryption operations of the file since this operation took the most overhead compared to other operations in the result so that this solution will be suitable to be used by thin clients.
- The current work will be enhanced to address more technical challenges such as enabling the legal modification of the secured file content without decrypting it in the cloud and the proposed solution can be integrated with the cloud service security approach so that an efficient searching capability is attained.

## References


[1] H. Schulze, "Cloud Security Report," Cybersecurity Insiders, 2018.

[2] K. Hashizum, D. G. Rosado and e. b. fernandez, "An analysis of security issues for cloud computing," *Journal of Internet Services and Applications,* feb, 2013.

[3] "en.wikipedia.org," [Online]. Available: https://en.wikipedia.org/wiki/Cloud_computing. [Accessed 20 Sep 2017].

[4] Mell, peter and T. Grance, "the NIST definition of cloud computing," 2011.

[5] Chen, deyan and z. hong, "Data security and privacy protection issues in cloud computing," in *International Conference on computer Science and Electronics Engineering(ICCSEE)*, 2012.

[6] E. Safiriyu, O. Abiona, A. Oluwatope, A. Oluwaranti and a. L. K. Clement Onime, "A user identity managment protocol for cloud computing paradigm," *International Journal of communications, network and system science,* vol. 4, no. 03, p. 152, 2011.

[7] "Generally Accepted Privacy Principles," American Institute of Certified Public Accountants, Inc. and Canadian Institute of Chartered Accountants, August 2009.

[8] S. Nabil, "Cloud computing for education: A new dawn?," *International Journal of Information Management* , vol. II, no. 30, pp. 109-116, 2010.

[9] Massadeh, S. A. and M. A. Mesleh, "Cloud Computing in Higher Education in Jordan," *World of Computer Science & Information Technology Journal,* vol. III, no. 2, 2013.

[10] Mohamed, M. Eman, S. A. Hatem and E.-E. Sherif, "Enhanced data security model for cloud computing," *8th International Conference on informatics and Systems (INFOS),* pp. CC-12, 2012.

[11] M. S. Bajwa, "A Concern towards Data Security in Cloud Computing," *International Journal of Computer Applications* , no. 11, p. 114, 2015.

[12] K. Ramakrishnan, "Security and Privacy in Cloud Computing," Master's thesis, Western Michigan University, Michigan , 2017.

[13] "FERPA Compliance on AWS," Amazon Web Services, USA, may 2015.

[14] L. Fang, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger and D. Leaf, "NIST cloud computing reference architecture," NIST special publication 500, 2011.

[15] K. Bakshi, "Cisco Cloud Computing - Data Center Strategy, Architecture and Solutions," Cisco

Systems, USA, 2009.

[16] L. S. Ranjan and B. Nayak, "Enhancing Data Security in Cloud Computing Using RSA Encryption and MD5 Algorithm," *International Journal of Computer Science Trends and Technology (IJCST),* vol. 2, 2014.

[17] Parmar, N. J. and P. K. Verma, "A Comparative Evaluation of Algorithms in the Implementation of an Ultra-Secure Router-to-Router Key Exchange System," *Security and Communication Networks,* 2017.

[18] A. H. M. M. Aysan Shiralizadeh, "Presenting a new data security solution in cloud computing," *Journal of Scientific Research and Development 2,* pp. 20-36, 2015.

[19] A. E. Youssef and M. Alageel, "A Framework for Secured Cloud computing," *International Journal of Computer Science Issues(IJCSI),* vol. 9, no. 4, pp. 487-500, July 2012.

[20] L. Greg., "Qualitative and quantitative research: How to choose the best design," 2007.

[21] C. R. Kothari, "Research methodology: methods and techniques," New Age International, 2008.

[22] Y.-Y. Chen, "Architecture for data-centric security," *Diss. Princeton University,* 2012.

[23] L. Chen and D. B. Hoang, "Active data-centric framework for data protection in cloud environment," *23rd Australasian Conference on Information Systems,* pp. 3-5, Dec 2012.

[24] D. Agrawal, A. E. Abbadi and S. Wang, "Secure and Privacy Preserving Data Services in the Cloud: A Data Centric View," *Proceedings of the VLDB Endowment,* vol. 5, no. 12, August 2012.

[25] I. S., "General secret sharing based on the Chinese remainder theorem," *IACR cryptology ePrint Archive,* p. 166, 2006.

[26] K. O. Geddes, S. R. Czapor and G. Labahn, "Algorithms for computer algebra," *Springer,* 1992.

[27] K. Y., S. J., G. J., J. R. and Y. P., "A Cryptographic Solution for General Access Control," *international conference on Information Security,* vol. 3650, pp. 461-473, 2005.

[28] "www.rosettacode.org," Chinese remainder theorem, [Online]. Available: https://rosettacode.org/wiki/Chinese_remainder_theorem. [Accessed 20 june 2018].

[29] "Java Cryptography Architecture Reference Guide," [Online]. Available: https://docs.oracle.com/javase/7/docs/technotes/guides/security/crypto/CryptoSpec.html#KeyPair. [Accessed 22 june 2018].

# Appendix I: Interview questions

**Introduction**

I am a Master of Science student at Adama science and Technology University undertaking MSC in Software engineering. And currently I am undertaking a thesis for the completion of the degree. Collecting data from your organization will help me to address the problem and finalizing the thesis successfully. The purpose of this questioner is to collect a structured data that will be used as an input in providing a solution for data security and privacy problems when universities try to adopt the cloud.

**Introductory questions**

- ✓ What is your name?
- ✓ Would you please tell me your education level and work experience?
- ✓ What is your position in the organization?

**Main Interview Questions for EthERNet team leaders**

**About cloud services**

1. Could you please tell me how many Higher education Institutions are members of EthERNet?
2. Could you please tell the different services given to these Higher education Institutions?

**About security issues**

3. What type of security issues concern you the most?
4. Who is responsible for the security of the entire cloud system and how is the responsibility carried out?
5. What type of mechanisms you use to keep the data secure and private?
6. How is the handling of personal data? Because processing of personal data must be with in national network.
7. What will be your opinion on the existing system data security and users privacy?

**Concerning the security strategy**

8. Is there any security policy in your organization to secure its asset, application and data? If so, could you please mention some of them?
9. What is your opinion on policies and rules that should be added to improve data security and user privacy?

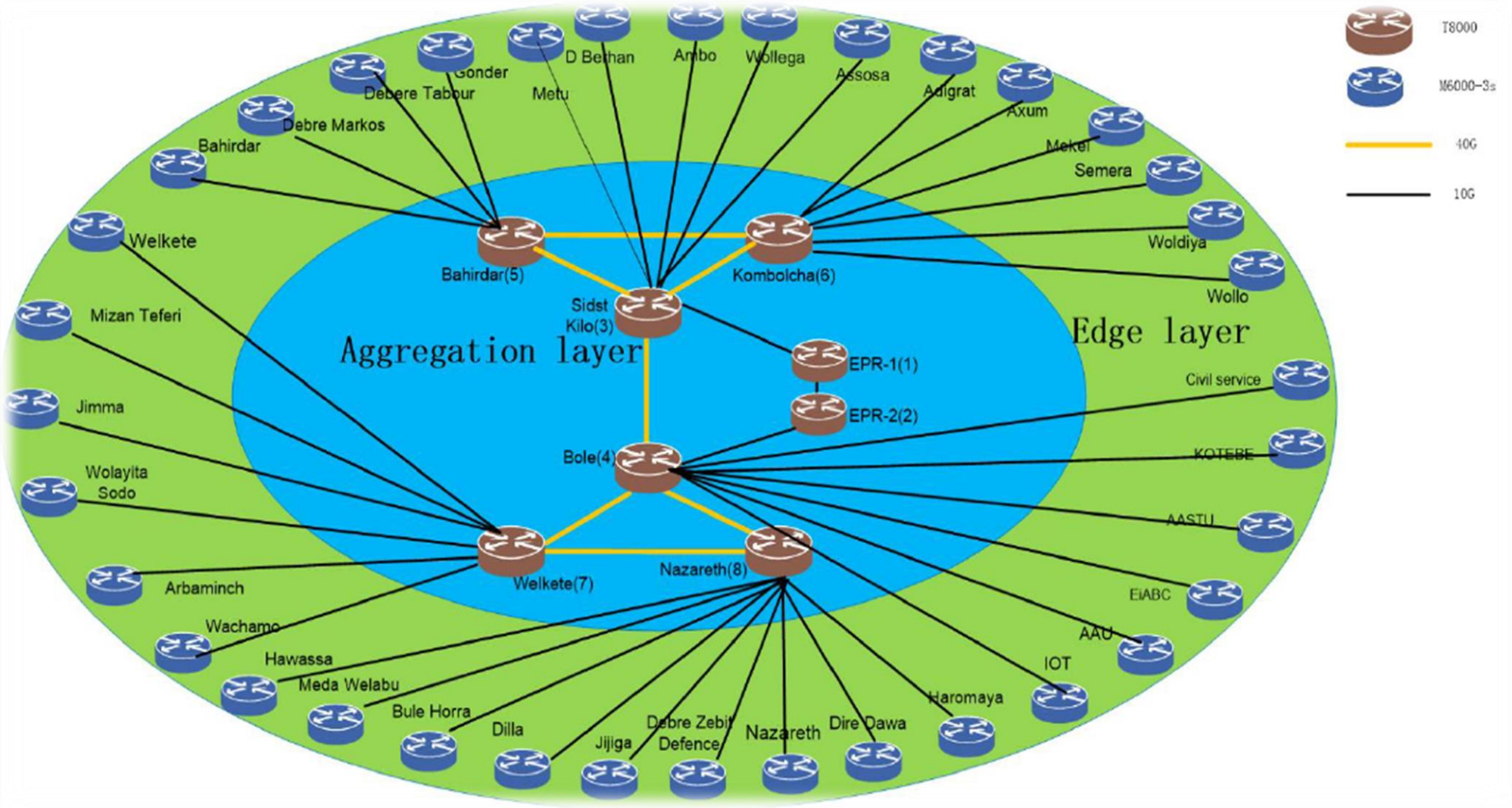
**Conclusion**

10. Do you believe that there is a proper protection of user data and privacy on the existing system (with relate to the security approaches used)? How?
11. Is there any security improvement plan for the future? Why? How?
12. Is there anything else you would like to add to what we have just discussed?

**Interview questions for ICT department of Ethiopian universities**

1. What does the general datacenter technology looks like? The operating system and the types of servers used in the ICT?
2. What does the current network architecture looks like?
3. What type of security issues concern you the most?
4. What type of mechanisms you use to keep the data secure and private?
5. What is your opinion on the existing system data security and users privacy?
6. What is your expectation on data security aspects that should be addressed on cloud computing?
7. Anything constrictive to add?

## Appendix II: Backbone Architecture of EthERNet



## Appendix III: Sample Source Code

Code for finding the CRT solution

```
import java.math.BigInteger;
public class CRTforBigInteger {
	// Finding CRT solution Xr given Xr = a1 (mod n1), x = a2 (mod n2) ...
		public static BigInteger findSolution(BigInteger[] a, BigInteger[] n)
		{
			if(a==null || n ==null)
		{		System.out.println("a or n must have at least 2 elements");
			 return null;
		}
		if(a.length<2 || n.length<2)
		{		System.out.println("Length of a or n are less than 2");
			 return null;
		}
		if(a.length != n.length)
		{		System.out.println("Length of a and n are not same");
				return null;
		}
		BigInteger x = a[0];
		BigInteger nInverse, y, z;
		BigInteger ni = BigInteger.ONE;
		int i = 0;
		while(i< a.length-1)
		{
		ni=ni.multiply(n[i]);
		//check if the n's are relatively prime
		if(!BigInteger.ONE.equals(n[i+1].gcd(ni))){
		System.out.println("The n's are not relatively prime->"+n[i+1]+","+ni);
		return null;
		}
		// calculating the modular multiplicative inverse
		nInverse = CryptoBig.inverse(n[i+1], ni);
		if(nInverse.compareTo(BigInteger.ZERO) == -1 )
		nInverse = nInverse.add(n[i+1]);
		z = a[i+1].subtract(x);
		z = z.multiply(nInverse);
		y = z.remainder(n[i+1]);
		if(y.compareTo(BigInteger.ZERO) == -1) // y < 0
		y = y.add(n[i+1]);
		x = x.add(ni.multiply(y));
		i++; }
		return x; // CRT solution Xr
		}
}
```

Code to calculate the modular multiplicative reverse

implimentation ▸ src ▸ implimentation ▸ CryptoBig ▸

```
3  import java.math.BigInteger;
4  public class CryptoBig {
5          //returns g^-1 (mod p)
6          public static BigInteger inverse (BigInteger p, BigInteger g)
7          {
8          //g inverse = g ^ (p-2) (mod p)
9          BigInteger[] gInverse = ExtendedEuclid(p, g);//crypto.fastSq(p, g, p-2);
10         if (gInverse[2].compareTo(BigInteger.ZERO) != -1)
11         return p.add(gInverse[2]);
12         return gInverse[2];
13         }
14         public static BigInteger[] ExtendedEuclid(BigInteger a, BigInteger b)
15         /* This function will perform the Extended Euclidean algorithm to find the
16         GCD of a and b. We assume here that a and b are non-negative (and not
17         both zero). This function also will return numbers j and k such that
18         d = j*a + k*b
19         where d is the GCD of a and b.
20         */
21         {
22         BigInteger[] ans = new BigInteger[3];
23         BigInteger q;
24         // if (b == 0)
25         if (b.equals(BigInteger.ZERO)){
26         ans[0] = a;
27         ans[1] = BigInteger.ONE;
28         ans[2] = BigInteger.ZERO;
29         }
30         else
31         { /* Otherwise, make a recursive function call */
32         q = a.divide(b);
33         ans = ExtendedEuclid (b, a.remainder(b));
34         BigInteger s = ans[2].multiply(q);
35         BigInteger temp = ans[1].subtract(s);// - ans[2]*q;
36         ans[1] = ans[2];
37         ans[2] = temp;
38         }       return ans;
39         }
```

Code for generating Public and Private key Pairs and store them in a file

```
import java.security.KeyFactory;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.spec.RSAPrivateKeySpec;
import java.security.spec.RSAPublicKeySpec;

public class GenRSAK {

	public static void main(String[] args) throws Exception {
			generateKeys();
			}
			public static void generateKeys() throws Exception {
			KeyPairGenerator kpg = KeyPairGenerator.getInstance("RSA");
			kpg.initialize(1024); // 2048 or 1024
			KeyPair kp = kpg.genKeyPair();
			PublicKey publicKey = kp.getPublic();
			PrivateKey privateKey = kp.getPrivate();
			System.out.println("keys created");
			KeyFactory fact = KeyFactory.getInstance("RSA");
			RSAPublicKeySpec pub = (RSAPublicKeySpec)
fact.getKeySpec(publicKey,RSAPublicKeySpec.class);
			RSAPrivateKeySpec priv = (RSAPrivateKeySpec)
fact.getKeySpec(privateKey,RSAPrivateKeySpec.class);
					// create files for the private and public keys
			RSAfile.saveToFile("public5.key",
pub.getModulus(),pub.getPublicExponent());
			RSAfile.saveToFile("private5.key",
priv.getModulus(),priv.getPrivateExponent());
					System.out.println("keys saved");
					}

}
```

Code for RSA encrypting/decrypting an array of byte

```
import java.io.BufferedInputStream;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.ObjectInputStream;
import java.security.spec.RSAPrivateKeySpec;
import java.security.spec.RSAPublicKeySpec;
import java.math.BigInteger;
import java.security.Key;
import java.security.KeyFactory;
import javax.crypto.Cipher;
public class RSAforBytes {
```

```
	static Key readKeyFromFile(String keyFileName) throws IOException {
		InputStream in = new FileInputStream(keyFileName);
		ObjectInputStream oin = new ObjectInputStream(new BufferedInputStream(in));
		try {
		BigInteger m = (BigInteger) oin.readObject();
		BigInteger e = (BigInteger) oin.readObject();
		KeyFactory fact = KeyFactory.getInstance("RSA");
		if (keyFileName.startsWith("public"))
		return fact.generatePublic(new RSAPublicKeySpec(m, e));
		else
		return fact.generatePrivate(new RSAPrivateKeySpec(m, e));
		} catch (Exception e) {
		throw new RuntimeException("Spurious serialisation error", e);
		} finally {
		oin.close();
		System.out.println("Closed reading the Key file.");
		}
	}
	// file_source is the key file name
	public static byte[] rsaEncrypt(byte [] byteData, String file_source) throws
Exception
	{
	System.out.println("start encyption");
	Key pubKey = readKeyFromFile(file_source);
	Cipher cipher = Cipher.getInstance("RSA");
	cipher.init(Cipher.ENCRYPT_MODE, pubKey);
	// use object for encryption
	byte[] encryptedByteData = cipher.doFinal(byteData);
	System.out.println("encrypted data Done");
	return encryptedByteData;
	}
	// Use this PublicKey object to initialise a Cipher and decrypt some data
	public static byte[] rsaDecrypt(byte[] encryptedByteData, String key_file) throws
Exception
	{
	System.out.println("start decryption");
	Key priKey = readKeyFromFile(key_file);
	Cipher cipher = Cipher.getInstance("RSA");
	cipher.init(Cipher.DECRYPT_MODE, priKey);
	// use object for encryption
	byte[] decryptedByteData = cipher.doFinal(encryptedByteData);
	System.out.println("decrypted data Done");
	return decryptedByteData;
	}
}
```

Code for integrity proof

```
import java.io.*;
import java.security.*;

public class Hash {

    static byte[] createFileHash(String filename, String method) {
        try {
        try {
        InputStream fis = new FileInputStream(filename);
        byte[] buffer = new byte[1024];
        MessageDigest complete = MessageDigest.getInstance(method);
        int numRead = 0;
        while (numRead != -1) {
        numRead = fis.read(buffer);
        if (numRead > 0) {
        complete.update(buffer, 0, numRead);
        }
        }
        fis.close();
        return complete.digest();
        }
        catch(NoSuchAlgorithmException nsae) {
        return null;
        }
        // Do nothing for it.
        }
        catch(IOException e ) {
        return null; } // Do nothing for it.
        } // End of creating file hash (or digest)

}
```