Providing Restrictions Against Attack And Congestion Control In Public Infrastructure Clouds

K. Aravinth[1], R. Gowtham[2], S. S. Sugania[3]
[1, 2, 3] Department of Computer Science
[1, 2, 3] JEPPIAAR SRR Engineering College, Chennai

Abstract- *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. In this project, we describe 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. The cloud user files are uploaded and stored in domain based. To provide resistance against cloud based web application attacks (focusing on session hijacking and broken authentication). Authentication is a critical aspect of this process, but even solid authentication mechanisms can be undermined by flawed credential management functions, including password change, forgot my password, remember my password, account update, and other related functions. In either case, if the session tokens are not properly protected, an attacker can hijack an active session and assume the identity of a user. Creating a scheme to create strong session tokens and protect them throughout their lifecycle has proven elusive for many developers. For session management, we have implemented cookie management and idle timeout. Also in our proposed system encryption keys are maintained outside of the IaaS domain. For key encryption we proposed RSA algorithm. For data owner file encryption, we use camellia algorithm. Finally the files are stored in public cloud named CloudMe.*

Keywords- Cloud Computing, Key seed Mechanism, File encryption, Domain Based Storage Protection, AES Algorithm, Camellian Algorithm.

I. INTRODUCTION

Cloud computing has progressed from a bold vision to massive deployments in various application domains. However, the complexity of technology underlying cloud computing introduces novel security risks and challenges.[1]Threats and mitigation techniques for the IaaS model have been under intensive scrutiny in recent years, while the industry has invested in enhanced security solutions and issued best practice recommendations. From an end-user point of view the security of cloud infrastructure implies unquestionable trust in the cloud provider, in some cases corroborated by reports of external auditors. [2]While providers may offer security enhancements such as protection of data at rest, end-users have limited or no control over such mechanisms. There is a clear need for usable and cost-effective cloud platform security mechanisms suitable for organizations that rely on cloud infrastructure. Another relevant security mechanism is encryption of virtual disk volumes, implemented and enforced at compute host level. We introduce a domain-based storage protection protocol to allow domain manager's store encrypted data volumes partitioned according to administrative domains.[3]We introduce a list of attacks applicable to IaaS environments and use them to develop protocols with desired security properties, perform their security analysis and prove their resistance against the attacks.

II. EXISTING SYSTEM

In existing system, to the best of our knowledge, none of the solutions provides cloud tenants a proof regarding the integrity of compute hosts supporting their slice of the cloud infrastructure. Also no secure framework in cloud environment to provide security against cloud attacks. In existing system, all data owners' files are stored in a single virtual machine and congestion happens during file request and response. Datacentres can choose to encrypt data on the operating system (OS) level within their VM environments. However, this approach suffers from several drawbacks: first, the underlying compute host will still have access encryption keys whenever the VM performs cryptographic operations; second, this shifts towards the tenant the burden of maintaining the encryption software in all their VM instances and increases the attack surface; third, 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.The existing system support data encryption at rest is offered by several cloud providers and can be configured by tenants in their VM instances, functionality and migration capabilities of such solutions are severely restricted. In most cases cloud providers maintain and manage the keys necessary

for encryption and decryption of data at rest. This further convolutes the already complex data migration procedure between different cloud providers, disadvantaging tenants through a new variation of vendor lock-in. Also in existing system attacks or threats to the infrastructure is not efficiently handled.

III. PROPOSED SYSTEM

In this system, we describe 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. The cloud user files are uploaded and stored in domain based. To provide resistance against cloud based web application attacks (focusing on session hijacking and broken authentication).Authentication is a critical aspect of this process, but even solid authentication mechanisms can be undermined by flawed credentialmanagement functions, including password change, forgot my password, remember my password, account update, and other related functions. In either case, if the session tokens are not properly protected, an attacker can hijack an active session and assume the identity of a user. Creating a scheme to create strong session tokens and protect them throughout their lifecycle has proven elusive for many developers. For session management, we have implemented cookie management and idle timeout.Also in our proposed system encryption keys are maintained outside of the IaaS domain. For key encryption we proposed RSAalgorithm. For data owner file encryption, we use camelliaalgorithm. Finally the files are stored in public cloud named CloudMe.

Architecture Diagram

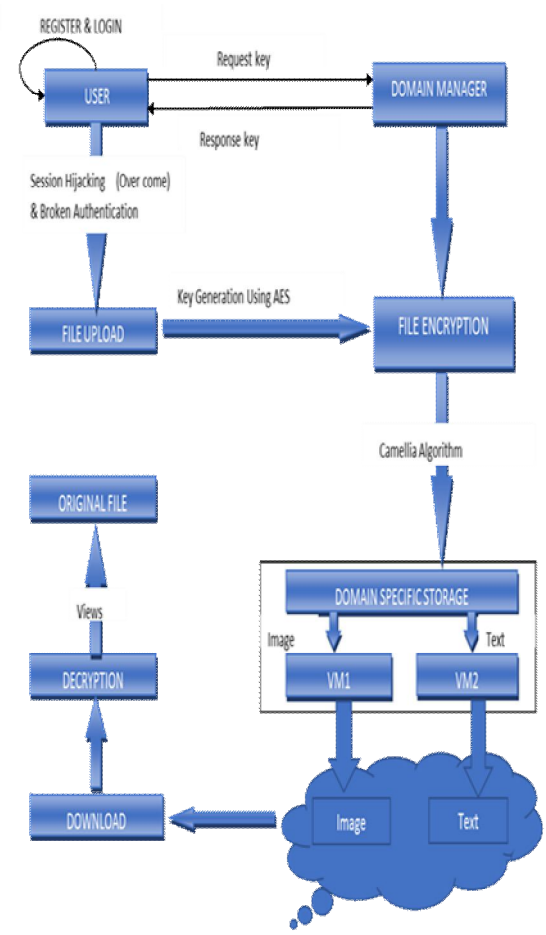

Fig.[1] Architecture diagram for infrastructure cloud

IV. MODULES

1) Resistance against Cloud attacks
2) Key Seed Mechanism
3) File encryption
4) Domain Based Storage Protection
5) Cloud Storage

1. Resistance against Cloud attacks

Managing user name and passwords has become a cumbersome task in today's internet-driven world. However, managing user name and passwords is a necessary evil with the rapid growth in data, advancements in mobile and cloud technologies and the increasing plethora of security breaches seeming to happen every other day. As a result, authentication has become more advanced to protect the data, systems and networks against intruders. A brute force attack is a trial-and-error method used to obtain information such as a user password or personal identification number (PIN).Our application provides field and security validation against

broken authentication attacks.Session hijacking is the exploitation of a valid computer session, it is also to gain unauthorized access to information or services in a computer system. It's nothing but hijacking a session.Session IDs are exposed in the URL (e.g., URL rewriting). Session IDs are vulnerable to session fixation attacks. The Session Hijacking attack consists of the exploitation of the web session control mechanism, which is normally managed for a session token. The Session Hijacking attack compromises the session token by stealing or predicting a valid session token to gain unauthorized access to the Web Server. The session ID is normally stored within a cookie or URL. For most communications, authentication procedures are carried out at set up. Session hijacking takes advantage of that practice by intruding in real time, during a session.

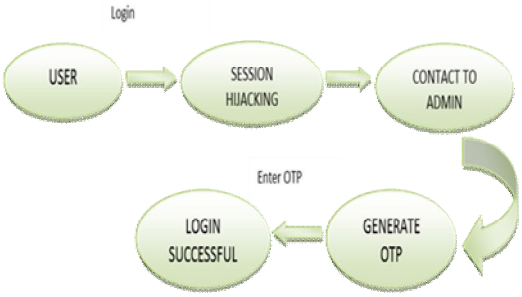

Figure 2. Resistance against Cloud attacks

2. Key Seed Mechanism

The SEED key generation mechanism is a key generation mechanism for SEED. This is the techniques to generate and hide the key to or from the user. In this mechanism keys such as encrypt and decrypt keys are generated by the ked seed mechanism and also it give the keys if the user forget or request the key from admin. The key seed mechanism mainly used for hiding the key from attacker .It will gives the keys only authorized users.The Advanced Encryption Standard (AES), also known as Rijndael (its original name), is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001. AES is a subset of the Rijndaelcipherdeveloped by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, who submitted a proposal to NIST during the AES selection process. Rijndael is a family of ciphers with different key and block sizes.AES operates on a 4 × 4 column-major order matrix of bytes, termed the state, although some versions of Rijndael have a larger block size and have additional columns in the state. Most AES calculations are done in a special finite field.AES consists of several rounds of several processing steps that include substitution, transposition and mixing of the input plaintext and transform it into the final output of ciphertext.

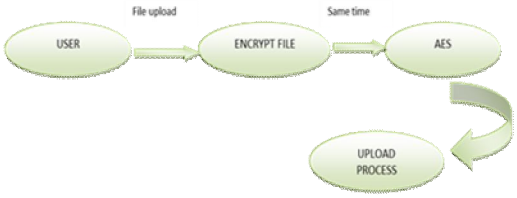

Figure 3. Resistance against Cloud attacks

3. File encryption

Camellia was jointly developed by Nippon Telegraph and Telephone Corporation and Mitsubishi Electric Corporation in 2000 [CamelliaSpec]. Camellia specifies the 128-bit block size and 128-, Encryption Standard (AES).[6]Camellia is characterized by its suitability for both software and hardware implementations as well as its high level of security. From a practical viewpoint, it is designed to enable flexibility in software and hardware implementations on 32-bit processors widely used over the Internet and many applications, 8-bit processors used in smart cards, cryptographic hardware, embedded systems, and so on [CamelliaTech].Moreover, its key setup time is excellent, and its key agility is superior to that of AES. Camellia has been scrutinized by the wide cryptographic community during several projects for evaluating crypto algorithms. In particular, Camellia was selected as a recommended cryptographic primitive by the EU NESSIE (New European Schemes for Signatures, Integrity and Encryption) project [NESSIE] and also included in the list of cryptographic techniques for Japanese e-Government systems which were selected by the Japan Cryptography Research and Evaluation Committees(CRYPTREC).[7]Camellia can be divided into "key scheduling part" and "data randomizing part".

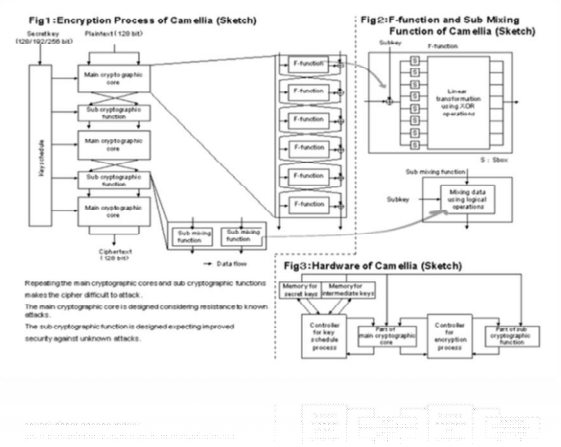

Figure 4. File encryption

4. Domain Based Storage Protection

In DBSP (domain-based storage protection). In this all the data owners' files are analyzed deeply and stored in virtual machines in specific to the domains.It would provide some advantages this would eliminate bottle neck problems and congestion, response time would be fast when compared to the traditional approach and easy to analyze and provide heap memory space based on the usage. Thus experimentally, we have connected virtual machines to a server (domain manager) using same connection. Example both server and virtual machines should be connected to a same internet connection to upload and access the user files.

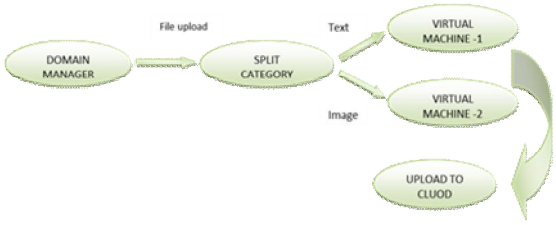

Figure 5. Domain Based Storage Protection

5. Cloud Storage

Cloud storage is a model of data storage where the digital data is stored in logical pools, the physical storage spans multiple servers (and often locations), and the physical environment is typically owned and managed by a hosting company.[8]These cloud storage providers are responsible for keeping the data available and accessible, and the physical environment protected and running. People and organizations buy or lease storage capacity from the providers to store user, organization, or application data. The group user can upload the files in real cloud server named cloudMe. Duplication of files are checked and the files is been uploaded in the cloud server. To get a file, the user needs to send a request to the cloud server. The cloud server will also check the user's identity before issuing the corresponding file to the user. During file access the user key has to match by the group manager and the requested file can be downloaded by the group users.

Figure 6. Cloud Storage

V. LIMITATION OF EXISTING SYSTEM

The underlying compute host will still have access encryption keys whenever the VM performs cryptographic operations. This shifts towards the tenant the burden of maintaining the encryption software in all their VM instances and increases the attack surface.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.

VI. APPLICATIONS

- Data centers
- Cloud servers i.e, Bank, Social media websites
- Organisations with centralized servers

VII. 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 the IaaS security model. In this proposed work, we presented a framework for trusted infrastructure cloud deployment, with the focus points are VM deployment on trusted compute hosts, domain-based protection of stored data, Resistance against cloud attacks and Secure cloud storage.

VIII. FUTURE ENHANCEMENTS

Important topics for future work are strengthening the trust model in cloud network communications and applying searchable encryption schemes to create secure cloud storage mechanisms. With reasonable engineering effort the framework can be integrated into production environments to strengthen their security proper

REFERENCES

[1] Nicolae Paladi, Christian Gehrmann, and Antonis Michalas, "Towards trusted cloud computing," in Proceedings of the IEEE Transactions on Cloud Computing (Volume: 5, Issue: 3, July-Sept. 1 2017).

[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] 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.

[9] 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.

[10] 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.

[11] 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.

[12] A. Sahai, "Ciphertext-policy attribute-based encryption," in In Proceedings of the IEEE Symposium on Security and Privacy, 2007.