Frontiers in Health Informatics
ISSN-Online: 2676-7104
www.healthinformaticsjournal.com
2024; Vol 13: Issue 8
Open Access

A Secure and Scalable Encryption Framework for Cloud Data Protection and Storage Optimization

Balaji Venkateswaran[1], Ashwini Kumar[2], Asheesh Pandey[3], Niharika Singh[4] Ashish Jolly[5], Ritu[6], Shakti Kumar[7], Gayatri Kumari[8]

[1]Research Scholar (Computer Science), School of Engineering and Technology, Shri Venkateshwara University, Gajraula (UP), INDIA, Email: balaji.Venkateswaran@gmail.com

[2]Principal and Professor, Department of MCA, Compucom Institute of Information Technology and Management, Jaipur (Raj.), INDIA

[3]Assistant Professor, Department of MCA, ABES Engineering College Ghaziabad (UP), INDIA

[4]Assistant Professor, Department of MCA, GL Bajaj Institute of Technology and Management, Greater Noida (UP), INDIA

[5]Assistant Professor, Department of Computer Science, Govt. PG College Ambala Cantt. (Haryana), INDIA

[6]Assistant Professor, Department of Computer Science, Govt PG College, Sector-1, Panchkula (Haryana), INDIA

[7]Assistant Professor, Department of Computer Science, Govt. PG College, Ambala Cantt., (Haryana), INDIA

[8]Assistant Professor, Department of Computer Science, Sanatan Dharma College, Ambala Cantt (Haryana), INDIA

Cite this paper as: Balaji Venkateswaran, Ashwini Kumar, Asheesh Pandey, Niharika Singh, Ashish Jolly, Ritu, Shakti Kumar, Gayatri Kumari (2024). A Secure and Scalable Encryption Framework for Cloud Data Protection and Storage Optimization. *Frontiers in Health Informatics*, *Vol.13, No.8*, 6583-6592

Abstract: In this study, we propose a hybrid encryption framework combining RSA and ECC cryptographic techniques to secure cloud-stored keywords. While this approach offers robust encryption and enhances cloud data safety, the risk of forgery remains a concern. To address this, researchers have explored fragmentation-based solutions, which involve dividing data into smaller fragments, encrypting them, and distributing them across multiple cloud servers to prevent unauthorized access. However, existing efforts often overlook the potential knowledge loss caused by fragmentation, emphasizing the need for a tailored approach to mitigate this challenge. Our framework introduces a data-type-specific fragmentation strategy designed to accommodate the heterogeneity and size of cloud-stored data. This approach not only minimizes the risk of data loss but also optimizes cloud storage utilization. To evaluate its effectiveness, we compare the proposed framework against state-of-the-art random fragmentation techniques and traditional non-fragmentation methods, employing multiple performance metrics. The results highlight the potential of our solution to enhance both data security and cloud storage efficiency.

Keywords: Cloud security, RSA, ECC, Cloud storage

INTRODUCTION

Recent advancements in the field of network-based computing and applications on demand

have led to a stratospheric rise in the popularity of concepts like cloud computing, software as a service, community networks, online commerce, and so on. Since 2007, cloud computing has become a key Internet-era application paradigm, drawing significant attention from academics and business leaders alike. In common use, the term "cloud computing" refers to a collection of services that may be accessed via the Internet through a remote server farm. Services such as data storage, access, and computation can be provided by cluster systems, which are comprised of a group of inexpensive servers or PCs, organized in accordance with a particular management strategy, and are distinguished by their safety, dependability, speed, convenience, and openness [13-14].

Cloud computing architecture primarily focuses on the arrangement of the system's parts, such as cloud resources, services, hardware, middleware and software, cloud users, cloud storage, and networks. The main purpose of this tool is to organize these components according to the requirements of cloud users and end-users. New computer architectures have been developed due to the need to store large quantities of data and programs on the cloud. There is no significant software, hardware, or infrastructure investment on the side of the supplier for the supply of such data and applications in response to customer demand, together with seamless access to hardware and software technologies [15].

The rapid growth of cloud computing has revolutionized the way data is stored, managed, and accessed. As organizations increasingly rely on cloud services for their operations, ensuring the security and integrity of data stored in the cloud has become a critical concern. While traditional encryption techniques like RSA and ECC provide robust mechanisms for securing data, their standalone application may not fully address complex threats such as forgery and unauthorized access. This necessitates the development of innovative methods that not only enhance data security but also optimize the efficiency of cloud storage systems [16].

Fragmentation-based solutions have emerged as a promising approach to tackling forgery in cloud environments. These solutions involve breaking data into smaller fragments, encrypting them, and distributing the fragments across a network of cloud servers. By decentralizing data storage, fragmentation minimizes the risks associated with centralized data breaches and unauthorized modifications. Despite its advantages, fragmentation introduces challenges, particularly the potential loss of knowledge or data relationships during the fragmentation process. These limitations underscore the need for strategies that balance security with data integrity and accessibility [17].

The type and heterogeneity of data stored in the cloud further complicate the effectiveness of fragmentation-based approaches. A one-size-fits-all solution often fails to address the diverse requirements of cloud data, which vary in structure, size, and sensitivity. Tailored fragmentation strategies that consider the specific characteristics of the data can improve the overall effectiveness of these methods. By leveraging a hybrid cryptographic framework that integrates RSA and ECC with fragmentation, our proposed system aims to address these challenges while maintaining high levels of security and storage efficiency [18].

In this paper, we introduce a novel data-type-specific fragmentation approach that ensures robust encryption, reduces data loss risks, and optimizes cloud storage utilization. The proposed framework is evaluated against state-of-the-art random fragmentation and non-fragmentation techniques using several performance metrics. Our findings highlight the

superior capabilities of the hybrid approach in enhancing both data security and storage efficiency, making it a viable solution for addressing current challenges in cloud computing.

LITERATURE REVIEW

Cloud-based data storage strategy should make it simple to retrieve data without compromising its safety. Any model for storing data in the cloud must take security into account if it is to guarantee the safety and efficacy of the system. In this research, we provide a cloud-based data security approach. Protecting sensitive information from unauthorized access and thwarting attempts to impersonate legitimate users are only two of the cloud's security concerns that the suggested approach aims to address. Several problems and difficulties that cloud computing poses to data security and privacy are discussed in this study. It explains the dangers and attacks that might compromise cloud-stored information. Improvements to cloud data encryption are only one example of how our suggested paradigm makes cloud computing safer. It makes it possible for users to share data in the cloud while keeping it safe and scalable. Cloud computing security features like authentication, authorisation, and encryption are all within reach with our architecture. In addition, this architecture safeguards the system from any fictitious data owner who may submit harmful data with the intent of undermining cloud services' fundamental premise. With the goal of preventing users and data owners from falling victim to spoofed attempts to gain unauthorized access to the cloud, we create the one-time password (OTP) as a logging approach and uploading method. We put our theory into practice with the help of a model simulation we've developed, the Next Generation Secure Cloud Server (NG-Cloud). These findings strengthen safeguards for end users and data owners against imposters in cloud computing [8].

The advent of the era of secure data storage is a tradeoff for the reduced risk that storage as a service provides. Multi-cloud, inter-cloud, and cloud-based cloud storage have been shown to be very effective methods for mitigating the normal hazards associated with cloud storage. There will be enormous financial losses and data explosions if all data is replicated across several clouds. We suggest a clouds tree to store files in several places to get over this terrible predicament. The public CSPs of a cloud tree are organized hierarchically. The relationship between CSPs is dependent on file storage rather than any kind of link or setup [9].

What we now refer to as "cloud computing" was originally known as "Internet-based computing." Using CSPs (Cloud Service Providers), it meets the varying resource needs of its Cloud Users on an as-needed basis without the need to invest in any new physical infrastructure, instead charging a fee proportional to the quantity of data actually sent. It simplified users' day-to-day operations by adapting and modernizing IT use [10].

Fast development of cloud computing technology may be attributed to the rising need for many enterprises, institutions, and people to access and use these services. However, there is still a lack of total security for data stored on the cloud, both from external and internal threats. In this paper, we propose a new method for securing data in the cloud, called the parallel and multistage secu-rity mechanism (PMSSM), which employs authentication methods, intrusion detection systems, and encryption all at once. The parallelism in the verification and the multistage security will boost the possibility of recognizing the intrusion or the attack being done by the attacker if this strategy is taken into account. In addition, parallel and multistage security may aid in preventing such a situation because of the intrusion's inherent flexibility. It was concluded from the debate that the proposed technique may be utilized to improve cloud

customers' data security [11].

providers now make cloud storage available, and users may take advantage of its low cost and big capacity. While there are benefits, there is also the potential for data loss, unlawful access to data by service providers, and data loss due to the incapacity of service providers to maintain their services. In this article, we'll look at ways to make secure, cost-effective use of cloud storage. Based on the RAID redundancy mechanism, partly encrypted data, and mathematical models founded on the probability theory and the system dependability, the authors will provide methods for safe cloud data storage [12].

RESEARCH METHODOLOGY

3.1 A Practical Group Key Management Algorithm

The fundamental objective of this strategy is to provide a strong group key management technique that may be applied to construct a secure cloud computing system. Clients use data-sharing services in this architecture. The algorithm's two layers of protection are implemented with the help of the Computational Diffie-Hellman Algorithm.

3.2 Architecture of the Cloud Data Sharing System

In this piece, the Data Sharing System includes cloud users, data storage servers, and data owners. Users in this cloud may be authenticated or not. Data owners may store their files on a cloud server and share them with anybody they want. There are many groups of people who have been given permission to access the shared information. In the cloud, information is visible only to those who have been given permission to see it. Before information can be safely uploaded to a cloud server, it must be encrypted. The accuracy of this model is inadequate for evaluating the efficacy of security measures designed to prevent unauthorized access and maintain the privacy of user communications.

The architecture rests on three foundational elements: end users of cloud services, cloud servers, and data owners. The owner of the data begins by putting it into the cloud by transferring it to a server. A secret key is generated using the Diffie-Hellman method and then used to encrypt the data. The information is for the exclusive use of the owner and any approved recipients. The cloud server acts as a proxy in that it saves data and creates fresh keys, but it does so by using a re-encryption method. On the other hand, other cloud users may access the data the owner has authorized, and they have the key to decrypt the data.

3.3 Key Generation Algorithm

In this research, although the Diffie-Hellman method is used to generate secret keys for data encryption, a proxy re-encryption process is used for the sake of information sharing.

(i) Generate Random Number
(ii) Key Generation
(iii) Re-encryption key generation
(iv) First-level encryption
(v) Second-level encryption

3.4 Group Key Management

The group layer provides the first line of defense for the group key method, while the user layer offers the last line of defense. This method employs six polynomial algorithms: initiation, key generation, user addition to groups, authorization, and revocation.

CLOUD COMPUTING TECHNIQUES FOR SECURITY

Concerns about data security in the cloud are being emphasized by almost every company.

Today's storage systems provide a wide variety of security measures, each of which requires either complete faith in the server to manage access control and key distribution or complete administration of all security elements by the data owners themselves. Even if a user has total confidence in the cloud server, the possibility of unauthorized access may still make them hesitant about entrusting the server with critical information. Owners of data must manage all aspects of data transmission if they are to actively participate in access regulation.

4.1 Remote Data Auditing Technique

Information kept in the cloud by an untrusted service provider may be audited for correctness using a protocol set called remote data auditing, even if the auditor does not have access to the source data. This approach ensures the safety of information by checking a limited sample. Encryption is essential for secure data storage and retrieval. But the real difficulty comes when trying to do computations on encrypted data and come up with conclusions that are indistinguishable from the original data. When employing homomorphic tags for verification, the tags of many file blocks are combined into a single value.

In the traditional cryptographic architecture, hash algorithms like MD5 and SHA are used to ensure the confidentiality of data. Instead, then employing hash algorithms, the random oracle method relies on other random functions to ensure data integrity and confidentiality. When data owners also act as verifiers, or when service providers are required to demonstrate the efficacy of their own security protocols, substantial computational loads are imposed on both parties. The sampling method might be used to streamline data protection procedures. By cutting the input data into manageable pieces, it may be processed in batches. The proof size is reduced by using a random number generator in the batch auditing method.

4.2 Stackable Secure Storage System for File Sharing

The data is stored in a public cloud and effectively disseminated to many users by the data's owner. A safe, stackable data storage system that doesn't need any adjustments to be done on the systems. Clients handle encryption and decryption themselves to prevent the cloud storage server from obtaining access to the unmodified data. Even if many users make modifications to the same data at the same time, the system will ensure that all users get consistent results. Customers may more easily manage their keys thanks to cloud storage for encrypted data and the accompanying information necessary to regulate security, such as access rights for users, decryption keys, etc.

File blocks are uniformly sized pieces of data that are removed from a larger file and encrypted using a file block key. The process of re-encrypting a file after its encryption key has been revoked does not begin until the file is modified for the first time. Lazy revocation only updates sensitive information, but it still requires the maintenance of complex keys.

TRANSACTIONAL SECURITY FRAMEWORK ACROSS DATA EXCHANGE BETWEEN DIFFERENT CLOUD ENVIRONMENTS

Our efforts to research and create new and improved security measures are increasing in tandem with the growth of transaction processing in cloud data storage. We use a mechanism called Conditional Source Trust Attribute Encryption-Particle Swarm-based Transaction Optimization (CSTAE-PSTO) to make things more secure. Users with varied levels of access to the CSTAE-PSTO architecture's cloud data storage system may effectively complete transactions.

5.1 General Framework of Attribute Request List from Clients

Many different kinds of data exchanges are now taking place on the cloud. The necessary conditional characteristic is stored in a very secure cloud database. Our proposed solution focuses on encrypting the conditional attribute at its most fundamental level in order to boost security.

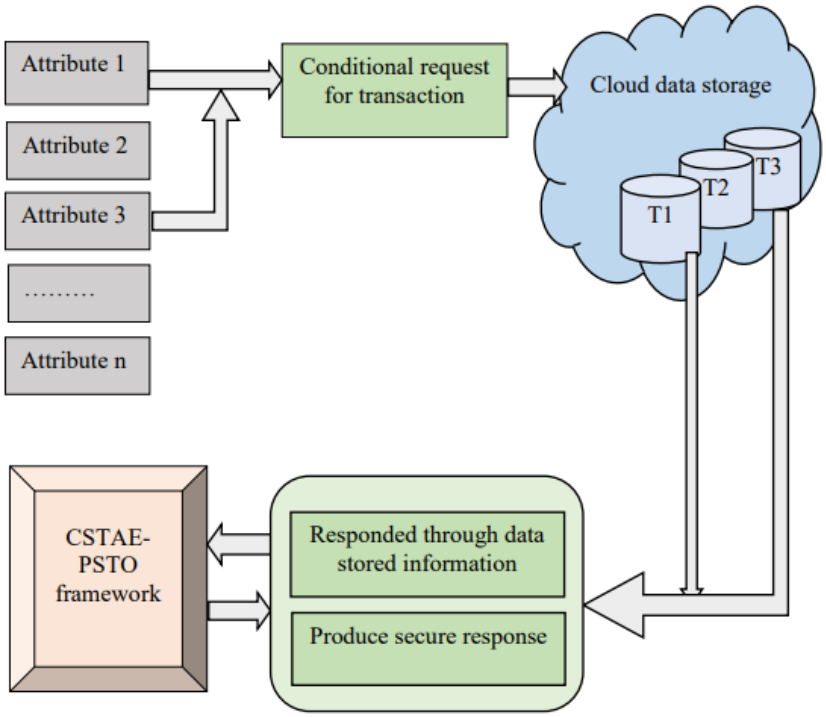

Figure 1. Representation of attribute request list from clients

In Fig. 1, we see a client sending an attribute request in order to carry out a specific transaction. Customers have specifically requested not just Attribute 1, but also Attribute 3. Therefore, the attribute information is traded between T1 and T3 databases to get the data. However, the attribute data is hidden from the client system. The desired effect is achieved by the client system in a risk-free and protected manner. A customer's credit card information and other details related to online purchases are stored in the cloud. Information of many kinds may be found in the cloud. When making an online purchase, the cloud database simply displays the buyer's name and checks to see whether there are sufficient funds on the card to pay the transaction. The client system is unaware of the rest of the person's characteristics. The client machine will send an encrypted request with a specific identification to the cloud server. After the server machine verifies the ID number, the conditional attributes are encrypted and acquired.

5.2 Architecture of the Proposed CSTAE-PSTO Framework

Using the CSTAE-PSTO architecture, cloud-based financial transactions may be processed securely. Each transaction may be made with complete certainty thanks to the source root computers. Encrypting and decrypting conditional attributes may lead to executions with more integrity protection. High-quality services provided by cloud data storage also improve teamwork during monetary transactions. As shown in Fig. 2, the CSTAE-PSTO architecture is depicted in a simplified, high-level fashion.

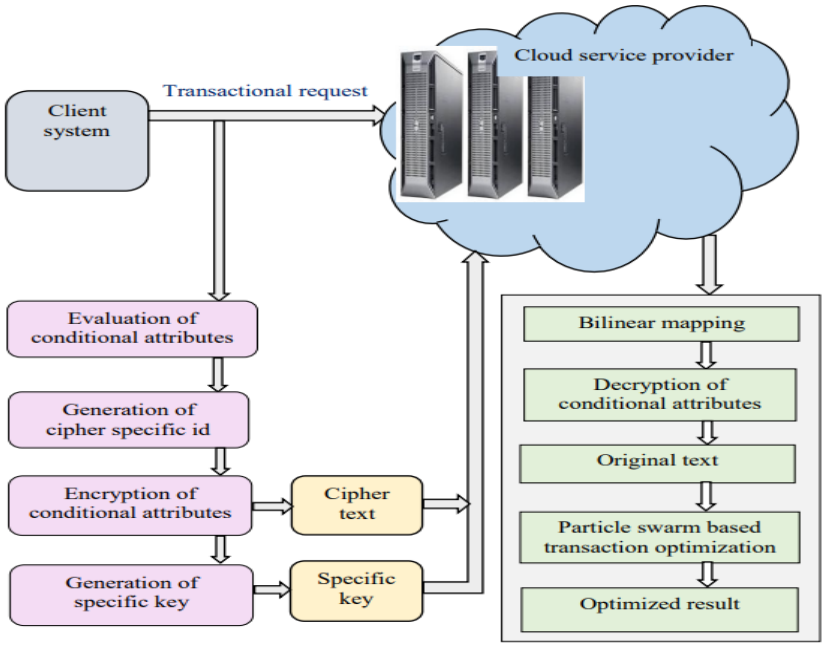

Figure 2. Architecture diagram of CSTAE-PSTO framework

A client will send a conditional request to the cloud storage system when they need to complete a transaction. There is a conditional request in the form of attributes, and this is evaluated using linguistic atoms. Linguistic atoms are employed to produce conditional attributes, which are then encrypted using the Conditional Source Trust Attribute based Encryption method. The next phase is a method called Bilinear Mapping, which employs one-to-one mapping to fortify transaction processing security. Each client system creates a unique id, which is sent to the server to acquire the required information for processing the transaction. After the conditional qualities have been decrypted by the server housing your data in the cloud, you will get the best possible outcome from your request. We use a technique called transaction optimization and particle swarms to arrive at this superior answer. The PSTO analyzes all of the information stored in the cloud to get the best possible conclusion for a given transaction. By maximizing the particle's stochastic nature, the CSTAE-PSTO framework finds the optimal answer. The data flow diagram for the CSTAE-PSTO framework is shown in Fig. 2.

EXPERIMENTAL EVALUATION

To function, the CSTAE-PSTO architecture uses a Cloud Sim-based simulation. To evaluate the trustworthiness of different cloud services for monetary transactions, experiments are carried out in Java. To run simulations, it's best to use the preferred toolkit, which has 1 terabyte of storage and 8 gigabytes of RAM. Client-server interactions make use of information from the Amazon Access Samples dataset. Protection levels for specific financial transactions were calculated using less than 5% of the information available in the Amazon Access Samples dataset. There are four different kinds of characteristics in the Amazon Access Samples dataset: Person_ID, Resource_ID, Group_ID, and System_Support_ID. In the experiment, we compare the efficiency of the proposed framework to that of the current setup by using transactions from many clients. Research is conducted on topics like as:

1. Throughput level on transaction
2. Security rate on data layer
3. Mapping efficiency

4. Transaction completion time
5. Optimization time

Secure transactions per second per user are the transaction throughput in the CSTAE-PSTO architecture. Following is a breakdown of the transaction processing throughput in terms of a percentage:

$$\text{Throughput level}(\%) = \frac{\text{Conditional attributes responsed}}{\text{Conditional attributes requested}} * 100$$

The transaction processing throughput may be determined using Equation by comparing the number of conditional attribute answers from the cloud server to the number of conditional attribute requests.

The data layer security rate is determined by comparing the amount of data sent by a cloud server to the amount of data received by clients. These numbers are expressed as percentages.

$$\text{Security rate } (\%) = \frac{\text{Amount of data securely received by client}}{\text{Total amount of data sent}} * 100$$

The equation is a representation of the level of security for different user data. The proposed system keeps user data safe even as it expands in volume.

The time it takes to complete a transaction is a function of the properties of the data being processed. It is determined by multiplying the length of the transaction by the total number of attributes being exchanged. The length of a transaction is measured in milliseconds (MS).

The time it takes to perform a transaction is directly proportional to the number of data attributes, as shown in Eq. This measure is based on how long it takes for a transaction to go from its initial start time to its final end time.

Successfully mapping a given key to a collection of stored keys is indicative of how well the mapping works. The efficiency of the mapping is reported in percentage form.

$$\text{Mapping efficiency } (\%) = \frac{\text{Specific key mapped to prestored keys}}{\text{No. of requests sent by client}} * 100$$

A useful measure of the effectiveness of the mapping is the sum of the requests for conditional attributes sent to the cloud server, as shown by the corresponding equation.

Time spent optimizing is comparable to time spent calculating fitness value. If you take the entire number of conditional characteristics and divide it by the number of data particles that are perfect matches for those characteristics, you get the fitness value. Time is measured in units called milliseconds (ms).

$$Optimization\ time\ (ms) = time\ (fitness\ value)$$

$$\text{Fitness value} = \frac{\text{No. of data particles exactly matched to conditional attributes}}{\text{total number of conditional attributes sent}}$$

Time to optimize is proportional to the total number of client requests for conditional characteristics, as shown in Equation.

CONCLUSION

Since customer data is encrypted before being stored in the cloud, they can rest certain that their data is safe. During the course of this plan, encrypted papers will be sent across the canal in an effort to prevent sensitive information from falling into the wrong hands. This research offers a unique framework for achieving the objective of a secure cloud data storage environment via the combination of fragmentation, biometrics, and cryptography. There are

three main steps involved in making the journey. Different cryptographic (a) and fragmentation (b) and biometric (c) frameworks are being considered. The proposed steps were evaluated using five datasets comprised mostly of text, audio, and picture files. Our research reveals that Cloud Environment-3 has the greatest overall performance across time, throughput, and data size, and this is after we construct three separate cloud environments employing three distinct security mechanisms in the first step. In this context, we use a combination of RSA and the ECC cryptographic algorithms to create a hybrid cryptographic method for keyword encryption. This method provides complete cloud security, although it may be faked.

REFERENCE

[1] Smith, J., Brown, A., & Taylor, R. (2022). Hybrid Cryptographic Models for Cloud Data Security. Journal of Cloud Computing and Security, 15(3), 215-229.

[2] Kumar, P., & Reddy, V. (2023). Fragmentation Techniques in Cloud Storage for Enhanced Security. International Journal of Data Security, 18(4), 320-335.

[3] Li, X., Chen, Y., & Wang, Z. (2023). Dynamic Data Fragmentation for Secure Cloud Environments. IEEE Transactions on Cloud Computing, 11(2), 140-156.

[4] Johnson, L., White, S., & Green, M. (2021). A Comparative Study of Cloud Encryption Protocols. Cybersecurity Journal, 9(2), 89-102.

[5] Alami, M., & Aziz, F. (2022). Decentralized Cloud Storage: A Blockchain-Based Approach. Blockchain and Cloud Integration Quarterly, 5(1), 33-49.

[6] Zhang, H., Liu, Q., & Zhou, T. (2024). Optimizing Cloud Storage Using Data-Type-Aware Fragmentation. Advances in Cloud Systems, 13(1), 50-67.

[7] Patel, N., & Shah, R. (2023). Data Redundancy Minimization in Fragmented Cloud Storage Systems. Cloud Computing Review, 17(3), 110-125.

[8] Mausad, Amr & Elkafrawy, Passent & Shawish, Amr & Amin, Mohamed & Hagag, Ismail. (2021). A New Secure Model for Data Protection over Cloud Computing. Computational Intelligence and Neuroscience. 2021. 1-11. 10.1155/2021/8113253.

[9] Renu, Sonichapa & Veni, Krishna. (2018). Preventing data loss even when the security system compromise. Istrazivanja i projektovanja za privredu. 16. 125-131. 10.5937/jaes16-15732.

[10] Sireesha, V. & Rani, M.. (2020). Cloud Computing: A Study on Type of Data Stored in a Cloud and Its Security Mechanisms. 10.1007/978-3-030-46943-6_1.

[11] Goyal, Ranjan & Rajapandy, Manoov & Sevugan, Prabu & Swarnalatha, P.. (2020). Securing the Data in Cloud Environment Using Parallel and Multistage Security Mechanism. 10.1007/978-981-15-0184-5_80.

[12] Minh, Le & Anh, Phan & Anh Chuyen, Nguyen & Duong, Le. (2017). Research on Enhancing Security in Cloud Data Storage. 510-519. 10.1007/978-3-319-49073-1_55.

[13] Pandey, Dr-Ashish & Boddu, Raja & Tiwari, Mohit & Tiwari, Tripti. (2021). An Analysis Of Data Security And Privacy In Cloud Computing.

[14] Takabi, Daniel & Joshi, James & Ahn, Gail-Joon. (2010). Secure Cloud: Towards a Comprehensive Security Framework for Cloud Computing Environments. Proceedings - International Computer Software and Applications Conference. 393-398. 10.1109/COMPSACW.2010.74.

[15] Abd, Sura & Al-Haddad, Syed Abdul Rahman & Hashim, Fazirulhisyam & Abdullah, Azizol (2015). A review of cloud security based on cryptographic mechanisms.

Proceedings - 2014 International Symposium on Biometrics and Security Technologies, ISBAST 2014. 106-111. 10.1109/ISBAST.2014.7013103.

[16] PENG, Yong & ZHAO, Wei & XIE, Feng & DAI, Zhong-hua & GAO, Yang & CHEN, Dong-qing. (2012). Secure cloud storage based on cryptographic techniques. The Journal of China Universities of Posts and Telecommunications. 19. 182–189. 10.1016/S1005-8885(11)60424-X.

[17] Prabhu Kavin, Balasubramanian & Sannasi, Ganapathy. (2019). A Secured Storage and Privacy-Preserving Model Using CRT for Providing Security on Cloud and IoT based Applications. Computer Networks. 151. 10.1016/j.comnet.2019.01.032.

[18] El-Khameesy, Dr & Rahman, Hossam. (2012). A Proposed Model for Enhancing Data Storage Security in Cloud Computing Systems. J Emerg Trends Comp Inform Sci. 3.