**CLOUD COMPUTING SECURITY ENHANCEMENT**

**A PROJECT REPORT**

*Submitted in partial fulfillment for the award of the degree*

of

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE AND ENGINEERING**

by

**DEBANJAN DEB (Reg. No.:RA1511003010534)**

**AVIRAL VERMA (Reg. No.:RA1511003010688)**

*Under the guidance of*

**Ms. J.V VIDHYA**

Associate Professor, CSE



**FACULTY OF ENGINEERING AND TECHNOLOGY**

SRM Nagar, Kattankulathur- 603 203

Kancheepuram Dist

**MAY 2019**

**BONAFIDE CERTIFICATE**

Certified that this project report titled **“CLOUD COMPUTING SECURITY ENHANCEMENT”** is the bonafide work of **Debanjan Deb (Reg. No. RA1511003010534)** and **Aviral Verma (Reg. No. RA1511003010688)** who have carried out the project work under my supervision.Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

|  |  |
| --- | --- |
| Signature of the Guide  Ms. J.V VIDHYA  **SUPERVISOR**  Associate Professor  Department of CSE  SRM IST | Signature of the HOD  Dr. B. AMUTHA  **HEAD OF THE DEPARTMENT**  Department of CSE  SRM IST |

Internal Examiner External Examiner

Date:

Department of Computer Science and Engineering

**SRM Institute of Science & Technology**

**Declaration Form**

**Degree/ Course :**

**Student Name :**

**Registration Number :**

**Title of Work :**

I / We hereby certify that this assessment compiles with the University’s Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

* Clearly references / listed all sources as appropriate
* Referenced and put in inverted commas all quoted text (from books, web, etc)
* Given the sources of all pictures, data etc. that are not my own
* Not made any use of the report(s) or essay(s) of any other student(s) either past or present
* Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
* Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalised in accordance with the University policies and regulations.

# ABSTRACT

Cloud Computing is a technology that has seen a spurious growth recently and is being deployed for personal as well as business purposes. Cloud computing is defined as the delivery of computing services – namely servers, storage, databases, networking, software, analytics and intelligence over the Internet to offer faster innovation, flexible resources and economies of scale.

With the feature becoming increasingly integral to various services provided across the inter-connected digital world, it is imperative that its susceptibility be assessed and security made impenetrable to protect the sensitive information Cloud servers store. Cloud security has been vulnerable to threats and in several cases has led to Data Loss, Information Hacking and Denial of Services. These incidents have given rise to widespread concern regarding the data security that these Cloud Services employ. However, security models and security tools are being continually enhanced. This project aims to define the term “Cloud Computing”, its functionality and implementation, define the utility and essentiality of Cloud Security and refer to its existence.

Index terms Cloud Security, Diffie-Hellman Key Exchange, AES Encryption Standard, Multiple Blocks Cloud Storage.

**ACKNOWLEDGEMENT**

We express our humble gratitude to **Dr. Sandeep Sancheti**, Vice Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. C. Muthamizhchelvan**, Director, Faculty of Engineering and Technology, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank **Dr. B. Amutha**, Professor & Head, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for her valuable suggestions and encouragement throughout the period of the project work.

We are extremely grateful to our Academic Advisor **Dr. K. Annapurani,**Associate Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for her great support at all the stages of project work.

We would like to convey our thanks to our Panel Head**, Dr. M Murali,**Assistant Professor Department of Computer Science and Engineering, SRM Institute of Science and Technology, for his / her inputs during the project reviews.

We register our immeasurable thanks to our Faculty Advisor, **Ms.M.Gayathri,** Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide **Ms. J.V. Vidhya** Assistant Professor,**,**Department of Computer Science and Engineering, SRM Institute of Science and Technology, for providing me an opportunity to pursue my project under his mentorship. He / She provided me the freedom and support to explore the research topics of my interest. Her / His passion for solving the real problems and making a difference in the world has always been inspiring.

We sincerely thank staff and students of the Computer Science and Engineering Department, SRM Institute of Science and Technology, for their help during my research. Finally, we would like to thank my parents, our family members and our friends for their unconditional love, constant support and encouragement.

Debanjan Deb

Aviral Verma

**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**

**BONAFIDE i**

**DECLARATION FORM ii**

**ABSTRACT iii**

**ACKNOWLEDGEMENT iv**

**LIST OF TABLES vii**

**LIST OF FIGURES viii**

**LIST OF SYMBOLS AND ABBREVIATIONS ix**

**1 INTRODUCTION** 1

1.1. Background 2

1.2. Innovation Idea 2

1.3. Evolution 3

## 2 LITERATURE SURVEY 4

2.1. Security Provenance: The Bread & Butter 4

of Data Forensics in Cloud Computing

2.2. Cloud Computing: Security & Risk 4

Classification

2.3. Cloud Computing Issues, Research & 5

Implementation

2.4. Towards a Data Centric View of Cloud 5

2.5. Security Audits of Multi-Tier 6

Infrastructures in Public Infrastructure

2.6. Transparent Security for Cloud 6

2.7. Cloud Hooks: Security & 7

Privacy Issues in Cloud

2.8. Managing Security of Virtual Machine 7

Images in Cloud Environment

2.9. A Client-Based Privacy Manager for 8

Cloud Computing

2.10. Data Protection Models for 8

Service Provision in Cloud

2.11. Literature Survey Summary 9

**3 MODULES** 11

3.1 Algorithms

3.1.1. Data Encryption Module 11

3.1.2. Key Exchange Module 13

3.1.3. Data Partitioning Module 14

3.2 Architecture Diagram 15

3.3 ER Diagram 16

**4 CLIENT SIDE ENCRYPTION PHASES** 18

**5 RESULTS** 22

**6 CONCLUSION** 25

**REFERENCES** 26

**APPENDICES** 27

# 

# LIST OF TABLES

TABLE NO. TITLE PAGE NO.

2.1.1 Literature Survey Table 9

# 

# LIST OF FIGURES

Figure No. TITLE PAGE NO.

3.1 System Architecture Diagram 15

3.2 Process Flow Diagram 16

3.3 ER Diagram 17

5.1 Original Directory 31

5.2 Server Side Program 32

5.3 Partitioned Files 33

5.4 Encrypted Directory 33

5.5 Uploaded Files on Tonido Server 34

5.6 Downloaded Files Client Side (Zipped) 34

5.7 Downloaded Files Client Side (Extracted) 35

5.8 Client Side Program 36

5.9 Decrypted Directory 37

5.10 Merged Files 38

**ABBREVIATIONS**

**AES-256** Advanced Encryption Standard 256 bits key

**DH** Diffie – Hellman

**EU** End User

**TPA** Third Party Auditor

**CSP** Cloud Service Provider

**CHAPTER 1**

**INTRODUCTION**

**CLOUD COMPUTING:**

Cloud Computing is defined as the distribution of computing amenities—storage, servers, networking, analytics, software, databases and intelligence across the Internet to offer faster innovation, supple resources and scalar economies.

The cloud services are available in the form of the following representations:

1. Software as a Service (SaaS)
2. Platform as a Service (PaaS)
3. Infrastructure as a Service (IaaS)

Cloud Computing has been in a steep ascent in the recent years due to the flexibility the service provides to each of the Service Providers and the Customers. It provides a shared pool of resources that are accessible over the internet over multiple devices. Data is stored on a server on the internet instead of storing locally. This enables easier access of data as well as frees up local storage space.

Cloud computing also enables a number of resources to be interconnected and work on a single resource. There are various benefits associated with Cloud Computing. These are be summarized as below,

**Cost:** Cloud Computing eliminates the expenses of software and hardware in bulk at datacentres as the service requires limited components. There is no requirement of setting up on-site data centres; instead, all the data is accessible across a shared platform on the internet.

**Speed:** Cloud Computing increases the speed of service as the service is provided over the internet on a pool of resources. Thus, the service is not monopolised and independent of a single system.

**Productivity:** Cloud computing minimises on-site operation such as “racking and stacking”—setting up hardware, patching software and other laborious IT management tasks.

**Performance:** The greatest benefit of Cloud computing comes in the enhanced performance of the service. As these services are provided over a large network of datacentres, there is regular Updation of software and gives reliable performance.

**CLOUD SECURITY:**

**Cloud Security** refers to the mechanisms, technologies and policies arrayed to safeguard data, communication and infrastructure of the Cloud service. Cloud Security is an integral measure of the Cloud computing service. It is a sub-category of Information Security and Network Security.

* 1. **BACKGROUND**

As per the current system, cloud security is still an area of concern. With the help of cloud computing, organizations can utilize services and information is stored at a physical location that outreaches their control. This utility gave rise to various security questions like privacy, confidentiality, integrity and demand a reliable computing atmosphere wherein data discretion be upheld. In order to sufficiently rely on the service of computing, there is an imminent call for a system that performs verification, authentication and encrypted transfer of data, thereby preserving data privacy. Here in our project we are trying to point out some measures, which will make it more safe and secure.

* 1. **INNOVATION IDEA**

The idea that motivates this project is the collaboration of cryptography with cloud computing to ensure an efficient security mechanism that successfully protects data storage as well as authenticates data access.

At its core, Cloud Services are nothing but widely inter-connected network of configurable system resources.

Cryptography is equipped with various protocols, in our focus namely the Diffie-Hellman Key Exchange and Advanced Encryption Standard (AES-256) that are utilised for secure data transfer over a network.

The idea is to implement this feature of cryptography on cloud servers to resolve the concerns over Cloud Security.

Moreover, the purpose is to build an implementation that is capable of partitioning the various data for storage over distributed server systems. The partitioning of data along with encryption and access protocol will provide an efficient encryption scheme for cloud computing enhancing its security and providing more power to client.

* 1. **EVOLUTION**

The work done until date can be distributed into three periods:

1. Idea period
2. Pre cloud period
3. Current Cloud period

During1960s, John McCarthy, Douglas Parkhill, and others reconnoitred the idea of computing in the form of a public service. The idea was to use commodity hardware and software, computing resources to delivered an infinite elastic online public utility.

In the late 1960s, J.C.R. Licklider inspired the ARPANET, and in the early 1970s global networking became a reality. In the early 1980s, the TCP/IP suite emerged as the protocol for ARPANET, and the Domain Name System (DNS) established naming designations for websites.

The now widely recognised cloud phase began the year 2007 when the cataloguing of IaaS, PaaS, and SaaS were officially finalized. The cloud-computing chapter of history has seen some peculiar and intriguing breakthroughs initiated by the leading web officialdoms of the digital world.

**CHAPTER 2**

## LITERATURE SURVEY

**2.1 SECURITY PROVENENCE: THE ESSENTIAL BREAD AND BUTTER OF DATAFORENSICS IN CLOUD COMPUTING**

Rongxing and his team in their publication presented in the ASIACCS suggested a new security system wherein they applied the algorithm called Bilinear Pairing Method. They proposed of a novel security and data provenance mechanism for cloud security systems. In their proposition, secret documented files shall be included on the tree of the users file system, and the tool will be capable of supporting data security. They provided an authentication mechanism to verify user access so that unauthorized user access can be filtered away. The provenance prospect is a process of the ‘bilinear pairing method’, in which blocks of data forensics are built within the environment. Data is then paired with these blocks for security and authentication. The idea is to verify access by resolving disputes of data forensics. They tested their model using multiple security techniques and were able to prove functionality. However, they could not implement the said system as there were occurrences of complexities on the mathematical models used.

**2.2 CLOUD COMPUTING: SECURITY AND RISK CLASSIFICATION**

In their research paper published in the ACMSE, R. La’Quata Sumter stated that the evolution of cloud computing implementation amounts to internet security doubted and constant increase of threats. Clients of cloud vendors and services are earnestly discouraged by the weakness of cloud security to protect sensitive information. Users distrust the security mechanisms that are implemented by the service providers. In order to reassure clients, they have proposed a model that keeps track of every move and process that is taking place on the information stored on the servers. To achieve this, they necessitated a security capture device that will provide the said functionality and make their model work. The advantages of their system are that they have been dealing with customer reinforcement about the security concerns. But the model is limited only to small Cloud Environments as it loses its practicality on large multi-server cloud systems.

**2.3 CLOUD COMPUTING ISSUES, RESEARCH AND IMPLEMENTATION**

Mladen A. Vouch, in the publication said that cloud computing evolved as a system after several practical years on networking and computer technology. The paper primarily focuses on concerns in respect to ‘cloud computing with virtualization, cyber infrastructure, service oriented architecture and end users’. They undertook and stated key concerns and implementation research that made their work important. But dissatisfaction on the user’s side led them to write theoretical papers based on security concepts and issue authentication.

**2.4 TOWARDS A DATA CENTRIC VIEW OF CLOUD SECURITY**

Wenchao et al. in their exploration offered an alternate point of view of arrangements through information driven data the executives. They completely examined the security necessities of verifying information and sharing of information through online applications. They included dialogs on criminology, framework investigation and information the board. They proposed of another security stage known as Declarative Secure Distributive Services condensed as DS2, a stage that bolsters the elements of the offered information verifying strategies. 'Secure Network Data log (SeNDlog)' a programming language that forms systems administration and access control rationale based errands deals with the system conventions and security strategies. Helped by Rapid Net definitive systems administration motor, they had the capacity to create DS2 model and included provenance support, as per their conviction that it will make the security level progressively steady. The strong point of their work lies in the information driven security that outcomes in secure question handling, framework examination and legal sciences, proficient start to finish check of information. The work done by them anticipates assessment from expert cloud sellers.

**2.5 SECURITY AUDITS OF MULTI-TIER INFRASTRUCTURES IN PUBLIC INFRASTRUCTURE CLOUD**

The benefits associated with cloud computing services such as safety, security and privacy, compelled Soren et al to explain the majority of the influence of cloud services. To provide invulnerability to Cloud systems against threats, complex and good management of web interfaces is essential. Their implementation was over the platform, “Amazon’s Elastic Compute Cloud (EC2)”. They connected a security examination device and reenacted it to genuine variables. They proposed complex abnormal state inquiry language and utilized it to portray the necessities of the arrangement. Python and EC2 were the unmistakable programming used. The system recognizes any ruptures on the safe segments of the foundation and afterward continues to educate the chairmen to break down the issue. In this manner it very well may be said that the product works like an antivirus program. While they researched each conceivable security assault with the proposed apparatus, the burden of the device is that the product is connected to work just with the Amazon EC2 and no other general frameworks.

**2.6 TRANSPARENT SECURITY FOR CLOUD**

Flavi and Roberto suggested an Architecture and Transparent Cloud Protection System (TCPS) to ensure greater security. They claimed to have accomplished integrity in relation to cloud privacy issues. To recognize these, they built feasibly secure architecture called TCPS. The system had potential to be used in order track every host transfer but as well as maintain the virtualization and transparency of the server. Their resultant work is that they managed to create an intrusion detection mechanism built in the architecture but failed to tackle realistic scenarios thereby lacking validation.

**2.7 CLOUD HOOKS: SECURITY AND PRIVACY ISSUES IN CLOUD COMPUTING**

Wayne stated the essentiality of configuring security on critical systems. Facing security issues from end user perspective is mandatory. Security policies with strong commands should keep data checked for dangerous actions and prevent unauthorized access to both clouds and data servers. Their paper focuses on public clouds. Key factors are end user trust, insider access, visibility, risk management, client-side protection, server-side protection, and access control and identity management. The weakness in their work is that they did not outcome of a tool, or a solution on real infrastructure.

**2.8 MANAGING SECURITY OF VIRTUAL MACHINE IMAGES IN A CLOUD ENVIRONMENT**

Jingpeng et al proposed a paper on cloud’s image database. Their design addresses the risks and can be easily implemented and prove success. Filters in the system infrastructure capture malware and secondly all sensitive data to crack passwords are removed and replaced by stronger ones. Clients can choose the required images. Repository maintenance decreases the possibility of running illegal software. The testing’s of this papers show that filters work efficiently in the image management system. They proposed a system “different” from other cloud architectures and showed with aid of filters and scanners that they could detect malicious traffic. The weakness is that captures of filters are not 100% accurate and could lead to legitimate issues as also the scanner cannot capture every type of virus and it has to be updated constantly.

**2.9 A CLIENT-BASED PRIVACY MANAGER FOR CLOUD COMPUTING**

Miranda and Siani are facing problems of data seepage user complain about. This issue puts a serious obstacle on the acceptance of the implementation of cloud and its growth on the market. Some scenarios have been taken under consideration. A client-based privacy manager tool for processing sensitive information inserted in the cloud is proposed. The tool reduces security issues as simultaneously increases privacy safety. The tool has been tested successfully and used in many environments.

**2.10 DATA PROTECTION MODELS FOR SERVICE PROVISIONING IN THE CLOUD**

Dan and Anna proposed a data protection framework for sensitive information. Their proposed framework contains three basic keys: policy ranking, integration and enforcement. Various models have been described for each part. They presented security data models but also cost functions. Their work is tested and simulated but not validated on real environments.

**LITERATURE SURVEY TABLE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Author/Year** | **Title** | **Main Focus** | **Algorithm** | **Drawbacks** |
| Rongxing et al  **Publisher :**  ASIACCS  **Year :**  2010 | Security Provenance: The Essential Bread & Butter of Data Forensics in Cloud Computing | New Security and Provenance Data Forensics Tool | Bilinear Pairing Method | Complex Mathematical Models |
| R. La’Quata Sumter  **Publisher :**  ACMSE  **Year :**  2009 | Cloud Computing: Security & Risk Classification | Store Information of every Process on Cloud Servers | Security Capture Device | Practical only for Small Cloud Environments |
| Mladen A. Vouch  **Publisher :**  IEEE  **Year :**  2008 | Cloud Computing Issues, Research & Implementation | Cloud Computing with Virtualization, Cyber Infrastructure, SOA and End Users | Issue Authentication | User Dissatisfaction |
| Wenchao  **Publisher :**  IEEE  **Year :**  2010 | Towards a Data Centric View of Cloud Security | Data Centric Perspective, Forensic, System Analysis and Data Management | Declarative Secure distributed Systems (DS2),  Secure Network Data Log (SeNDlog) | Complex Implementation. Awaits Evaluation from Cloud Vendors |
| Soren Bleikertz  **Publisher :**  CCSW  **Year :**  2010 | Security Audits of Multi-Tier Infrastructures in Public Infrastructure Clouds | Implementation of Cloud Security Analysis and simulation to real factors | Complex Query Language over Amazon’s Elastic Compute Cloud (EC2) and Python | Software connected to work only with EC2 and not generic. |
| Flavio Lombard, Roberto Di Pietro **Publisher :**  IEEE  **Year :**2010 | Transparent Security for Cloud | Transparent Cloud Protection System (TCPS) for better Security Management | Secure Architecture system of TCPS | Failed to deploy realistic scenarios and validate their work |
| Wayne A. Jansen  **Publisher :**  IEEE  **Year :**  2011 | Cloud Hooks: Security and Privacy Issues in Cloud Computing | The essentiality of configuring security on critical systems | Revamped Security Policies with Strong Commands | Unknown outcome of tool or a solution on real infrastructure |
| Jinpeng et al  **Publisher :**  CCSW  **Year :**  2009 | Managing Security of Virtual Machine Images in a Cloud Environment | Cloud’s Image Repository. | Use a Filter to capture malware and replace all sensitive passwords with stronger ones. | Captures weren’t 100 % accurate and could lead to legitimate issues. Required regular Updation |
| Miranda & Siani  **Publisher :**  IEEE  **Year :**  2009 | A Client-Based Privacy Manager for Cloud Computing | Manager tool for information processing in the cloud | Client-side Key Authentication Mechanism | Lack of implementation in all scenarios |
| Dan Lin, Anna Squicciarini  **Publisher :**  SACMAT  **Year :**  2010 | Data Protection Models for Service Provisioning in the Cloud | Data Protection Framework for sensitive information | Policy Ranking, Integration and Enforcement | Not validated on real environments. |

**Table 2.1**

## CHAPTER 3

## MODULES

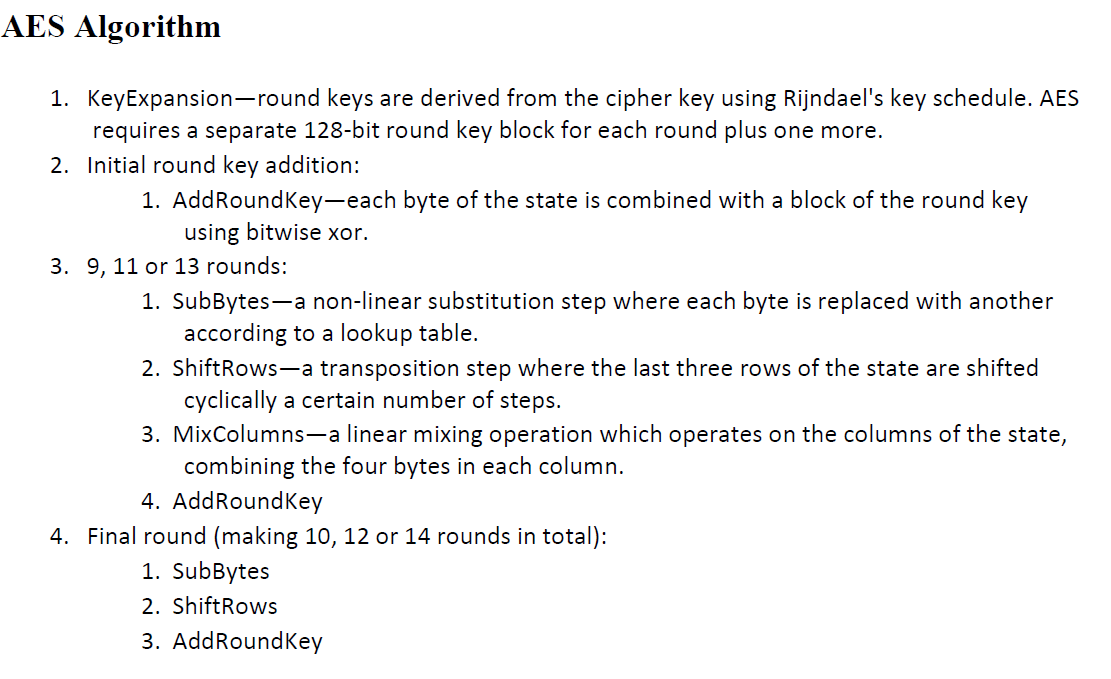
## 3.1 ALGORITHM

## 3.1.1 DATA ENCRYPTION MODULE

The Data Encryption Module is responsible for encryption and decryption of a data at the client side. It is a simple module that serves the purpose of encrypting each file individually. It is designed with cloud storage in mind.

The Data Encryption Module uses the AES-256 (32 Bit Key) for encrypting the data. Using the EncryptFS module, the client can specify a chosen directory that contains the various files required to be encrypted before upload on the cloud storage. The said directory can also be the primary cloud sync folder.

The module has been built using Python 2.7 and the PyCrypto module. At execution it asks for the complete address of the directory containing sensitive files and demands a 16/24/32-bit key. The key specified by the user is then used to create and encrypted index file and various files with randomly generated names. The data in the files along with its metadata is safely encrypted. Later, the existing EncryptFS can be used to decrypt the files in the index.

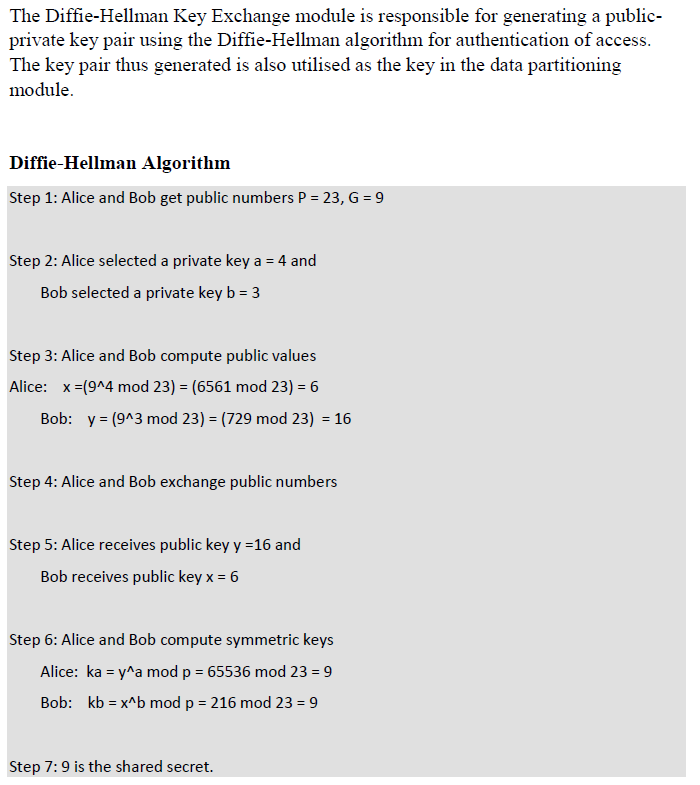
****

This module also provides with renaming the various files in a directory with a random 1-bit hex number thereby hiding the original file names and metadata such as file type, date etc.

Indexing is used to communicate the original filenames to the receiver’s end for proper decryption. The module creates a dictionary with the original file names and new file names as a key-value pair. This index is stored as a Java Synchronised Object file and further encrypted to be uploaded along with the other files.

**3.1.2 DIFFIE-HELLMAN KEY EXCHANGE MODULE**

The Diffie-Hellman Key Exchange module is responsible for generating a public-private key pair using the Diffie-Hellman algorithm for authentication of access. The key pair thus generated is also utilised as the key in the data partitioning module.

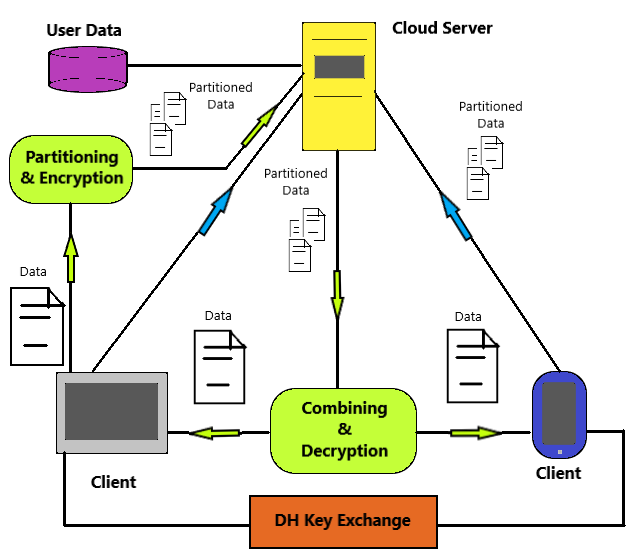


**3.1.3 DATA PARTITIONING MODULE**

The Data Partitioning Module serves the purpose of partitioning the various files to be stored on the cloud into unrecognisable chunks that can be then regrouped at the receiver’s end only through access by the key generated by the Diffie-Hellman process. In the existing system, the data is stored on the cloud utilising dynamic operations on data along with computations that requires the user to create a copy for further updation and verification of the loss in data. An efficient distributed storage auditing mechanism is proposed which has the ability to overcome the limitations of data loss handling.

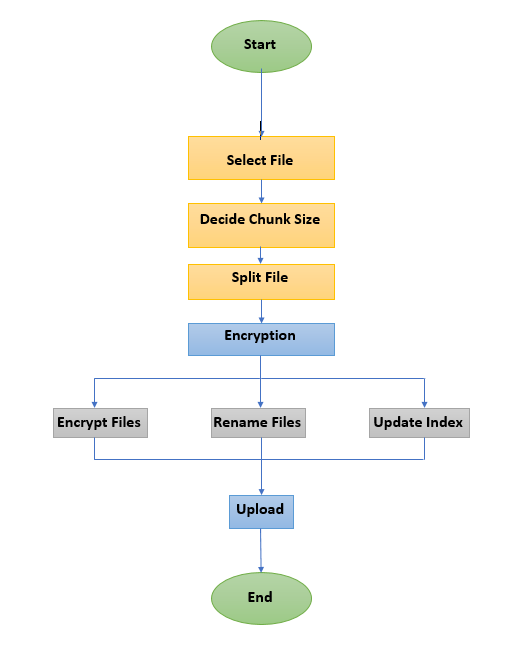
The files once decrypted are matched using regular expressions to facilitate the organising of the various part files matched as objects into a single file that replicates the original.

**3.2 ARCHITECTURE DIAGRAM**



**Figure 3.2 Architecture Diagram**

**3.3 PROCESS FLOW DIAGRAM**



**Figure 3.3 Process Flow Diagram**

**3.4 ER DIAGRAM**



**Figure 3.4 ER Diagram**

**CHAPTER 4**

**CLIENT SIDE SECURITY ENHANCEMENT PHASES**

The process involved in proposed system is comprised of the following modules,

i. Partitioning

ii. Encryption

iii. Key-Exchange

iv. Decryption

v. Merging

Each section varies greatly in the methodologies.

**4.1. PARTITIONING**

a) The first phase of the system proposes partitioning of the data that is to be stored over a cloud storage server into several chunks.

b) The said functionality is achieved by splitting files of all formats such as \*.txt, \*.jpg, \*.mp4 etc. into smaller fixed size chunks using the partitioning module.

c) The system reads the files selected by the user and requests the chunk size i.e. the maximum size of each of the chunks the file will be divided into.

d) The optimal chunk size is dependent on the cloud service the customer is utilising. For example, Amazon S3 requires chunk size of 5 MB minimum whereas Azure demands they must be much less than 4 MB.

e) The program then reads the source file up to the specified chunk size and writes on to an output part file iteratively until the end of file is reached.

f) Thus, the client is provided with numerous divisions of a single file that can be further encrypted and uploaded on to a cloud storage.

**4.2. ENCRYPTION**

a) The Data Encryption Module holds the responsibility of encryption and decryption of a data at the client side. It is a simple module that serves the purpose of encrypting each file individually. It is designed with cloud storage in mind.

b) The Data Encryption Module uses the AES-256 (32 Bit Key) for encrypting the data. Using the module, the client can specify a chosen directory that contains the various files required to be encrypted before upload on the cloud storage. The said directory can also be the primary cloud sync folder.

c) The module has been built using Python 2.7 and the PyCrypto module. At execution, it asks for the complete address of the directory containing sensitive files and demands a 32-bit key.

d) The key specified by the user is then used to create and encrypted index file and various files with randomly generated names.

e) The data in the files along with its metadata is safely encrypted.

**4.3. KEY-EXCHANGE**

a) The Diffie-Hellman Key Exchange module is responsible for generating a public-private key pair using the Diffie-Hellman algorithm for authentication of access.

b) The key pair thus generated is used to exchange the 256 bit AES key for decryption.

**4.4. DECRYPTION**

a) At the receiver’s end, the encrypted files when downloaded are decrypted using the AES key and the index.

b) The index provides the information pertaining to various source file names and their corresponding randomly generated names.[7] It also gives information regarding original file formats.

c) The files are then decrypted in similar manner as the AES Decryption process.

d) The result is a series of decrypted files that require merging to access the original files.

**4.5. MERGING**

a) The system merges the various part files by accessing the manifest to gain information regarding the original file names.

b) When the user provides the original source file name along with the command to merge it’s part files, the system obtains all the part files downloaded.

c) The system uses regular expression to match all the part files in the directory with the source file name.

d) For each chunk matched, the system reads the data stored in them and writes them onto a new file that replicates the source fil, thereby merging all the chunks.[12]

e) The part files are then deleted from the directory.

**CHAPTER 5**

**RESULTS**

**5.1 SERVER SIDE**

In several test runs across multiple devices, we proceeded to select a directory containing various files of different sizes and types such as images, videos, documents, PDFs, audios etc.

The directory as stated by the user in the program is then used as the home directory of the File Splitting module. For each file present in the directory, the program demands of the user the optimal chunk size that will become the maximum size of each chunk file the original file is split into.

In our test run depicted in the Appendix 2 of this report, we selected three files namely, *696.pdf* (PDF Document), *Batman.mkv* (Video)and *Sunrise.mp3* (Audio). With an optimal chunk size provided by the user of 500 KB, the said files were split into 4, 3 and 2 parts respectively.

Each chunk file limited at a maximum of 500 KB while few with the residual size observed as the last part files. These part files are named in the format *file\_name-(part number).file\_type.*

Thus once all the files in the directory are partitioned, the program moves onto the second module i.e. the Directory Encryption Module.

The user is then prompted by the program to provide with a key of amenable size.

In this next stage, the directory as provided by the user is encrypted using the AES-256 algorithm with a 32 bit key generated by the password provided by the user and an Initialisation Vector produced using SHA algorithm.

The key value is then used to encrypt all the files in the directory also renaming each file by a random 16 bit Hexagonal number. This function adds to the security feature while maintaining the metadata associated with each file. Eg, *26fci97hjres45asutahijgiod887.enc*

The program also creates an index of all the files containing a dictionary with a key-value pair that imitates the *original filename – new filename* storage. This index is stored in the directory into an *index.json* file wich is also further encrypted.

The index file is deemed necessary for decoding original file names at the receiving end.

The user is finally prompted to communicate the password through DH Key Exchange Protocol as stated previously.

The whole directory is then ready to be uploaded onto any cloud server.

**5.2 CLIENT SIDE**

The client then downloads the said files onto their local machine from the cloud server. The process of decryption starts with the Directory Decryption Module.

In this module, the user now is asked of their keys regarding the DH Key Exchange Process to retrieve the password required to decrypt all the files. With the correct password in place the user enters this password prompted by the program to initiate the decryption process.

Firstly, the index file is read to rename all the files in the directory from random number to their original post encryption file names. The file is read onto the program and the dictionary is retrieved to splat out the required information.

The Decryption process then begins using the same 32 bit key to decrypt all the files at each thread into original data. As the process continues each file is restored to its original form i.e. their original names and types.

Thus once the decryption is complete we are left with the various part files such as *696.pdf-1, Batman.mkv-3* etc. The index file is the only exception of the process. It is left as is.

The part files are then merged in the File Merging Module wherein the user is prompted to provide the filename they wish to be merged. Using regular expressions, all the part files of the file demanded are matched and are rewritten as a single file.

As seen, *Batman.mkv-1, Batman.mkv-2* and *Batman.mkv-3* are merged into singular file *Batman.mkv.*

Similarly all the files are merged in the directory and the original data is restored.

## 5.3 PERFORMANCE ANALYSIS

Implementing Threading into the Encryption Process enhances the performance of the program as depicted by the graph below.

**Figure 5.3 Performance Analysis Graph**

## CHAPTER 6

## CONCLUSION

In conclusion, we have represented the security means that possess the ability to secure the data of users stored on cloud servers against malicious entities. Data Partitioning is observed providing an efficient scheme to split the customers’ data into changeable chunks.

Data Encryption is covered through utilisation of the AES Encryption Algorithm to encrypt the user data on client machine. Files are further protected through renaming into random file names that are indexed and communicated across the servers. The index file further encrypted is a source of valuable information.

Insecure breach into the server is completely avoided by Diffie-Hellman key exchange algorithm. Implementation of AES cryptographic algorithms over a cloud computing environment is also ventured. The data partitioning scheme provides a dynamic perspective towards protection & security over cloud systems.

An efficient program is presented that promises to give the much required power and control to the most at risk party of Cloud Computing, i.e. the customer.

Providing security to the customer’s data on the cloud will empower the standing of Cloud Computing & Storage.

## REFERENCES

[1]. V. Fusenig, A. Sharma 2012. Security architecture for cloud networking, in: 2012 International Conference on Computing, Networking and Communications (ICNC). Presented at the 2012 ICNC Conference.

[2]. B.R. Kandukuri, V.R. Paturi, A. Rakshit, 2009. Cloud Security Issues, in: IEEE International Conference on Services Computing, SCC 2009.

[3]. S. Ramgovind, M.M. Eloff, E. Smith, 2010. The management of security in Cloud computing, in: Information Security for South Africa (ISSA), 2010.

[4]. Karun Handa, Uma Singh, 2015. Data Security in Cloud Computing using Encryption and Steganography. International Journal of Computer Science and Mobile Computing. IJCSMC, 2015.

[5]. H. Tianfield, 2011. Cloud computing architectures, in: 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC). Presented at the 2011 IEEE SMC Conference.

[6]. Op – ed : Encryption, not restriction, is the key to safe cloud computing. https: / / www. nextgov.com/cloud-computing/2012/10/oped-encryp-tion-not-restriction-key-safe-cloudcomputing/58608

[7]. “ Cloud Security and Privacy ” , Tim Mather, Shahed Latif and Subra Kumaraswamy. – O’Reilly Book.

[8]. David Talbot. “ How Secure Is Cloud Computing?” Technology Review [Online].Available: http://www.technologyreview.com/computing/23951/ 2009

[9]. Tania Gaur, Divya Sharma, 2016. A Secure and Efficient Client-Side Encryption Scheme in Cloud Computing. I.J. Wireless and Microwave Technologies. IJWMT 2016.

[10]. Elminaam, DiaaSalama Abdul et al. "Performance Evaluation of Symmetric Encryption Algorithms." IJCSNS International Journal of Computer Science and Network Security 8.12. 2008

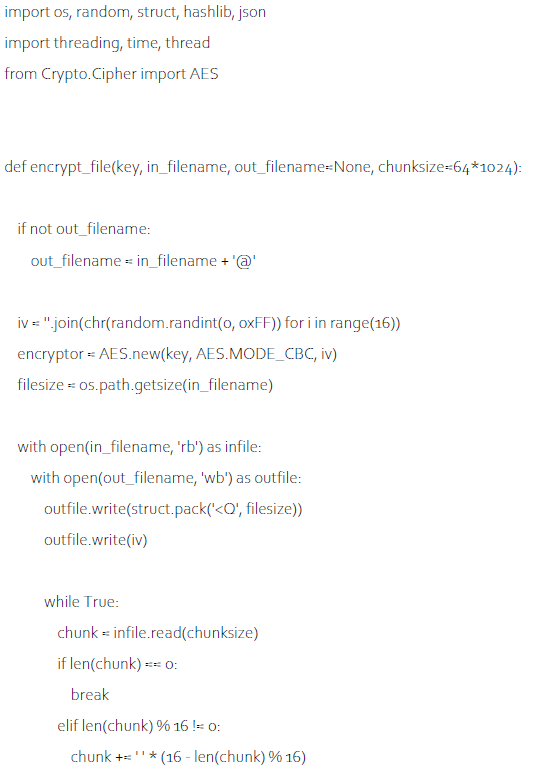
[11]. “Heuristics in Physical Design Partitioning “, Bhargab Sinha, Naushad Manzoor Laskar, Rahul Sen.

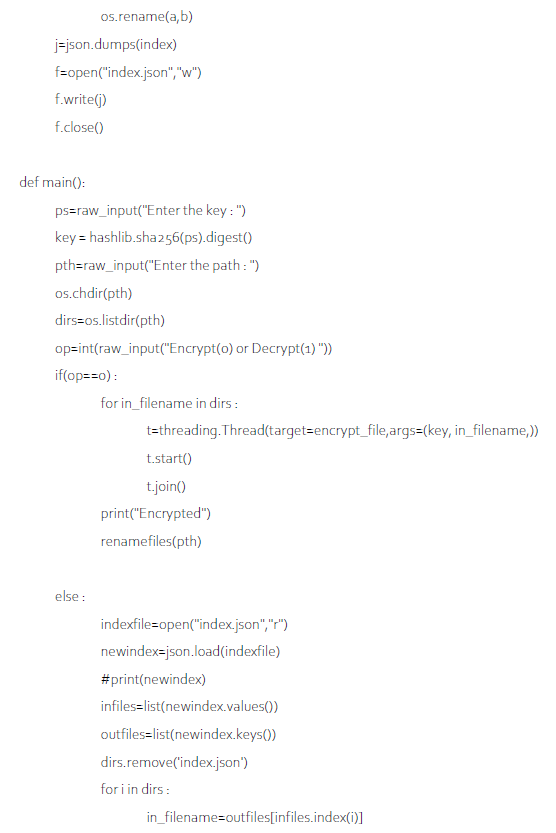
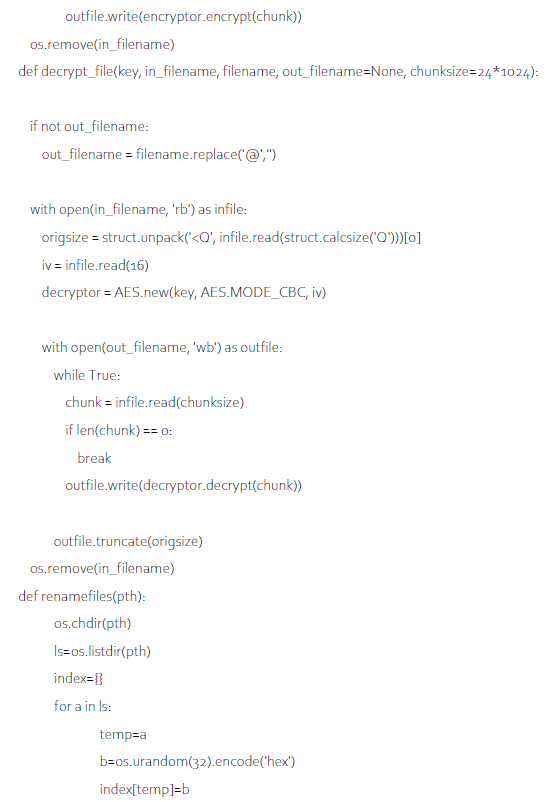
[12]. “Security Aware Partitioning for efficient file system search”, Aleatha Parker Wood, Christina Strong, Ethan L. Miller.

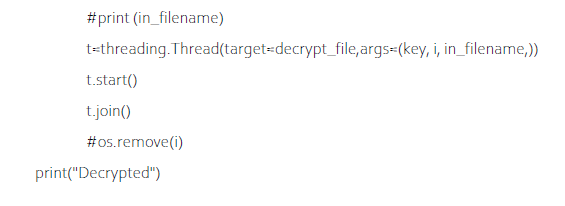
**APPENDIX 1**

**CODE**

**encrypt\_directory.py**







……

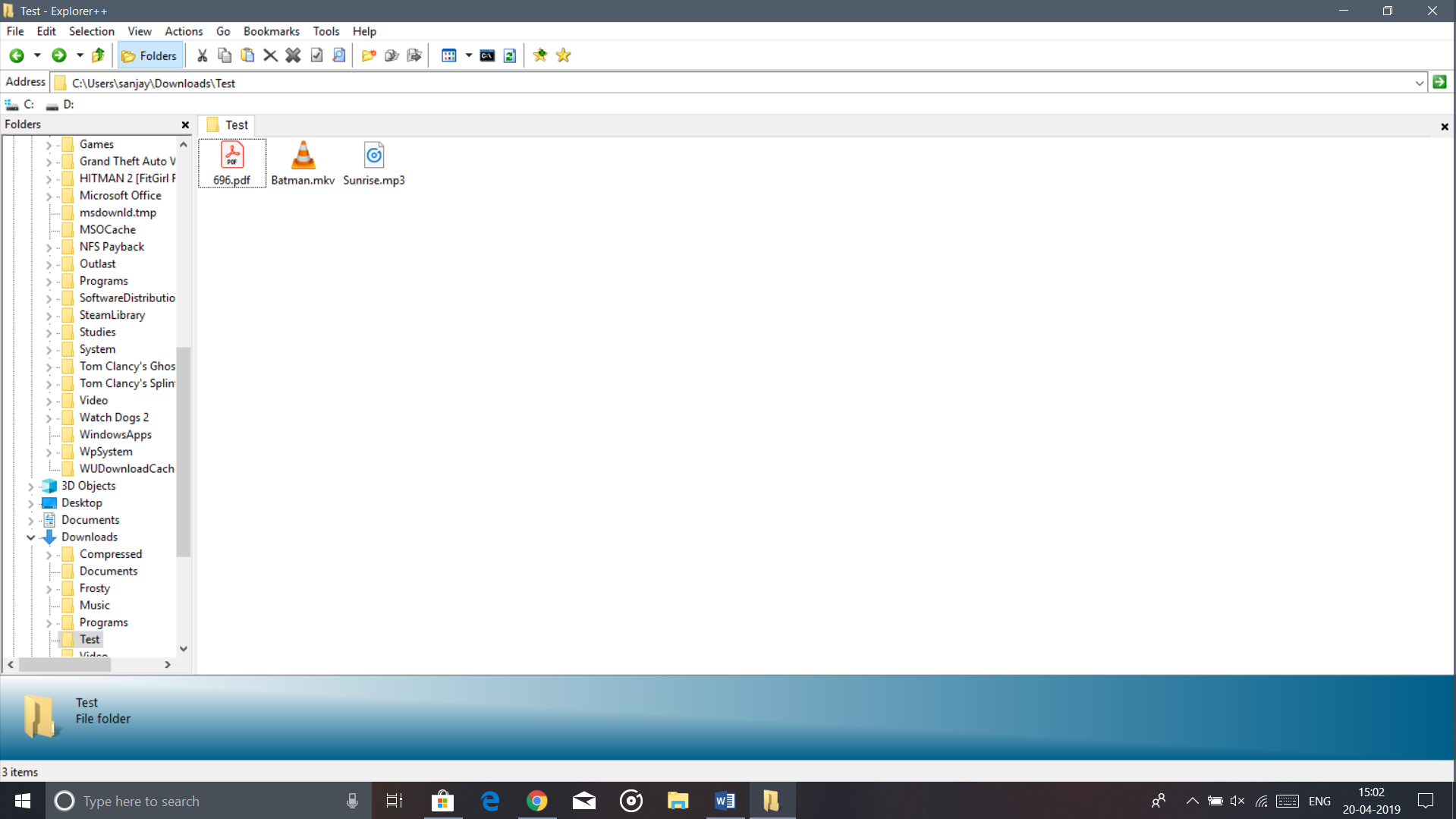
…...

..

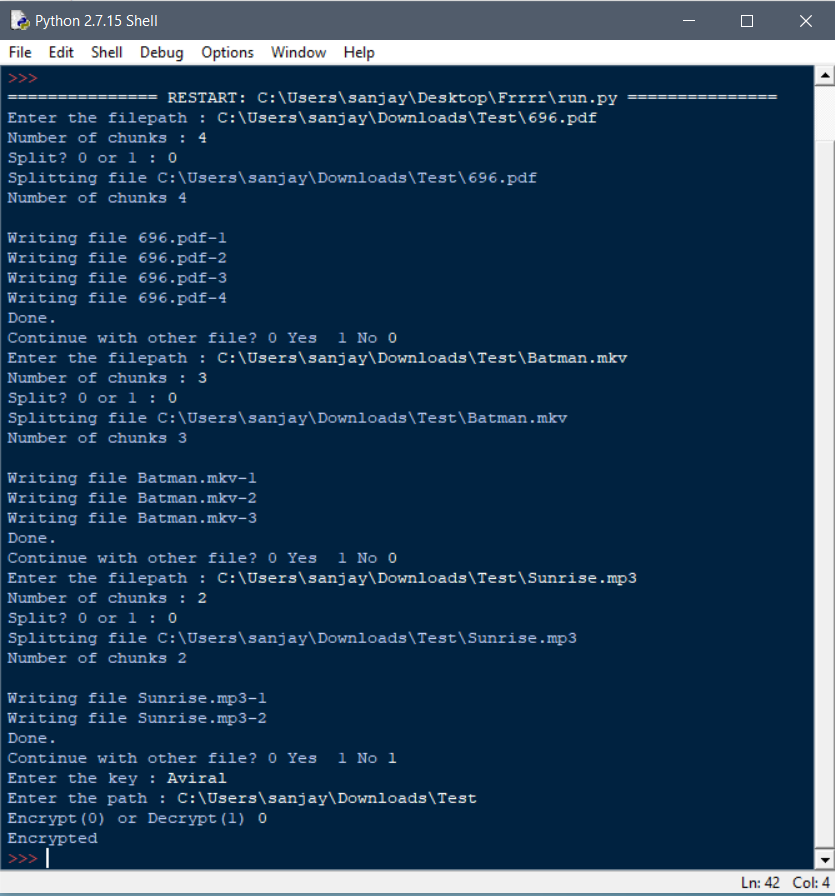
.

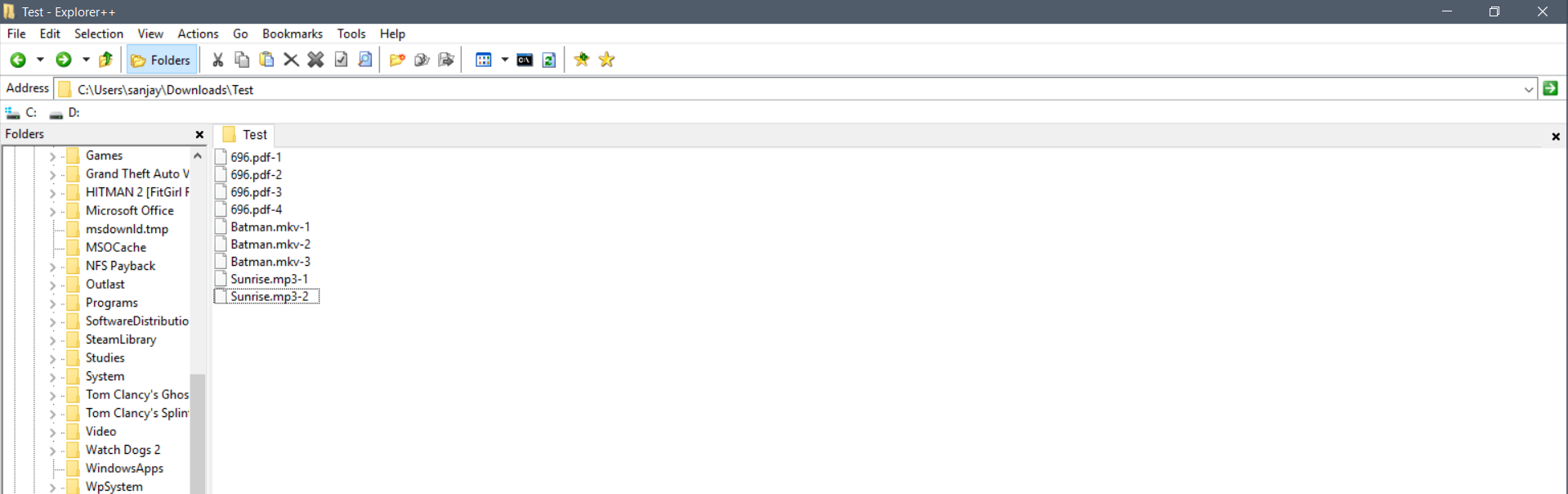
**APPENDIX 2**

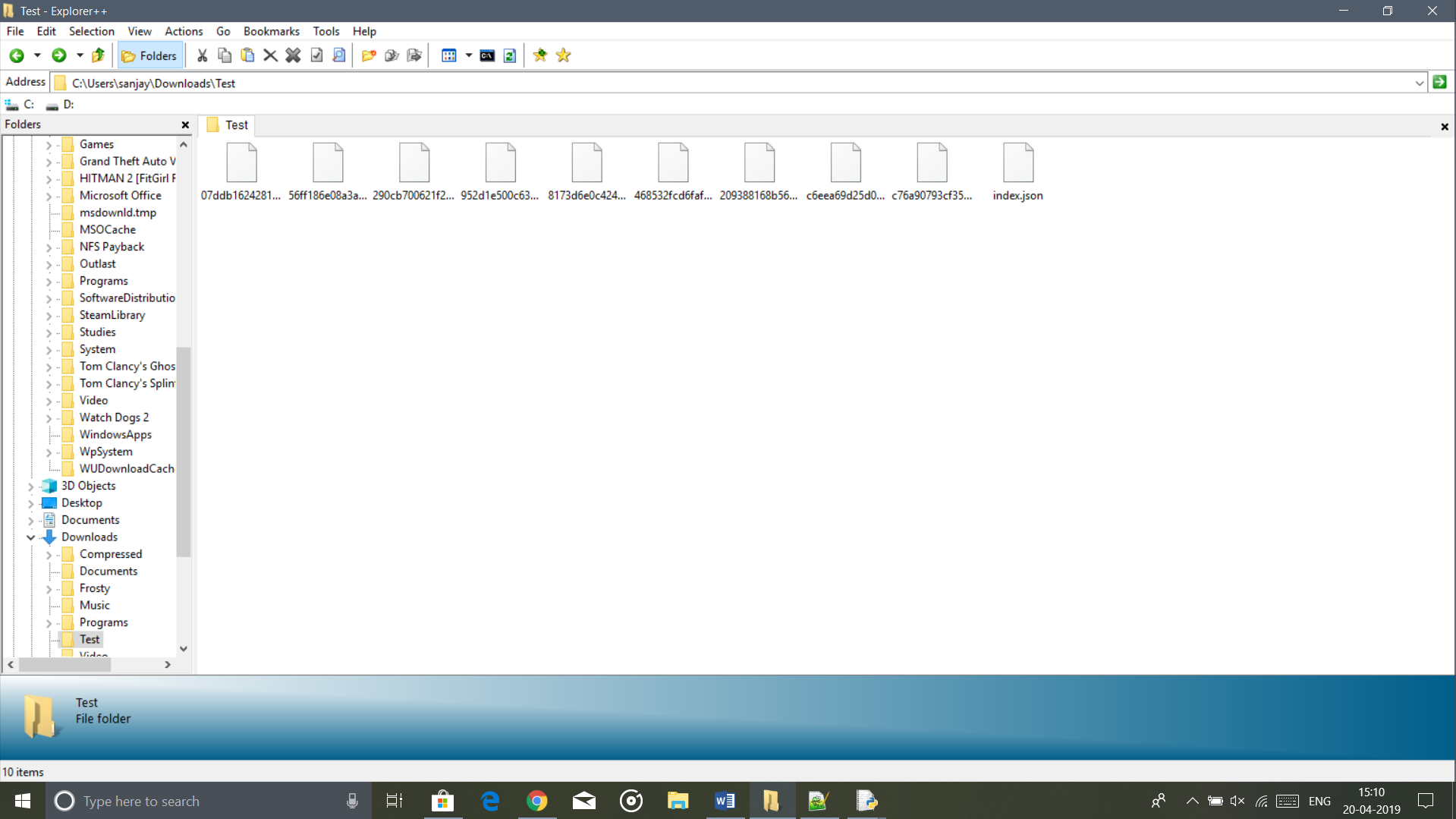
**PROCESS SCREENSHOTS**

****

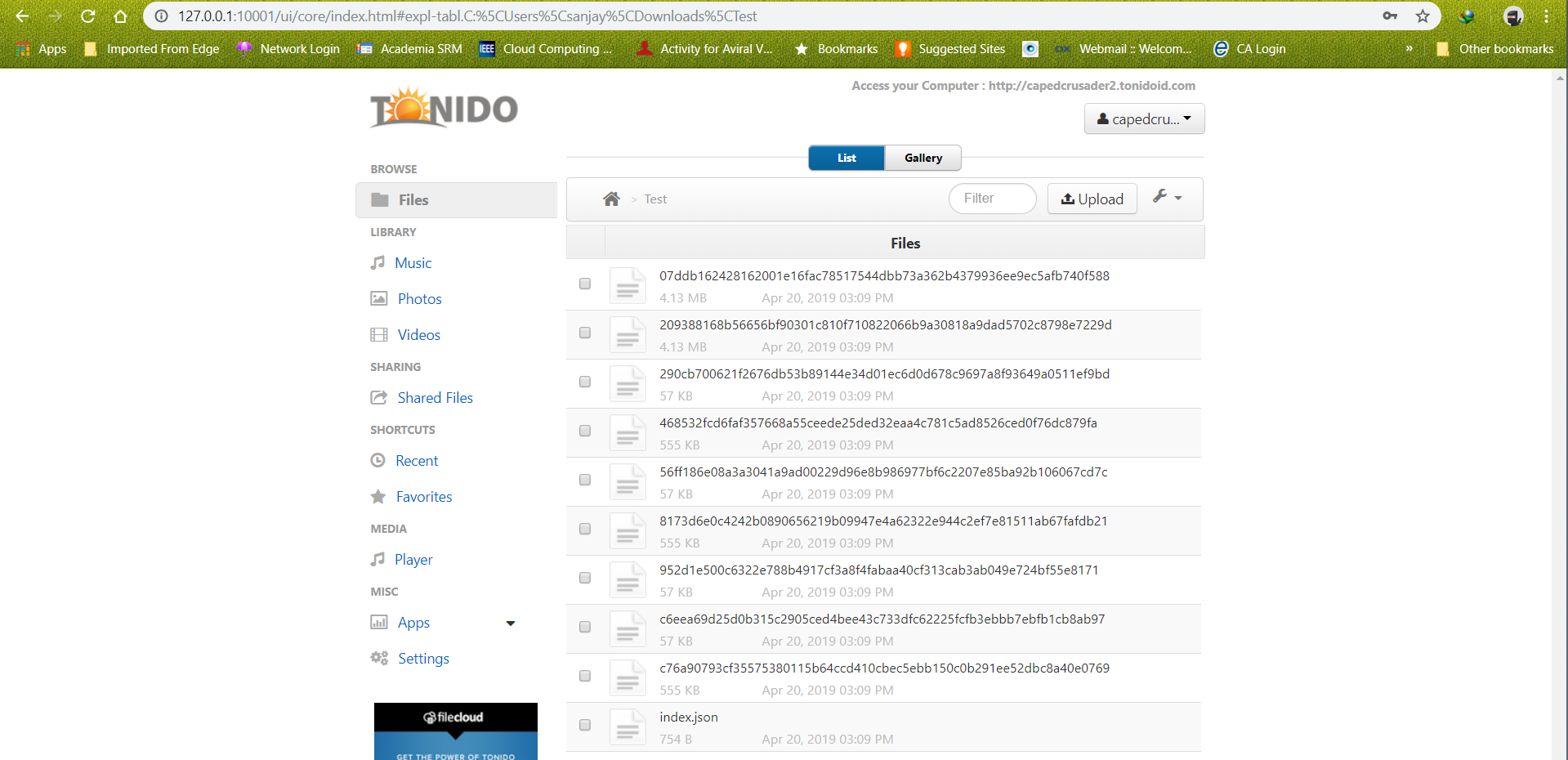
**Fig 5.1 Original Directory**

**Fig 5.2 Server-Side Program**

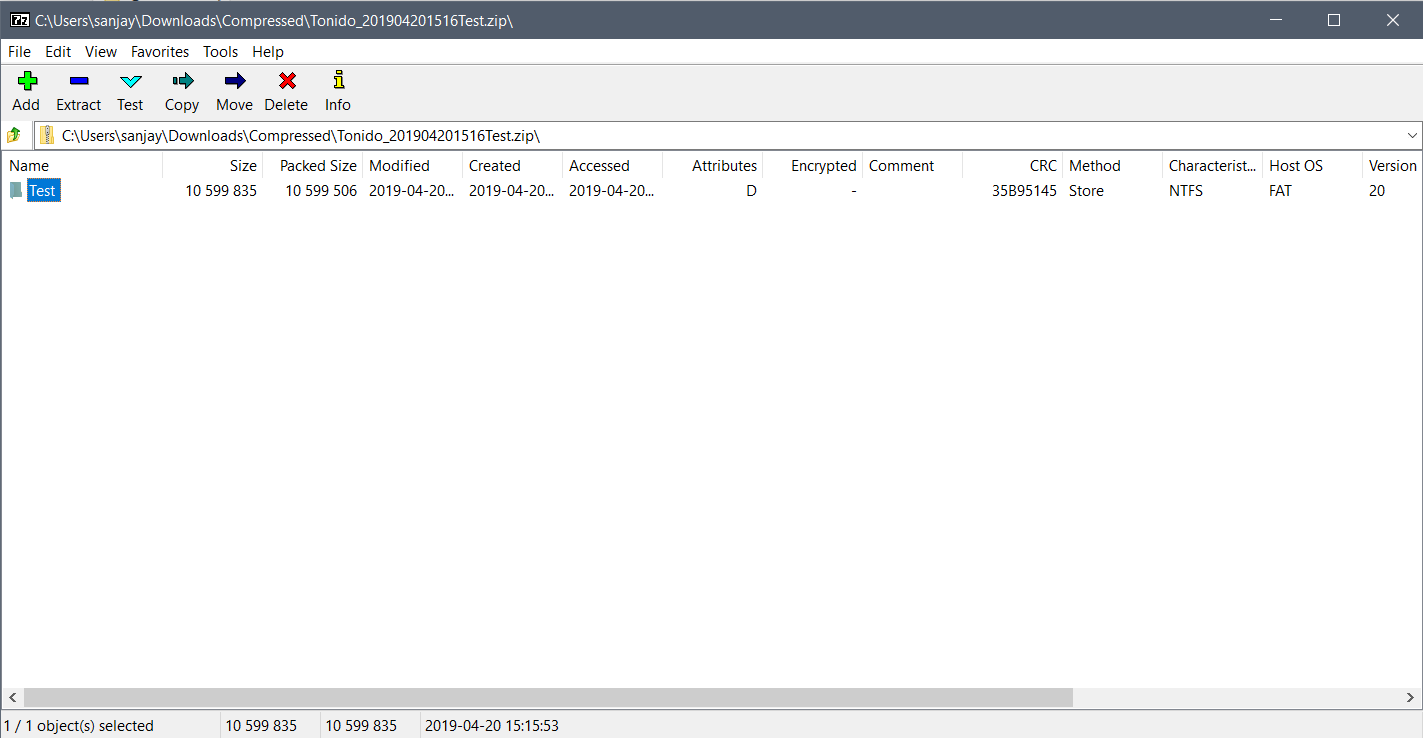
**Fig 5.3** Partitioned Files



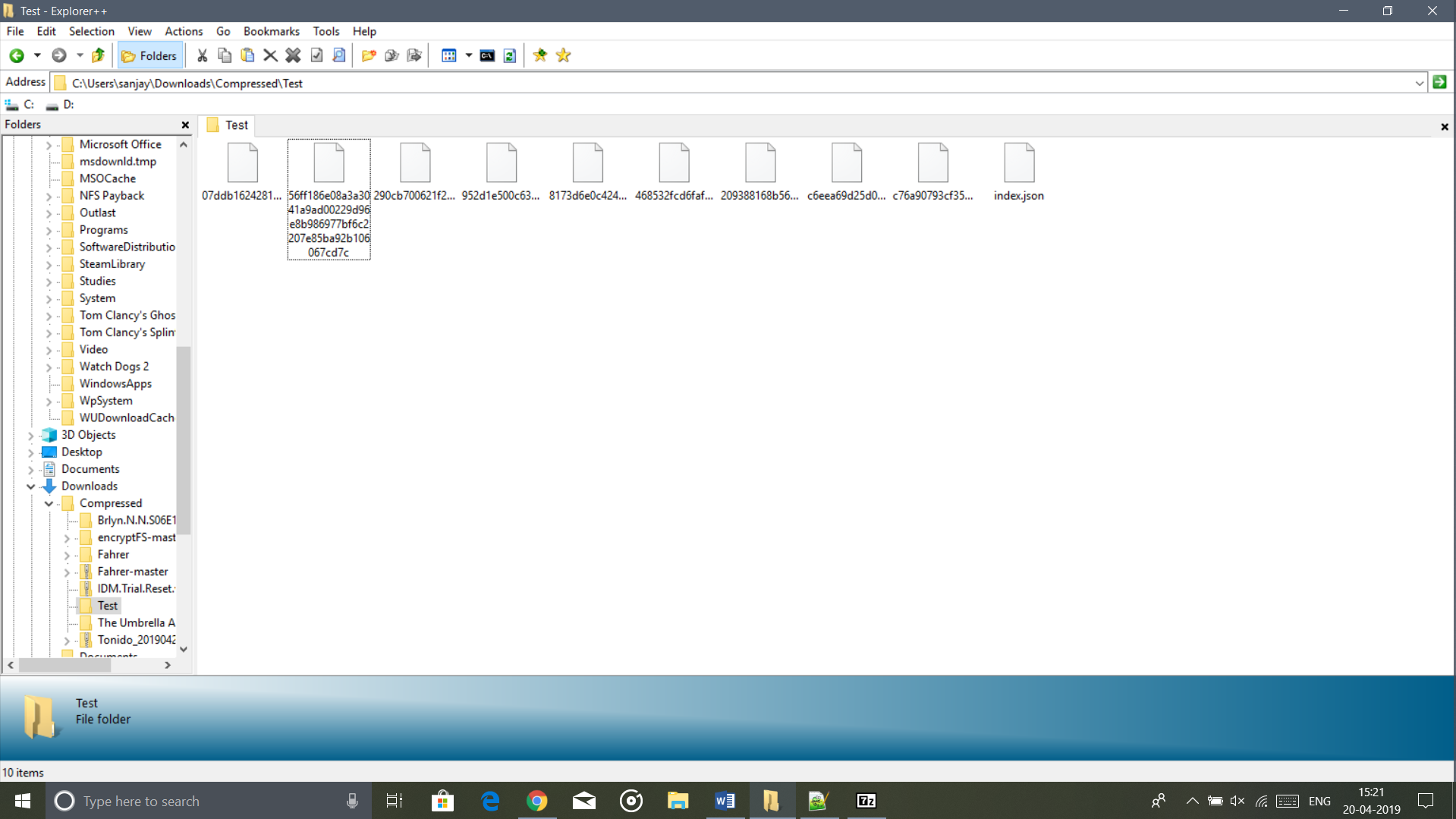
**Fig 5.4 Encrypted Directory**



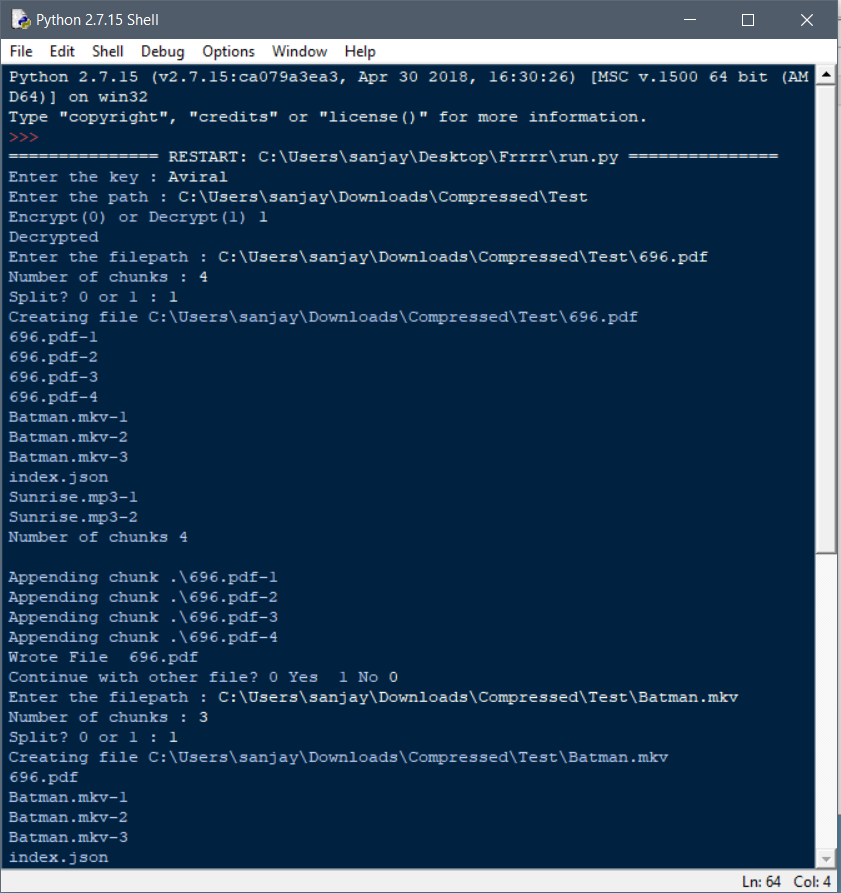
**Fig 5.5 Uploaded Directory on Tonido Server**



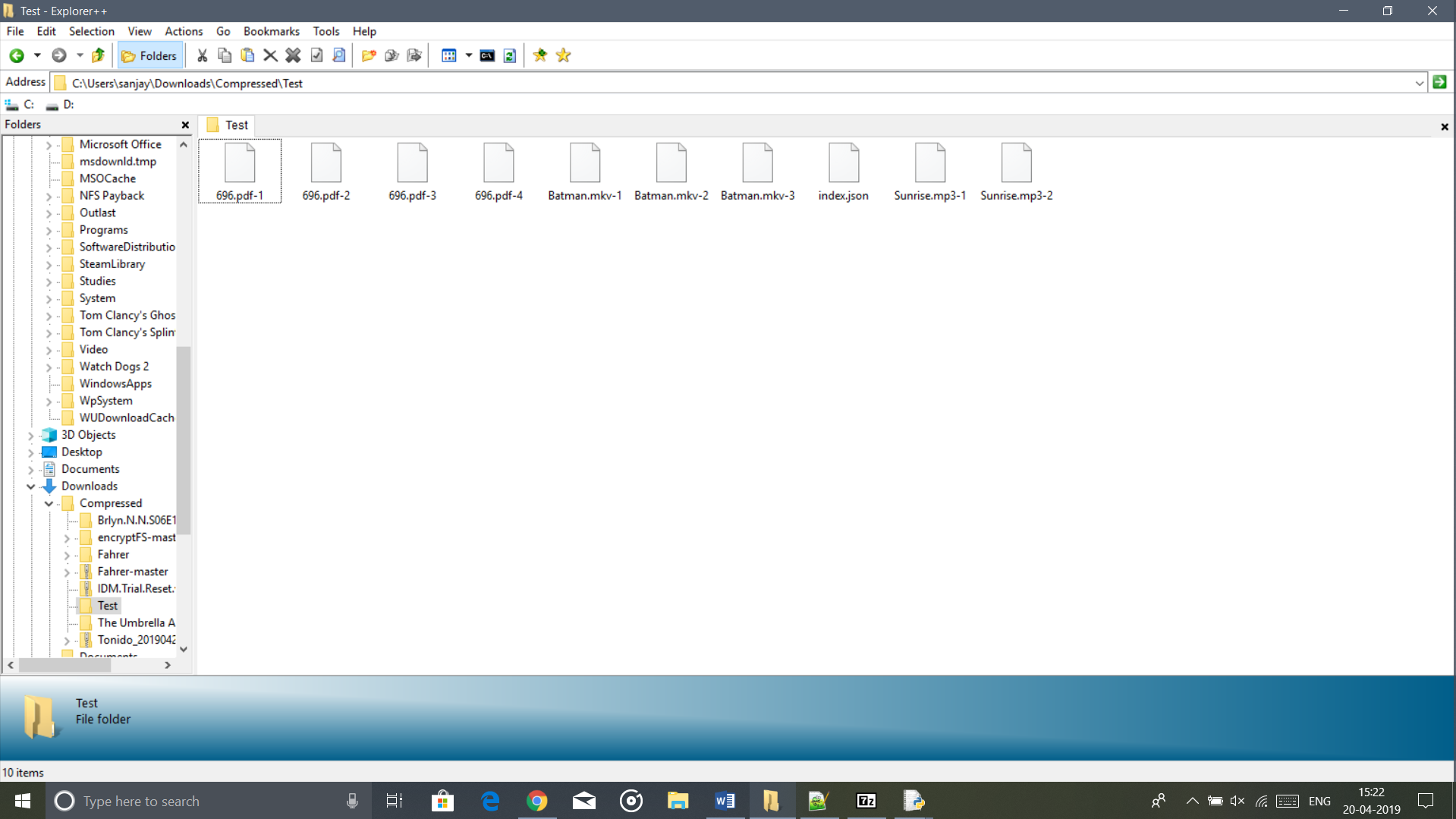
**Fig 5.6 Downloaded Files Client Side (Zipped Download)**



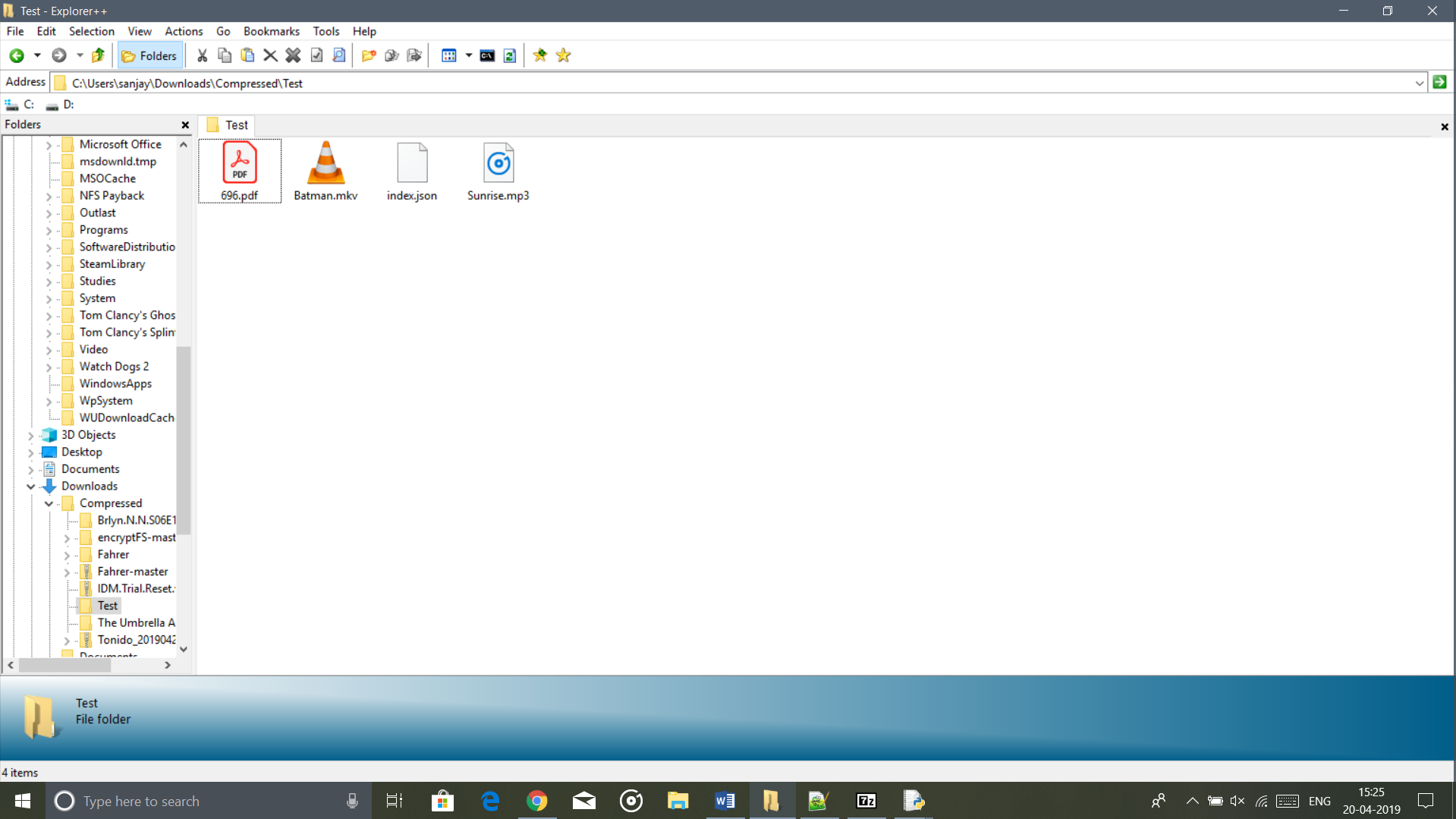
**Fig 5.7 Downloaded Files Client Side (Extracted)**



**Fig 5.8 Client-Side Program**



**Fig 5.9 Decrypted Directory**



**Fig 5.10 Merged Files**

**CONTRIBUTION SUMMARY**

The table below summarises the contribution of each student in the formation of this Project.

|  |  |
| --- | --- |
| **Debanjan Deb**  **RA1511003010534** | 1. Diffie Hellman Key Exchange Module. 2. UI Design of each Module. |
| **Aviral Verma**  **RA1511003010688** | 1. Directory Encryption Module. 2. File Partitioning Module. |