Research in Information Security

Project Report

Enhanced ML Based Security Framework for Cloud Security

Team 8

1. T V S Vishnu Vardhan (M.Tech. CSIS, 2023202021)
2. L Sai Chaitanya Reddy (M.Tech. CSIS, 2023202023)
3. V Subhash Chandra Bose (M.Tech. CSIS, 2023202012)

November 05, 2024

Abstract

In a cloud environment, since sensitive and confidential information is stored, we can ensure integrity, confidentiality and availability of systems only by handling the cryptographic keys securely. As the cloud service is growing day by day and people who use it are increasing rapidly, it is necessary to ensure reliable and secure key management service. Taking an example of a financial institution like a bank, their data is highly confidential, and should not be revealed. But, if they use cloud services, then their data should be secure. Access control, data security, network security, life cycle management of keys, protocols for encryption and security of keys are important components of overall security of the cloud environment. In this work, security is enforced by using 2 parts. Permission Detection engine and Registration authorization engine. An ML Based algorithm is also used to detect anomalies and deviations. Using this, the secure access is ensured.

Keywords : Cloud Computing, Cloud Security, Access Control, Cryptogtaphic Keys, Clustering Algorithm, Anomalies, Cloud Services, Key Management Systems

Introduction

Cloud Computing is being used by many users across the world. This is increasing day by day. Since there is no requirement of maintaining separate resources, and the cloud service is cost-effective, many users are shifting towards cloud. While cloud computing provides many advantages, this also creates some security issues. These issues include security of sensitive data, and managing the services. Key management is a very important aspect in ensuring security. Key management consists of tasks like creation, storage, distribution, and revoking the keys required for cryptographic algorithms and techniques. Protecting the keys and their security is important because these will leads to some unauthorised persons accessing the data, data leaks, and may also leads to financial problems, and legal issues. Hence, it is very important to develop security protocols used for cloud security.
Strong key management policies, which are crucial for safeguarding key management services in cloud environments, have made cloud security much more important. Because of the particular requirements of cloud infrastructure, there are significant security challenges. In order to improve key management's security and dependability, a specialized framework with secure policies can be put into place.

Organizations frequently pay significant costs for additional hardware, software, and networking resources as digital data transmissions expand. Many are choosing cloud storage services as a way to control these expenses. Servers, databases, and software are just a few of the resources that may be accessed remotely via the internet thanks to cloud computing (CC). With this configuration, users can store information in remote databases that they can access from any location with a reliable internet connection.

Cloud services do, however, come with security and privacy hazards even if they are obviously more convenient. Data privacy is still a big worry, particularly when it comes to sensitive data like bank account information or medical records. Many businesses are hesitant to fully utilize cloud services because of these worries because they might not always ensure the required standards for data confidentiality, integrity, and dependability.

In this work, the main challenges that happen are with the cryptographic key management services in the cloud environment. The attacks that are possible include unauthorised access, data leaks, etc. Also, the communications that occur between client and server can be interpreted and some information can be retrieved from these. To avoid all these, an ML based security framework was developed for cloud security using KMeans algorithm.

KMS is a common component, which normally uses envelope encryption for key management. In development environment, server-side applications are built through VMs and deployed on cloud. In building mobile application, same environment is used, which makes client side REST API calls to server-side apps hosting REST APIs. The mobile apps can be any business apps like Amazon, Flipkart, etc. Applications are deployed on different mobile devices When apps are used by users, first the User authentication is done, then the client uses REST API calls to server-side applications on cloud VMs, securely using OAuth2.0 based protocol and HTTPS Layer security. Here, KMS Layer acts as key vault, client and server gets required keys from KMS. Cloud directory acts as OAuth2.0 authorization server. ML will be used to detect outliers during interactions of apps with KMS. Log data from KMS is stored in a big data repository. For each app that interacts with KMS, app will be placed in clusters. Checks will be made on a scheduled basis for any deviation using an algorithm like K Means. If any anomaly is found, the app will be blocked from using KMS. This guarantees security of cloud environment by reducing the probability of unauthorized access and security issues. But, this is not scalable to use cases large databases, requires internet access, and have a single point of failure issues. To avoid these issues, we have proposed our solution to make it robust to all these conditions.

Literature Review

In cloud security, there were many research contributions. In [1] by Mohammed et. al., they introduced a Machine Learning assisted Cloud Computing model, ML-CCM, which was developed for increasing data transmission speed and improving security. The data transmission speed has been improved significantly. They have used some supervised and unsupervised ML algorithms, This paved a way for using Machine Learning algorithms in Cloud security.

Then [2] by Lo'ai et. al. proposed a P2P cloud system P2CS, to handle processing of large databases. They also introduced a model for cloudlets. This also achieved a significant improvement over existing frameworks.

The [3] by Narayanan et. al. focused on Sharing of data and Secured authentication. They have used the Hashing algorithm SHA3 during the initial registration phase. During any login, they check the hash and the given input. This was similar to hashing a password.

Jayaraman et al. (2024) [4] proposed SECGURU for Network Policy Validation. SECGURU automatically verifies network connectivity policies in Azure using the Z3 solver and SMT bit vector theory. This automates the validation of network policies, reducing manual work. Strong use of formal verification (SMT) was used to ensure high accuracy.

Viswanath et al. (2021) [5] proposed Hybrid Encryption Framework for Multi-Cloud. This framework uses a unique hybrid encryption architecture to increase the security of multi-cloud storage. Data slicing, encryption, dissemination, and decryption procedures are all included. This achieved high rate of encryption, which makes it effective for big datasets. Security across several cloud storage providers is guaranteed via hybrid encryption.

Work	Advantages	Disadvantages
Mohammed et. al. [1]	• High Data transmission Rate • Used ML Algorithms • Computationally Efficient	• Not cost effective • Very complex implementation
Lo'ai et. al. [2]	• Improved Performance • Improved data processing and analysis • Efficient Resource utilisation	• Very complex implementation • Not scalable
Narayanan et. al. [3]	• Strong authentication using SHA3 • Improved Data Sharing and Data organization	• Computationally expensive • Less flexibility
Jayaraman et. al. [4]	• High accuracy • Automated policy validation	• Only suitable for Azure • Focuses only on network security
Viswanath et. al. [5]	• Scalable • Customizable • Flexible to any environment	• Complex process • Computationally costly • Suitable for multi cloud only

Proposed Work

The drawbacks that are observed in the given work was scalability, offline access and single point of failure. To avoid these, some steps needs to be taken. This might not be scalable, as Lambda functions and KMS are used for Key management. Since Lamda function is causing issue, Amazon EC2 instances with auto-scaling groups can be used. Based on active no. of users, the system will automatically scale. This also allows more flexibility in handling high loads. This also can be customised with instance types which are optimised for some applications. Elastic Kubernetes Service (EKS) service provided by Amazon also can be used for rapid scaling and isolation.

Coming to offline access, since a connection should be necessary and continuous interactions between client and server are required, an active internet connection is required, which gives no possibility of offline access. This may be a concern for some users whose network connection will not be stable. Hence, offline access is crucial for those users. This can be achieved by implementation of local caching, where we store essential data needed for communications. Amplify DataStore service can be used for this. This service synchronises data whenever there is connection. Another way is to implement for conditions with low bandwidth. Although internet connectivity is needed, low bandwidth also works. CloudFront Service can be used to implement this. Hence, the offline access will be possible.

For avoiding single point of failure, establishing a backup KMS using another service provider will be useful. Ex : Using Google KMS and AWS KMS. We can also deploy the services in multiple regions, so we can switch to another region whenever a region fails. This handles single point of failure issue.

For security, the sensitive data should be encrypted during rest and in transit so that it becomes more secure. Also, using a multi factor authentication will also help along with key, to achieve extra layer of security.

Future Directions

In future direction, we can study more on how we can use the database, where we are using MongoDB in current implementation, to achieve scalability and efficiency. Also, logging the every detail of the implementation and history will be very useful for analysing. Any roll back application can be useful for safely rolling back to a safe state. Implementing a robust monitoring system is also crucial for ensuring safety. All the implementation details will be needed to undergo a thorough study and checking of metrics for security efficiency. Hence, these will be the next research ideas.

References

[1] Mohammad, A.S., Pradhan, M.R.: Machine learning with big data analytics for cloud security. Comput. Electr. Eng. 96, 107527 (2021)
[2] Lo'ai, A.T., Saldamli, G.: Reconsidering big data security and privacy in cloud and mobile cloud systems. J. King Saud Univ. Comput. Inf. Sci. 33(7), 810–819 (2021)
[3] Narayanan, U., Paul, V., Joseph, S.: A novel system architecture for secure authentication and data sharing in cloud enabled Big Data Environment. J. King Saud Univ. Comput. Inf. Sci. 34(6), 3121–3135 (2022)
[4] Jayaraman, K., Bjørner, N., Outhred, G., Kaufman, C.: Automated analysis and debugging of network connectivity policies. Microsoft, Tech. Rep. MSR-TR-2014–102, July 2014.
[5] Viswanath, G., Krishna, P.V.: Hybrid encryption framework for securing big data storage in a multi-cloud environment. Evol. Intel. 14(2), 691–698 (2021)