SECURITY RISK OF CLOUD COMUTING

Peddati Divya[1]

Computer Science and Engineering,

Vidya Jyothi Institute of Technology, Hyderabad, India.

Dr. D.Venkateshwarlu,

Associate Professor,

Department of Computer Science and engineering,

Vidya Jyothi Institute of Technology,

Hyderabad, India.

Mail: Pdivyareddym27@gmail.com[1]

ABSTRACT

To make computing more convenient, cloud computing is rapidly gaining popularity. Cloud computing's main security hole is in the software and hardware components. Safe cloud computing may be accomplished by methodically investigating the hazards associated with cloud computing from the angles of service designers, service operators, cloud security designers, and cloud system certifiers. The paper examines the strengths and weaknesses of secure cryptography. In order to accomplish its research objectives, this paper investigates potential operational shortcomings and technological security vulnerabilities, and it proposes secure solutions for cloud systems.

Keywords: Cloud computing, cryptography, hardware vulnerabilities, and key management.

1. INTRODUCTON

More and more people are opting to adopt cloud computing. in order to provide pleasant computer experiences. The cloud's security flaws stem from both software and hardware problems. From the perspectives of service designers, service operators, cloud security designers, and cloud system certifiers, a variety of systematic methodologies might be utilized to evaluate the risks associated with cloud computing in order to achieve safe cloud computing. Disadvantages and security mechanisms in safe cryptography are discussed in this

article. In order to achieve its research goals, this paper offers solutions for safe cloud systems after investigating operational inadequacies and technology security weaknesses.

2. CLOUD COMPUTING

Cloud computing refers to the practice of accessing software and hardware resources over a network, often the Internet, as a service. The term comes from the cloud-shaped visual that system diagrams frequently utilize to show the intricate architecture they include. Remote services can access user data, apps, and processes using cloud computing. Cloud computing involves making resources, such as hardware and software, available online through regulated third-party services. Users of these services frequently have access to modern software developers and server networks.

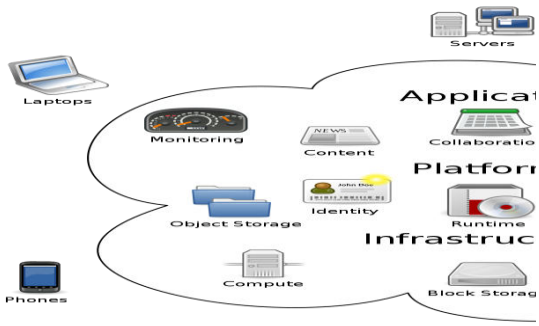


Fig. 1.1Structure of Cloud Computing.

3. EXISTINGSYSTEM

A new cryptographic primitive called attribute-based encryption (ABE) provides a fascinating method for secure and flexible data transmission. The one-to-many nature of ABE means that different ciphertexts can be decrypted with different keys, or the same key can decode several ciphertexts. Crucial regulations Crypttext policy ABE (CP-ABE) and ABE (KP-ABE) are the two kind of ABE. A ciphertext encoding the access policy and the attribute set together form the CP-private ABE key. A cipher text encapsulates the attribute specified in KP-ABE, whereas a private key encodes the access policy. Through the use of CP-ABE, data owners are able to define their own access controls. Previous key management approaches in attribute-based data sharing systems mainly aimed upon proxy re-encryption, key upgrades, and outsourcing decryption, as mentioned above. Unreliable key authorities have been associated with key escrow concerns, according to a research that provided remedies. Although front-end devices, particularly mobile ones, are inherently vulnerable to unauthorized access, there isn't enough proof to suggest that they should be treated with far more suspicion than authority. A more serious problem called "key exposure" happens when front-end devices still have private keys in full, which puts private key security

at risk. In addition, recipients of the data had a greater decryption cost, but key management security was enhanced using most attribute-based data sharing strategies. Therefore, we are unhappy with the previous primary management techniques' efficacy and safety.

4. PROPOSED SYSTEM:

We introduce CKM-CP-ABE, a novel collaborative key management protocol, to enhance the safety and efficiency of key management in cloud data sharing systems. We provide a novel protocol for teamwork. The key authority, the cloud server, and the often accessing client all work together to enable the distributed generation, distribution, and storage of private keys. So, installation is made easier, and safe key management is guaranteed, unlike previous multi-authority systems that required additional physical infrastructure. We build the procedure for updating the private key using attribute groups that we establish. An unique key is assigned to each attribute group, including those with similar properties, such as clients. Changing the key of the attribute group allows for fast and accurate attribute revocation. Both the key escrow problem and the less-discussed key exposure pose risks to the privacy of private keys. When compared to previous key management protocols for cloud-based attribute-based data sharing systems, our suggested protocol's collaborative key management successfully resolves both of the aforementioned difficulties. Last but not least, we back up the proposed approach with security evidence. The collaborative technique greatly reduces the cost of client decryption by utilizing a decryption server to conduct much of the decryption operation without submitting any information to it.

5. LITERATURE SERVEY

The utilization of cloud computing has been advantageous for big data applications, including the processing of medical data. Big data processing services may see a dramatic improvement in QoS as a consequence of the many resources made accessible by cloud platforms. Disruptive networks or misleading advertising might make it so that the quality of service that service providers promise isn't always delivered. This is why it's crucial to evaluate the service's quality based on trustworthy metrics, such the service's historical QoS records. However, the review system would be inefficient and fail to meet customers' needs for rapid answers if all service records are collected for quality evaluation.

Additionally, due to the fact that different records have varied invocation circumstances, it would be risky to evaluate

all the data in the same way, since it might lead to the "Lagging Effect" or inaccurate evaluations. In light of these challenges, this study suggests a novel approach known as Partial-HR (Partial Index Terms—cloud computing, big data, context-aware service assessment, historical quality of service record, weight). Method for evaluating services using historical records. Each prior QoS record in Partial-HR is assigned a weight according on the specifics of the service request. Consequently, quality is evaluated based on subsets of the data that are deemed meaningful. Finally, we run a battery of tests to make sure our proposal can be implemented efficiently and accurately.

There is a lack of quantitative weight models for historical QoS data or only limited consideration of context components in the existing research. Because of this, developing a quantitative weight model that accounts for all context-related factors becomes necessary for an efficient and successful evaluation of big data service quality. In response to this problem, this work introduces Partial-HR, a new method for evaluating services that takes into consideration all relevant context aspects of service invocation, including invocation time, input size, and user location, while still satisfying the Volatility Effect and Marginal Utility. To improve the efficiency and accuracy of the evaluation process, we may use Partial-HR to choose sections of key historical QoS data for service assessment. We do a battery of tests to make sure our idea will work.

In a cloud environment, the stated quality of service data from big data suppliers isn't necessarily correct. Consequently, looking at QoS data from the past is crucial for evaluating service quality. A large number of academics have lately examined this matter and provided suggestions. In answer to the question of QoScredibility, it was proposed that historical QoS data be used to determine the actual quality of service. Literature reviews determine the reliability of a service's quality of service (QoS) by comparing the service's stated SLAs (Service Level Agreements) with QoS statistics from the past. Composition, evaluation, suggestion, and selection are just a few examples of the dependable service-oriented applications that have expanded their use of historical QoS records of services. Nevertheless, the aforementioned literatures address the matter of balancing different sets of historical QoS data.

6. Module Description

Client:

Anyone planning to use a front-end device to access data stored in the cloud is referred

to as a client. In anticipation of the growth of mobile cloud services, the majority of front-end devices are on the go. The client will be granted permission to alter the plaintext if its attribute set matches an access policy associated with the ciphertext. Since the majority of mobile devices are prone to experiencing performance difficulties, users run the danger of overlooking vital information.

Key Authority:

One crucial component is the system's central authority. The key authority is responsible for performing calculations such as key generation and key updating. We take it as read that our system's principal authority is semi-trusted, meaning it recognizes the value of plaintext but isn't motivated to alter it.

Encryption:

To improve the efficiency and safety of key management in cloud data sharing systems, we provide a new protocol for collaborative key management in cipher text policy attribute-based encryption (CKM-CP-ABE). A new protocol for collaboration is introduced. In order to facilitate the distributed generation, issuance, and storage of private keys, the key authority, cloud server, and often accessing client collaborate. Compared to previous multi-authority systems, it is easier to implement due to the fact that safe key management is ensured without requiring extra physical infrastructure.We construct the system for updating private keys by first establishing attribute groups.Key exposure and the key escrow problem, which have received little attention in the literature, pose serious risks to private key security, as we show.

Decryption:

There is a lot of computing power on the decryption server. It separates and absorbs the majority of the decoding work, albeit not entirely. Assuming the decryption server is semi-trusted and the decryption server access route is susceptible, we may conclude that CKM-CP-ABE is sufficient to guarantee data security. Collaborative decryption drastically cuts down on client encryption overhead by utilizing a decryption server to perform the bulk of the decryption and then erasing all traces of it.

Data Owners:

An authorized user who holds data that may be submitted is known as a data owner. The data owner decides which clients they deem suitable to have access to plaintext based on their own stated access restrictions. The data owner initially approached us to request that we upload our information with enhanced security from many major organizations. Once the request is submitted, get upload

file keys from several key authorities. Data encryption and storage on a cloud server is an important part of this procedure for many data owners.

7. ALGORITHMS

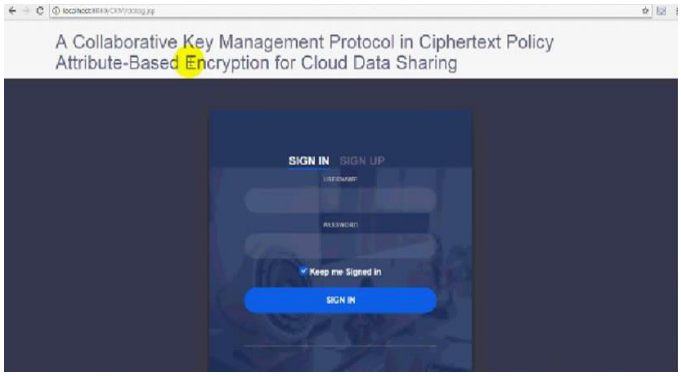
Key Generation Algorithm:

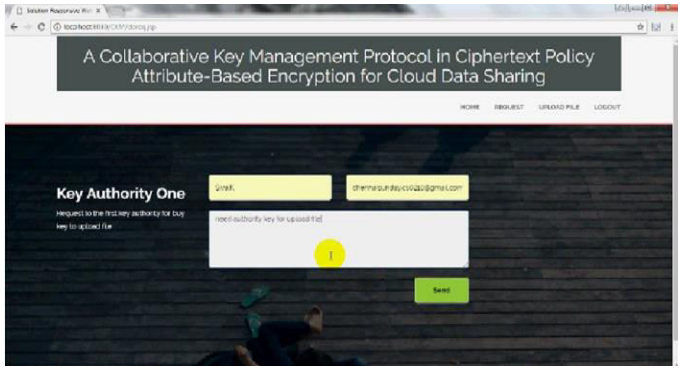
Key generation is the term used to describe the process of creating keys in cryptography. A key is required to unlock or decode any data. Any piece of software or program that can produce keys is called a key generator, or keygen. Starting from an unexpected (and frequently large and random) number, an asymmetric key method can create the right pair of keys. With an asymmetric key encryption method, everyone can encrypt messages using the public key, but only the owner of the matching private key can decipher them.

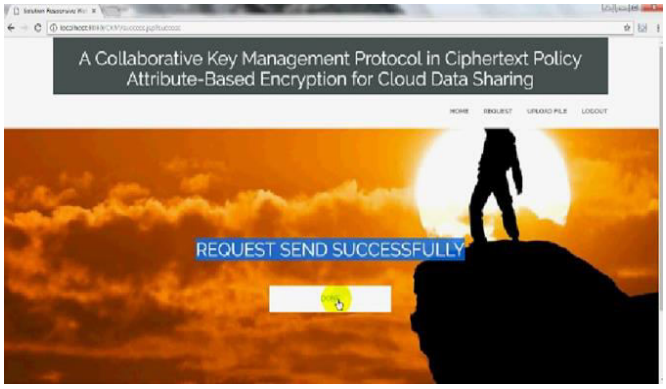
Encryption Algorithm:

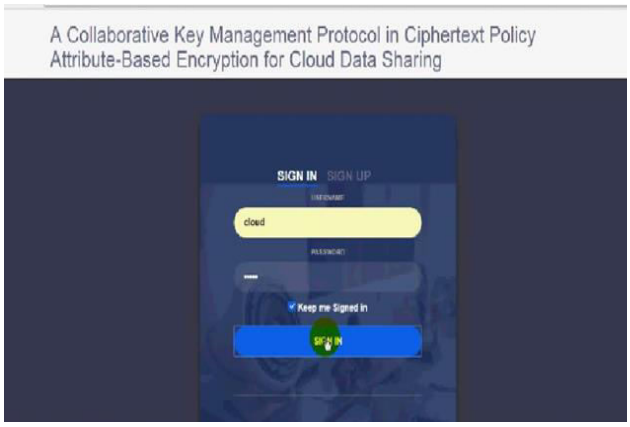
The purpose of our innovative CKM-CP-ABE collaborative key management protocol is to improve the efficiency and safety of key management in cloud-based data sharing platforms.

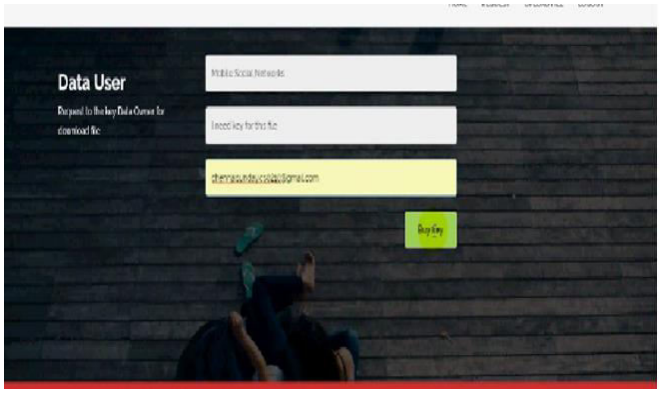
8. Results and Output Screenshot

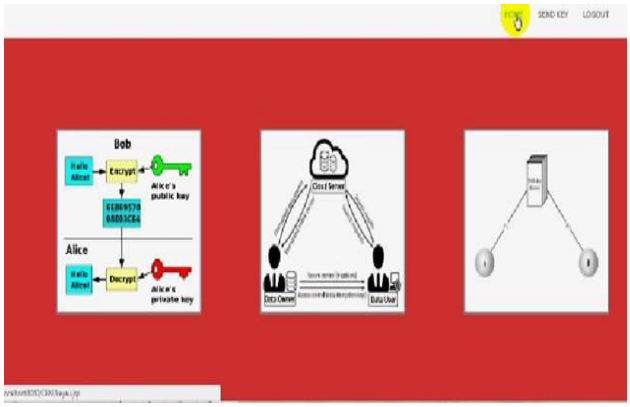


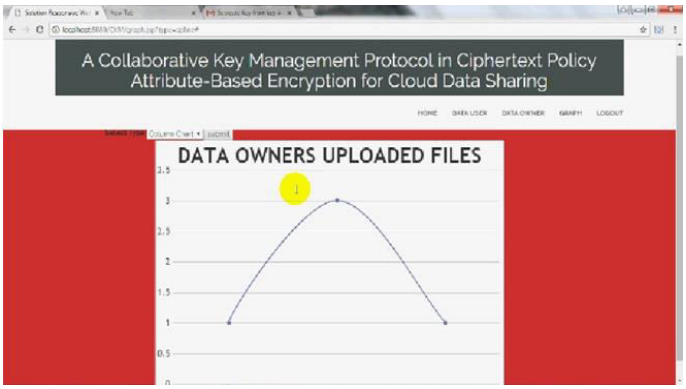


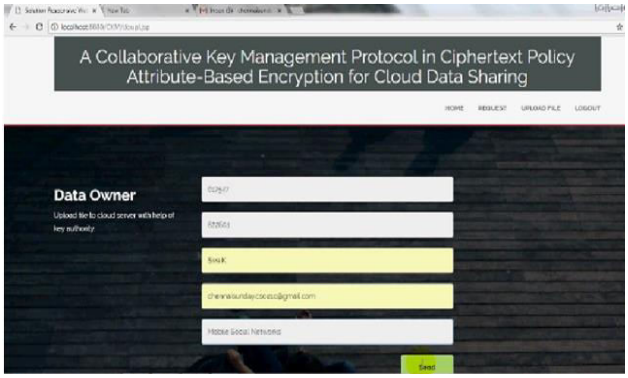












9. Conclusion

Cipher text policy attribute-based encryption is a cutting-edge cryptographic method that allows for granular control over who may access protected cloud storage. In this paper, we provide a novel collaborative key management protocol to enhance the efficacy and safety of key management in cloud data sharing systems that use cipher text policy attributes for encryption. Distributed key generation, issuance, and storage does not need supplementary physical infrastructure. We provide a private key updating technique and establish attribute groups for efficient and granular attribute revocation. Not only does the proposed collaborative system efficiently

manage key escrow, but it also tackles key exposure, a more pressing issue that has received less attention in previous research. As a result, the client has less work to do during decryption, which improves their experience overall. The proposed approach is thus more effective and secure in cloud data sharing systems that support several front-end devices with limited processing power. Improving the proposed system requires addressing three outstanding concerns with attribute-data sharing: reducing the size of the cipher text, the cost of encryption, and the cost of decryption. Building upon the preliminary findings of this study, we will work to address these issues.

REFERENCES:

[1]A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Proc. EuroCrypt, 2005, pp. 457-473.

[2] J. Bethencourt, A. Sahai, and B.Waters, "Ciphertext-policy attribute-based encryption," in Proc. IEEE Symp. Secur. Privacy, 2007, pp. 321-334.

[3] N. Attrapadung and H. Imai, "Conjunctive broadcast and attribute-based encryption," in Proc. Int. Conf. Pairing-Based Cryptography, 2009, pp. 248-265.

[4] B. Waters, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," in Proc. Public Key Cryptography, 2011, pp. 53-70.

[5] M. Green, S. Hohnberger, and B. Waters, "Outsourcing the decryption of ABE ciphertext," in Proc. USENIX Secur. Symp., 2011, pp. 34.

[6] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," IEEE Trans. Inf. Forens. Security, vol. 8, no. 8, pp. 1343-1354, 2013.

[7] S. Lin, R. Zhang, H. Ma, and M. Wang, "Revisiting attribute-based encryption with verifiable outsourced decryption," IEEE Trans. Inf. Forens. Security, vol. 10, no. 10, pp. 2119-2130, 2015.

[8] M. Chase, and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in Proc. ACM CCS, 2009, 121-130.

[9] G. Zhang, L. Liu, and Y. Liu, "An attribute-based encryption scheme secure against malicious KGC," in Proc. TRUSTCOM, 2012, pp. 1376-1380.

[10] J. Hur, "Improving security and efficiency in attribute-based data sharing," IEEE Trans. Knowl. Data. Eng., vol. 25, no. 10, pp. 2271-2282, 2013.

[11] P. P. Chandar, D. Mutkurman, and M. Rathinrai, "Hierarchical attribute based proxy reencryption access control in cloud computing," in Proc. ICCPCT, 2014, pp. 1565-1570