

Implementation of Cloud Computing Data Security Based on Hybrid Elliptical Curve Cryptography

¹ N. Krishnamoorthy, ² S. Umarani

Submitted: 25/05/2023

Revised: 07/07/2023

Accepted: 26/07/2023

Abstract: The adoption of cloud storage has increased in a rapid pace across all kinds of applications. Though cloud storage has been used frequently, there are two key worries the consumer has when using the cloud storage. Recently, the storage and retrieval of data in the cloud architecture is an attractive research field in cloud computing. Besides those, economical resource use and secure data transfer are additional limits over the numerous internet services. The rise of cryptographic encryption and decoding technologies assure the privacy protection of cloud uploaded data. An important activity prior to an encryption mechanism is to understand the levels of security, storage and data access risks. Cloud computing, a globalized platform to allow the virtual execution of process lead to outsourcing of numerous services such as data processing, remote execution and storage to cloud servers. This work comprises several sequential procedures including key evaluation mechanism, production of key derivation policies with two stages namely, master key generation and private key generation and key verification. The key assessment mechanism reveals that the inclusion of attributes of data and data owner decides the execution of both encryption and decryption. Hybrid Elliptical Curve Cryptography (HECC) to bring security into Web Application based cloud services while providing security for hybrid cloud networks and the data they store and retrieve from the clouds. The proposed security frameworks have been implemented and also compared the effectiveness of the system with the existing models in terms of encryption and computational burden along with the security level when storing, retrieving and accessing the data in cloud environment.

Keywords: Cloud Storage, Cryptography, Encryption, Privacy, Private key, Security, Web


1. Introduction


Cloud computing is a new technology that is gaining traction because of its many advantages, such as the ability to access data from any location. The platform, hardware, and software are all provided as a service in this technology. A cloud service provider responds to a customer's request for a service [1]. The scheduling of user requests is an important issue in the cloud, which means how to allocate resources to these requests so that the requested tasks can be completed in the least amount of time and cost. Services can be used from various and widely distributed resources rather than from remote servers or local machines in the case of Cloud computing [2].

Cloud computing has no accepted definition. Many distributed servers, collectively known as masters, provide requested services and resources to a variety of clients, collectively known as customers, in a network with data centre scale and reliability. On-demand services are provided by the distributed computers.

Organizations are interested in outsourcing their data to cloud storages to reap the benefits of cloud services because

service providers offer a highly scalable computing platform for users to build a wide range of applications [3]. Because sensitive data are stored on shared servers and accessible by both internal and external users, data owners must implement security measures to protect their data. Ultimately, Authentication and Access control plays a crucial role in protecting data against unauthorised access. For any type of computing, security is the most important consideration, making it clear that cloud computing is no exception [4]. It is essential to manage and verify the user's identity in cloud computing because sensitive information is stored both on the client's side and in cloud servers. To avoid security breaches that go undetected for an extended period of time, it is critical to ensure that only qualified users have access to the cloud's authentication mechanisms and that these mechanisms are properly protected [5]. A probable authentication scenario for cloud infrastructure is exemplified in Figure 1.

¹Department of Computer Science, College of Science and Humanities, SRM Institute of Science and Technology, Ramapuram, Chennai- 600 089, TamilNadu, India, , krishnan@srmist.edu.in

²Department of Computer Applications, College of Science and Humanities, SRM Institute of Science and Technology, Ramapuram, Chennai- 600 089, TamilNadu, India, , umaranis@srmist.edu.in

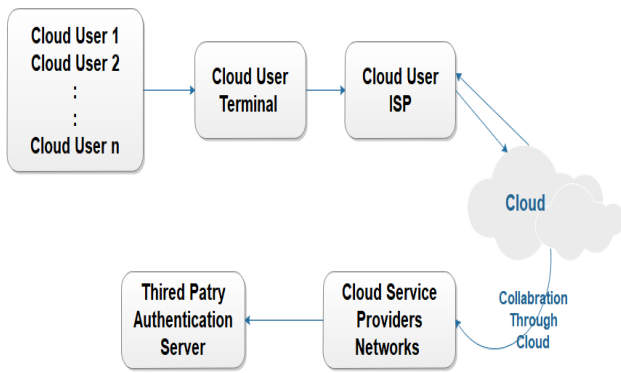


Fig. 1. Cloud Authentication Process.

2. Background

The front end and back end of the cloud computing architecture can be categorised. Apps and interfaces used by users to access the CC platforms are referred to as front-end components [6-8]. The cloud itself is referenced in the backend. Everything that is needed to run the CC services can be found in this group. Large data storage, virtual machines, security mechanisms, service offerings, deployment models, servers, and so on are all part of this infrastructure.

The front-end and back-end are connected via a network, most commonly the internet [9]. There are a number of different network entities in the cloud architecture model.

- (i) Users/Clients
- (ii) Cloud Service Provider (CSP)
- (iii) Third Party Auditor (TPA).

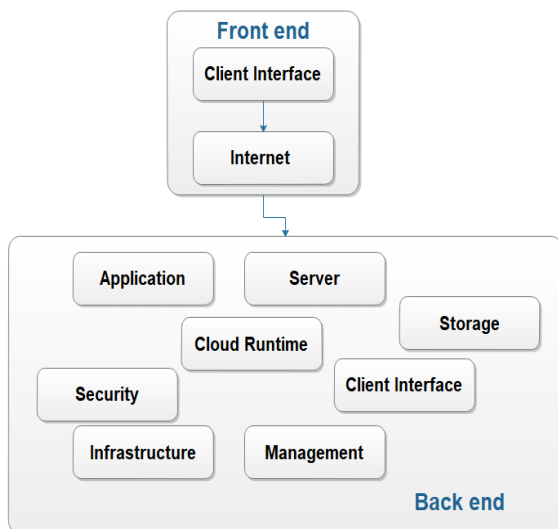


Fig. 2. Architecture of Cloud Computing

There are two types of customers who use the cloud for information storage and computation: individual purchasers and organizations. It is a discretionary substance, a TPA, who has the ability and capacities that clients might not have, who is trusted to inspect distributed storage benefits

and reveal dangers on behalf of clients when solicited [10-14]. Clients can store their data on multiple cloud servers that are running simultaneously, collaboratively, and disseminated by a cloud service provider (CSP). As the client's information grows in quantity and importance, a strategy of eradication remedying code can be used to withstand difficulties or server crashes. For application purposes, the client connects to the cloud servers via CSP to retrieve or access his data. Customers no longer have access to their data locally, thus it is critical to ensure that their information is being effectively stored and maintained [15]. That is, customers should be given security measures so that they may verify the accuracy of their stored data even if there are no nearby copies.

In this section, we'll go through how the proposed system's security base would be implemented. The primary objective of this security framework is to prevent an adversary from accessing users' data, therefore preserving their privacy [16]. The following are the main points of the proposed cryptographic scheme:

- **Securing keys exchange.**
- **Securing file encryption.**

ECIES, ECDH, ECDSA, and SHA all enable safe key exchange and confidentiality, whereas HECC and ECIES also provide data integrity and authentication, respectively.

3. Methodology

The proposed authentication process for cloud security applications consists of three stages: startup, registration, and authentication. The trustworthy server produces and publishes the system parameters in the first phase. The trusted server is used to provide the public and private keys to the users in the second stage. In the final step, a shared key is created between the server programme and the user's machine [17]. Figure 3 depicts the proposed design for the new building type. When using this proposed method of obfuscation, the key is generated from the plain text itself, rather than a separate procedure. The key is extracted from the current plain text and used to encrypt the preceding plain text. Language processing techniques are used to select the key term from the provided plain text in order to implement this. The suggested method incorporates a language processing technique for chunking the entire text into characters in order to produce a repeating key [18].

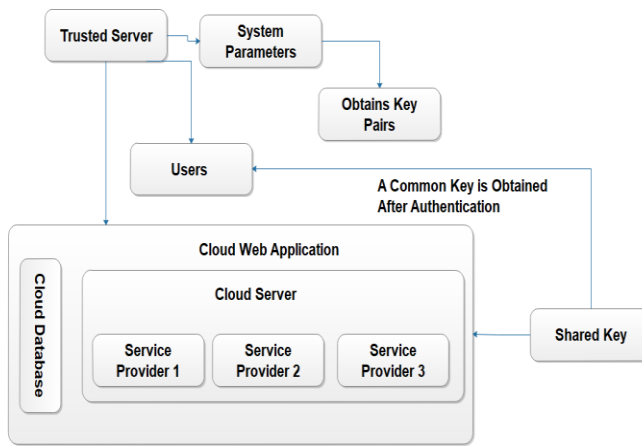


Fig. 3. Overview of authentication in web application using HECC

A block of plain text is broken down into individual words, and each sequence of ten words is counted as a segment [19]. The first five words of a section will be encoded using the following five words. If we use the words length to compare characters, we may find that they are not all of the same length. The data-sharing subsystem is handled by the service provider. Key generation, file encryption, key verification, and file decryption are all part of this subsystem.

Key management scheme

Users can speak directly with the cloud controller once they discover their secret keys. For each group of users, manager selects a random key and uploads tuple containing key parameters to cloud service provider [20]. All users can view the tuple in a public directory, and the data section DS is assigned to it. It is possible to store the key database outside of the organization's firewall, either in a private cloud or on a trusted server. It is not enough to know these key parameters in order to decrypt a message. To gain access to the data, the manager gives out the secret keys to everyone in the group.

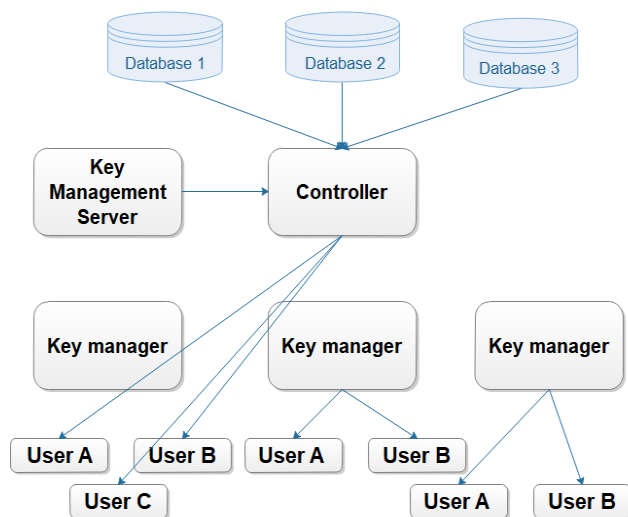


Fig. 4. Key management server based on cloud controller

The manager distributes secret keys to the appropriate users after verifying their identity through the authentication process [21]. The first step in this process is for the manager to authenticate each user by verifying the user's identity using the ACL. The symmetric partition group key is used to encrypt a message as a ciphertext. Then, the cloud authenticates the user and stores the ciphertext in the data portion. The cloud data can only be accessed by the recipient, who has the key to open it. The recipient can then use the symmetric partition group key to decrypt the ciphertext after it has been authenticated. When a new member of the group joins, the existing tuple is replaced with a new one that includes the new member and is submitted to the service provider. Current users do not need to receive a copy of the updated tuple as they already possess the symmetric partition group key. Figure 5 dissipate the key management work flow.

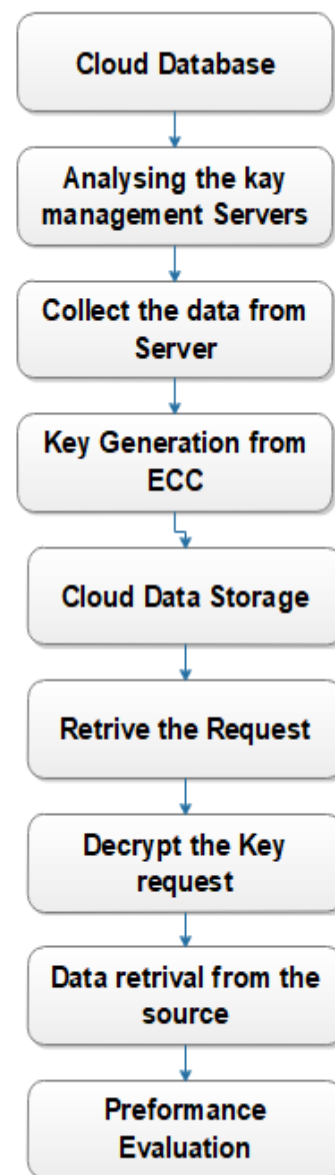


Fig. 5. Flowchart for the key management system

Key Generation Module

Encrypting each fractured block and storing it on a separate server with its unique identification number is the primary goal of this module [22]. When it comes to encryption, the RSA and HECC algorithms are combined to form a hybrid algorithm. According to the analysis, this combination delivers a higher level of security. In order to decrypt a file, perform these steps:

Algorithm for Key generation module

- 1: Start
- 2: Input
 - Initial Key size “n”
 - Create the public key
 - Range 1 to n-1
 - Point of Curve H
- 3: Encrypt block using key
 - Key generation using HECC
- 4: Encrypt → Encrypt block using key of HECC
- 5: Output → Encrypt output using generated key of HECC
- 6: Stop

4. Results

The proposed method's performance is evaluated based on how long it takes to encrypt and decode data. In the following diagram, each performance statistic is described in detail:

Encryption Time: Encryption time is the amount of time it takes the algorithm to convert the original text into ciphertext. The maximum encryption time is caused by the variety in file sizes and the amount of key attributes. In order for a method to be considered the best, the amount of time it takes to encrypt a file and its key properties must be minimised.

Decryption Time: Decryption time is the amount of time it takes the algorithm to convert the encrypted text to the plain text. Decryption times might take a long time because of the wide range of file sizes and key properties. As file sizes and key properties increase, the method only becomes more efficient.

Security Performance: The level of security's performance is a measure of how accurately the encrypted text's original content is received. When it comes to greater security, the time necessary for encryption and decoding is the deciding factor. Algorithms with short computation times provide the best security performance.

Because data is encrypted before it is posted to the server, it is possible to estimate the amount of time the cloud

environment will take to process it. As can be seen in Figure 6, Dataset-5 took the longest to process due to the sheer volume of files it includes. As a result, the time it takes to upload data increases as the quantity of files increases.

Original File Size in Cloud platform (KB)	Encryption time (ms)	Decryption time (ms)	Encrypted file size (KB)	Decrypted file Size (KB)
10	0.4	0.3	12.34	10
20	0.46	0.32	21.28	20
30	0.52	0.38	35.72	30
40	0.56	0.44	43.67	40
50	0.63	0.49	54.29	50

Table 1: Possible to estimate the amount of time the in cloud environment

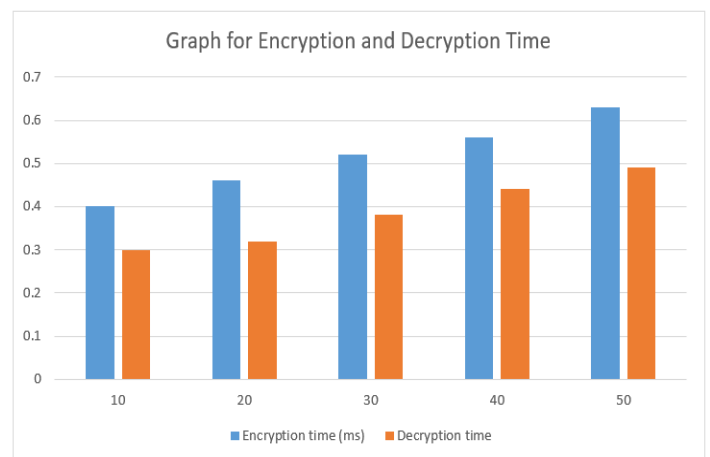


Fig. 6. Graph for encryption and decryption time



Fig. 7. Encrypted File Size in cloud environment

	Cloud Platform 1	Cloud Platform 2	Cloud Platform 3
Database 1	0.053	0.033	0.013
Database 2	0.064	0.053	0.016

Database 3	0.073	0.042	0.035
Database 4	0.042	0.063	0.054
Database 5	0.059	0.073	0.062

Table 2: Data Loss in cloud platform

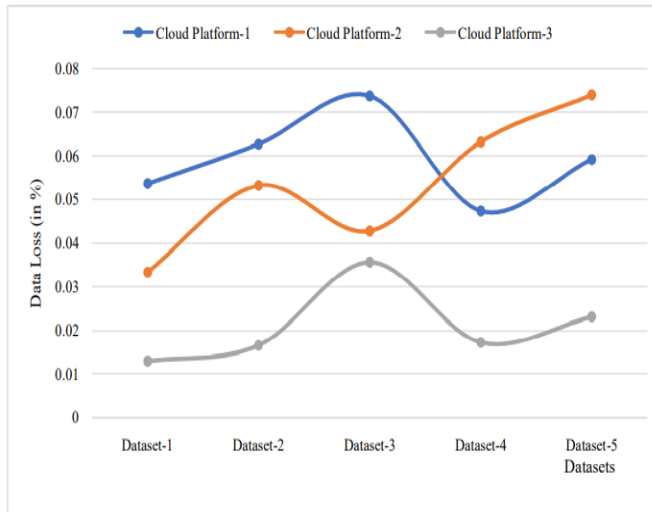


Fig 8. Graph for data loss in cloud environment

After the encryption procedure, the original file may still contain a number of additional bits (Key) as cypher text. There must be no additional text or missing information while the file is being decrypted.

5. Conclusion:

All aspects of the HECC text encryption method, including encryption and decryption time, message loss, and key braking time, have been thoroughly examined and analysed. Because it maintains network anonymity, this approach offers a promising perspective on this rapidly expanding and difficult topic. This shows the usefulness of the CS-based public key when it is used in an efficient manner. The prior results confirm the loss of data. By uploading five data sets with different types of files to a local cloud environment, the proposed efficient HECC framework is compared to an existing framework that uses a random fragmentation mechanism along with the same encryption mechanism to determine its performance, resulting in a lower time, data size, and data loss rate.

Data storage time and storage requirements can be reduced by using this method, which reduces the size of encrypted data while retaining the least amount of data. All of these schemes rely on standard encryption algorithms with a slew of new parameters added on top of them. When employing the standard technique, there are various compromises that the user must consider. It is the implementation of these algorithms or the management of the encryption and

decryption keys that causes issues and is vulnerable to errors or attacks, not the algorithms themselves.

In addition to the difficulties encountered, the additional overhead created by standard methods is significant. These algorithms are more difficult to implement because of their complexity. The user has to make a trade-off, sacrificing time in order to gain a high level of security. In order to address these challenges, the following chapter discusses models that use lightweight strategies to secure cloud storage data security.

References

- [1] T. D. Dang, D. Hoang and D. N. Nguyen, "Trust-Based Scheduling Framework for Big Data Processing with MapReduce," in *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 279-293, 1 Jan.-Feb. 2022, doi: 10.1109/TSC.2019.2938959.
- [2] F. Yi, O. Jeong and I. Moon, "Privacy-Preserving Image Classification With Deep Learning and Double Random Phase Encoding," in *IEEE Access*, vol. 9, pp. 136126-136134, 2021, doi: 10.1109/ACCESS.2021.3116876.
- [3] M. Zhang, Y. Chen and J. Huang, "SE-PPFM: A Searchable Encryption Scheme Supporting Privacy-Preserving Fuzzy Multikeyword in Cloud Systems," in *IEEE Systems Journal*, vol. 15, no. 2, pp. 2980-2988, June 2021, doi: 10.1109/JSYST.2020.2997932.
- [4] Y. Wang, S. Yang, X. Ren, P. Zhao, C. Zhao and X. Yang, "IndustEdge: A Time-Sensitive Networking Enabled Edge-Cloud Collaborative Intelligent Platform for Smart Industry," in *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2386-2398, April 2022, doi: 10.1109/TII.2021.3104003.
- [5] S. Fugkeaw, "A Fine-Grained and Lightweight Data Access Control Model for Mobile Cloud Computing," in *IEEE Access*, vol. 9, pp. 836-848, 2021, doi: 10.1109/ACCESS.2020.3046869.
- [6] W. Guo, J. Li, X. Liu and Y. Yang, "Privacy-Preserving Compressive Sensing for Real-Time Traffic Monitoring in Urban City," in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14510-14522, Dec. 2020, doi: 10.1109/TVT.2020.3042794.
- [7] W. M. Danquah and D. T. Altılar, "UniDRM: Unified Data and Resource Management for Federated Vehicular Cloud Computing," in *IEEE Access*, vol. 9, pp. 157052-157067, 2021, doi: 10.1109/ACCESS.2021.3127521.
- [8] B. D. Deebak and F. Al-Turjman, "Smart Mutual Authentication Protocol for Cloud Based Medical Healthcare Systems Using Internet of Medical Things," in *IEEE Journal on Selected Areas in Communications*,

vol. 39, no. 2, pp. 346-360, Feb. 2021, doi: 10.1109/JSAC.2020.3020599.

- [9] X. Xu et al., "Secure Service Offloading for Internet of Vehicles in SDN-Enabled Mobile Edge Computing," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3720-3729, June 2021, doi: 10.1109/TITS.2020.3034197.
- [10] Y. Sun, Q. Liu, X. Chen and X. Du, "An Adaptive Authenticated Data Structure With Privacy-Preserving for Big Data Stream in Cloud," in *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3295-3310, 2020, doi: 10.1109/TIFS.2020.2986879.
- [11] W. DING, Z. Yan and R. H. Deng, "Privacy-Preserving Data Processing with Flexible Access Control," in *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2, pp. 363-376, 1 March-April 2020, doi: 10.1109/TDSC.2017.2786247.
- [12] L. Lan, R. Shi, B. Wang and L. Zhang, "An IoT Unified Access Platform for Heterogeneity Sensing Devices Based on Edge Computing," in *IEEE Access*, vol. 7, pp. 44199-44211, 2019, doi: 10.1109/ACCESS.2019.2908684.
- [13] Z. Wen, R. Qasha, Z. Li, R. Ranjan, P. Watson and A. Romanovsky, "Dynamically Partitioning Workflow over Federated Clouds for Optimising the Monetary Cost and Handling Run-Time Failures," in *IEEE Transactions on Cloud Computing*, vol. 8, no. 4, pp. 1093-1107, 1 Oct.-Dec. 2020, doi: 10.1109/TCC.2016.2603477.
- [14] S. Guan, R. E. De Grande and A. Boukerche, "A Multi-Layered Scheme for Distributed Simulations on the Cloud Environment," in *IEEE Transactions on Cloud Computing*, vol. 7, no. 1, pp. 5-18, 1 Jan.-March 2019, doi: 10.1109/TCC.2015.2453945.
- [15] J. George, C. Chen, R. Stoleru and G. G. Xie, "Hadoop MapReduce for Mobile Clouds," in *IEEE Transactions on Cloud Computing*, vol. 7, no. 1, pp. 224-236, 1 Jan.-March 2019, doi: 10.1109/TCC.2016.2603474.
- [16] J. Yuan and Y. Tian, "Practical Privacy-Preserving MapReduce Based K-Means Clustering Over Large-Scale Dataset," in *IEEE Transactions on Cloud Computing*, vol. 7, no. 2, pp. 568-579, 1 April-June 2019, doi: 10.1109/TCC.2017.2656895.
- [17] D. C. Nguyen, P. N. Pathirana, M. Ding and A. Seneviratne, "Blockchain for Secure EHRs Sharing of Mobile Cloud Based E-Health Systems," in *IEEE Access*, vol. 7, pp. 66792-66806, 2019, doi: 10.1109/ACCESS.2019.2917555.
- [18] X. Yuan, J. Weng, C. Wang and K. Ren, "Secure Integrated Circuit Design via Hybrid Cloud," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 8, pp. 1851-1864, 1 Aug. 2018, doi: 10.1109/TPDS.2018.2807844.
- [19] Q. Huang, W. Yue, Y. He and Y. Yang, "Secure Identity-Based Data Sharing and Profile Matching for Mobile Healthcare Social Networks in Cloud Computing," in *IEEE Access*, vol. 6, pp. 36584-36594, 2018, doi: 10.1109/ACCESS.2018.2852784.
- [20] M. Sookhak, F. R. Yu and A. Y. Zomaya, "Auditing Big Data Storage in Cloud Computing Using Divide and Conquer Tables," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 5, pp. 999-1012, 1 May 2018, doi: 10.1109/TPDS.2017.2784423.
- [21] X. Li, J. Yuan, H. Ma and W. Yao, "Fast and Parallel Trust Computing Scheme Based on Big Data Analysis for Collaboration Cloud Service," in *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 1917-1931, Aug. 2018, doi: 10.1109/TIFS.2018.2806925.
- [22] Z. Xia, Y. Zhu, X. Sun, Z. Qin and K. Ren, "Towards Privacy-Preserving Content-Based Image Retrieval in Cloud Computing," in *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 276-286, 1 Jan.-March 2018, doi: 10.1109/TCC.2015.2491933.
- [23] Zellar, P. I. . (2021). Business Security Design Improvement Using Digitization. *International Journal of New Practices in Management and Engineering*, 10(01), 19–21. <https://doi.org/10.17762/ijnpme.v10i01.98>
- [24] Al-Ansi, A. M. . (2021). Applying Information Technology-Based Knowledge Management (KM) Simulation in the Airline Industry . *International Journal of New Practices in Management and Engineering*, 10(02), 05–09. <https://doi.org/10.17762/ijnpme.v10i02.131>
- [25] Anand, R., Ahamad, S., Veeraiah, V., Janardan, S. K., Dhablya, D., Sindhwani, N., & Gupta, A. (2023). Optimizing 6G wireless network security for effective communication. *Innovative smart materials used in wireless communication technology* (pp. 1-20) doi:10.4018/978-1-6684-7000- 8.ch001 Retrieved from www.scopus.com