ANALYSIS OF DATA INTEGRITY PROOF TECHNIQUES IN CLOUD STORAGE

By

WATSALA HERBERT

SEP15/COMP/0638U

SCHOOL OF COMPUTING AND ENGINEERING

Supervisor

Dr. D.P. Mirembe

A Proposal submitted to the School of Computing and Engineering

For the Study Leading to a Master's Dissertation in Partial Fulfillment of the Requirements for the Award of Masters Degree in Computing of Uganda Technology and Management University

January 2017

TABLE OF CONTENTS

CONTENTS **PAGE**

INTRODUCTION 1

1.1 Background to the study 1

1.2. Statement of the Problem 3

1.3 Objective of the Study 4

1.3.1 General Objective 4

1.3.2 Specific Objectives 4

1.4 Research Questions 4

1.5 Scope of the Study 4

1.6 Significance of the Study 5

LITERATURE REVIEW 6

2.1 Introduction 6

2.2 Data Integrity 7

2.3 Current Data Integrity Proving Schemes with their advantages and limitations 7

2.4 Provable Data Possession (PDP) 7

2.4.1 Basic PDP Scheme based on MAC 9

2.4.2 Scalable PDP 10

2.4.3 Dynamic PDP 10

2.5 Proof of Retrievability (PoR) 11

2.5.1 PoR based on keyed hash function hk(F) 12

2.5.2 PoR Based on Selecting Random Bits in Data Blocks 13

2.6 High Availability and Integrity Layer (HAIL) 13

METHODOLOGY 15

3.1 Introduction 15

3.2 Research Design 15

3.3 Data collection methods 15

3.4 Analysis and Interpretation 16

3.5 Quality Control 16

3.6 Ethical issues 17

References 18

Appendix: 20

Work Plan: 21

INTRODUCTION

Cloud Computing is an emerging technology aimed at providing scalable, fast, flexible, and cost effective technology platforms for IT enabled services over the internet. The technology provides cost effective computing resources like; Memory, storage and processor all physically located and hosted at the cloud service provider's premises. Cloud Computing is continuously showing consistent growth in the field of computing. It's emerging from recent advances in technologies such as hardware virtualization, Web services, distributed computing, utility computing and system automation. The technology represents one of the magnificent shifts in information technology which can enhance collaboration, agility, scaling, availability, and provide the potential for cost reduction through optimized and efficient computing [1, 2].

1.1 Background to the study

Cloud Data Storage is an attractive means for storing and managing outsourced data remotely through the internet. Data modification can be done by unauthorized user or some malicious activity without the knowledge of the data owner. So to ensure integrity of remotely stored data, various integrity checking techniques for cloud data storage have been proposed. This study presents an investigation of different Integrity Checking Techniques for remotely stored data on cloud. Comparison of all techniques with their advantages and disadvantages are provided.

In 2008 Amazon faced a downtime and was not able to recover the original data. [3] In 2006 Gmail also faced mass deletion of email which resulted into data loss. [4] Cloud Service providers like Amazon, have since then decided to have explicit statements like they are responsible for any kind of data loss or data damage to save their reputation. [5] However these

statements are not enough to ensure correctness of user data hence it's necessary for user to check the integrity of their remotely stored data using reliable techniques.

The Cloud is characterized by a large group of virtual servers networked to enable users store, share and access computing resources virtually from large Data Centers. The technology has a combination of five essential characteristics (on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service), three service models (Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS)), and four deployment models (Public, Private, Hybrid, Community) [6]. In SaaS, users are provided access to application software. IaaS refers to the sharing of hardware resources for executing services, typically using virtualization technology. In the PaaS model, cloud providers deliver a computing platform, typically including operating system, programming language execution environment, database, and web servers. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers.

Data integrity refers the maintenance of intactness of any data during transactions like transfer, retrieval or storage to ensure it's unaltered, correct and consistent. The data may change if and only if an authorized operation is valid on the data [7]. However since the client's data is outsourced to a cloud provider with no physical access to the cloud storage servers by the data owner, there is a high chance of data deletion or modification in one way or the other, therefore the need for periodical checking of data to guarantee its intactness. Cloud Data Centers mainly rely on internal and external techniques to examine outsourced data and emphasis is put on developing unique techniques with low cost overheads to protect data integrity. To protect data integrity, I propose to analyze techniques that guarantee data integrity proof in the cloud

environment. In this study I shall analyze and evaluate effective basic schemes of Proof of Retrievability (PoR) and Provable Data Possession (PDP). Currently these are known schemes to check data integrity in the cloud storage environment; however they have limitations that shall be exposed in this study.

1.2. Statement of the Problem

Cloud computing is a new computing paradigm in which dynamically scalable resources are provided as a service over the Internet. Virtualized resources are dynamically reconfigured and robustly deployed with low maintenance cost, effort and high efficiency. End users outsource and access these services through the internet without knowing their location of storage and management. However this unique technology poses a challenge of maintaining the correctness of user data outsourced for storage. Currently maintaining the correctness of user data in the cloud storage servers is the most important aspect. Existing Data Integrity Proof schemes need to be examined in detail so that an effective technique is suggested to user to check the correctness of their data to ensure vital not deleted or altered in any way by malicious entities or behaviors. In this study, data integrity proof schemes in cloud storage will be analyzed and suggest to end users better schemes for checking integrity of their data while outsourced to cloud providers. Each user has different security levels for outsourced data to CSPs. Currently, to the best of my knowledge and statistics from previous work show that Cloud providers never consider client's data sensitivity seriously and instead store it in the same manner as less sensitive data hence the need for this study to investigate and examine techniques for proving data integrity in the cloud.

1.3 Objective of the Study

1.3.1 General Objective

The general objective of the study is to carry out a detailed examination on data integrity proof schemes to check the correctness of user data stored in the cloud environment.

1.3.2 Specific Objectives

(i) To examine the current state of the art data integrity proof schemes.

(ii) To compare existing data integrity proof schemes.

(iii)To evaluate data integrity proof schemes in the cloud environment.

1.4 Research Questions

(i) What technique is currently used to check user data correctness in the cloud?

(ii) What are the weaknesses of the existing data integrity proof schemes?

(iii)What are the solutions to the weaknesses identified?

1.5 Scope of the Study

The study focuses on examining and techniques of data integrity proof in the cloud storage environment by providing a broad overview of some of the basic schemes that check user data integrity stored in the cloud. I will further establish the current state of the art as far as checking the correctness of user data stored in cloud environment. The study will also suggest an efficient and effective data integrity proof scheme in cloud environment. Therefore this study will identify challenges with existing data integrity proof schemes in cloud storage servers and suggest to users effective techniques to check the correctness of their data stored in the cloud.

1.6 Significance of the Study

The study will examine the existing data integrity proof schemes for sensitive data stored and processed in the cloud. Users outsource data to cloud providers and lose control to determine its correctness. This study will therefore suggest techniques that can help users to check the correctness of their sensitive data stored in the cloud.

LITERATURE REVIEW

2.1 Introduction

Cloud computing is increasingly attracting individuals and organizations to move and maintain their sensitive data from local to remote cloud servers. In addition to major cloud infrastructure providers, such as Amazon, Google, and Microsoft, more and more third-party cloud data service providers are emerging which are dedicated to offering more accessible and user friendly storage services to cloud customers. It is a clear trend that cloud storage is becoming a pervasive service. Along with the widespread enthusiasm on cloud computing, concerns on data security with cloud storage are arising due to unreliability of the service.

For example, recently more events on cloud service outage or server corruption with major cloud infrastructure providers are reported to be it caused by malicious attacks. Such a reality demands for reliable data storage to tolerate certain outage. In particular, the cloud storage service should offer cloud customers with capabilities of: 1) timely detection of any server corruption event, 2) correct retrieval of data even if a limited number of servers are corrupted, and, 3) repair of corrupted data from uncorrupted data.

Although existing techniques have provided solutions, the main challenge for cloud storage service is to simultaneously provide these capabilities at minimal cost. This is because in cloud computing both data storage and transmission are charged in the “pay-as-you-use” manner. Solutions of high cost will discourage user engagement and be of less practical use. Moreover, it is important to set cloud customers free by minimizing the complexity imposed on them in terms of computation cost and burden of being online.

Cloud Computing aims to provide reliable, customized and guaranteed computing in a dynamic storage environment to end users with virtualized resources that are dynamically reconfigured to maintain a variable load. End users access these services through the internet without knowing their location and management.

2.2 Data Integrity

Data correctness, legality, and security are very important while considering storing data in the cloud environment. Data integrity is a guarantee that data can only be accessed or modified by those authorized to do so. Users depend on cloud service providers to provide reliable services by ensuring that their sensitive data and applications are stored in a secure manner. However this is not the case as service providers are at times dishonest and delete data that is rarely accessed to save storage space, therefore data integrity is a major concern and the need for techniques that can check data correctness as a solution to this concern. [7]

2.3 Current Data Integrity Proving Schemes with their advantages and limitations

There are many traditional methods for checking data integrity; however these methods cannot be directly applied to large data files stored on untrusted remote cloud servers. It’s not logical to download an entire file to perform integrity checking due to high cost of computation and bandwidth consumption. The basic schemes for data integrity in the cloud are PDP and PoR.

2.4 Provable Data Possession (PDP)

A PDP scheme checks a file which consists of a collection of n blocks retrieved by a remote cloud server. The data owner processes a data file to generate some metadata to store it locally. The file is then sent to the server and the owner deletes the native copy of the file. In this scheme, a client that has stored data on untrusted cloud server can verify that the server

possesses the original data without retrieving it. Ateniese et al. [8] first considered a public audit ability called provable data possession model to ensure possession of data files on untrusted cloud storages using Homomorphic Verifiable Tags. However, Ateniese et al. doesn't consider dynamic data storage, and the direct expansion of their scheme from static data storage to dynamic case may suffer design and security vulnerability. In their subsequent work [9], Ateniese et al. proposed a dynamic version of the prior PDP scheme problems. Wang et al. [10] considered the proposed challenge response protocol that can both determine data correctness and locate possible errors. Erway et al. [11] were the first to explore constructions for dynamic provable data possession. They extend the PDP model in [8] to support provable updates to stored data files using rank-based authenticated skip lists. This scheme is essentially a complete dynamic version of the PDP solution. They remove the index information in the "tag" computation in Ateniese's PDP model [9] to support update for block insertion, and employ authenticated skip list data structure to authenticate the tag information of challenged or updated blocks first before the verification procedure. Feifei Liu [12] also proposed an improved dynamic model that reduces the computational and communication complexity to constant by using Skip-List, Block, Tag and Hash method.

The PDP scheme operates in two stages that include setup and challenge stages. In the setup stage, the client generates a pair of matching keys public and secret key using a probabilistic key generation algorithm. The public key along with the file will be sent to the cloud storage by the client and he deletes the file from its local storage. In the challenge stage, the client challenges the cloud server for a proof of possession for a subset of blocks in the file. The client later checks the response from the cloud server.

Advantages:

- The server does not actually have to access the file blocks, hence supporting block less verification.
- The scheme allows public verifiability.

Limitations:

- Lack of error-correcting codes to address concerns of corruption.
- Lack of privacy preservations.
- No dynamic support.
- Unbound number of queries.

2.4.1 Basic PDP Scheme based on MAC

This scheme ensures data integrity of any file stored on the cloud storage server by enabling the data owner compute a Message Authentication Code (MAC) of the whole file with a set of secret keys and stores them locally before outsourcing it to the Cloud provider [13]. Users store only the computed MAC on their local storage, sends the file to the cloud provider, and deletes the local copy of the file. Whenever a user needs to verify the integrity of their stored data, they send a request to retrieve the file from the cloud, reveal a secret key to the cloud server and request to compute another MAC with the previously stored value.

Limitations:

- Number of verifications allowed is limited by the number of secret keys.
- The data owner has to retrieve an entire file to compute MACs.

- Public auditability not supported since private keys are required for verification.

2.4.2 Scalable PDP

It's an improved version of the original PDP that uses symmetric encryption while the original PDP uses public key to reduce computation overhead. The scheme can have dynamic operations on remote data and doesn't require bulk encryption. It relies on the symmetric-key which is more efficient than public key encryption and doesn't provide for public verifiability [8].

Limitations:

- A client can perform limited number of updates and challenges.
- It does not perform block insertions, only append type insertions are allowed.
- Problematic for large files since each update requires re-creating all challenges.

2.4.3 Dynamic PDP

This is a collection of seven polynomial-time algorithms (KeyGen DPDP, PrepareUpdate DPDP, PerformUpdate DPDP, VerifyUpdate DPDP, GenChallenge DPDP, Prove DPDP, and Verify DPDP) [14]. The scheme supports full dynamic operations like insert, update, modify, and delete but at relatively higher computational, communication and storage overhead. This technique uses rank based authenticated directories along with a skip list for inserting and deleting functions. The DPDP scheme has a computational complexity but still efficient, for example to perform a proof on a 500MB file, only 208KB proof data and a computational overhead of 15ms is required.

Limitations:

- The scheme has a computational complexity.
- It's not recommended to work on thin clients.
- It doesn't have provisions for robustness.

2.5 Proof of Retrievability (PoR)

Juels and Kaliski [15] describe a "proof of retrievability" model in which spot-checking and error correcting codes are used to ensure both "possession" and "retrievability" of data files on archive service systems. For detection purpose some special blocks called "sentinels" are randomly embedded into data files, and to protect the positions of these special blocks the file is encrypted.

Shacham and Waters [16] designed an improved PoR scheme with full proofs of security in the security model defined in [15]. They use publicly verifiable homomorphic authenticators built from BLS signatures based on which the proofs can be aggregated into a small authenticator value [17].

This is a cryptographic method for remotely verifying the integrity of files stored in the cloud without keeping a copy of the user's original files on the local storage. In this scheme, the user backups his data files together with some authentication data to a potentially dishonest cloud storage server. The user can check the data for its integrity while it's stored with the cloud service provider using an authentication key without retrieving the data file from the cloud. [7]

PoR works in two phases of setup and sequence of verification. In the setup phase, user preprocesses his data file using his private key to generate an authentication code and later sends

the data file together with this code to the cloud storage server and wipes it from his local storage completely. Therefore the user retains only the private key on his local storage while the cloud provider retains both the data file and its corresponding authentication code. In the sequence of verification phase, each sequence the user generates a random challenge query and the cloud provider is supposed to produce a proof upon receiving the challenge query based on the user's data file and the corresponding authentication information. In this case the user verifies the cloud provider's response using his private key and decides to accept or reject response.

Limitations:

- Works only with static data sets.
- Supports only a limited number of queries as a challenge.
- It deals with a finite number of check blocks.
- It doesn't provide prevention to files stored with the cloud provider.

2.5.1 PoR based on keyed hash function hk(F)

A keyed has function is very simple and easily implementable as it provides a strong proof of integrity. The user pre-computes a cryptographic hash of a data file F using hk(F) before outsourcing the data file to the cloud provider storage and stores a secret key K along with its computed hash. The user releases the secret key K to the cloud provider to check the integrity of the file F and asks it to compute and return the value of hk(F). If the user needs to check the integrity of the file F for multiple times he has to store multiple has values for different keys.

Limitations:

- Verifier needs to store keys for each of the check performed with its hash value.
- High implementation costs since hashing is performed on an entire file.
- Computing hash values for large data files is difficult on thin clients.

2.5.2 PoR Based on Selecting Random Bits in Data Blocks

This technique requires encryption of a few bits of data per data block instead of encrypting the whole data file F, hence reducing on the computational power and bandwidth requirements on the cloud storage clients [19]. In his scheme, the user doesn't store any data on his local machine and only required to store a single cryptographic key and two random sequence functions. Before outsourcing data to the cloud provider, the use preprocesses the data file and appends some Meta data to the data file and stores it at the cloud provider's storage servers. During verification, the verifier uses the Meta data to verify the integrity of the data. [20]

Limitations:

- The technique is only suitable for static data.
- No data prevention mechanism used in this technique.
- No implementation of data prevention mechanism in this technique.

2.6 High Availability and Integrity Layer (HAIL)

The scheme allows the user to store data on multiple servers in the cloud to enable data integrity checking performed on redundancy data [18]. The technique uses Message Authentication Codes

(MACs), pseudorandom function, and the universal hash function to ensure integrity process. The proof generated by this method is independent of its data and compact size.

Limitations:

- Mobile adversaries easily attacks and corrupt data files.
- The scheme is applicable to static data only.
- It requires high computational power.
- Not suitable for thin clients due to high processing power.

METHODOLOGY

3.1 Introduction

The selection criteria through which the study sources will be evaluated will base on the research experience, and in order to select these sources I will consider certain constraints: studies to be included in the selected sources will be written in English and these sources are available on the web. The following list of sources will be considered: Science Direct, ACM digital library, IEEE digital library, Scholar Google and DBLP. Later, i will refine the results and include important works that has not been recovered in these sources and update the work taking into account other constraints such as impact factor, received cites, important journals, renowned authors, to mention but a few.

3.2 Research Design

This research begins by studying existing theories and techniques related to research problem area hence deductive approach is used. In order to carry out an analysis on data integrity in cloud environment, the study will adopt a simulation method because it offers the possibility to investigate systems or regimes that are outside of the experimental domain or the systems that is under invention or construction.

3.3 Data collection methods

Quantitative data collection methods will be used in this study, and particularly the self-administered questionnaire, which is mainly close-ended with a few open-ended questions to accommodate the qualitative aspect of the data collection. Semi structured interviews and observation methods will also be used. Methodologically, information need research in the public domain has equally moved from an early reliance on positivist surveys to the use of

diverse methodologies in a mix of quantitative and qualitative research tools; enabling a more holistic view to emerge from the researcher getting close to the data, thereby developing the analytical, conceptual and categorical components of explanation from the data itself'. The three methods, (questionnaire, interview and observation) will therefore be considered appropriate for this study.

3.4 Analysis and Interpretation

Descriptive analysis on variables such as techniques of data integrity, the current status of data integrity, the impact of data integrity and the hindrances to data integrity in cloud environment. This is achieved through using bar chart and line graph for visual interpretation. This is carried out for both local and offloaded task execution times. Local execution time is the time taken to solve the task when no LAN connection is available or when offloading is infeasible while offloading task execution time is the actual time that the whole process of offloading the task takes. Finally, tables, charts graphs and pie-charts will be applied as part of inferential statistics to rigorously ascertain an analysis of data integrity in cloud environment.

3.5 Quality Control

Epstein (1977) notes that the quality of a research report depends to a large degree on the accuracy, reliability, and validity of the measures it employs. He clarifies that measurement accuracy refers to the degree of freedom of error in the measuring process that is achieved in the study. This is concerned with whether or not mistakes are made in the clerical processing and tabulation of the data. In this study, a computerized data analysis package (SPSS version 16) will be used for easy and quick processing of the data and to ensure accuracy. The tabulated chi-square values will also be computed to confirm accuracy.

Epstein (1977) explains reliability as the consistency in response to a given set of measurements and the freedom from bias. In this study, three data collection instruments (questionnaire, interview and observation) will be used and in each of these, an effort will be made to ensure that the data collected on particular concepts in the study confirm the general conclusions derived from the responses given from the other. This will yield sound reliability and less bias in the study. This will also ensure triangulation of the instruments.

Validity ensures that the data sets collected or items used are pertinent or relevant to the research. Validity according to Epstein (1977) refers to the extent to which a measure measures what it is supposed to be measuring (i.e. the measurement device should directly be relevant to the concept being measured specifically referred to as content validity). This measure is considered appropriate because, only a measure of association of the variables will be required in the study.

3.6 Ethical issues

Approval and permission will first be sought to conduct the study with introductory letters from the department. Respondents' consent will be also first sought with a brief introductory letter as part of the questionnaire, with confidentiality of the information to be collected assured. The respondents will be assured that this study is meant for only academic purposes and that permission will be sought for future use of the findings. The purpose of the data collection will therefore clearly be explained to the respondents to ensure that their responses are not biased but genuine. The respondents' privacy of information will be ensured by not disclosing the names of the individuals and every effort made not to exploit the respondents' responses.

References

[1] J. Ruiter and M. Warnier, Privacy regulation for cloud computing, compliance and implementation in theory and practice, article.

[2] P. Metri and G. Sarote, Privacy Issues and Challenges in Cloud Computing, International journal of Advanced Engineering Sciences and Technologies, Vol. No. 5, Issue No. 1,001-006.

[3] Amazon.com, “Amazon s3 Availability Event: July 20, 2008,” July 2008. http://status.aws.amazon.com/s3 20080720.html

[4] M. Arrington, “Gmail Disaster: Reports of Mass Email Deletions,” Dec. 2006. http://www.techcrunch.com/2006/12/28/gmail-disasterreports-of-mass-email-deletions/

[5] http://aws.amazon.com/agreement/

[6] Luo, Jun-Zhou, et al. "Cloud computing: architecture and key technologies. "Journal of China Institute of Communications 32.7 (2011): 3-21.

[7] Rajkumar Chalse, Ashwin Selokar, Arun Katara, A New Technique of Data Integrity for Analysis of the Cloud Computing Security, CICN 2013, pp.469-472.

[8] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, “Scalable and Efficient Provable Data Possession,” in Proceedings of SecureComm ’2008.

[9] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, “Scalable and Efficient Provable Data Possession,” Proc. Fourth Int’l Conf. Security and Privacy in Comm. Networks (SecureComm ’08), pp. 1-10.

[10] Wang, Q. Wang, K. Ren and W. Lou, “Ensuring Data Storage Security in Cloud Computing,” Proc. 17th Int’l Workshop Quality of Service (IWQoS ’09), 2009.

[11] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia,“Dynamic Provable Data Possession,” Proc. 16th ACM Conf. Computer and Comm. Security (CCS ’09), 2009.

[12] Feifei Liu, Dawu Gu, Haining Lu," An Improved Dynamic Provable Data Possession Model," 978-1-61284-204-2/11/$26.00 ©2011 IEEE

[13] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to keep online storage services honest," in Proceedings of the 11th USENIX workshop on Hot topics in operating systems, 2007.

[14] Berkeley, CA, USA, 2007, pp. 1–6. C. Erway, A. K¨up¸c¨u, C. Papamanthou, and R. Tamassia. Dynamic provable data possession in Proceedings of the 16th ACM conference on Computer and communications security, CCS '09, New York, NY, USA, 2009.

[15] A. Juels and B.S. Kaliski Jr., "Pors: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 584-597, 2007

[16] H. Shacham and B. Waters,"Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '08), pp. 90-107, 2008.

[17] D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," Proc. Seventh Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '01), pp. 514-532, 2001.

[18] K.D. Bowers, A. Juels, and A. Oprea, HAIL: A high availability and integrity layer for cloud storage, in Proceedings of 16th ACM conference on Computer and communications security, 2009.

[19] R. Sravan kumar and Saxena,"Data integrity proofs in cloud storage" in Proceedings of IEEE 2011.

[20] R. Pandya, K. Sutaria, "An analysis of privacy techniques for data integrity in the cloud environment", International Journal of Computer and Electronics Engineering, (Dec 2012) ISSN: 0975-4202

Appendix:

Budget (Ugandan Shillings)

ITEM	QTY	UNIT COST (UGX)	TOTAL (UGX)
1) Data Collection			
a) Transport	1	900,000	900,000
b) Airtime/Internet Bundles	1	500,000	500,000
2) Stationary			
a) Printing	500 Pages	200/=	100,000/=
b) Binding	6 Copies	12,000/=	72,000/=
c) Photocopying	300 Pages	100/=	30,000/=
d) Pens	5	500/=	2,500/=
e) Books (Note Books)	3	5000/=	15,000/=
3) Equipments			
a) Laptop	1	1,500,000/=	1,500,000/=
4) Miscellaneous			300,000/=
		GRAND TOTAL	**3,419,500/=**

Work Plan:

No.	ACTIVITY	START DATE	END DATE
1	**Proposal Writing**		
	a) Problem Formulation & Structuring	1st April, 2016	2nd April, 2016
	b) Literature Review	3rd April, 2016	4th April, 2016
	c) Methodology	5th April, 2016	6th April, 2016
	d) Compilation & Printing	7th April, 2016	8th April, 2016
2	**Proposal Presentation**		
	a) Submission	24th January, 2017	25th January, 2017
	b) Defense	N/A	N/A
3	**Implementation**		
	a) Data collection	15th April, 2016	20th January, 2017
	b) Editing	21st April, 2016	25th February, 2017
4	**Final Dissertation**		
	a) Dissertation Writing	16th February, 2017	26th February, 2017
z	b) Compilation & Printing	27th February, 2017	27th February, 2017
	c) Submission	30th February, 2017	30th September, 2017
	d) Defense	N/A	N/A