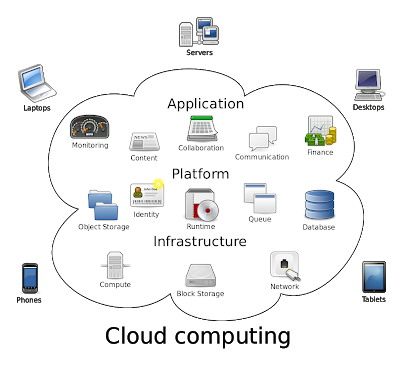
**CHAPTER 1**

**INTRODUCTION**

* 1. **PROJECT OVERVIEW**

Cloud computing is an Information Technology (IT) paradigm that enables ubiquitous access to shared pools of configurable system resources and higher level services that can be rapidly provisioned with minimal management effort, often over the Internet. Cloud computing relies on sharing on resources to achieve coherence and economic of scale, similar to a utility over a network. Cloud computing, or in simpler short hand just “The cloud”, also focuses maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per demand. This approach should maximize the use of computing power thus reducing environmental damage as well since power less, air conditioning, rack space, etc., are required for variety of functions.

Third party clouds enable organizations to focus on their core businesses instead of expending resources on computer infrastructure and maintenance. Advocates note that cloud computing allows companies to avoid or minimize up-front IT infrastructure costs. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance and that it enables IT teams to more rapidly adjust resources to meet fluctuating and unpredictable demand. Cloud providers typically use a “pay-as-you-go” model, which can lead to unexpected operating expenses if administrators are not familiarized with cloud pricing models.



The infrastructure cloud (IaaS) service model offers improved resource flexibility where tenants rent computing resources to operate complex systems. Many organizations operating on sensitive data avoid migrating operations to IaaS platforms due to security concerns. This project describes a framework for data and operation security in IaaS, consisting of protocols for a trusted launch of virtual machines and domain-based storage protection. Thus once the user is authenticated they will be launched in virtual machines where they initiate the upload process into the cloud. Also in our proposed system encryption keys are maintained outside of the IaaS domain. RSA algorithm proposed for key encryption. In the proposed system, implementing virtual machines and key management secure data access has been provided.

* 1. **AIM:**

The main aim of the project is to provide file security in the public cloud environment and to provide a mechanism to prevent the common web vulnerabilities attacks. It not only provide file storage but also block the intruder to avoid future attacks possibilities by providing security using encryption.

**PROBLEM DEFINITION**

The idea is to demonstrate file security in the public cloud environment and to provide security for file storage by using Camellia algorithm and RSA.

* 1. **OBJECTIVE:**

The objective of the project is to create a virtual environment where infrastructure can be rendered as a service to the end users. By setting up virtual machines, file access has been made secure and unauthorized access has been prohibited. Also the project is to provide secure encryption and cloud storage environment. For each user, private key has been rendered from the admin side for accessing the downloaded file. To improve security dual encryption algorithms are used, for key encryption RSA is used & for data encryption camellia algorithm is used.

**CHAPTER 2**

**SYSTEM ANALYSIS**

System analysis is a problem solving technique that decomposes a system in to its component pieces for the purpose of studying how well those component parts work and interact to accomplish their purposes. System analysis is the process of studying a procedure or business in order to identify its goals and purposes and create systems and procedures that will achieve them in an efficient way. Analysis and synthesis, as a scientific methods, always go hand in hand; they complement one another. Every synthesis is built up on the results of a preceding analysis, and every analysis requires a subsequent synthesis in order to verify and correct its results.

* 1. **EXISTING SYSTEM**

The existing system support data encryption at rest is offered by several cloud providers where functionality and migration capabilities of such solutions are severely restricted. In most cases cloud providers doesn’t maintain and manage the keys necessary for encryption and decryption of data. This further convolutes the already complex data migration procedure between different cloud providers, disadvantage tenants through a new variation of vendor lock-in. Tenants can choose to encrypt data on the operating system (OS) level within their VM environments and manage the encryption keys themselves.

* + 1. **Drawbacks of Existing System**

1. The underlying compute host will still have access encryption keys whenever the VM performs cryptographic operations.
2. This shifts towards the tenant is the burden of maintaining the encryption software in all their VM instances and increases the attack surface.
3. This requires injecting, migrating and later securely withdrawing encryption keys to each of the VM instances with access to the encrypted data, increasing the probability than an attacker eventually obtains the keys.
   1. **NEED FOR PROPOSED SYSTEM**

This proposed system also focus on broken authentication and session management. Authentication and session management includes all aspects of handling user authentication and managing active sessions. Authentication is a critical aspect of this process, but even solid authentication mechanisms can be undetermined by flawed credential management functions, including password change, forgot my password, remember my password, account update, and other related functions.

* 1. **FEATURES OF PROPOSED SYSTEM**

In this project, DBSP (domain-based storage protection), a virtual disk encryption mechanism where encryption of data is done directly on the compute host, while the key material necessary for re-generating encryption keys is stored in the volume metadata. This approach allows easy migration of encrypted data volumes and withdraws the control of the cloud provider over disk encryption keys. Also to access the downloaded file in the virtual machines admin has to provide the secure private key to the user, if the user finds to be genuine and authorized. The user secret key is encrypted using RSA algorithm.

* + 1. **Feasibility Study**

The feasibility of the project is analyzed in this phase and business proposal is but forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

1. Economical Feasibility

2. Technical Feasibility

3. Social Feasibility

**1. Economical Feasibility**

This study is carried out to check the economic impact that the system will have on the organization. The amount of found can company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the development system as well within the budget and this was achieved because most of the technologies are freely available. Only the customized products had to be purchased.

**2. Technical Feasibility**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**3. Social Feasibility**

The aspect of study is to check the level of acceptance of the system by user. This includes the process of training the user to the system efficiently. The user must not feel the threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods are employed to educate the user about the system and to make him familiar with nit. His level of confidence must be raised but also able to make some constructive criticism, which is welcomed, as the final user of the system.

* 1. **REQUIREMENT ANALYSIS**

**2.4.1 HARDWARE REQUIREMENTS**

Processor : Pentium –I Core

RAM : 2 GB

Hard Disk : 1.28GB

* + 1. **SOFTWARE REQUIREMENTS**

Operating System : Windows 7/8/10

Front End : Java

Script : JavaScript

Front End Tool : Net Beans 8.0.1

Application Server : Tomcat 5.0/6.X

Back End : MySQL

Back End Tool : SQLYog

* + - 1. Tools:

**NET BEANS 8.0.1**

Net Beans IDE is the official IDE for Java 8. With its editors, code analysers, and converters, you can quickly and smoothly upgrade your applications to use new Java 8 language constructs, such as lambdas, functional operations, and method references. Batch analyser and converters are provided to search through multiple applications at the same time, matching patterns for conversion to new Java 8 language constructs. With its constantly improving Java Editor, many rich features and an extensive range of tools, templates and samples, Net Beans IDE sets the standard for developing with cutting edge technologies out of the box.

**APACHE TOMCAT SERVER 5.0/6.X**

Tomcat is a Java servlet container and web server from the Jakarta project of the Apache Software Foundation (http://jakarta.apache.org). A web server is, of course, the program that dishes out web pages in response to requests from a user sitting at a web browser. But web servers aren’t limited to serving up static HTML pages they can also run programs in response to user requests and return the dynamic results to the user’s browser. Tomcat can be used stand-alone, but it is often used “behind” traditional web servers such as Apache http, with the traditional server serving static pages and Tomcat serving dynamic servlet and JSP requests.

**SQLYOG**

SQLYOG is a GUI tool for the RDBMS MySQL. It is developed by Webyog, Inc. based out of Bangalore, India and Santa Clara, California. SQLYOG is being used by more than 30,000 customers worldwide and has been downloaded more than 2,000,000 times.

**JAVA JDK.1.7**

Computers connected to the net are from many different manufacturers, running on different operating systems and they differ in architecture, computing power and capacity. By considering this point SUN Microsystems Corporation felt the need for a new programming language suitable for this heterogeneous environment and java was the solution. This breaks barriers between different computers, chips and operating systems.

* + - 1. Programing Languages:

**JSP**

The most significant of the many good reasons for this is that it is amazingly easy to develop sophisticated Web sites with JSPs. Anyone who can write HTML can quickly create rich, dynamic, and responsive Web sites that enable users to get the most out of their online time. Through a mechanism called JavaBeans, JSPs have made it possible for large teams or individuals working on complex projects to divide the work in such a way as to make each piece simple and manageable, without sacrificing any power. JSPs also provide a great deal of flexibility when generating HTML, through the ability to create HTML-like custom tags.

**JavaScript:**

JavaScript often abbreviated as JS, is a [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) [programming language](https://en.wikipedia.org/wiki/Programming_language). It is a language which is also characterized as [dynamic](https://en.wikipedia.org/wiki/Dynamic_programming_language), [weakly typed](https://en.wikipedia.org/wiki/Weak_typing), [prototype-based](https://en.wikipedia.org/wiki/Prototype-based_programming) and [multi-paradigm](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language). Alongside [HTML](https://en.wikipedia.org/wiki/HTML) and [CSS](https://en.wikipedia.org/wiki/CSS), JavaScript is one of the three core technologies of [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web) [content engineering](https://en.wikipedia.org/wiki/Content_engineering). It is used to make dynamic Webpages interactive and provide online programs, including video games. The majority of [websites](https://en.wikipedia.org/wiki/Website) employ it, and all modern [web browsers](https://en.wikipedia.org/wiki/Web_browser) support it without the need for [plug-ins](https://en.wikipedia.org/wiki/Browser_extension) by means of a built-in [JavaScript engine](https://en.wikipedia.org/wiki/JavaScript_engine). Each of the many JavaScript engines represent a different implementation of JavaScript, all based on the [ECMAScript](https://en.wikipedia.org/wiki/ECMAScript) specification, with some engines not supporting the spec fully, and with many engines supporting additional features beyond ECMA.

**HTML:**

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

**CSS:**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging Webpages, user interfaces for web applications, and user interfaces for many mobile applications.

* + - 1. Database:

**MySQL**

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

* 1. **ASSUMPTIONS AND DEPENDENCIES:**
* To use this system the user must connected to internet.
* Assumptions is made in such a way that the file server is on working/running mode.
* Each User can store files up to 3 GB in cloud storage for free and after that need to pay for additional charges for more consumption.

**CHAPTER 3**

**SYSTEM DESIGN**

**3.1 UML DIAGRAMS**

The Unified Modelling Language is a standard language for specifying, visualizing, constructing, and documenting the artifacts of the software systems, as well as for business modelling and other non-software systems. The UML represents a collection of the best engineering practices that have proven successful in modelling of large and complex systems.

**3.1.1 Use Case Diagram**

A use case is a set of scenarios that describing an interaction between a user and system. A use case diagram displays the relationship among the actors and use cases. The two main components of a use case diagram are use cases and actors. Here figure 3.1 describes the use case diagram for providing user file security which deals with actor such as admin and users.

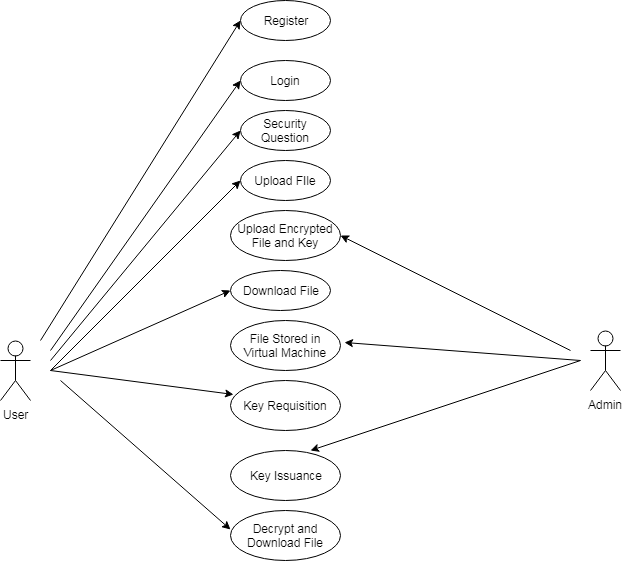
****

Figure 3.1 Use case diagram

**3.1.2 Class Diagram**

Class diagrams are the most common diagrams used in UML. Class diagram consists of classes, interfaces, associations and collaborations. Class diagrams are basically represent the object oriented view of the system which is static in nature. Active class is used in the class diagram to represent the concurrency of the system. This is the most widely used diagram at the time of system construction. In figure 3.2 describes the class diagram for register class, login class, upload file class, download file class and encryption class.

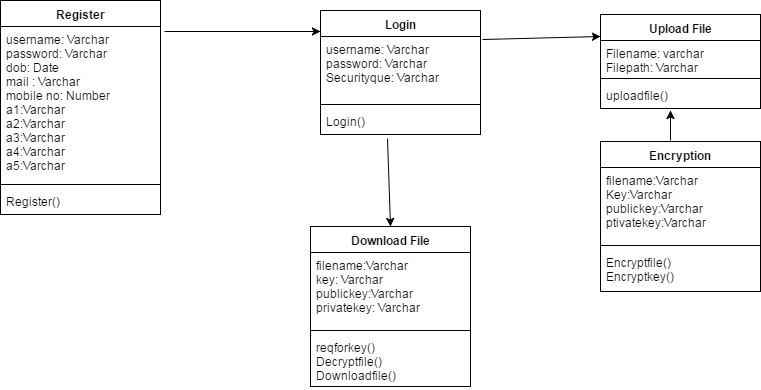
****

Figure 3.2 Class diagram

**3.1.3 Sequence Diagram**

A sequence diagram is an interaction diagram. Sequence diagram is used to visualize the sequence of calls in a system to perform a specific functionality. In figure 3.3 represents the sequence of steps happened between user, admin, database (DB) and cloud storage.

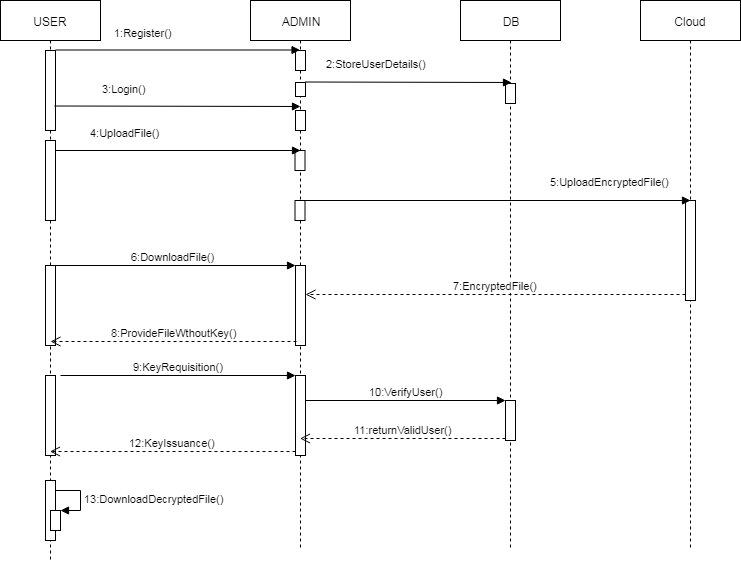
****

Figure 3.3 Sequence diagram

**3.1.4 Activity Diagram**

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. This diagram is basically a flow chart to represent the flow from one activity to another activity. The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behavior of the system. Activity is the particular operation of the system. Figure 3.4 denotes the activities that are performed by the user for providing file security in cloud infrastructure.

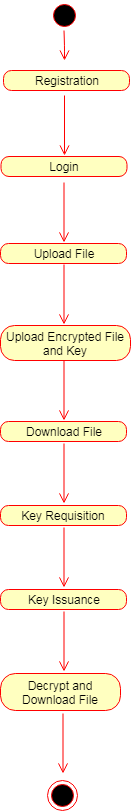
****

Figure 3.4 Activity diagram

**3.1.5 ER Diagram**

An entity–relationship model (ER model for short) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types. An ER model for the project is described in this figure 3.5 which consist of all entities with their relationships.

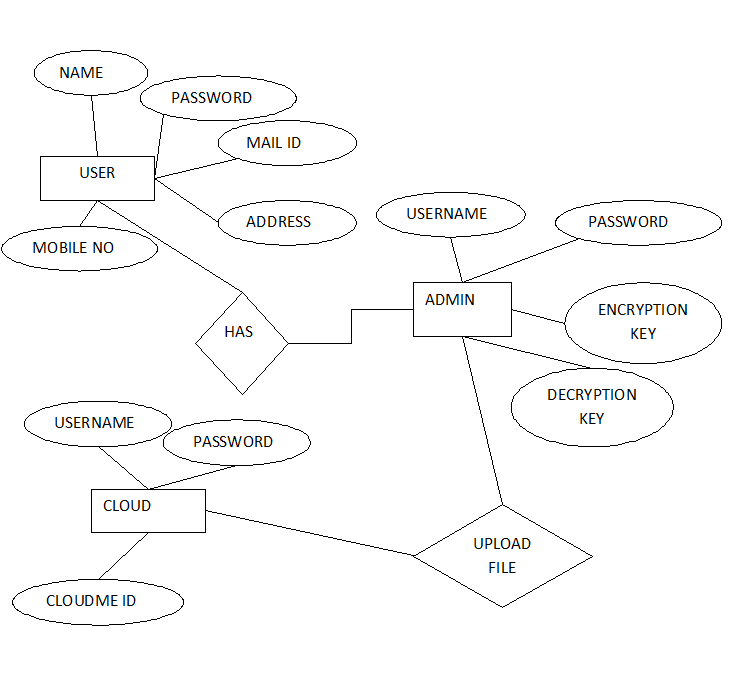


Figure 3.5 ER diagram

* + 1. **System Architecture Diagram**

This diagram represents the overall structure of the system. It also shows the behavior of the system in an abstracted way. The System architecture diagram for file security in cloud infrastructure which clearly describes the process of how the file is storing securely and retrieving back in cloud infrastructure is shown in figure 3.6. User should register for file upload in cloud and then he should login by two way verification. Then he will upload the file, it will be encrypted and the file will be uploaded in cloud infrastructure by admin. When user needs to download the file, the file will be downloaded and stored in virtual machine and that file can’t be visible for user. To access that file user need to get the private key from the admin by sending request to admin and admin will perform identity management. Then the response is sent to the user by providing private key. With the help of private key we can download the file from the virtual machine by entering the virtual key and the file will be stored in local repository.

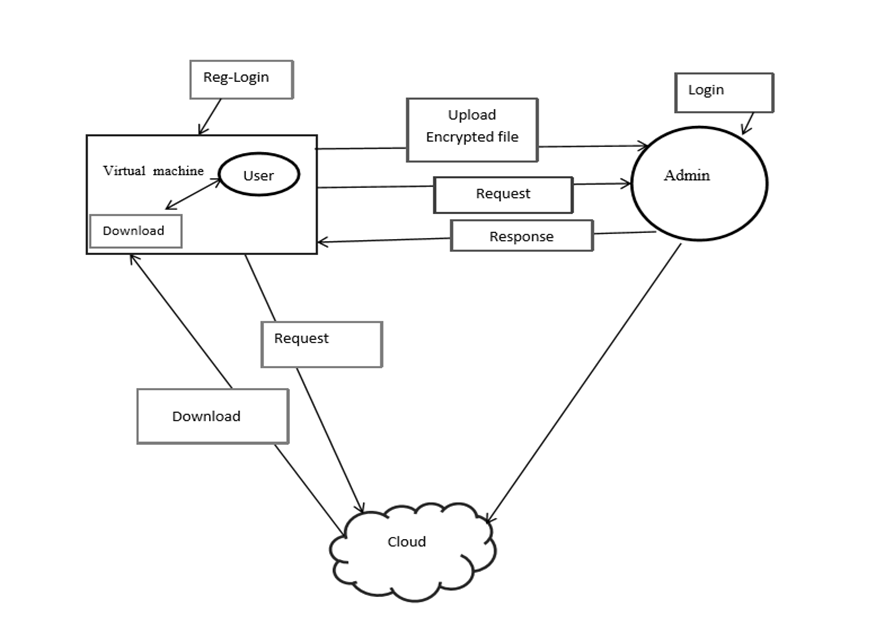


Figure 3.6 System architecture diagram

* 1. **DATABASE DESIGN:**

**3.2.1 Registration Table**

Here the Registration table 3.1 is consists of Field name, Data type and description for each field used in registration form. User name denotes the name of the user, password for user login should be entered for User file upload, DOB denotes Date of Birth, mail and mobile are to be filled by user. A1, A2, A3, A4, A5 are security question that are used for 2 step verification for users. All are in varchar data type.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| User Name | Varchar(22) | User Name |
| Password | Varchar(22) | Password |
| Dob | Varchar(22) | Date Of Birth |
| Mail | Varchar(222) | Mail Id |
| Mobile | Varchar(22) | Mobile Number |
| A1 | Varchar(22) | User first pet name |
| A2 | Varchar(22) | User childhood friend name |
| A3 | Varchar(22) | User childhood super hero |
| A4 | Varchar(22) | User math class teacher |
| A5 | Varchar(22) | User favourite uniform colour |

Table 3.1 Registration Table

* + 1. **File Table**

The File table (3.2) consists of each field that is required for file upload in cloud infrastructure. It consists of File Id denotes the Id of file, cover name, File name denotes the name of the file, username, private key and public key are generated by the admin and sent it to the each user and status describes whether the file is uploaded or downloaded or key request or response.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| Fileid | Int(22) | File Id |
| Covername | Varchar(222) | Cover Name |
| Filename | Varchar(222) | File Name |
| Username | Varchar(22) | User Name |
| Privatekey | Varchar(22) | Private Key |
| Publickey | Varchar(22) | Public Key |
| Status | Varchar(22) | Status |

Table 3.2 File Table

**CHAPTER 4**

**IMPLEMENTATION AND TESTING**

**4.1 MODULE DESCRIPTION**

**4.1.1 Identity Management:**

This module is used to determine and control the illegal users who tries to access the file. Only a legal user can access its own data fields. Any irrelevant entity cannot recognize the exchanged data and communication state even it intercepts the exchanged messages via an open channel. The additional security question is asked for every user who tries to login is shown in Figure 4.1. The user answer for the security question in another redirected page is shown in Figure 4.2.

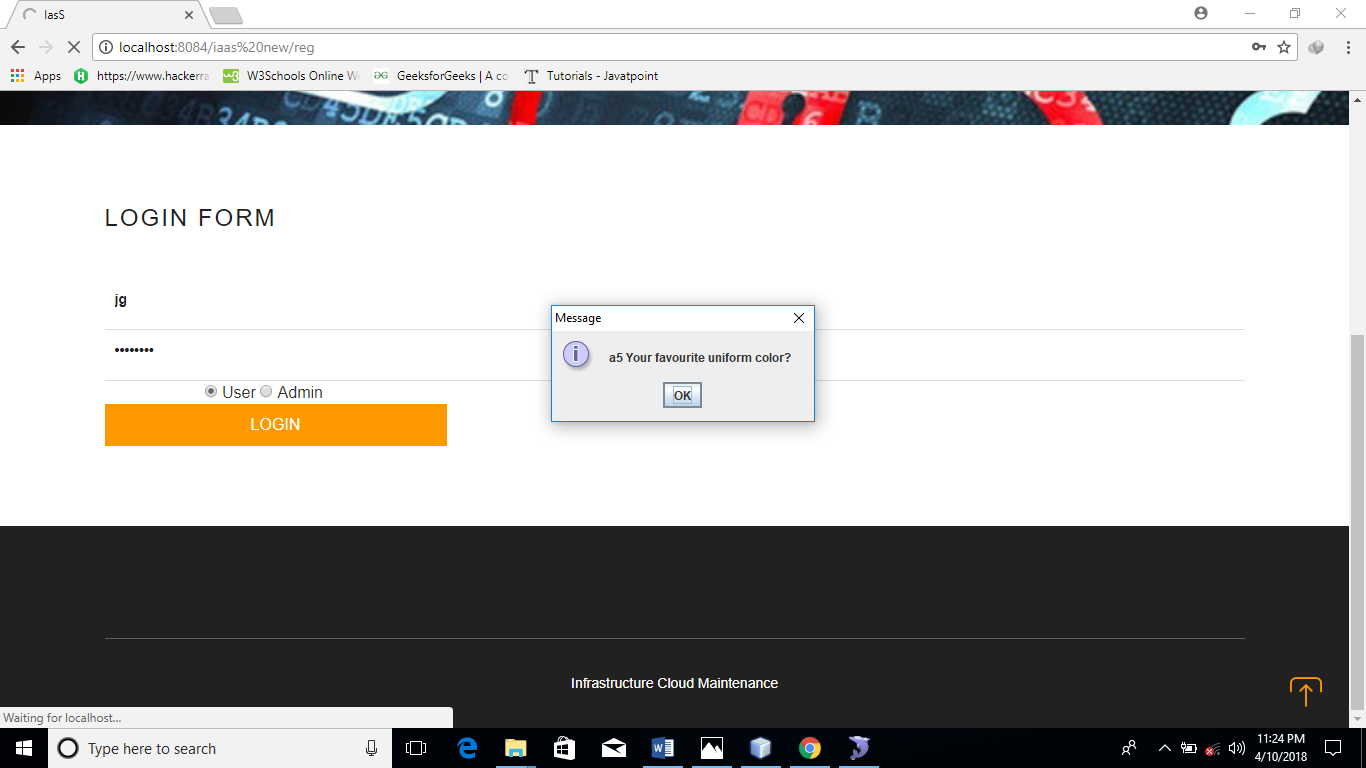


Figure 4.1 Generation of security question

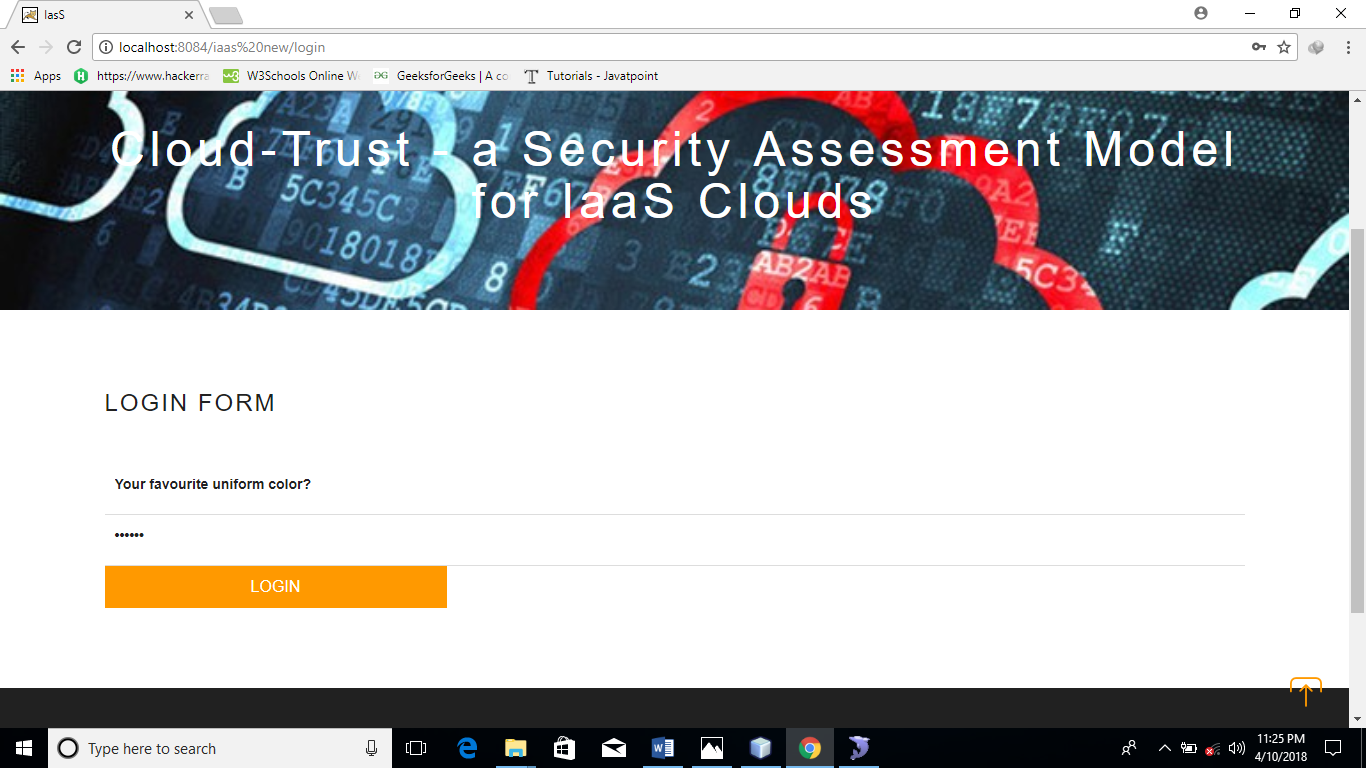
****

Figure 4.2 security question page

**4.1.2 Key Generation and Maintenance Scheme:**

Virtual machine concept is included in the project to maintain all the keys. There are three keys involved in this project, they are randomly generated for encryption purpose. The key must be maintained for 2 purpose. First for privacy as well as for user convenient so the private and public keys are generated for uploading and downloading the file. Second, for the user hard to remember all keys at all time, in this proposed work just the user may remember the private keys alone to check the file status is shown in Figure 4.3. If the user want to download the file they should give request to the Admin before downloading, after getting the key they may download the appropriate file is shown in Figure 4.4.



Figure 4.3 User file list

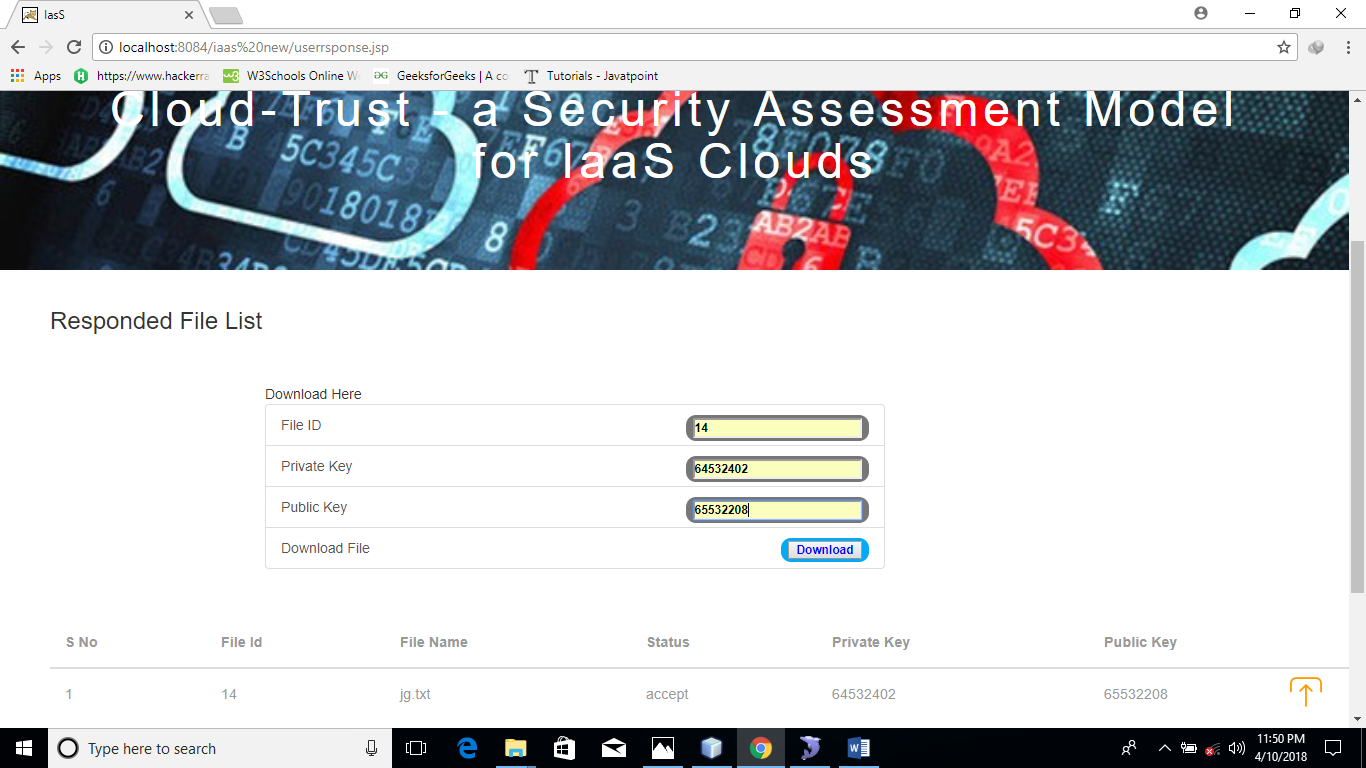


Figure 4.4 User download page

**4.1.3 Virtual Machine:**

This project uses the new technique as a virtual machine for generating the keys while uploading the files (Key Manager) in the cloud, which provide tenants with a proof that the requested VM instances were launched on a host with an expected software stack. The VM performs cryptographic operations and it manages the encryption keys themselves. This virtual machine may act as the inter-mediator that handles all key functions such as private and public keys which are randomly generated is shown in Figure 4.5. Virtual machine under the control of Admin side, but the user can’t access the file or key directly from the cloud or from the VM without getting the key from the Admin. The key is provided to the valid users who gave request for the key by the admin is shown in Figure 4.6. All encryption and decryption is done at the virtual machine.

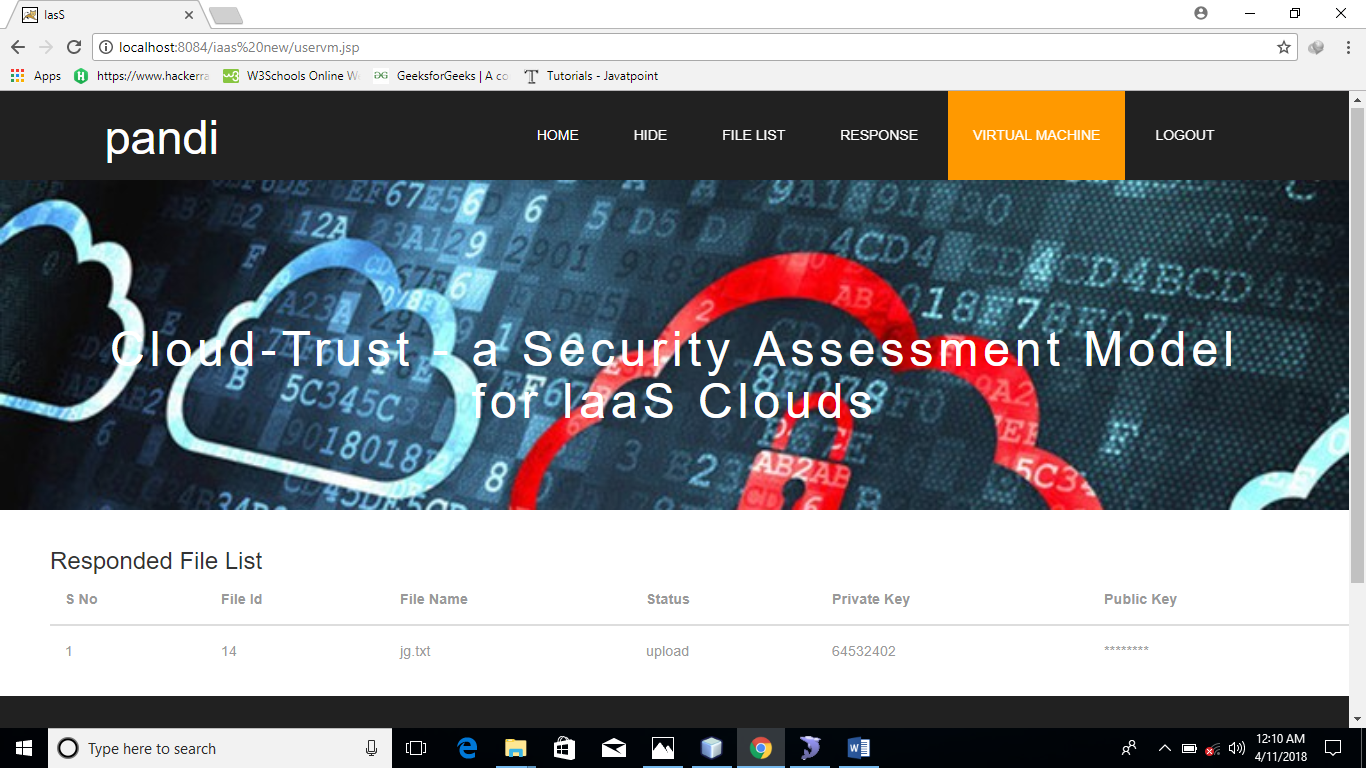


Figure 4.5 public key and private key in VM

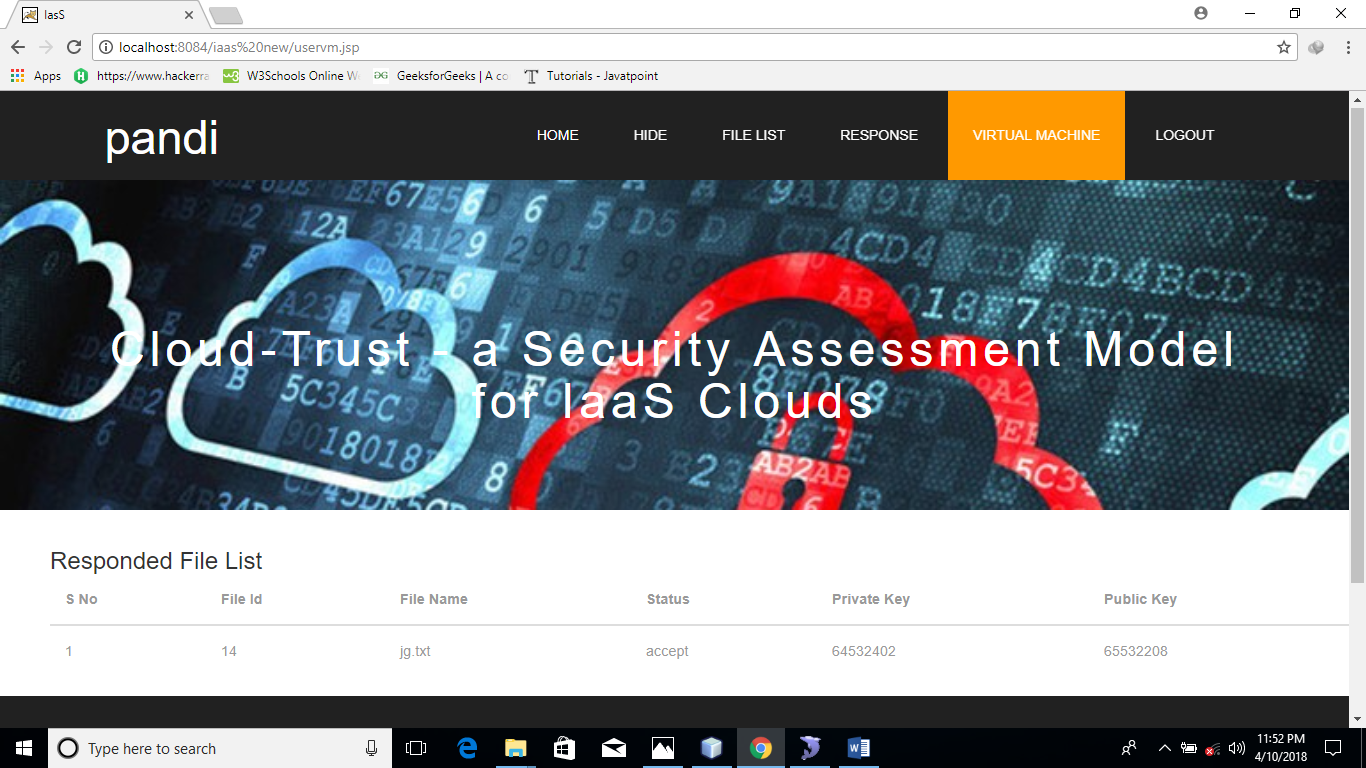


Figure 4.6 public key visible to the user in VM

**4.1.4 Cloud storage**

Cloud storage is a model of data storage where the digital data is stored in logical pools, and it is owned and managed by a hosting company. These cloud storage providers are responsible for keeping the data available and accessible, and the physical environment protected and running. Users can store their data on the cloud and access their data anywhere at any time. This project uses the cloud data storage in which encrypted file is stored and protected from access by other users. This project uses Cloudme cloud for the storage as a public cloud. Files placed in this folder are also accessible via the Cloudme website and mobile apps is shown in the Figure 4.7 as follows

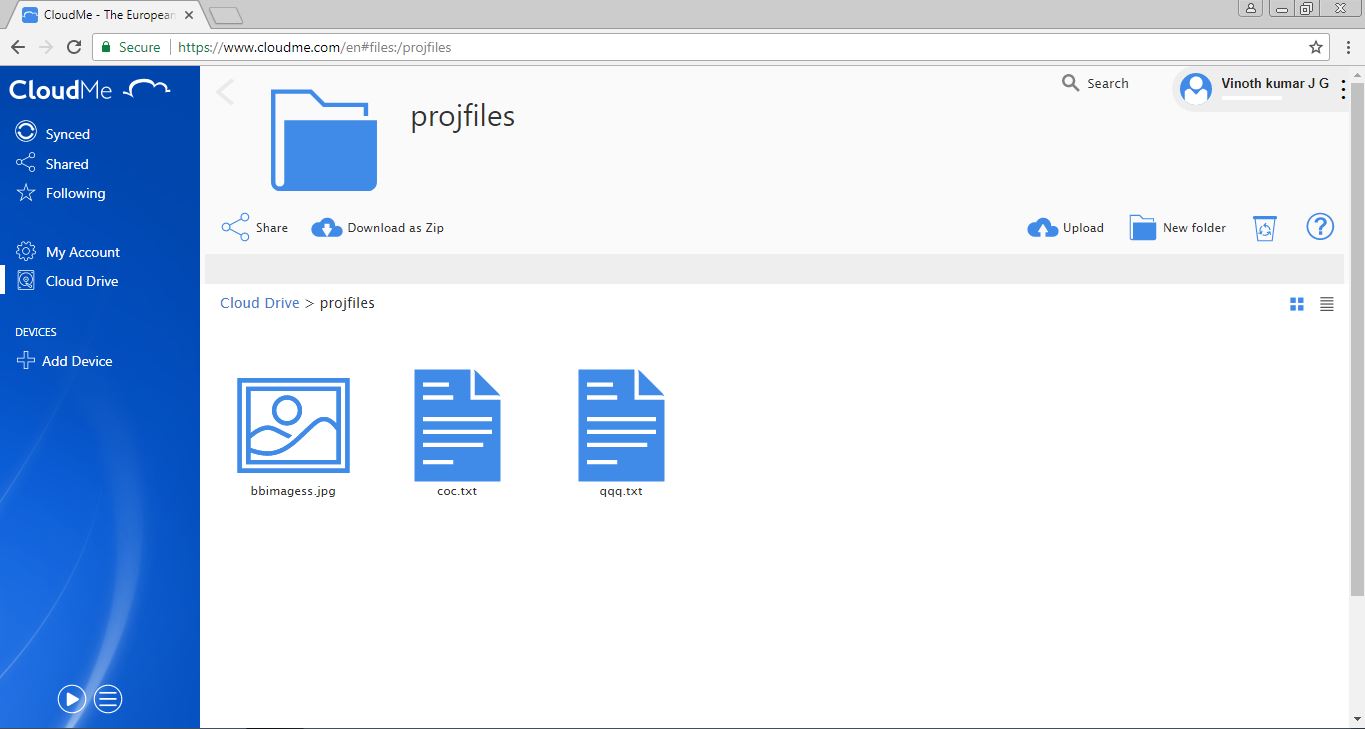
****

Figure 4.7 Cloudme webpage

**4.1.5 Encryption Algorithm**

For more security, this project uses two encryption algorithms namely RSA which is asymmetric technique and camellia which is symmetric technique. In this project, encryption and decryption is done at VM. Initially original data in the file is encrypted using camellia algorithm and the key used for camellia algorithm is encrypted using RSA algorithm. For camellia technique one symmetric key is used for encryption and decryption and for RSA algorithm public key and private key is used for encryption and decryption respectively. To obtain original data RSA followed by camellia decryption technique is used. This reduces the intruders to access the data as well as the key.

**RSA**

RSA is a cryptosystem for public-key encryption, and is widely used for securing sensitive data, particularly when being sent over an insecure network such as the Internet. RSA was first described in 1977 by Ron Rivest, Adi Shamir and Leonard Adleman. Public-key cryptography, also known as asymmetric cryptography, uses two different but mathematically linked keys, one public and one private. The public key can be shared with everyone, whereas the private key must be kept secret. It provides a method of assuring the confidentiality, integrity, authentication and non-reputability of electronic communications and data storage.

The pseudo code for RSA encryption is follows

**Pseudocode**

This is the original algorithm.

1. Generate two large random primes, *p* and *q*, of approximately equal size such that their product n = pq is of the required bit length, e.g. 1024 bits.
2. Compute n = pq and (phi) φ = (p-1)(q-1).
3. Choose an integer *e*, 1 < e < phi, such that gcd(e, phi) = 1.
4. Compute the secret exponent *d*, 1 < d < phi, such that ed ≡ 1 (mod phi).
5. The public key is (n, e) and the private key (d, p, q). Keep all the values d, p, q and phi secret. [We prefer sometimes to write the private key as (n, d) because you need the value of n when using d. Other times we might write the key pair as ((N, e), d).]

Where, n is known as the modulus.

e is known as the public exponent or encryption exponent or just the exponent.

d is known as the secret exponent or decryption exponent.

**Camellia Algorithm**

Camellia was jointly developed by Nippon Telegraph and Telephone Corporation and Mitsubishi Electric Corporation in 2000. Camellia specifies the 128-bit block size and 128-Encryption Standard (AES).Camellia is characterized by its suitability for both software and hardware implementations as well as its high level of security. Moreover, its key setup time is excellent, and its key agility is superior to that of AES. Camellia is considered a modern safe cipher. Even using the smaller key size option (128 bits), it's considered infeasible to break it by brute-force attack on the keys with current technology. There are no known successful attacks that weaken the cipher considerably. Camellia can be divided into "key scheduling part" and "data randomizing part". The encryption steps is shown in Figure 4.8. The decryption steps is shown in Figure 4.9.

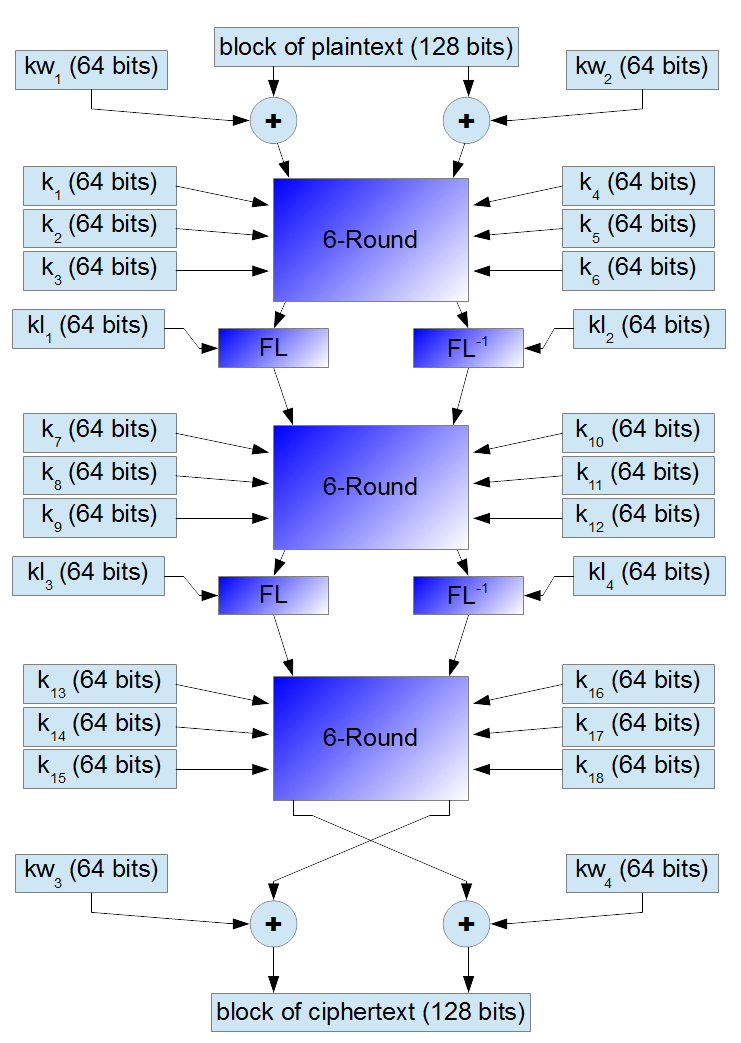


Figure 4.8 Encryption algorithm

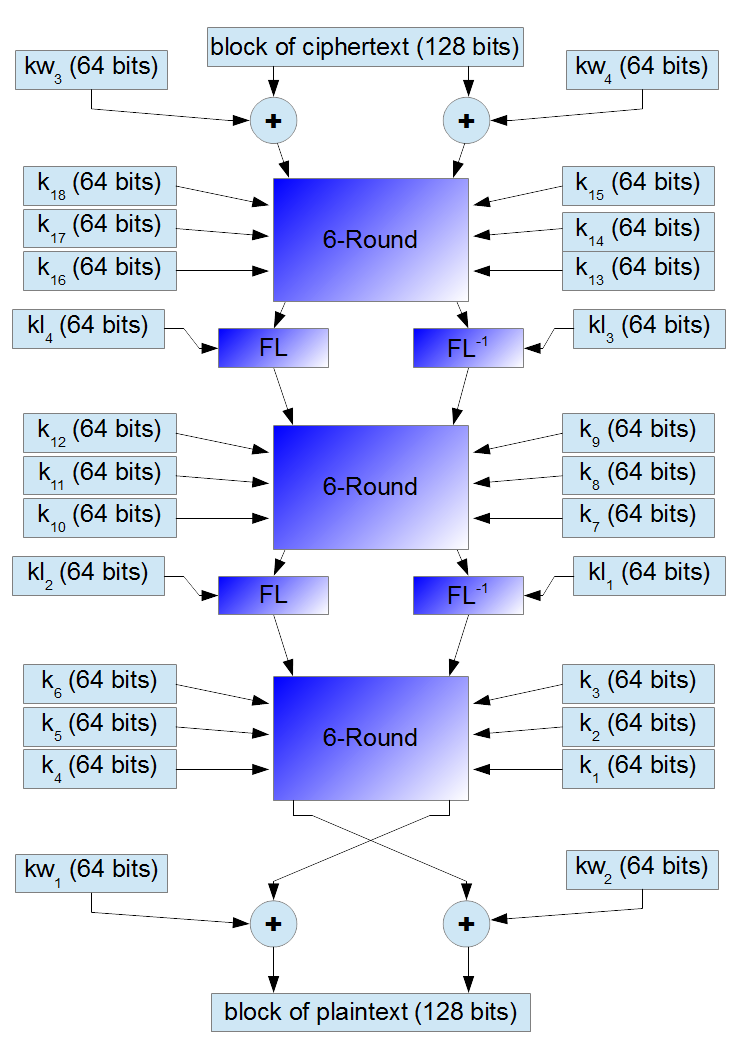
****

Figure 4.9 Decryption Algorithm

**4.2 TESTING**

Since the error in the software can be injured at any stage. So, we have carry out the testing process at different levels during the development. The basic levels of testing are,

* Unit Testing
* Integration Testing
* Validation Testing
* Functional Testing
* Structural Testing

**4.2.1 Unit Testing**

Unit testing was used to test individual units in the system and ensure that they operate correctly. Alternate logic analysis and screen validations were tested in this to ensure optimum efficiency in the system. The procedures and functions used and their association with data were tested.

* + 1. **Integration Testing**

This testing process focuses on identifying the interfaces between components and their functionality. The bottom up approach was adopted during this testing. Low-level modules are integrated and combined as a cluster before testing. This allowed identifying any wrong linkages or parameters passing early in the development process as it just can be passed in the set of data and checked if the result returned is an accepted one.

* + 1. **Validation Testing**

Software testing and validation is achieved through a series of block box tests that demonstrate conformity with requirements. A test procedure defines specific test cases that will be used to demonstrate conformity with requirements. Both, the plan and the procedure are designed to ensure that all functional requirements are achieved, documentation is correct and other requirements are met. After each validation test case has been conducted, one of the two possible conditions exists.

* + 1. **Functional Testing**

Functional testing, also known as block box or closed box testing, is normally applied to HDL (High-Level Data Link) code that operates concurrently and concentrates on checking the interaction between modules, blocks or functional boundaries. The objective here is to ensure that `correct results” are obtained when `good inputs” are applied operates in a predictable manner. Functional testing can therefore be considered as concentrating on checking that the data paths operate correctly. The coverage measurements that fall into this category are toggle, triggering, and signal trace coverage.

* + 1. **Structural Testing**

Structural testing, are known as white box or open box testing, is normally applied to sequential HDL (High-Level Data Link) code and concentrates on checking that all executable statements within each module have been exercised and the corresponding branches and paths through that module have been covered. If there is a section of HDL code that has never been exercised then there is a high possibility that it could contain an error that will remain undetected.

* 1. **DRIVING TEST CASES**

A test case is a set of conditions or variables under which a tester will determine if a requirement upon an application is partially or fully satisfied. The types of testing that are to be carried out on the system is as follows.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case no** | **Description** | **Pre-conditions** | **Pass/Fail** | **Expected results** |
| IAAS\_001 | Validate user Registration | New user only allowed | Pass | Registered Successfully |
| IAAS\_002 | Validate user login | Registered user only allowed | Pass | Login successfully |
| IAAS \_003 | User select secret and hide image | Registered user only allowed | Pass | Image send successfully |
| IAAS\_004 | User select secret text and cover text and hide the text | Registered user only allowed | Pass | Text successfully |
| IAAS\_005 | Admin upload file | User file are uploaded | Pass | Upload successfully |
| IAAS\_006 | User send request | Valid user only allowed | Pass | Send request successfully |
| IAAS\_007 | User send request | Registered user only allowed | Fail | Return to home page |
| IAAS\_008 | Admin accept valid user’s request | Only send response to valid user | Pass | Send key successfully |
| IAAS\_009 | User get the response | Get the response from admin | Pass | Get the key successfully |

4.1 Test case Table

**CHAPTER 5**

**CONCLUSION AND FUTURE WORK**

**5.1 CONCLUSION**

From a tenant point of view, the cloud security model does not yet hold against threat models developed for the traditional model where the hosts are operated and used by the same organization. However, there is a steady progress towards strengthening this security model. This project presented a framework for trusted infrastructure cloud deployment, with two focus points VM deployment on trusted compute hosts and domain-based protection of stored data. Thus in the security analysis, introduced a series of attacks and proved that the protocols hold in the speciﬁed threat model. To obtain further conﬁdence in the semantic security properties of the protocols, this project modelled and veriﬁed them. Finally, this project performance test had shown that the protocols introduce an insigniﬁcant performance overhead. This work has covered only a fraction of the infrastructure cloud attack landscape. Thus the platform software integrity guarantees for tenants and efﬁciently isolate their data using established cryptographic tools. With reasonable engineering effort the framework can be integrated into production environments to strengthen their security properties.

**5.2 FUTURE WORK**

Now this project focuses only on uploading the cloud file and images. Future work will be developed by uploading audio and video files. One of the keys to success for maintenance management is to collaboration with manufacturing. Maintenance needs equipment to be available for preventive maintenance and repair, while production needs equipment available to meet user demands. This project improves collaboration by providing schedules for production review to agree upon the timing and duration of work to be performed. In future to improve this project the breakdown information, tracking VM to encrypted files, optimizing spare parts inventory, request the cloud facilitating planned maintain and improve the cloud storage.

**REFERENCES**

[1] N. Santos, K. P. Gummadi, and R. Rodrigues, “Towards trusted cloud computing,” in Proceedings of the 2009 Conference on Hot Topics in Cloud Computing, HotCloud’09, (Berkeley, CA, USA),USENIX Association, 2009.

[2] J. Schiffman, T. Moyer, H. Vijayakumar, T. Jaeger, and P. McDaniel, “Seeding Clouds With Trust Anchors,” in Proceedings of the 2010 ACM Workshop on Cloud Computing Security, CCSW ’10, (New York, NY, USA), pp. 43–46, ACM, 2010.

[3] N. Paladi, A. Michalas, and C. Gehrmann, “Domain based storage protection with secure access control for the cloud,” in Proceedings of the 2014 International Workshop on Security in Cloud Computing, ASIACCS ’14, (New York, NY, USA), ACM, 2014.

[4] M. Jordon, “Cleaning up dirty disks in the cloud,” Network Security, vol. 2012, no. 10, pp. 12–15, 2012.

[5] Cloud Security Alliance, “The notorious nine cloud computing top threats 2013,” February 2013.

[6] B. Bertholon, S. Varrette, and P. Bouvry, “Certicloud: a novel tpm- based approach to ensure cloud IAAS security,” in Cloud Computing.

[7] M. Aslam, C. Gehrmann, L. Rasmusson, and M. Bjorkman, “Securely launching virtual machines on trustworthy platforms in a public cloud - an enterprise’s perspective.,” in CLOSER, pp. 511– 521, SciTePress, 2012.

[8] D. Song, E. Shi, I. Fischer, and U. Shankar, “Cloud data protection for the masses,” IEEE Computer, vol. 45, no. 1, pp. 39–45, 2012.

**URL**

1. Source :https://www.w3school.com
2. Source :https://www.tutorialspoint.com
3. Source :https://www.javatpoints.com
4. Source :http://www.di-mgt.com.au/rsa\_alg.html
5. Source :http://crypto-it.net/eng/symmetric.camellia.html