IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

FUNCTIONALITY, SECURITY, AND RELIABILITY IN SOLUTION ARCHITECTURE

[1]Roshan Mahant, [2]Sumit Bhatnagar

[1]Senior Technical Consultant, [2]Vice President
[1]Michigan Gaming Control Board,
[1]Launch IT Corp, Dallas, USA

Abstract - The infrastructure cloud (IaaS) service model offers businesses and organizations the ability to access and utilize computing resources on-demand without having to invest in and maintain physical hardware infrastructure. While the current cloud security model may not fully align with traditional threat models developed for on-premises environments, there is ongoing progress toward strengthening the security posture of IaaS platforms. From the perspective of a tenant using cloud services, the current cloud security model is not yet fully equipped to withstand threat models that were traditionally developed for environments where hosts are operated and used by the same organization. However, significant progress is being made towards strengthening the security model of Infrastructure as a Service (IaaS) platforms. This model provides improved resource flexibility and availability, allowing tenants to rent computing resources to deploy and operate complex systems without worrying about the details of hardware maintenance. However, despite the benefits offered by IaaS platforms, many organizations, particularly those dealing with sensitive data, are hesitant to migrate their operations to the cloud due to security concerns. In response to these concerns, this paper proposes a framework for enhancing data and operation security in IaaS environments. The framework consists of secure and flexible resource allocation with the trusted launch of virtual machines and domain-based storage protection. These aim to establish trust by ensuring the integrity of the host platform configuration before launching guest virtual machines and by ensuring the confidentiality of data stored remotely, with encryption keys maintained outside of the IaaS domain. This analysis helps to establish reliability and effectiveness and enhance system security in IaaS environments.

Index Terms: Infrastructure as a Service (IaaS), Cloud computing, Data security, Virtual machines

I INTRODUCTION

The transition from cloud computing being an aspiration aim to widespread implementations in a variety of application domains has occurred. These additional security risks and challenges are brought about by the complex structure of the technology that underpins cloud computing. Over the last several years, there has been a significant amount of focus placed on potential dangers and strategies for mitigating them for the infrastructure as a service (IaaS) paradigm. In addition to providing ideas for best practices, the industry has made investments in improving security solutions. The security precautions implemented by the cloud provider, which are often backed by the conclusions of external auditors, must be completely trusted by end-users. Data encryption is one example of a security feature that may be provided by providers; however, end users have very little to no control over these methods. The need for cloud platforms to include security measures that are both feasible and inexpensive, and that are acceptable for businesses that use cloud infrastructure, is unquestionably there. Verifying the platform integrity of the compute hosts that are offering the virtualized cloud architecture is one method that may be used. Several of the most prominent cloud service providers have shown working uses of this strategy, primarily for the purpose of protecting the cloud infrastructure against advanced persistent attacks and insider threats. When it comes to these implementations, there are two primary areas that need improvement. It is not possible to divulge the specifics of private systems, which prevents rival cloud platforms from implementing and augmenting them. Regarding the second point, to the best of our knowledge, none of the solutions provides cloud tenants with a guarantee of the integrity of the compute hosts that are responsible for supporting their share of the cloud architecture.[1] The safe commencement of virtual machines (VM) in Infrastructure as a Service (IaaS) is facilitated by a set of protocols that we provide. Tenants are provided with proof that the virtual machine instances that were requested were launched on a host that consisted of the software stack that was expected. The encryption of virtual disk volumes is yet another essential security step that is carried out and enforced at the level of the hosting computer. Tenants have the ability to set up data encryption while it is stored on their virtual machine instances, which is supported by a number of cloud providers. Nevertheless, the capabilities of these platforms in terms of migration and functionality are restricted. These encryption keys, which are necessary for the protection of stored data, are often managed and supervised by cloud providers.[2-3] To put renters at a disadvantage, this introduces a new sort of vendor lock-in, which makes the process of transferring data between different cloud providers more complicated. In their virtual machine (VM) settings, tenants have the option to encrypt data at the operating system (OS) level and to manage encryption keys separately. This customization is available to them. However, this method has a number of drawbacks, including the fact that the compute host keeps access to encryption keys while cryptographic operations are being performed, the tenant is responsible for managing encryption software in all virtual machine instances, which increases the vulnerability to attacks, and the process of injecting, migrating, and withdrawing encryption keys to each VM instance that accesses encrypted data increases the likelihood that attackers will be able to receive access to the keys. With the help of this study, DBSP (domain-based storage protection) is presented as a way for encrypting virtual disks. This approach encrypts data on the compute host and keeps the key material that is required to renew encryption keys in the volume metadata. The cloud provider loses control over the encryption keys for disks when using this strategy, which also makes the transmission of encrypted data volumes more reliable.[4]

Additionally, DBSP significantly reduces the probability of encryption keys being exposed and lessens the amount of maintenance that the tenant is required to do. In addition to this, it extends the range of options available for picking the compute host depending on the software stack it uses. We place a strong focus on the Infrastructure-as-a-Service model, which offers tenants a single platform that is powered by compute hosts that are running virtual machine guests and link to each other over a virtual network infrastructure. During the process of migrating a distributed electronic health record (EHR) system to an infrastructure as a service (IaaS) platform, the requirements that required the adoption of system model were discovered.

Contribution:

Propose a Trusted VM Launch (TL) protocol that enables domain managers to initiate VM instances exclusively on hosts with a verified platform configuration. This protocol enhances security by ensuring that VMs are deployed on trusted infrastructure, thereby reducing the risk of compromise. Introduce a protocol for domain-based storage protection, allowing domain managers to store encrypted data volumes partitioned according to administrative domains. This ensures the confidentiality and integrity of data stored in the cloud, mitigating the risk of unauthorized access or tampering. Describe the implementation of the proposed protocols on an open-source cloud platform and present extensive experimental results to validate their practicality and efficiency. These experiments demonstrate the effectiveness of our security mechanisms in real-world cloud environments, highlighting their potential for widespread adoption and deployment.

II RELATED STUDY RESEARCH

The purpose of this work is to provide a security with encryption approach for launching virtual machine instances. This technique gives domain administrators (tenants) the ability to guarantee that virtual machines are only launched on hosts that have an authenticated platform configuration. This protocol enhances the security of VM deployment by reliably verifying the integrity of the underlying infrastructure. Implementation and Experimental Results: The authors explain how they put into practice the suggested standards using a publicly available cloud platform. The study includes thorough experimental findings that confirm the feasibility and efficacy of the security measures, proving their usefulness in real-world cloud settings. **Tahi Ralyas et al. (2022)[5]** focus on the security challenges posed by cloud computing adoption, particularly emphasizing the vulnerabilities in virtualization technologies and the need for robust data protection measures. Their work underscores the importance of assessing the advantages and disadvantages of transitioning to cloud environments, especially concerning security concerns that may deter potential adopters.

Deepika et al. (2022)[6] provide SEFAI, a new framework that aims to improve cloud information security via the use of secure encryption algorithms. Their research addresses emerging security issues associated with accessing information from cloud systems and proposes cryptographic solutions to mitigate unauthorized access and data breaches.

S. Mahipal et al. (2021)[7] delve into the security challenges prevalent in cloud computing, particularly focusing on vulnerabilities at both the hypervisor and virtual machine levels. Their work highlights the need for effective countermeasures against attacks targeting virtualization, emphasizing the importance of maintaining quality of service (QoS) in cloud environments amidst evolving security threats.

Narayanasetti Mehar Dhinakar et al. (2023)[8] explore the security threats inherent in cloud-based systems, with a specific focus on cache-based side-channel attacks targeting virtualization. Their research investigates the vulnerabilities associated with cloud computing architecture and proposes methodologies to mitigate the risks posed by malicious actors targeting virtualized environments. **N.B. Kadu et al. (2022)[9]** conduct a comprehensive survey of major virtualization technology security concerns, emphasizing the importance of addressing open security flaws in virtualized environments. Their study covers various virtualization technologies available on the market and discusses security risks associated with virtual machines, providing insights into improving security measures in virtualized environments. **Saurabh Singhal et al. (2023)[10]** address security concerns in cloud computing by examining the performance of learning-based security applications using datasets. Their research underscores the challenges in evaluating machine learning models' performance across diverse datasets and contexts, emphasizing the need for comprehensive evaluations in cloud security research.

Peter Ntambu et al. (2021)[11] propose a proactive monitoring and detection model for anomalies in virtual machine resource usage in cloud environments. Their research aims to enhance security measures by detecting unauthorized activities and anomalies in virtual machine behavior, leveraging machine learning algorithms for anomaly detection and classification. **Raghunadha Reddi Dornala et al. (2023)[12]** introduce advanced load balancing with high-level security (HLS) for healthcare systems deployed in cloud environments. Their work emphasizes the importance of data protection, access control, and threat detection in healthcare cloud systems, proposing advanced load-balancing strategies tailored to healthcare workloads. **Fazalur Rehman et al. (2023)[13]** propose a hypervisor-based virtual machine introspection (HVMI) tool to detect and mitigate attacks on cloud platforms. Their research focuses on enhancing cloud security by leveraging hypervisor-based techniques for real-time threat detection and forensic analysis, aiming to improve incident response and minimize the impact of security breaches.**C. Kaushik et al. (2022)[14]** develop a Network Intrusion Detection System (NIDS) empowered by machine learning algorithms to enhance cyber-attack detection and prevention. Their research emphasizes the integration of powerful machine learning techniques into intrusion detection systems to improve accuracy and automate the detection process, facilitating effective network security management.

Problem Statement:

The system aims to provide a platform where domain managers can efficiently utilize the allocated allocation of resources to manage VM guests within their domains. Additionally, they have control over the creation, modification, and deletion of domains, as well as managing access permissions of VMs to data stored within the domains.

III SYSTEM MODEL

In the system model described, the infrastructure follows an IaaS paradigm.

Providers: These are entities that offer a set of resources including network, computation, and storage within the IaaS system.

Tenants (Domain Managers): Tenants, also known as domain managers, are entities that utilize resources provided by the providers. Each domain manager is allocated a quota of resources, which they can use to launch and manage Virtual Machine (VM) guests.

VM Guests: These are virtual machines that run on the infrastructure provided by the domain managers. Each domain manager can own multiple VM guests.

Domains: VM guests are organized into domains, which serve as logical groupings for cloud resources. Domains are managed by domain managers and correspond to specific organizational or administrative units.

Let DM={DM1,…,DMn} represent set of all domain managers in the IaaS system.
For each domain manager
VMi={vmi1,…,vmin} denote the set of all VMs owned by that domain manager.
Each domain manager DMi creates, modifies, and destroys domains.
Let Di={Di1,…,Din} represent set of all domains created by DMi.

Resource Management and Operations on VMs

The scheduler in the IaaS system handles activities on VMs including launching, migration, and termination. This scheduler manages resources by distributing them across available computing hosts using a resource management algorithm. Here are the main components:

Compute Hosts (CH): These are physical servers within the IaaS infrastructure, denoted as CH={CH1,…,CHn }.

VM Instances on Compute Hosts: Each VM instance $\boldsymbol{VM_i^l}$ running on a compute host CHi is denoted as $\boldsymbol{VM_i^l}$ CHi. Each VM instance has a unique identifier $\boldsymbol{VM_i^l}$ id $\boldsymbol{VM_i^l}$

Security Profile (SP): The security profile depends on the verified and measured implementation of a trusted computing foundation. The process includes a set of software components that may be quantified during the initialization of a system, and these measurements are saved in secure storage. Each compute host with a same security profile is shown as $\mathrm{CH}_{\mathrm{SP}}$i.

Virtual Network Overlay (SDN): VMs intercommunicate through a virtual network overlay, facilitated by a "software-defined network". Domain managers can create arbitrary network topologies within the same domain without affecting other domains' network topologies.

I/O Virtualization and Storage Resources (SR): Device aggregation is made possible by I/O virtualization, which allows many physical devices to be aggregated into a single logical device that can then be provided to a virtual machine (VM). It is via the use of this that cloud systems are able to combine various storage devices into highly accessible logical devices that have arbitrary storage capacities. The term "storage resources" (SR) refers to these logical devices, which might be any unit that is supported by disk encryption subsystem.

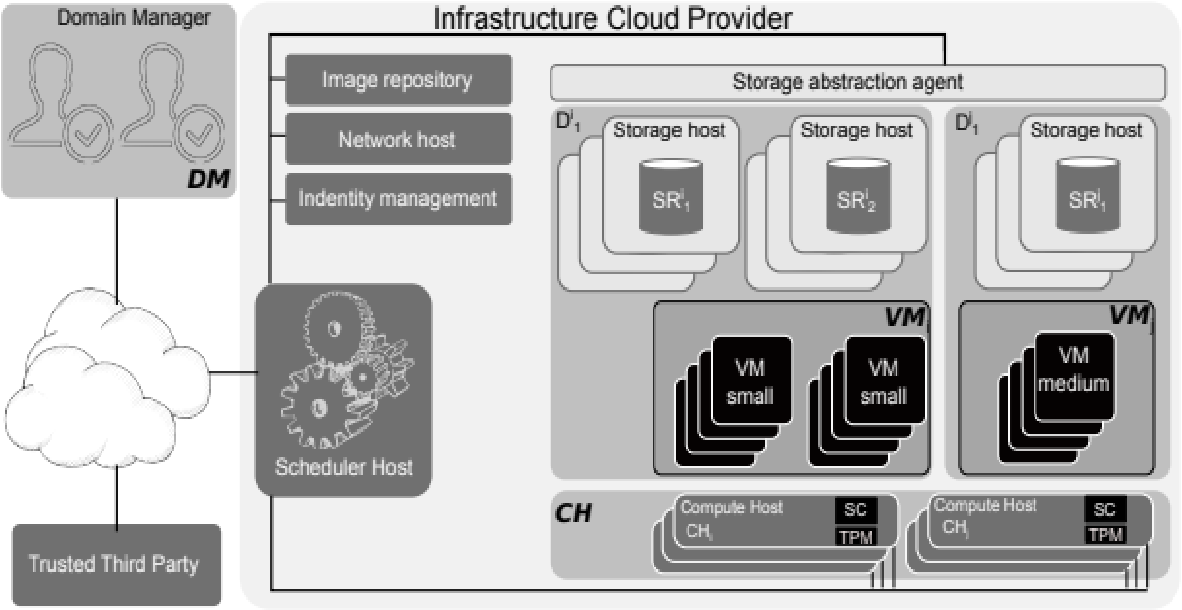


Fig. 1. High-level view of the IaaS model.

IV PROPOSED SYSTEM

The proposed system, Division, and Replication of Data in the Cloud for Optimal Performance and Security (DROPS), aims to enhance both performance and security in cloud computing environments. The methodology strategically fragments user files into pieces and replicates them across multiple nodes within the cloud. Unlike traditional methods that store entire files on a single node, DROPS ensures that no single node contains all of the data, thus minimizing the risk of data compromise in the event of a successful attack on any individual node. Cloud computing has become increasingly popular in recent years, offering distributed computing resources and application platforms shared over the internet on a pay-as-you-go basis. However, despite its benefits, cloud computing also introduces security threats and vulnerabilities that undermine user trust. These threats can potentially lead to the unauthorized access or manipulation of confidential data stored in cloud environments.

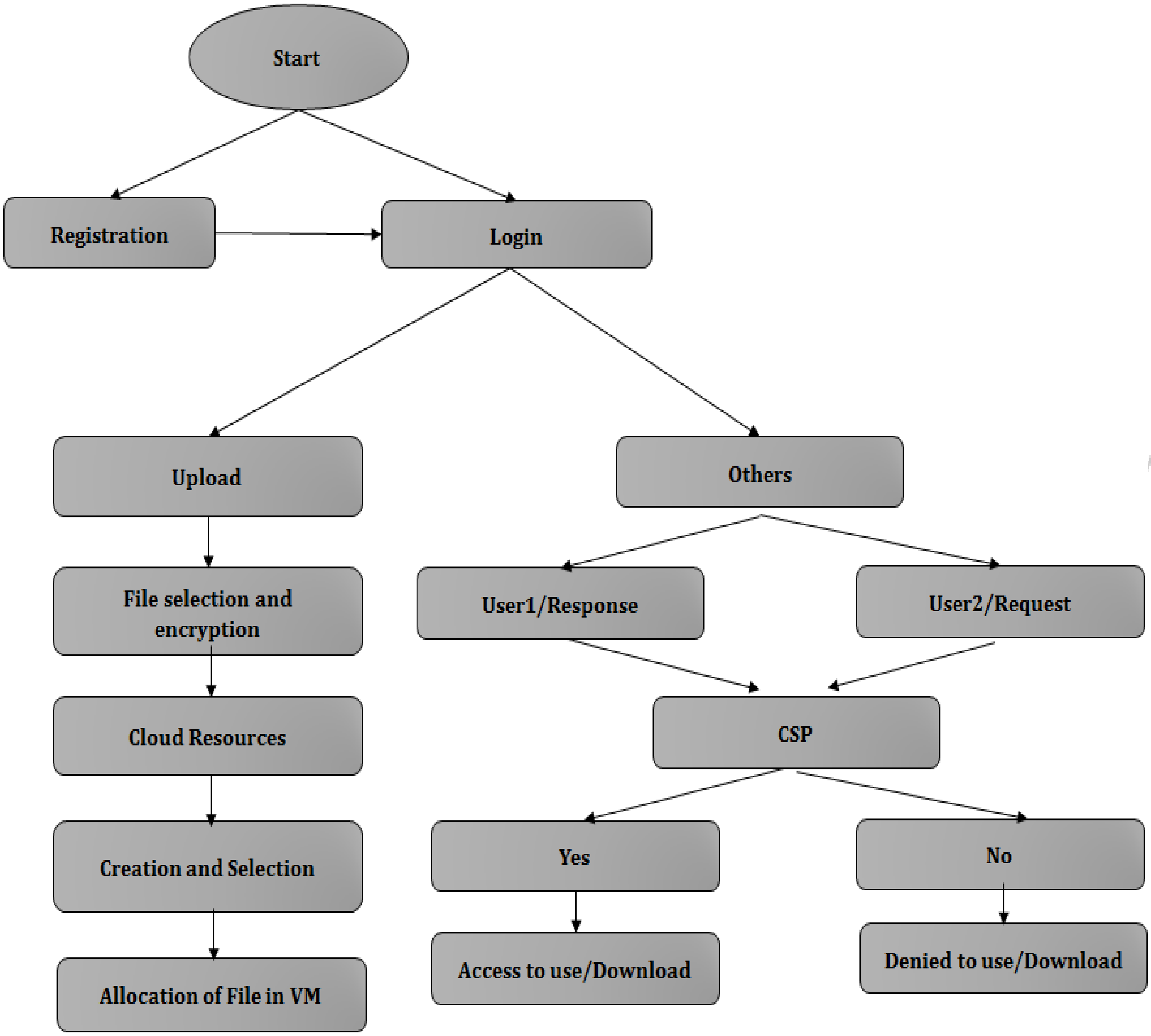


Fig.2 Design System

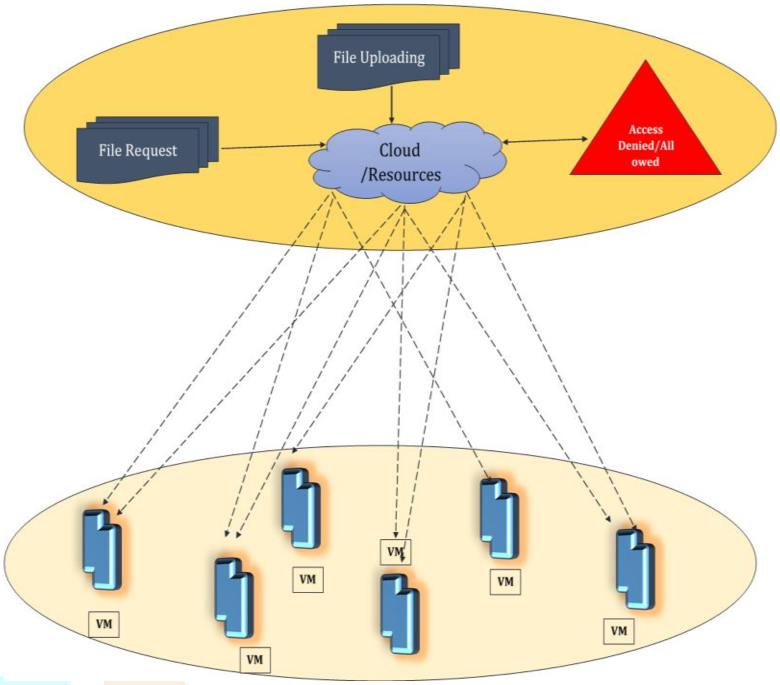


Fig. 3. System Architecture

MODULES:

- Initialization
- User File Uploading
- File Encryption (AES Algorithm)
- Cloud Resources Creation and Allocation
- User Request To the Friend File
- CSP Verification and Permission

Initialization:

- In Initialization module, new users are required to register their information throughout the procedure and get a username and password immediately.
- The procedure that allows users to upload files and makes requests for files whenever they need them.

User File Uploading:

- User enters the username and password with security key for process.
- In order to validate the user whose information is being uploaded, we employ a security key and create a one-time password (OTP) for each individual upload.
- The OTP produces a dynamic value for each time the user joins the procedure itself.
- Then, the User uploads the file to the server.

File Encryption (AES Algorithm):

- Then encrypt the file.
- For encryption, we use the AES algorithm, which encrypts the file in unreadable.
- The file is then prepared to be loaded into the cloud resources.
- Now, the file with full security is uploaded into cloud resources with a key.

The AES encryption process

Key Expansion: AES employs a key expansion process to generate round keys for each round of encryption. Let K denote the original encryption key. The key expansion algorithm transforms K into a series of round keys denoted as

$$K0,K1,...,Kn-1.$$

SubBytes Transformation: In this step, each byte of the input data is substituted with another byte from a fixed table (S-box). Let S represent the substitution table. The SubBytes transformation can be expressed as:

$$\text{OutputByte}=S[\text{InputByte}]$$

ShiftRows Transformation: This step cyclically shifts the rows of the state matrix. Let S denote the state matrix. The ShiftRows transformation can be represented as follows:

- Row 0 remains unchanged.
- Row 1 shifts one position to the left.
- Row 2 shifts two positions to the left.
- Row 3 shifts three positions to the left.

MixColumns Transformation: Each column of the state matrix is treated as a polynomial over GF (28) and multiplied with a fixed polynomial. Let M represent the MixColumns matrix. The MixColumns transformation can be expressed as:

$$\text{OutputColumn}=M\times\text{InputColumn}$$

AddRoundKey Transformation: This step XORs each byte of the state matrix with a corresponding byte from the round key. Let Kidenote the round key for the ith round. The AddRoundKey transformation can be expressed as:

$$\text{OutputByte}=\text{InputByte}\oplus Ki$$

Nr represents the number of rounds in the AES encryption (determined by the key length).
KeyExpansion, subBytes, shiftRows, mixColumns, and addRoundKey are functions implementing the respective AES transformations.

ALGORITHM

```
function encryptFile(file, key):
    expandedKey = keyExpansion(key)
    state = initializeState(file)
    addRoundKey(state, expandedKey[0])
    for round = 1 to Nr - 1 do:
        subBytes(state)
        shiftRows(state)
        mixColumns(state)
        addRoundKey(state, expandedKey[round])
    subBytes(state)
    shiftRows(state)
    addRoundKey(state, expandedKey[Nr])
    encryptedFile = stateToBytes(state)
    return encryptedFile
```

Cloud Resources Creation and Allocation:

- Resources in the cloud are now being set up for processing
- Under the cloud, we built virtual machines with specific characteristics, such as speed, RAM, bandwidth, and VM ID.
- We then sort the virtual machines according to their storage capacities.
- After that, we distribute the files to the cloud-based virtual computers based on their capacities.

User Request to Friend File:

- The user then contacts the CSP by sending a request along with the file's security key.
- Every time a user wants to download anything, we check their identity using their security key and create an OTP (one-time password).
- After the server receives a request, they save the relevant data in the database.
- Send the request to the CSP, and the CSP will send it to the file owner.

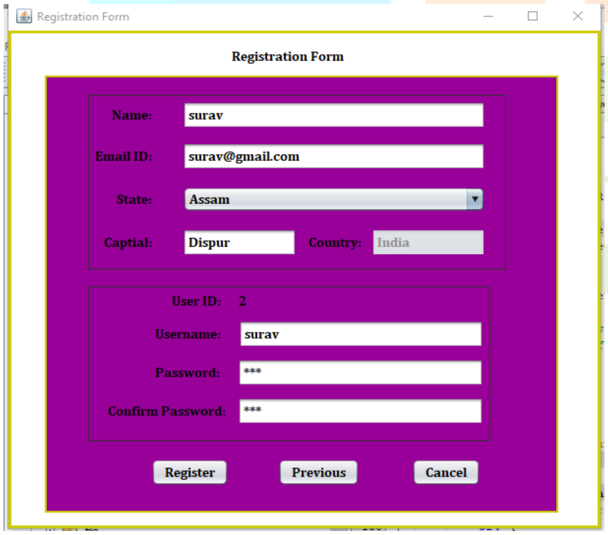
CSP Verification and Permission:

- The request is sent to the file owner using CSP.
- After that, the owner choose whether to provide access or not.
- CSP then verifies the answer after forwarding the request.
- Then grant the user permission to access and download the file.

File Encryption (AES Algorithm):

In this module, files are encrypted using the AES algorithm to ensure data security before being uploaded to cloud resources. Symmetric encryption algorithms like AES are often employed to protect sensitive data.

V EXPERIMENTS RESULTS



Registration Form

Registration Form

Name: surav

Email ID: surav@gmail.com

State: Assam

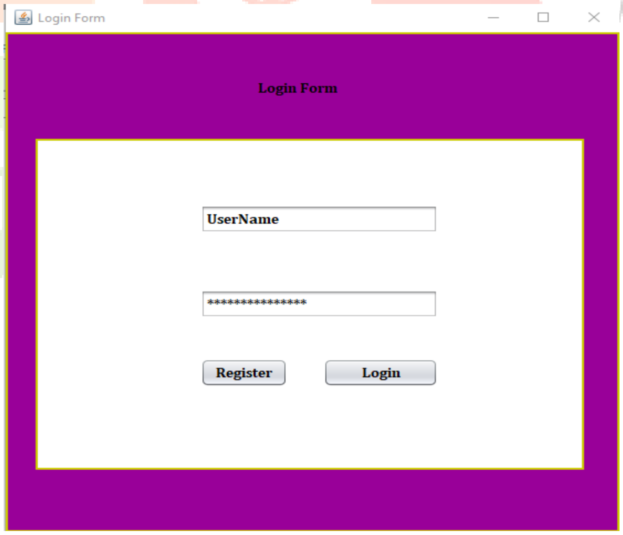
Captial: Dispur Country: India

User ID: 2

Username: surav

Password: ***

Confirm Password: ***



Login Form

Login Form

UserName

****

Fig. 4. Login Form

Fig. 5. Registration Form

Fig. 4 likely depicts a login form, where users can enter their credentials, such as username or email and password, to access a system or application. This form is typically used to authenticate users and grant them access to their accounts or the system's functionalities. Fig. 5 likely depicts a registration form, where users can provide their personal information, such as name, email address, and desired password, to create a new account. This form is typically used for onboarding new users and collecting necessary information to set up their accounts.

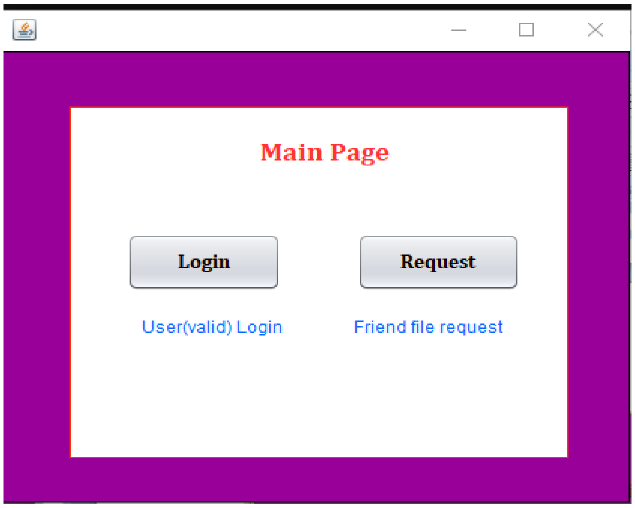
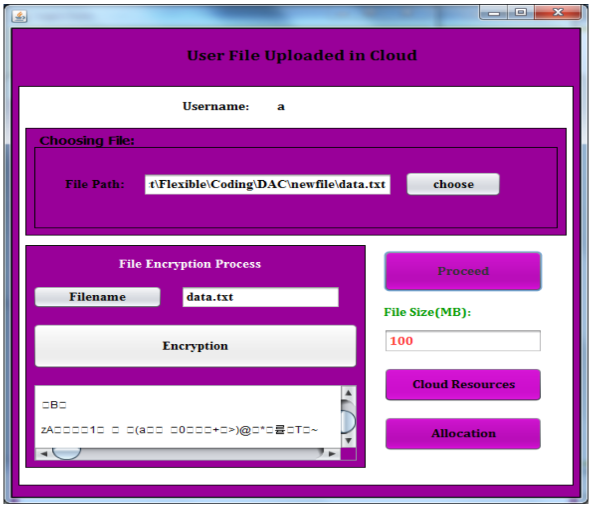

Main Page

Fig.6.Login user


User File Uploaded in Cloud

Username: a

Choosing File:

File Path: t\Flexible\Coding\DAC\newfile\data.txt

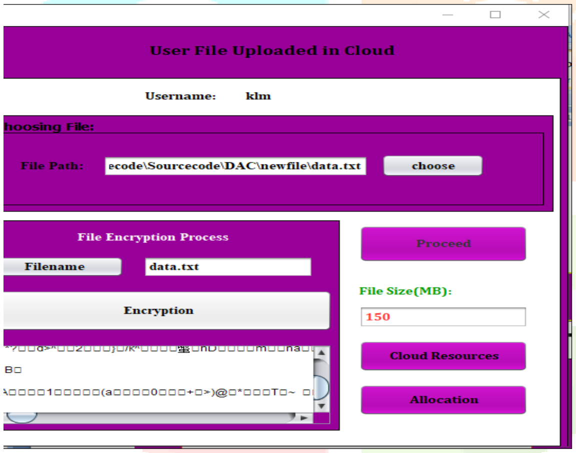
File Encryption Process

Filename: data.txt

File Size(MB): 100

Fig.7.Upload data

The process of registering a new user and obtaining their username and password takes place inside the Initialization module of the procedure. The act of uploading the files and requesting the file whenever they need. Fig. 5 shows what was accomplished in this procedure. A platform for cloud servers, with the goal of developing an integrated framework that establishes a trust chain between the domain manager and the virtual machine instances


User File Uploaded in Cloud

Username: klm

hoosing File:

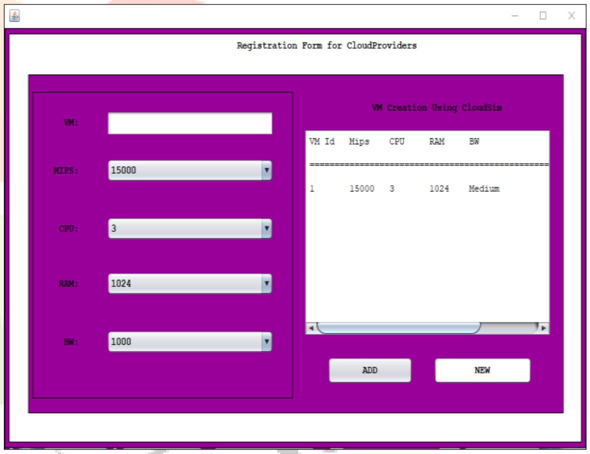
File Path: ecode\Sourcecode\DAC\newfile\data.txt

File Encryption Process

Filename: data.txt

File Size(MB): 150

Fig.8. user file uploaded in cloud


Registration Form for CloudProviders

VM:

MIPS: 15000

CPU: 3

RAM: 1024

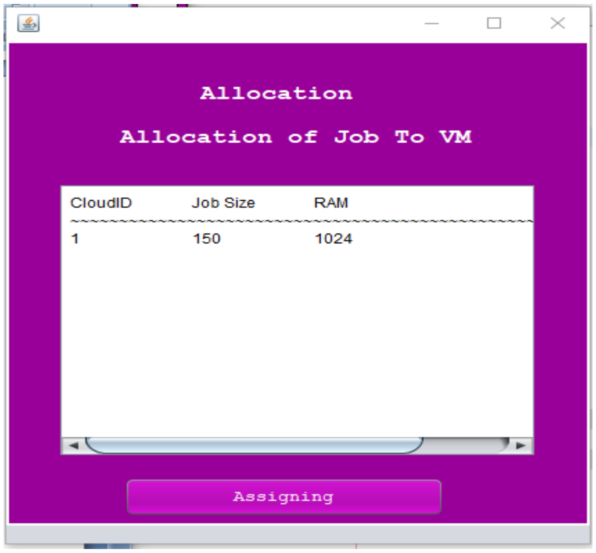
BW: 1000

VM Creation Using CloudSim

```
VM Id   Mips    CPU     RAM     BW
==============================================
1       15000   3       1024    Medium
```

Fig.9. Creating VM

In Fig.8, a user has uploaded a file to a cloud storage service. The specifics of the file and the cloud service being used. In Fig.9 creating a virtual machine (VM) for a cloud service, the amount of CPU and RAM allocated to the virtual machine, storage options, networking settings, and any additional configurations or customizations required.



```
CloudID        Job Size        RAM
------------------------------------------
1              150             1024
```

Fig.10. allocation job to VM

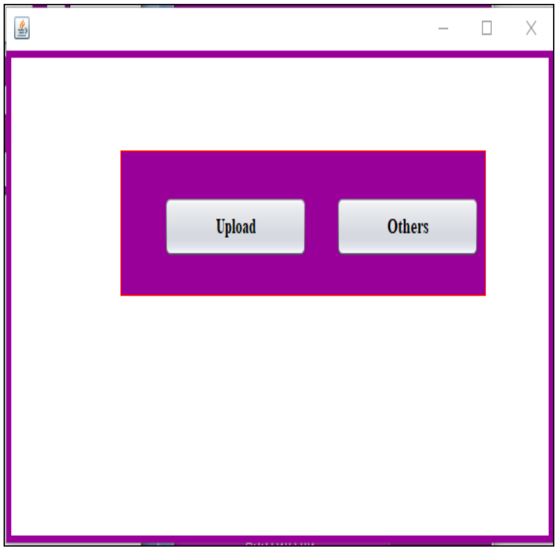
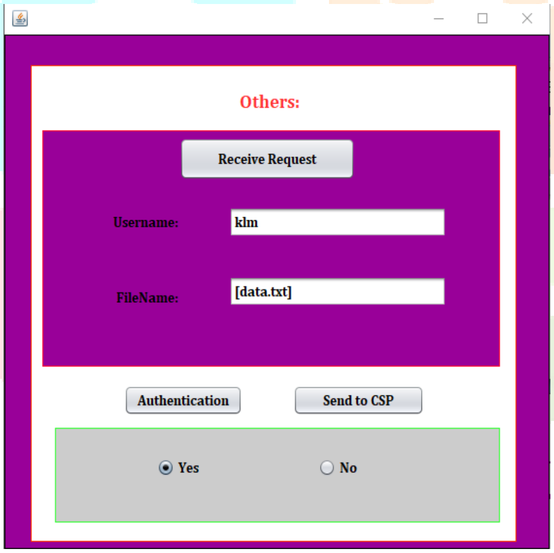


Fig.11. upload

Fig.10 likely depicts a user interface where jobs or tasks are being allocated to virtual machines (VMs). This environment where computational tasks are distributed among multiple VMs to optimize resource utilization and performance

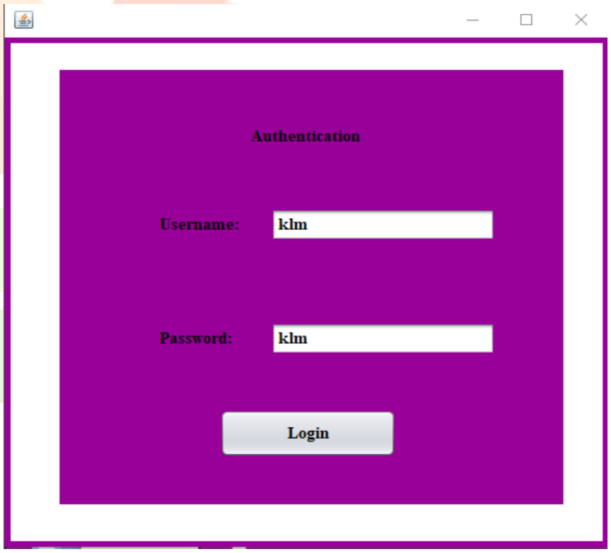


Username: klm

FileName: [data.txt]

● Yes ○ No

Fig.12. received request

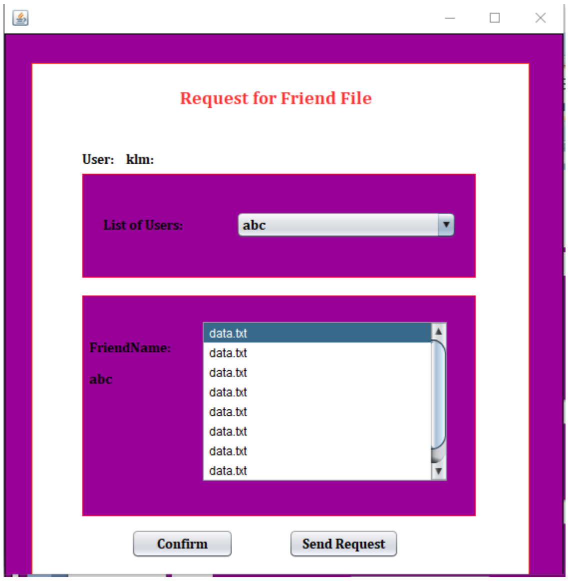


Username: klm

Password: klm

Fig.13. authentication form

Fig.12 depicts a user interface or system notification indicating that a request has been received. a request from a client, an application receiving a request for data, Fig.13 likely depicts an authentication form where users are required to provide their credentials to verify their identity before accessing a system or application.


Request for Friend File

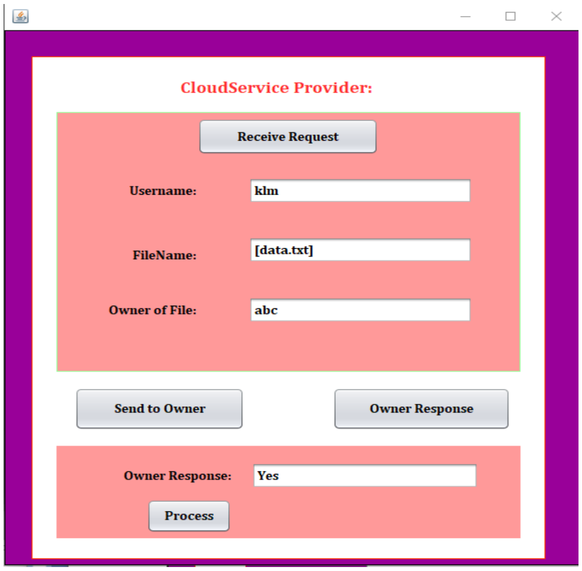
User: klm:

List of Users: abc

FriendName: abc

data.txt
data.txt
data.txt
data.txt
data.txt
data.txt
data.txt
data.txt

Fig.14. request for friend file


CloudService Provider:

Username: klm

FileName: [data.txt]

Owner of File: abc

Owner Response: Yes

Fig.15. cloud service provider

Fig.14 depicts a user interface or notification indicating that someone has requested access to a friend's file. a file-sharing platform or a cloud storage service where users can share files with others. Fig.15 depicts a scenario where a cloud service provider sends a notification to the owner of a file regarding a request made by another user. The owner then has the option to respond to the request, either granting or denying access to the file**.**

VI CONCLUSION

In conclusion, work has focused on addressing the security challenges faced by tenants in cloud computing environments, particularly within the Infrastructure as a Service (IaaS) model. In this work, presented a framework for deploying a trusted infrastructure cloud, focusing on two key aspects: the deployment of virtual machines (VMs) on trusted compute hosts and the domain-based protection of stored data. This framework aims to enhance security measures within the cloud environment to better protect tenant data and resources. Provided a detailed overview of the design, implementation, and security evaluation of protocols aimed at ensuring the trustworthiness of VM deployment and protecting stored data within specific domains. These protocols are crucial for establishing trust in the cloud infrastructure and mitigating potential security risks associated with unauthorized access or manipulation of tenant data.

REFERENCES

1.B. Bertholon, S. Varrette, and P. Bouvry, “Certicloud: a novel tpmbased approach to ensure cloud IaaS security,” in Cloud Computing, 2011 IEEE International Conference on, pp. 121–130, IEEE, 2011.

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

3.A. Cooper and A. Martin, “Towards a secure, tamper-proof grid platform,” in Cluster Computing and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on, vol. 1, pp. 8–pp, IEEE, 2006.

4.W. Wang, Z. Li, R. Owens, and B. Bhargava, “Secure and efficient access to outsourced data,” in Proceedings of the 2009 ACM workshop on Cloud computing security, pp. 55–66, ACM, 2009

5.TahirAlyas;KaramathAteeq;MohammedAlqahtani;SaigeetaKukunuru;NadiaTabassum;Rukshanda Kamran (2022)Security Analysis for Virtual Machine Allocation in Cloud Computing 2022 International Conference on Cyber Resilience (ICCR)Year: 2022

6. Deepika;Rajneesh Kumar;Dalip (2022)Security Enabled Framework to Access Information in Cloud Environment 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON) Year: 2022
7. S Mahipal;V. Ceronmani Sharmila(2021) Virtual Machine Security Problems and Countermeasures for Improving Quality of Service in Cloud Computing 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS) Year: 2021
8. Narayanasetti Mehar Dhinakar;Kasineni Kishan Rao;Neradi Jayanath;Repalle Devi Vara Prasad;Vijaya Chandra Jadala;Radhika Rani Chintala (2023) Defending against Cache-based Side-Channel Attack using Virtual Machine Migration in Cloud 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS) Year: 2023
9. N.B. Kadu;Pramod Jadhav;Santosh Pawar (2022) Virtual Machine Migration Techniques, Security Threats and Vulnerabilities 2022 5th International Conference on Contemporary Computing and Informatics (IC3I) Year: 2022
10. Saurabh Singhal;Rishabh Srivastava;R Shyam;Deepak Mangal(2023) Supervised Machine Learning for Cloud Security 2023 6th International Conference on Information Systems and Computer Networks (ISCON) Year: 2023
11. Peter Ntambu;Steve A Adeshina(2021) Machine Learning-Based Anomalies Detection in Cloud Virtual Machine Resource Usage 2021 1st International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS) Year: 2021
12. Raghunadha Reddi Dornala;Sudhir Ponnapalli;Adusumilli Ramana Lakshmi;Kalakoti Thriveni Sai(2023) An Advanced Cloud Security and Load Balancing in Health Care Systems 2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS) Year: 2023
13. Fazalur Rehman;Zia Muhammad;Sara Asif;Hameedur Rahman(2023) The next generation of cloud security through hypervisor-based virtual machine introspection 2023 3rd International Conference on Artificial Intelligence (ICAI) Year: 2023
14. C. Kaushik;T. Ram;C Ritvik;T. Lakshman(2022) Network Security with Network Intrusion Detection System using Machine Learning Deployed in a Cloud Infrastructure 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC) Year: 2022
15. 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.
16. S. Graf, P. Lang, S. A. Hohenadel, and M. Waldvogel, “Versatile key management for secure cloud storage,” in Proceedings of the 2012 IEEE 31st Symposium on Reliable Distributed Systems, pp. 469–474, IEEE Computer Society, 2012