Cloud Based Missing Document Finder System

P. S. Lagad[1], Shaikh Taskiya[2], Pawase Snehal[3], Sonawane Roshni[4]

[1]Professor, Dept. of Cloud Computing and Big Data, P.Dr.V.V.P. Institute of Technology and Engineering, Loni,
Maharashtra, India Author Department & College
[2,3,4] Final year Diploma Student, P.Dr.V.V.P. Institute of Technology and Engineering, Loni, Maharashtra, India

---***---

Abstract - An intelligent solution, the Cloud-Based Missing Document Finder System makes use of cloud computing and secure key generation techniques to efficiently help customers find misplaced documents. In this system, users who have misplaced papers can submit the specifics of those documents in the first person. An agent then processes the information and uses the Message Digest (MDS) algorithm to build a unique secret key. Users who discover misplaced documents (Second Person) also generate a new secret key by inputting the contents of the document they found. A secure cloud platform is used to match these keys within the system's database. When two users are found to be a good fit, the algorithm will decide to send them an email to securely connect them. Scalability, data security, and help with document recovery in real-time are all guaranteed by this cloud-based method. The system streamlines processes, protects user data, and offers a solid platform to help owners recover misplaced documents.

***Key Words*:** Cloud Computing, Missing Document Detection, Document Recovery System,Secret Key Generation, Message Digest Algorithm (MDS)

1.INTRODUCTION

The digital age makes it all too easy to lose track of important documents such as identification cards, certifications, or legal paperwork. This is especially true in today's world. Using traditional methods to locate papers that have been misplaced is typically a process that is both time-consuming and imprecise. This problem can be solved by using the Cloud-Based Missing Document Finder System, which offers a secure and efficient method of connecting individuals who have misplaced papers with other individuals who have located them.The employment of cloud computing for the purpose of centralized data storage and accessibility enables users to submit or retrieve information from any location at any time. This is made possible by the system's utilization of cloud computing. The algorithm known as Message Digest (MDS) is responsible for transforming each document entry into a one-of-a-kind secret key in order to safeguard sensitive information. These keys are utilized by the system in order to determine the likely associations between the document and its owner through the use of an automated decision-making process. After a match has been confirmed, both parties receive notifications via encrypted email, which enables them to safely communicate with one another and return documents.

This technology integrates intelligent key-matching with cloud-based storage, which not only simplifies the recovery process but also ensures the secrecy of data, ensures its accuracy, and in stills confidence in users. People are able to retrieve their misplaced documents in a secure and comfortable manner with the assistance of the missing document finder system that is hosted inside the cloud. Cloud computing is utilized for the purposes of document recovery and storage by this system. In order to achieve the goals of secrecy and matching, the MDS approach is utilized to convert each document into a secret key. If a match is found, the technology will immediately begin establishing a safe connection between the person who misplaced the document and the person who was able to retrieve it. This technique makes the process more reliable, secure, and expedient as compared to other approaches that are more typical.

The technology behind blockchain is the foundation upon which other upcoming cryptocurrencies, including Bitcoin, are constructed. The decentralized nature of blockchain technology is the primary selling point of this technology. In addition to facilitating decentralized peer-to-peer (P2P) transactions, it can also promote distributed system coordination and cooperation. This is accomplished without the requirement for trusted third parties or centralized control. Information encryption, time-stamping, distributed consensus algorithms, and financial incentive systems are some of the methods that are utilized in order to accomplish this goal. The business has been hampered for a long time by the high operational expenses, poor performance, and probable security dangers that are associated with statistics garage in typical centralized organizations. However, blockchain technology presents a fresh solution to addressing these issues.

[1] Sunil Kumar et al. proposed that Users without technical knowledge can nevertheless benefit from cloud computing and other information technology services. Data can be stored, managed, improved, and accessed remotely from any location with the help of third-party cloud service providers. Customers who use cloud services have a lot of choices. A user is any individual who utilizes a service offered by a cloud provider. Because of the affordable prices of the services, many people are able to access their data from anywhere. When you use cloud services, you won't need to carry your device about; instead, you can access them from any location. Cloud services aren't perfect; for example, there isn't enough protection for sensitive data, which means that providers of such services will need to come up with creative solutions. In order to simplify the activities involved in key creation, ECC is used. The small-key size of ECC makes its improvement better than other cryptographic techniques. Data optimization and security can be greatly enhanced when AES is used in combination with ECC. The necessity for extra security measures, including cryptographic methods, is rising in tandem with the popularity of cloud computing. Future improvements to the hybrid approach can focus on making it more secure. Increasing the number of security measures in the system will increase its efficiency and production.

[2] Ishu Gupta et al. addressed that the area of cloud computing and information security presents a formidable challenge when it comes to data protection. A mountain of research is analyzed to lessen the impact of this obstacle. Unfortunately, there isn't enough time to thoroughly examine all of the ongoing options. With this in mind, the study analyzed the state-of-the-art in cloud data security extensively, discussing the most important techniques for securing data sharing and the solutions that are now available. The most important and relevant details needed to understand the method's core, as well as any gaps in the current research and potential directions for future solutions, are brought to light. In addition, we compare and contrast all of the refereed methodologies and conduct a comprehensive study. Each method's applicability is assessed according to the situation. According to the research, no method can guarantee the complete safety of the data from all involved parties in the system on its own. By incorporating methods for offering comprehensive system security in a shared setting, a solid solution can be created. Furthermore, the highlighted analysis is expected to serve as a benchmark for future researchers in the field and other developing applications that require secure data storage and exchange. This is due to the exceptional solutions that have been handled.

[3] Sixu Guo et al. introduced that issue, the RBDC system is suggested as a solution because the majority of currently available searchable encryption schemes do not permit multi-keyword Boolean searches. The technique uses GM and Paillier encryption algorithms to create encrypted secure indices, building on traditional searchable encryption schemes. These indices have good search efficiency and storage efficiency. Next, in order to accomplish multi-keyword Boolean searches, encrypted indices of both individual keywords and keyword intersections are built in accordance with the principles of set theory. The search results are ordered by the third-party entity SCP, and the forward score indices are built using TF-IDF. By the same token, the approach can enhance the efficiency of numerous keywords while simultaneously updating them dynamically. Analysis of the algorithm's security features subsequently reveals that it is capable of fending off two distinct security threats. Last but not least, comparisons with other searchable encryption systems and analyses of its functions and performance have shown the RBDC scheme's superiority.

The second section of this book presents an analysis of prior research classified as a Literature Survey. Section 3, under "Proposed Methodology," offers a detailed account of the suggested technique. Section 4 explores the experimental assessment, Section 5 considers potential improvements, and Section 6 culminates the essay with a summary of the current plan.

2. LITERATURE SURVEY

[4] Somchart Fugkeaw et al. proposed a Secure single sign-on (SSO) authentication, dynamic authorization, and preventive-based access control with accountability in the cloud are all supported by a blockchain-based system known as the D2-IAM system. By utilizing smart contracts and blockchain technology, our solution was able to optimize the cost of the SSO-authorization and authentication process. Aside from accomplishing very efficient authentication and authorization, the document database also models the access policy, making it easy to maintain. Its content is

also guaranteed to be confidential thanks to the public key encryption. We present the results of the efficiency study and the experimental data that prove D2-IAM is practical and performs better than previous works. In order to ensure the security of cloud-based access policies, an auditing protocol must be developed for use in subsequent projects. Encryption does not eliminate the need to guarantee the policies' integrity. It might be wise to investigate the public cloud auditing methods [43], [44]. Also, instead of using the general cloud storage, policies and personally identifiable information databases can be saved on decentralized storage platforms like Inter Planetary File System (IPFS), which offers better file management and data indexing. Lastly, it's worthwhile to create a machine learning-based anomaly detection mechanism to spot attacks on the authentication protocol or the abuse of the SSO authentication ticket.

[5] Fang Xiao et al. studied that proposes a file prefetching framework based on file correlation for cloud storage data access. Using the skip-gram model to train the file correlation vectors and then establishing a file index table, the framework dynamically adjusts the allocation of cache space during prefetching. In the context of cloud storage environment, our approach aims to accelerate the access speed of cloud storage by enhancing the accuracy of predicting prefetched files and improving the efficiency of prefetching. As for the web environment, our method leverages user access data to enhance the efficiency of server-side prefetching, thereby improving the response speed of the web environment. Through actual data testing, the algorithm outperforms LRU and ARC algorithms, which serve as the baseline model in hit rate and maintains a high prediction accuracy even with more extended access intervals and smaller training data sizes.

[6] Jawad Sadek et al. studied that to establish the first resource for finding plant specimens in the Sloane Herbarium by the species name used by John Ray, and a procedure to extract data from Hans Sloane's copy of Ray's Historia Plantarum. At present, there is no digitally searchable version of the data included in Historia Plantarum, which is the principal taxonomic index to the Sloane Herbarium. Consequently, there is a chance that a dramatic improvement in data accessibility might result from the mobilization of these datasets. In order to accomplish the goals of this research, it is necessary to record three kinds of information: (i) the names of plants that appear in the text itself; (ii) references that are hand-written and can be found in the margins, footers, and heads of the document; and (iii) the relationships between the names of plants and the annotations that are associated with their specimens. Existing optical character recognition (OCR) and human-to-human text recognition (HTR) systems are unable on their own to meet the NHM's stringent standards for accurately identifying handwritten numerals, which makes a bespoke solution necessary.

[7] Jannatun Noor et al. studied that combines OpenStack Swift with machine learning capabilities to create a better search solution. We accomplish comprehensive system design by utilizing Elasticsearch. Our main goal is to improve Swift's search functionality. However, we also want to build a content-based image searching system that is user-centric and uses a text-based database [69]. Users can tailor the YoLOv4 and YoLOv8 algorithms to their liking in this setup, and neither Swift storage nor the Elasticsearch cluster will suffer as a result, because they function autonomously. Users have a variety of options to fit their needs with YOLOv4 and YOLOv8, which can detect objects in both pictures and live video feeds. Search methods based on metadata at the content level have not been thoroughly investigated in the OpenStack Swift literature. Thus, our study involves integrating a framework for object recognition and an Elasticsearch cluster with our Swift storage system. We put our model through its paces in order to see how well it works and how responsive it is. Although we are able to accomplish our goals with little delay, there is still potential for enhancement.

[8] Xiaojie Zhu et al. proposed that a parallel, efficient, verifiable, responsible, and privacy-preserving solution to the challenge of keyword search in a multitenant cloud. This was accomplished by creating a privacy-preserving inverted index that allowed for a search of verifiable ciphertext. The compressed MAC for all related documents, along with encrypted keyword and document identity pairs, are included in each entry. Next, we developed a method for fine-grained access control using token generation based on keywords. Additionally, in order to achieve user responsibility, we incorporated the user's identity within the token. The VAKSE scheme already included all of those components. We introduced the PVAKSE to further enhance search efficiency; it partitions the inverted index into small, simultaneously searchable chunks. Our proposed techniques were formally tested for security flaws, and we ran comprehensive tests to prove their

efficacy. We plan to significantly improve PVAKSE's security and performance in future work.

[9] Jing Wen et al. cryptanalyzed the discovered two security flaws in existing KASE schemes: offline keyword guessing attacks and permission abuse. In order to cryptanalyze several current KASE schemes, we first use the known keyword guessing attack methods for the former attacks. Additionally, we provide two unique approaches to keyword guessing attacks: (1) attack using modified ciphertext and (2) attack using verification equation. Following this, we provide two brand-new attack approaches that rely on keyword guessing and include cryptanalysis examples of each. Before moving on to the second set of attacks, we cryptanalyzed a number of preexisting KASE schemes using the well-established techniques for authorization abuse attacks. Additionally, we come up with a new way to attack where the hacker can have better search capabilities without coordinating with other authorized users—they can just update their own authorization individually. Next, we provide real-world examples of cryptanalysis applied to the new technique of authorization abuse attack. We are hopeful that future cryptography researchers will be able to use our study to create better secure KASE schemes.

[10] Salah T. Alshammari et al. aimed to find a way to address trust difficulties in models of access control in order to lower risk. An improvement in the decision-making process for both cloud operators and data owners could result from a more secure cloud storage system. This paper introduced a vocabulary for trust criteria and reputation attacks in cloud computing. In addition to outlining the most current technology, the paper lays forth some important ideas concerning the management of trust for services in the cloud. We used the three-layer structure of trust models, with their respective sets of dimensions, to evaluate several research prototypes of trust models in cloud computing settings. These models served as our standards and evaluation criteria. In addition, we showed how to construct a reliable and flexible cloud storage system based on trust. In this system, resource owners can assign responsibilities and tasks, and if data leaks, the system would stop the job or task immediately. Cloud operators and data owners may now make better decisions thanks to our model, which incorporates T-RBAC, a new type of access control, with a strong trust model. This model effectively reduces threats and provides excellent safety for cloud storage systems.

[11] Chunwei Lou et al. introduced a cloud-assisted Internet of Things applications using a safe key-aggregate keyword search strategy over encrypted data. Our suggested method allows users to safely transfer their search capabilities across several encrypted documents to a remote server in the cloud without exposing their keyword or trap door privacy. Finally, we demonstrated the model's security in the standard model and explicitly assigned it the four specified primitives. We show that our proposed method is effective and applicable to real-world scenarios by comparing it to existing theories and conducting comprehensive experiments. Looking ahead, our primary focus will be on incorporating forward security and verifiable search into our established architecture.

[12] Ahmadakmalaminuddin Mohdkamal et al. introduced two approaches to efficient and safe searchable encryption that combine secure computing with a secret sharing scheme with a (k,n) threshold and tools to regulate user access. By design, our system ensures that no one other than the data owner may execute legitimate search queries and operations. We also suggested a query-generation algorithm that uses a secure computation approach to randomize search queries, protecting users' search pattern privacy. In addition, we compared our suggested methods to other traditional secret sharing-based approaches and proved that they can be efficiently executed in Python within a reasonable amount of time. Because of this, our solutions can be easily implemented in setups with several servers, like cloud storage as a service. A logical search mechanism for multi-keyword searches and an improved way to provide a verification function to safeguard against hostile adversaries are our goals for future research. In addition, we will offer a more robust security framework for our approach after conducting a thorough survey of current adversarial attacks in areas like machine learning. This framework will successfully counter both overt and covert adversaries, including those who are honest but curious. Secure cloud computing environments are also placing a premium on the zero trust idea, which aims to accommodate the intricacies of the contemporary world by including the mobile workforce. In our upcoming research, we aim to investigate efficient fine-grained access control via secure computation, develop access control with stronger authentication and authorization that is cloud-deployable, and ensure that applications like IoT data access systems, which require complex layered access

management, are adequately protected. Also, we will think about doing a complete real/ideal security analysis using Curtmola et al.'s security framework.

[13] Franziska Jurosch et al. introduces that Here, we introduce a novel MTMCT architecture that can recognize the post-operative period, generate time stamps automatically, and estimate locations. Nineteen reenacted postoperative patient flows with over 150,000 frames were used to train a MOT algorithm for each camera.The object categories of operating room tables and care beds achieved good tracking results, whereas patients achieved moderate tracking outcomes. Despite this, in multi-patient scenarios, the whole architecture produced impressive outcomes for postoperative phase recognition (84.9±5.9%), timestamp generation (91.4±1.5%), and patient localization (92.0±3.6%).As a result, it improves clinical documentation and offers a foundation for real-time physician support, which in turn enhances patient care.Extending successful concepts from intraoperative cases to postoperative processes and developing whole new techniques to fully harness the benefits of camera-based solutions in postoperative settings are necessary steps beyond continuing optimization of this approach for the future.

[14] Tom Landman et al. trusted FL-based paradigm for the identification of unknown malware in Linux-based cloud settings. Our suggested methodology makes use of virtualization technology by reliably capturing volatile memory dumps over time through the use of various hypervisors connected to distinct virtual server instances. With the use of a hybrid analysis approach that combines dynamic and static analyses, our system is able to detect unknown malware with more generalizability. The first step is to dynamically obtain volatile memory dumps from all of the server community's virtual servers. The obtained volatile memory dumps show how each virtual server's programs really work and what their actual nature is. The next step is to transform the memory dumps into RGB pictures. By employing this process, the size is drastically reduced by around 99.99% (i.e., going from a 1.1 GB memory dump to a 20 KB visual image), which in turn reduces the storage and memory needs. One more perk of the visual transformation is that it can be used with convolutional neural networks (CNNs) to learn a discriminative function that can distinguish between benign and malicious files, automatically extracting their features and eliminating the need for cyber experts. The system's horizontal FL-based mechanism is its backbone; it lets the server community work together to learn cloud malware detection without exposing server privacy, regardless of whether they're dealing with benign or harmful families of malware. By decentralizing the learning process across multiple servers rather than relying on a single master server, FL improves system security by making it more difficult for attackers to compromise the system or cause it to fail at a critical point; doing so would necessitate not only additional knowledge but also substantial computational resources. Our CNN-based detector, MobileNetV2, is optimized for low-memory edge devices and boasts an advanced architecture with residual units, depth-wise and point-wise convolutions, etc., which allows it to efficiently perform detection and classification tasks. Due to the previously stated isolation of the virtual machine servers, the converted images derived from the volatile memory dumps obtained from each server do not include any information.Your name. Consequently, federated learning is a desirable option since each virtual server may meet apps that are either harmless or harmful, belonging to distinct families of malware. This could lead to an uneven distribution. Three experiments were conducted to evaluate the two types of servers using volatile memory dumps obtained during the execution of 112 benign and malicious applications. The trials were designed to be of varied degrees of complexity.A 5-fold cross-validation procedure was employed throughout the studies. First, we compared a centralized model to our suggested FL-based detection framework's performance in detecting unknown Linux malware in a federated setting, employing a decentralized community of ten homogeneous servers. In contrast to the centralized HTTP model, the federated model achieved an AUC of 98.3% in the HTTP scenario, leading to a 1.7% loss in δ-AUC. Compared to the centralized DNS model, the federated global model achieved an AUC of 93.2% in the DNS scenario, resulting in a 4.9% loss of δ-AUC. Regardless of the discrepancy, the results showed that the federated approach could safeguard the servers' confidentiality while reliably detecting hidden malware on both of them.

[15] Yuanchao Chen et al. addressed that An increasing number of websites are utilizing cloud services to store their files. One of the most common uses for cloud storage is the direct upload of user files. We take a close look at the potential security issues with this scenario in this study. In this scenario, we thoroughly examine the three crucial steps: downloading

credentials, verifying uploaded files, and receiving and responding to callback notifications. From the analysis, we can deduce that this scenario introduces six new types of vulnerabilities: unrestricted upload credential acquisition (V1), upload credentials validity flaw (V2), unrestricted file types and size (V3), file overwriting (V4), file stealing (V5), and callback notification spoofing (V6). Web services, consumers, and cloud storage services are all at risk of severe security breaches due to these flaws. We do a measurement of the existing prominent websites that employ public cloud services to explore the real-world implications caused by these vulnerabilities. To everyone's surprise, every single one of the evaluated websites has a security hole. During the measurement, we discovered 79 new vulnerabilities. We promptly reported them to the relevant security teams or communities, and some of them have responded positively. We are hopeful that our results can inform studies regarding the safety of cloud storage in the future.

3. METHODOLOGY

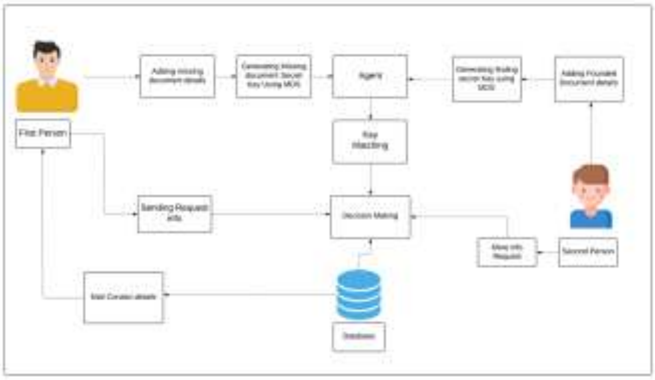
Adding missing document details
Generating missing document Secret Key Using MDS
Agent
Generating finding secret Key using MDS
Adding Founded Document details
First Person
Key Matching
Sending Request info
Decision Making
Second Person
Database

Fig 1: Overview Diagram

Figure 1 illustrates the comprehensive workflow of the proposed cloud-based document management system for missing and discovered items, utilizing a secure key-matching method. The methodology is structured to effectively discover and correlate missing documents with located documents via a centralized cloud architecture. The following steps describe the detailed workflow used to implement the proposed system

Step 1: User Registration and System Access

The process begins with the registration of Person 1 in the system by providing essential personal details such as name, email, and contact information. After successful registration, the user logs into the application. Upon login, the main dashboard is displayed, which includes four modules: Manage Profile, Missing Document, Found Document, and Exit. The Manage Profile module allows users to edit and update their personal information, ensuring data accuracy throughout the system.

Step 2: Reporting a Missing Document

Person 1 navigates to the Missing Document module and selects the option to add a missing document. The user enters all relevant details such as document type (e.g., Aadhaar card, PAN card), document number, and identifying information. Once submitted, the data is securely processed and prepared for storage.

Step 3: Data Storage using AWS RDS (SQL Database)

All user and document data are stored in a cloud-based AWS RDS (Relational Database Service) using an SQL database. AWS RDS ensures high availability, scalability, and data security. The missing document details entered by the user are stored in structured tables, enabling efficient querying and retrieval. Along with storage, a unique Missing Document Secret (MDS) key is generated using the document attributes, which acts as a secure identifier for matching purposes.

Step 4: Viewing and Managing Records

The system allows users to view previously submitted missing documents through the View Missing Document option. All records are fetched dynamically from the AWS RDS database, ensuring real-time access and consistency.

Step 5: Second User Registration (Finder)

Person 2, who has found a document, registers and logs into the system. After accessing the dashboard, the user navigates to the Found Document module to report the found document.

Step 6: Reporting a Found Document

Person 2 enters the details of the found document, similar to the missing document entry process. The system stores this data in the AWS RDS SQL database and generates a corresponding MDS key using the same logic. This ensures uniformity in the matching process.

Step 7: Key Matching and Decision Making

The system performs an automated comparison between the MDS key of the found document and the keys of all missing documents stored in the AWS RDS database. This matching process is handled by the decision-making module, which ensures accuracy and efficiency. If a match is found, the system confirms that the found document belongs to Person 1.

Step 8: Notification and Communication

Once a successful match is identified, the system sends automated email notifications to both users. Person 1 is informed that their missing document has been found, while Person 2 is notified about the rightful owner of the document. Relevant contact details may also be shared to facilitate communication and document return.

Step 9: Data Security and Reliability

Using AWS RDS ensures that all stored data is secure, encrypted, and backed up regularly. The use of MDS keys enhances privacy by avoiding direct exposure of sensitive document details during the matching process. The system maintains high reliability and performance due to cloud-based infrastructure.

4. RESULT AND DISCUSSION

The proposed method makes use of the Java programming language and the NetBeans IDE to develop a reliable Cloud-Based Missing Document Management System. The development environment consists of a laptop with the Microsoft Windows operating system, powered by an Intel Core i5 processor, 8 GB RAM, and 500 GB internal storage. For database management, the system utilizes Amazon RDS to securely store and manage user details, missing document records, found document reports, and secret matching keys generated for document verification.

The effectiveness of the proposed system has been evaluated under different operating conditions to ensure reliability, efficiency, and secure cloud storage. The experimental results obtained from the implementation of the system are discussed below.

Scalability Analysis of Cloud Database Transactions

The scalability of data processing in the proposed Cloud-Based Missing Document Management System is analyzed by evaluating database transaction performance in the cloud environment. The system allows multiple users to access the platform, where users can register, report missing documents, report found documents, and view document records through the system interface.

The system stores various types of information such as missing document details, found document reports, user registration information, and generated Missing Document Secret (MDS) keys in the cloud database. The transaction density and the time required for processing database operations are recorded and analyzed to measure system performance. The summarized results of these database transactions are presented in Table 1.

S. No	No. of Database Transactions	Time Taken (in Seconds)
1	250	0.48
2	520	0.81
3	780	1.19
4	960	1.54
5	1180	1.86
6	1420	2.14

Table 1: Cloud-Based Missing Document System Transaction Time Estimation Table

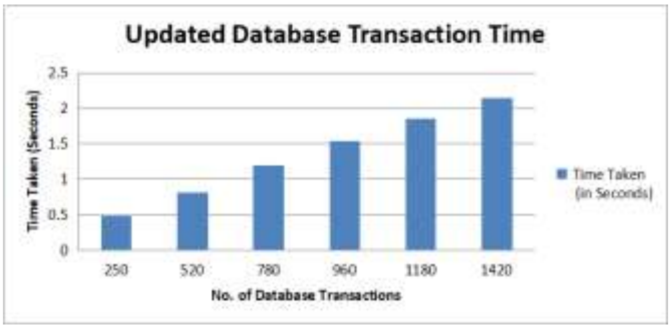

Figure 2: Cloud Database Transactions

The graph shown in Figure 2 is generated using the data from Table 1. The graphical representation illustrates the relationship between different system operations and the time required to complete them in the cloud database environment.

This analysis helps in understanding the efficiency of cloud-based storage and retrieval mechanisms used in the proposed system. The results show that database operations such as user registration, missing document submission, found document reporting, document retrieval, and secret key matching are performed efficiently with minimal processing time.

The findings demonstrate that the cloud database infrastructure is effectively utilized for handling missing document reporting and verification transactions. This improves the system's reliability, scalability, and data security while managing large volumes of document-related information.

5. CONCLUSIONS

The use of cloud storage to house website files is on the rise. Direct user file uploads are among the most popular uses for cloud storage. In this paper, we examine the possible security concerns with this scenario in detail. Downloading credentials, checking uploaded files, and receiving and responding to callback messages are the three critical stages that we analyse in detail in this case. The investigation led us to conclude that six new vulnerabilities are introduced by this scenario: unrestricted upload credential acquisition (V1), upload credentials validity flaw (V2), unrestricted file types and sizes (V3), file overwriting (V4), file stealing (V5), and callback notification spoofing (V6). Because of these vulnerabilities, cloud storage services, customers, and web services are all vulnerable to serious security breaches. In order to investigate the practical consequences of these vulnerabilities, we conduct a measurement of the most popular websites that now use public cloud services. Surprisingly, there is a security flaw in every single one of the tested websites. In the course of the evaluation, we uncovered 79 more security holes. They were reported to the appropriate security teams or communities right away, and we've heard back from a few of them. We are optimistic that future research on the security of cloud storage can be informed by our findings.

Future work can focus on designing secure-by-default cloud upload frameworks that eliminate the identified vulnerabilities through strict credential scoping, time-bound access tokens, and robust server-side validation. Advanced file inspection mechanisms using machine learning can be integrated to automatically detect malicious file types, abnormal file sizes, and overwrite attempts in real time. The callback notification process can be strengthened using cryptographic verification and mutual authentication to prevent spoofing attacks. Additionally, large-scale automated security testing tools can be developed to continuously monitor cloud-based file upload implementations across diverse web applications. Extending the study to emerging cloud platforms, edge computing environments, and multi-cloud architectures will further enhance the understanding and mitigation of cloud storage security risks.

REFERENCES

[1] S. Kumar and D. Kumar, "Securing of Cloud Storage Data Using Hybrid AES-ECC Cryptographic Approach," *Journal of Mobile Multimedia*, vol. 19, no. 2, pp. 363–388, 2022. doi: 10.13052/jmm1550-4646.1921.

[2] Gupta, A. K. Singh, C. -N. Lee and R. Buyya, "Secure Data Storage and Sharing Techniques for Data Protection in Cloud Environments: A Systematic Review, Analysis, and Future Directions," in *IEEE Access*, vol. 10, pp. 71247-71277, 2022, doi: 10.1109/ACCESS.2022.3188110.

[3] S. Guo, H. Geng, L. Su, S. He and X. Zhang, "A Rankable Boolean Searchable Encryption Scheme Supporting Dynamic Updates in a Cloud Environment," in *IEEE Access*, vol. 11, pp. 63475-63486, 2023, doi: 10.1109/ACCESS.2023.3284904.

[4] S. Fugkeaw, "Achieving Decentralized and Dynamic SSO-Identity Access Management System for Multi-Application Outsourced in Cloud," in *IEEE Access*, vol. 11, pp. 25480-25491, 2023, doi: 10.1109/ACCESS.2023.3255885.

[5] F. Xiao, S. Yu, and Y. Li, "Efficient Large-Capacity Caching in Cloud Storage Using Skip-Gram-Based File Correlation Analysis," in *IEEE Access*, vol. 11, pp. 111265-111273, 2023, doi: 10.1109/ACCESS.2023.3322725.

[6] Q. Groom, T. van Dooren, S. J. R. de Lame, E. Steeman, and M. Stefanaki, "Leveraging OCR and HTR cloud services towards data mobilisation of historical plant names," *Biodiversity Data Journal*, vol. 12, p. e115797, Mar. 2024, doi: 10.3897/BDJ.12.e115797.

[7] J. Noor *et al.*, "Sherlock in OSS: A Novel Approach of Content-Based Searching in Object Storage System," in *IEEE Access*, vol. 12, pp. 69456-69474, 2024, doi: 10.1109/ACCESS.2024.3401074.

[8] X. Zhu, P. Shen, Y. Dai, L. Xu and J. Hu, "Privacy-Preserving and Trusted Keyword Search for Multi-Tenancy Cloud," in *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 4316-4330, 2024, doi: 10.1109/TIFS.2024.3377549.

[9] B. Chen, L. Wu, H. Wang, L. Zhou and D. He, "On the Security of Key-Aggregate Searchable Encryption," in *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 1,

pp. 696-708, Jan.-Feb. 2023, doi: 10.1109/TDSC.2021.3135500.

[10] S. T. Alshammari, M. Al-Razgan, T. Alfakih and K. A. AlGhamdi, "Building a Comprehensive Trust Evaluation Model to Secure Cloud Services From Reputation Attacks," in *IEEE Access*, vol. 12, pp. 150754-150775, 2024, doi: 10.1109/ACCESS.2024.3471337.

[11] C. Lou *et al*., "A Secure Key-Aggregate Keyword Retrieval Scheme Over Encrypted Data in Cloud Computing," in *IEEE Access*, vol. 13, pp. 123429-123439, 2025, doi: 10.1109/ACCESS.2020.2980886

[12] A. A. Aminuddin Mohd Kamal, M. Okada and M. Fujisawa, "Privacy-Preserving Keyword Search With Access Control for Secret Sharing-Based Data Outsourcing," in *IEEE Access*, vol. 13, pp. 73625-73651, 2025, doi: 10.1109/ACCESS.2025.3562667.

[13] F. Itzel, J. -L. Stoll, J. -C. Rosenthal, P. Eisert and T. Neumuth, "Video-based multi-target multi-camera tracking for postoperative phase recognition," in *International Journal of Computer Assisted Radiology and Surgery*, vol. 18, no. 6, pp. 1105-1113, Jun. 2023, doi: 10.1007/s11548-023-02874-y.

[14] T. Landman and N. Nissim, "Securing Linux Cloud Environments: Privacy-Aware Federated Learning Framework for Advanced Malware Detection in Linux Clouds," in *IEEE Access*, vol. 13, pp. 30377-30394, 2025, doi: 10.1109/ACCESS.2025.3540955.

[15] Y. Chen *et al*., "Understanding the Security Risks of Websites Using Cloud Storage for Direct User File Uploads," in *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 2677-2692, 2025, doi: 10.1109/TIFS.2025.3544082