Multi Access Filtering using FOG Enivronment

[1]Gowtham S, [2]Mohammed Abdullah S, [3]Asha J, [4]Nawin Ram Shanker I and [5]Selwin Isac Neilsingh J
Sri Eshwar College of Engineering, Kinathukadavu, Coimbatore, India.
[1]gowtham.scse2020lateral@sece.ac.in, [2]mohammedabdullah2020cselateral@sece.ac.in, [3]asha.j@sece.ac.in,
[4]nawinramshanker2020cselateral@sece.ac.in, [5]selwinisacneilsingh.jcse@sece.ac.in

Article Info
A. Haldorai et al. (eds.), *2nd International Conference on Materials Science and Sustainable Manufacturing Technology,* Advances in Computational Intelligence in Materials Science.
Doi: https://doi.org/10.53759/acims/978-9914-9946-9-8_14
©2023 The Authors. Published by AnaPub Publications.
This is an open access article under the CC BY-NC-ND license. (http://creativecommons.org/licenses/by-nc-nd/4.0/)

Abstract—Fog computing is gaining popularity, even in traditionally conservative and delicate sectors like the Armed forces and nations.The interconnection of our community and technology breakthroughs like the IOT technology both have an impact on this. However, one of many crucial factors in fog computing adoption is guarding against privacy leaks. As a result, we present a multi-layer access filtration model in this study that safeguards privacy and was designed for such fog-based cloud environment, thus the name (FAF)fog-based access filter. The three major algorithms that make up FAF are the tuple reduction algorithm, the optimal privacy-energy-time methodology, and the access filter initialization method. The different protective aims are also distinguished using a hierarchical classification. Our experimental evaluation's conclusions show that FAF enables one to strike the perfect balance between privacy protection and computing costs.

Keywords—Cloud Computing, Fog Computing, Privacy, Security, Server, Resource, Performance, Upload, Access.

I. INTRODUCTİON

The rise of cloud computing has greatly enhanced the IT sector. It is among the most key techniques now in use and provides several benefits to any business or individual. It offers users more enticing data storage, on-demand services, all-encompassing network access, and practical data sharing. Regardless of local infrastructure limits, cloud computing provides data storage solutions and gives customers a platform to handle their data. This technology's major goal is to increase the efficiency of shared resources that are available in the cloud and to dynamically reallocate them as needed. Despite the technology's innumerable advantages, it lacks robust security features and compromises data owners' privacy. The main focus of traditional security measures is user authentication to verify that users only access their own data fields. When cloud users need to exchange and access one other's allowed information to reap the benefits, additional problems emerge in addition to the security-related ones. When that comes to storage outsourcing, one approach to protect the integrity of the information is to have verified data custody. In this study, we talk about the creation of a powerful PDP method for distributed cloud services to help with service scaling including data on migration, where we, take into account for the existence of several cloud service providers in order to jointly manage and store client data. We outline a technique called cooperative PDP (CPDP) that relies on a hierarchy of hash indexes and homomorphic verifiable responses. We use a multiple-prover zero-knowledge proof mechanism to show that our approach is secure. Comprehensiveness, information soundness, and zero-knowledge requirements can all be met by this system. In addition, we specify techniques for improving the performance of our approach.

In order to lower computation costs for the both clients and storage service providers, we present a quick method for selecting the best parameter values. Our testing shows that our technique results in lower computing and communication overheads when compared to non-cooperative approaches.

Several cloud components often communicate with one another in Cloud Infrastructure, the structure of the software products used to supply cloud computing, that used a loosely connected method like a messaging queue. The following essential traits of cloud computing are present:

Agility is increased by consumers' ability to replenish resources in the technical infrastructure. The Application programming interface (API) accessibility to software enables machines to communicate with cloud-based applications in a manner similar to how the user interface promotes communication between people and computers. As according claims that the cost will reduce, capital expenditure is converted to operating expenditure in a public cloud platform delivery model. Users can utilize a web browser to access systems no matter where they are or what device they are using thanks to device and location independence (e.g., PC, mobile phone). Increased usage is made possible through virtualization technology, which enables sharing of servers and storage resources. From one physical server to another, applications can be moved with ease. Resource and cost sharing among a wide group of users is made possible by multitenancy.

Multiple redundant sites boost reliability, making well-designed cloud computing appropriate for business continuity and catastrophe recovery. Scalability and stretchability are accomplished through flexible ("on-demand") resource provisioning on a very good, self-service basis almost in real-time, eliminating the need for users to prepare for peak needs. Performance is tracked, and web services are used as the computer interface to build consistent, loosely coupled systems. Data centralization, more resources devoted to security, and other factors may strengthen security, however concerns about losing control over sensitive data and unsecured kernel storage may still exist. Security is frequently on par with or superior to other conventional methods. As they can be accessed from numerous places and do not need to be loaded on each user's computer, programs for cloud technology require less maintenance. **Fig 1** shows cloud computing.

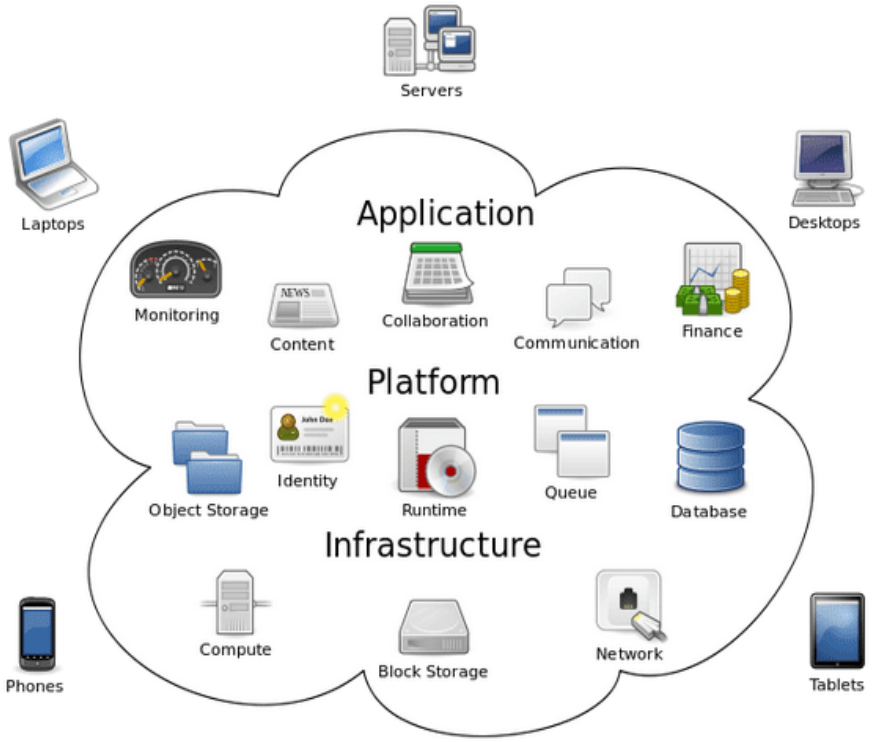

Fig 1. Cloud Computing.

II. LİTERATURE SURVEY

Cloud computing [1] is the anticipated upcoming architecture for IT businesses. The databases and software applications are moved to centralized, sizable data centers, where the information and service administration may not be completely reliable. Many new security concerns are presented by this distinct paradigm, many of which are poorly understood. This article examines the problem of maintaining data integrity when employing cloud computing. The major objective of our work is to enable third-party auditors (TPAs) to certify the integrity of dynamic data stored in the cloud on behalf of cloud clients. The introduction of TPA removes the client's involvement by verifying whether his cloud-stored data is indeed intact. Since cloud computing services are not just for archiving or backing up data, supporting data dynamics through the most fundamental data operations, like block modification, insertion, and deletion, is an crucial step toward becoming a reality. While earlier studies on maintaining remote data integrity typically lack either the aid of open auditability or dynamic data operations, this research succeeds in both. We first outline the challenges and potential security issues with straight extensions with completely dynamic data updates from prior works before illustrating how to construct a comprehensive validation mechanism for the seamless integration of these two essential components in our approach.[2] Cloud storage users don't have to worry about maintaining their local hardware and software because they have access to high-quality cloud applications whenever they need them. Using this type of service entails consumers giving up direct possession on their data confidentiality, which, despite the obvious advantages, inevitably generates further security concerns regarding the accuracy of the data kept in the cloud. In this study, we provide an adaptable shared file integrity auditing approach that is based upon homomorphic token and distributed erasure-coded data to solve this new issue and advance the creation of a safe and reliable cloud storage service. Users can audit cloud storage using the suggested technique for very little connection and computation expense. The auditing reports not only provide a reliable confirmation of the correctness of cloud storage, but also enable rapid localization of data errors or server mis-behaviour. The suggested method also enables secure and effective dynamic operations of out data, such as block change, deletion, and addition, given that cloud data is dynamic in nature. The suggested approach is extremely effective, resistant to Byzantine failure, malicious data alteration assaults, and occasionally even server collusion attempts, according to the study. [3] In this post,

we offer a dynamic audit solution for assessing the dependability of outsourced storage that is unreliable. Utilizing techniques from index-hash tables, random sampling, and fragment architecture, our audit service was created, enabling verifiable modifications to outsourced data and quick anomaly identification. We also provide a potent technique for enhancing the effectiveness of audit services, which is based on statistical query and periodic verification. In addition to proving the effectiveness of our approaches, our experimental findings show that the audit system requires less additional storage and has a low calculation cost for the audit metadata. The traditional cryptographic solutions, which focus on hash functions as well as signature algorithms for data availability and integrity, cannot be employed with the outsourced data. Due to the pricey connections, especially for large files, downloading data to validate it is not a realistic solution. Moreover, auditing the veracity of the information in a cloud infrastructure can be challenging and costly for cloud users. In order for data owners to regularly audit the outsourced data, it is important to actualize public audit capability for CSS. This is so that independent auditors can perform assessments that typical users cannot. For digital forensics and cloud credibility, this auditing service is absolutely essential. Some scholars have suggested the concepts of evidence of retrievability and possessing proven data in order to create public audit capability.

They used a probabilistic proof technique, which is a way for a storage provider to demonstrate that the data of their customers is still intact. [4] The current ways for setting access control on it involve the quite well encryption of the secret material. When employing such methods, it is up to the file owners to encrypt their files before uploading them to the cloud and to do so again whenever user identities or permission guidelines vary. As a result, transmission and processing costs for data owners are significant. To reduce the burden on data owners and ensure the privacy of data in the cloud, it would be better to delegate the implementation of fine-grained authentication mechanisms to the cloud. We suggest a strategy based on two levels of encryption to meet this criterion. In our system, the coarse-grained encryption is carried out by the data controller, and the fine-grained encryption is carried out by the cloud on top of the previously encrypted data. How to disassemble Access Control Policies (ACPs) to permit the use of two layer encryption is a difficult problem. To demonstrate the NP-completeness of this problem and suggest fresh optimization techniques. We use a powerful group key administration system that encourages persuasive ACPs. Our solution fully trusts the cloud to handle authorization enforcement while protecting user privacy and ensuring data confidentiality. [5-10] Users can take advantage of high-quality programs and services on demand from a shared pool of customizable computer resources with the help of cloud storage without being concerned about preserving and keeping their data locally. It's challenging to maintain the integrity of cloud-based data that has been outsourced, especially for users with low computing resources, because users they no longer physically possess the outsourced data. Users should also not have to worry about verifying the privacy of cloud storage. They should just be able to utilize it as if it were local. Customers can utilize the Third Party Analyzer (TPA) to confirm the accuracy of data that has been outsourced, it is essential for cloud storage to have public audit capability.[11-16] This will put them at peace. The auditing procedure should not create any new online burdens for users or vulnerabilities affecting user data privacy in order to introduce a TPA securely and effectively. We present a private-public auditing approach for a secure cloud storage system in this study. We further extend our finding such that the TPA may effectively and simultaneously conduct audits for a number of consumers. The proposed techniques are provably secure and extremely effective, according to a thorough investigation of security and performance.

III. EXİSTİNG METHOD

Through the provision of dynamically scalable and frequently virtualized services and resources through the Internet, cloud computing offers a fresh way to augment the present consumption and distribution paradigm for IT services obtained from the Internet. Among these notable personal and commercial cloud computing services currently offered are Yahoo, Google, Microsoft, Salesforce, and Amazon. Users are not believed to be expertise in technology infrastructure and are distanced from the complexity of the services offered. Additionally, users might not be aware of the hardware used to process and store their data. Consumers start to worry about loosing ownership of their own information in addition to enjoying the ease that this modern technology brings.

The handling of personally identifiable data is one of the many accountability challenges that contracting the data processing onto clouds typically produces. Such worries are starting to pose a serious obstacle to the widespread use of cloud services.

Highly scalable services can now be quickly and readily used as-needed through the Internet thanks to cloud computing. The fact that member data is frequently processed remotely on unidentified infrastructure that clients do not control or operate is one of the essential elements of cloud services. Despite the ease that this new, emerging technology affords, users' concerns regarding loosing access to their personal information, particularly their health and financial information, can constitute a substantial barrier to the broad usage of cloud services.

IV. PROPOSED SYSTEM

Multi-access filtering is a technique used in privacy-preserving fog computing to improve data privacy and security in a multi-user environment. In this approach, fog nodes act as intermediaries between the end devices and the cloud, filtering out sensitive data before sending it to the cloud.

The multi-access filtering technique involves three main components:

Data Encryption: To ensure that sensitive data is protected from unauthorized access, fog nodes encrypt the data before

transmitting it to the cloud.

Access Control: Access control is used to restrict access to sensitive data only to authorized users. Fog nodes use access control policies to decide who can access what data.

Data Filtering: Fog nodes filter out sensitive data that is not relevant to a particular user or application. For example, if a user only needs access to certain data elements, the fog node will filter out the other data elements before sending the data to the cloud.

Using multi-access filtering in fog computing provides several benefits for data privacy and security. Firstly, it helps to prevent unauthorized access to sensitive data by encrypting it and controlling access to it. Secondly, it reduces the amount of data that needs to be transmitted to the cloud, which can help to improve network bandwidth and reduce latency. Finally, it provides greater control over data privacy by allowing users to filter out data that is not relevant to their needs.

We provide a brand-new, highly decentralised information accountability architecture to monitor how users' data is being used in the cloud. Our Object-Centered design, in particular, permits confining our logging method along with user data and regulations.

In order to build a portable, adaptable object and guarantee that any access to user data will result in automated logging as well as local JAR authentication, we made utilization of JAR's programmed characteristics. We also offer distributed auditing tools to give users more control. We offer in-depth experimental research that show how successful and efficient the suggested strategies are. On a real cloud that has been tested, we run experiments. The outcomes show how effective, scalable, and precise our method is. We also go through the dependability and strength of our architecture and offer a thorough security analysis.

The classical approach, the client/server methodology, component-based approach are the three main methods for developing applications over time.

Traditionally, a single program handled display strategy, business rules, and database interaction. These programs were also known as monolithic program. The disadvantage of this method was that the application software needed to be included and recompiled whenever even a minor update, extend, or upgrading was necessary.

The client/server design, often known as a two-tier architecture, was established as a result of the drawbacks of the conventional method. In this design, the client-side and central locations where the data is stored are separated, with the central location serving as the server. The business logic is combined with the display logic either at the consumer side or at the server-side using the database connectivity code.

If enterprise functionality and representation theory are combined at the client's end, the client is said to be a "fat client." The server is referred to as a fat server is if business logic and database server are coupled.

An application is split into 2 parts as a result of a 2-tiered architecture: The GUI and the Database. A request is made to the server by the client. Due to its greater computing capacity, the server handles all data retrieval and processing, sending the desired outcome only response to the customer for last-minute modifications.

The back-end server system's capabilities are essentially shared by less capable machines. Faster server-side execution results in less web traffic as well as lengthier loading times for the program to gather and process the data. The Client/Server System, however, also had several drawbacks:

Any modification to the business logic was necessary whenever the company policies changed. Depending on where the business logic is located, it may be necessary to update the display logic or the database connectivity code.

Due to the few database connections that the client has access to when employing a 2-Tier design, applications could be challenging to scale up. The server simply rejects requests for connectivity that exceed a particular threshold.

The 3-Tier Architecture was developed as a result of the limitations of the client/server architecture. In a three-tier architecture, the client controls the presentation logic, the server manages the database, and the middle layer houses the business logic. The application server is the name given to this layer of business logic. Server-centric architecture is another name for this design.

The workload is distributed evenly among the client, database server, and server performing the business logic because the middle-tier handles that functionality. Additionally, this architecture offers effective data access. The problem with database connection restrictions is reduced since the database simply views the business logic layer rather than all of its clients.

In a two-tier software, a database connectivity is established and maintained early. However, in a three-tier application like this, it is only generated when data access is required, and it is released as soon as the data is acquired or sent to the server.

Distributed applications are those in which the database, presentation logic, and business logic all reside on different systems.

User registration

Users must register their information, including their user ID and password, before visiting the home page. To prevent anonymous users, this registration will be used. The user must choose the constant factor; a different random salt would be created for each user. During the registration process, the user must choose the random function that is made available to him.

Login

In this module, the user must choose a specific digit from their password to serve as the value for the variable a, a specific digit from their random salt password to serve as the value for the variable b, and the constant value they chose at registration to serve as the constant value.

The user-selected function is then tested using the values of a, b, and the constant value. When a user enters a value that exactly matches the password that the system generates, the system verifies the user's identity and allows them to log in. **Fig 2** shows data flow diagram level 0.

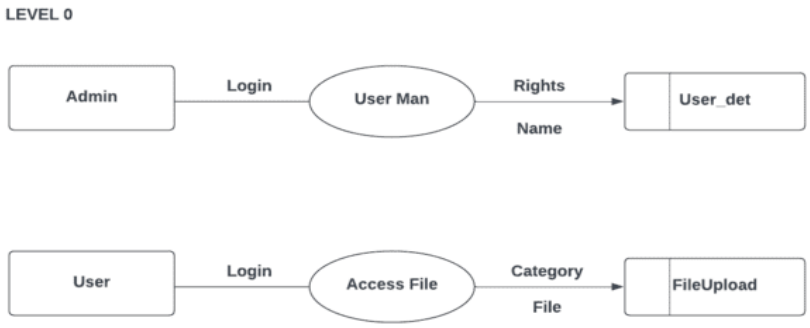

Fig 2. Data Flow Diagram Level 0.

File Download

Certain files of Content are available for download from the Cloud. These files are uploaded by data owner. If user have a all rights user download the files. If doesn't download rights user unable to download cloud files user can only view the files. Here we are maintaining accountability of user's account. In an organization some peoples only have all rights. Else have specific rights.

Admin Module

User Account

Here administrator has to register the Cloud Users. We are gathering specific details from user for registration. Admin can view the user details even edit and delete also. We use two rights the first one is downloading and another one is view for priority users. And we generate encryption key for every user. User cans logging their account using by this encryption key. **Fig 3** shows data flow diagram level 1.

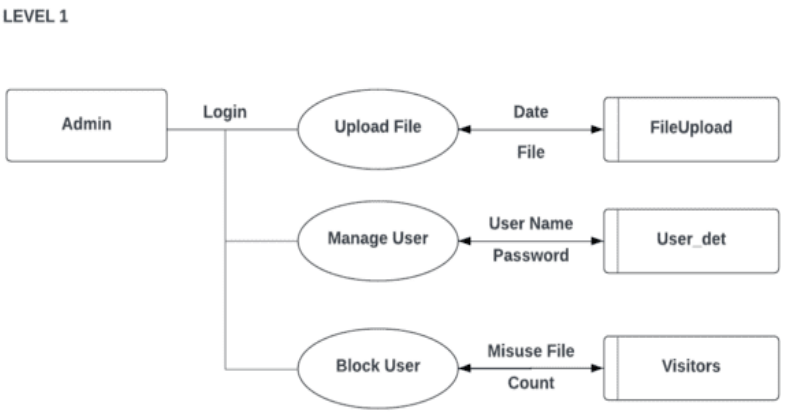

Fig 3. Data Flow Diagram Level 1

Uploading files

If you are using a personal computer to sign onto a network and you want to send files across the network, you must upload the documents from your PC. The administrator uploads each file to the cloud. Similar to the electrical grid, internet-based cloud computing is a way of providing software, common resources, and information to systems and other devices as needed.

Fig 4 shows data flow diagram level 2.

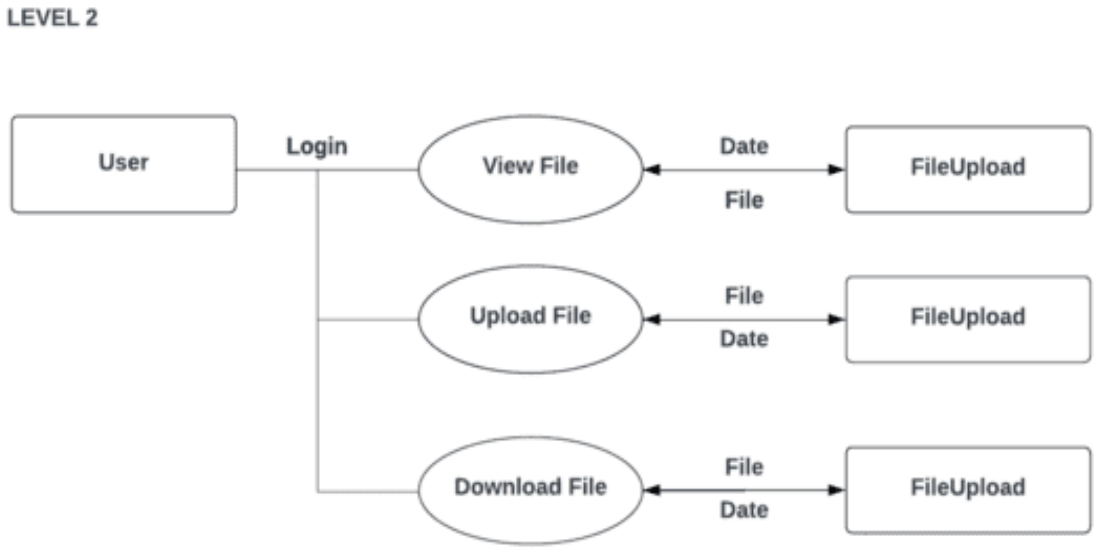

Fig 4. Data Flow Diagram Level 2

File Download
The behavior or method of copying data in this way. to allow admin to download the files he submitted. The cloud server stores uploaded files. So the files could be downloaded from the cloud by authorized users. Files can only be viewed by unauthorized users.

Cloud Visitors
When this strategy is applied in the cloud, when a user misbehaves in the cloud, other users respond swiftly, strongly, and respectfully. Before behavior spirals out of control, small issues are addressed. The administrator can monitor the bad user using the log file. After logging in, the user's login file is automatically created. However, the user is unaware of these log files. Admin can then follow the user. These guidelines aid in securing the cloud.

V. RESULT AND ANALYSİS

The Advanced Encryption Standard (AES) algorithm is a widely-used symmetric-key encryption algorithm that is used to protect sensitive data. It was selected as the standard encryption algorithm by the National Institute of Standards and Technology (NIST) in 2001.

AES operates on fixed-length blocks of data, which are typically 128 bits in length, although it can also work with blocks of 192 or 256 bits. The algorithm uses a secret key to transform plaintext into ciphertext, and vice versa.

The encryption process in AES involves several rounds of substitution and permutation operations, which mix up the data in a way that makes it difficult for an attacker to decipher without knowing the secret key. The number of rounds depends on the length of the key and the block size, with more rounds being used for longer keys and larger blocks.

One of the key advantages of AES is that it is very fast and efficient, even on relatively low-powered devices. This has made it popular for use in a wide range of applications, including secure communication protocols, encrypted storage devices, and secure payment systems.

Overall, AES is considered to be a highly secure encryption algorithm that offers strong protection against unauthorized access to sensitive data. However, like all encryption algorithms, its security can be compromised if the key is stolen or if there are weaknesses in the implementation of the algorithm.

The proposed strategy includes the following five algorithms:

(1) Setup (K). K is an input for the security parameter of the system setup algorithm. It produces the public key and Master Key (MK) (PK).
(2) CreateAttributeAuthority (PK, AA) (PK, AA). With the AA request as input, the GA (Central Authority) runs this algorithm. With a collection of characteristics, Sid, a secret authority key, SKAid, and a functional identification, Aid, for

the AA, it produces a result. The Ministry of Health classifies the AAs based on their functionalities before determining the qualities for those who would use those functionalities.
(3) AttributeKeyGenerator (PK, SKAid, Sid) (PK, SKAid, Sid). The Aid domain authority runs this algorithm. It requires the domain authority's secret key, SKAid, as well as the list of attributes, Sid, as input. It outputs the user SKUj's attribute secret keys.
(4) Encrypt (PKU, P, M, PK). The encrypt method takes as inputs the PK, Message (M), an Access Policy (P), and then a collection of the Public User Keys (PKUs) that match each attribute in P. The text message CT cypher is created.
(5) Decryption (SKUj, PK, CT , SKA, P). The decode method takes as inputs the PK, a cipher-text message CT, the privilege P used for encryption, the secret user key SKUj, and a group of hidden attribute key sets SKA.

VI. CONCLUSİON

In this study, mainly focused on the potential privacy leakage issue brought on by overzealous data collecting by unreliable parties. The suggested paradigm uses fog computing like a secure interface and offers a number of encryption methods for access filtering. An ideal schedule (dynamic programming) was created to maximize the capability for protecting privacy in resource-constrained environments. The findings of evaluations demonstrated the potential utility of the model.

References

[1]. K. Gai, L. Zhu, M. Qiu, K. Xu, and K.-K. R. Choo, “Multi-Access Filtering for Privacy-Preserving Fog Computing,” IEEE Transactions on Cloud Computing, vol. 10, no. 1, pp. 539–552, Jan. 2022, doi: 10.1109/tcc.2019.2942293.

[2]. Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, “Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing,” IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 5, pp. 847–859, May 2011, doi: 10.1109/tpds.2010.183.

[3]. Cong Wang, Qian Wang, and Kui Ren Department of ECE Illinois Institute of Technology. This work was supported in part by the US National Science Foundation under grant CNS-0831963, CNS0626601, CNS0716306, and CNS-0831628.

[4]. Md. Shamsheer, K. Sunıl Kumar, “Towards Secure and Dependable Storage Services in Cloud Computing” in International Journal of Innovative Technologies Volume.06, Issue No.01, JanuaryJune, 2018, Pages: 0128-0131.

[5]. Boyang Wang, Baochun Li, and Hui Li, “Oruta: privacy-preserving public auditing for shared data in the cloud,” IEEE Transactions on Cloud Computing, vol. 2, no. 1, pp. 43–56, Jan. 2014, doi: 10.1109/tcc.2014.2299807.

[6]. C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing,” 2010 Proceedings IEEE INFOCOM, Mar. 2010, doi: 10.1109/infcom.2010.5462173.

[7]. Yan Zhu, Gail-Joon Ahn, Hongxin Hu, S. S. Yau, H. G. An, and Chang-Jun Hu, “Dynamic Audit Services for Outsourced Storages in Clouds,” IEEE Transactions on Services Computing, vol. 6, no. 2, pp. 227–238, Apr. 2013, doi: 10.1109/tsc.2011.51.

[8]. Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, “Dynamic audit services for integrity verification of outsourced storages in clouds,” Proceedings of the 2011 ACM Symposium on Applied Computing, Mar. 2011, doi: 10.1145/1982185.1982514.

[9]. H. Wu, G. Li, and L. Zhou, “Ginix: Generalized inverted index for keyword search,” Tsinghua Science and Technology, vol. 18, no. 1, pp. 77–87, Feb. 2013, doi: 10.1109/tst.2013.6449411.

[10]. M. Nabeel and E. Bertino, “Privacy Preserving Delegated Access Control in Public Clouds,” IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 9, pp. 2268–2280, Sep. 2014, doi: 10.1109/tkde.2013.68.

[11]. C. Wang, S. S. M. Chow, Q. Wang, K. Ren and W. Lou, "PrivacyPreserving Public Auditing for Secure Cloud Storage," in IEEE Transactions on Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013, doi: 10.1109/TC.2011.245.

[12]. Yang, J. Xu, J. Weng, J. Zhou, and D. S. Wong, “Lightweight and Privacy-Preserving Delegatable Proofs of Storage with Data Dynamics in Cloud Storage,” IEEE Transactions on Cloud Computing, vol. 9, no. 1, pp. 212–225, Jan. 2021, doi: 10.1109/tcc.2018.2851256.

[13]. Haldorai, A. Ramu, and S. A. R. Khan, Eds., “Business Intelligence for Enterprise Internet of Things,” EAI/Springer Innovations in Communication and Computing, 2020, doi: 10.1007/978-3-030-44407-5.

[14]. S. Murugan and Anandakumar H., “Privacy Information Leakage Prevention in Cognitive Social Mining Applications,” Cognitive Social Mining Applications in Data Analytics and Forensics, pp. 188–212, 2019, doi: 10.4018/978-1-5225-7522-1.ch010.

[15]. M. Li, L. Zhu, and X. Lin, “Efficient and Privacy-Preserving Carpooling Using Blockchain-Assisted Vehicular Fog Computing,” IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4573–4584, Jun. 2019, doi: 10.1109/jiot.2018.2868076.

[16]. W. Teng, G. Yang, Y. Xiang, T. Zhang, and D. Wang, “Attribute-Based Access Control with Constant-Size Ciphertext in Cloud Computing,” IEEE Transactions on Cloud Computing, vol. 5, no. 4, pp. 617–627, Oct. 2017, doi: 10.1109/tcc.2015.2440247.