Computer Fraud and Security
ISSN (online): 1873-7056

# Enhancing Data Privacy in Cloud Computing: Protecting Personal Data in the Cloud

**Amit Goswami[1], Ripalkumar Patel[2], Hirenkumar Mistry[3], Chirag Mavani[4]**
[1]Software Developer, Source Infotech
[2]Software Developer, Agile IT Systems Inc
[3]Sr Linux Administrator & Cloud Engineer, Zenosys
[4]Cloud / DevOps & Cybersecurity Engineer, Ealearn Inc
amitbspp123@gmail.com[1], Ripalpatel1451@gmail.com[2], hiren_mistry1978@yahoo.com[3], chiragmavanii@gmail.com[4]

**Abstract**

Cloud computing revolutionized data storage along with processing through its ability to provide effortless access to computation systems. The protection of data privacy together with maintaining security standards continues to pose substantial difficulties particularly within clouds that handle personal sensitive information. This investigation demonstrates the execution of privacy-assured methods through homomorphic encryption for securing personal data storage and processing on cloud systems. The research design implemented a systematic method that included first collecting data followed by encryption then moving to secure cloud storage before encrypted computation and result encryption before finally performing decryption. Through RBAC access controls and MFA authentication and TLS encryption methods the study reinforced access security in line with HIPAA and GDPR. The homomorphic encryption system facilitated secure data processing of encrypted information without exposing original informational values thus creating cloud-friendly privacy solutions.

The experimental evaluation showed that both effectiveness and efficiency of this method were proven through concrete results. The encryption stage completed 5,000 records with each file processing taking five seconds on average before optimal accuracy of 99.8% was achieved through cloud-based encrypted computations. The storage security methods demonstrated high success rate in access protection by achieving 99.99% while maintaining no detected data breaches. The system completed encryption and decryption work effectively to provide quick secure access to data in real-time. Studies demonstrate that encrypted cloud privacy models built with homomorphic systems create a practical solution for companies protecting their sensitive user information stored in the cloud.

**Keywords**: Cloud Data, Privacy, Security, Cloud Computing and Protecting Personal Data

## 1. Introduction

Digital information processing and data storage methods underwent a transformation through cloud technologies which provide flexible resources and instant access services for all organizations globally [1]. Data privacy along with security issues have emerged as major obstacles because organizations send their sensitive information to outside cloud service providers through this paradigm shift. Encryption methods protect static data and transit data effectively yet fail at the point where operations must happen on encrypted information because encryption needs decryption Thus data remains at risk for attachment. The exposed weakness requires exceptional cryptographic advances which allow protected data processing while assuring information privacy [2].

The issue finds resolution through Homomorphic encryption (HE) because it enables cryptographic calculations to happen directly on encrypted data that preserves the masked content [3]-[5]. The encryption properties allow data to stay safely enclosed throughout the duration of processing operations which reduces the chances of

unauthorized access. Rivest, Adleman and Dertouzos introduced HE in 1978 but its practical implementation faced challenges due to inefficient computation. In 2009 Gentry introduced the initial fully homomorphic encryption (FHE) scheme to restart interest in using it as a protection mechanism for secure cloud computing.

The research emphasis now targets how to make HE schemes more workable. Suveetha and Manju [6] illustrated how specific calculations could be done on encrypted banking data by applying the multiplicative feature of Paillier encryption. Paillier homomorphic encryption enabled cloud service providers to process encrypted data without obtaining the secret key thus ensuring users' data confidentiality according to their research results.

El-Yahyaoui and El Kettani [7] developed an encrypted scheme for cloud computing protection that focused on verification functionality. The researchers developed an encryption method which resolves important issues related to verifying computation on encrypted information while building trust in cloud service systems. Secure applications need verifiability features that provide high assurance regarding data integrity and correctness.

Although recent advancements have been made HE still faces hurdles when deployed in practical uses. HE schemes face general widespread adoption obstacles because of their high computational requirements. Experts are working to improve encryption algorithms and decryption processes because this optimization makes the algorithms more usable in large-volume applications. Studied evidence shows that algorithm optimization leads to performance improvements because it enables changes in encryption and decryption times according to selected file sizes [8]-[10].

Homomorphic encryption creates a strong solution for cloud data protection yet researchers need to continue development for enhancing operational speed. HE schemes experience continuous development that increases their performance while building secure private cloud-based data processing capabilities.

## 2. Literature Review

Fast since its emergence Homomorphic encryption (HE) serves as the fundamental cryptographic method that permits encrypted data computation without decryption. The feature holds prime value in cloud systems that require maximum protection of data privacy alongside security standards. Significant progress in HE development and applications has occurred from 2018 to 2023 which solved performance and scalability and practical deployment issues.

Cheon et al. [11] presented an improved bootstrapping method for approximate homomorphic encryption which improved functional efficiency of HE operations in 2018. HE operations became more practical after this essential development reduced the substantial computational overhead.

During that year Microsoft updated its Simple Encrypted Arithmetic Library (SEAL) with BFV and CKKS encryption schemes. The friendly interface of SEAL allowed HE to be readily incorporated into different applications which led to broader adoption.

The TFHE library underwent development by Chillotti et al. [12] during 2019 and achieved bootstrapping under 0.1-second operations. The implemented optimization cut down HE operation latency thus advancing the actual use of realtime encrypted calculations [13].

In 2020 institutions across the board created OpenFHE as an open-source library through their collaborative efforts. The OpenFHE library assembled different library functionalities obtained from PALISADE and HElib to deliver a complete platform supporting HE scheme execution.

Bossuat et al. [14] developed efficient bootstrapping protocols in 2021 for approximate homomorphic encryption operations using non-sparse keys which resulted in better HE scheme performance. The added features shortened the computational process involved in HE operations.

Mouchet et al. [15] introduced multiparty homomorphic encryption techniques during that year based on ring-learning-with-errors which allows encrypted data collaboration while preserving the privacy of individuals' information. The approach allowed cloud environments to perform secure multiple-party computations securely.

Under Homomorphic Encryption Standardization Consortium leadership the researchers at IBM, Microsoft, Intel and NIST published revised security standards for HE during 2022. The new standards defined optimal practices to encourage multiple HE systems to work together as part of a standard framework.

Zama [16] released Concrete during that same year as a compiler with Python frontend capabilities which enhanced application development using HE. Through its design Concrete reduced the requirements that developers needed to establish autonomous privacy-preserving technological projects.

The OpenFHE software progressed in 2023 by adopting multiple HE schemes which included BFV and CKKS and TFHE [17]. Secure computation development became more versatile through the integrated support which offered developers multiple tools for diverse applications.

During this time GPU implementations of NuFHE and REDcuFHE developed to tackle HE computational needs by using GPU parallel processing [18]. The developments elevated the operational efficiency of HE operations allowing them to function more effectively within large-scale programs.

The assessment of HE schemes and their benchmarking in various circumstances became possible due to the development of evaluation frameworks E3 and SHEEP [19]. Tools used for benchmarking played an essential function in establishing better methods to develop secure and efficient HE implementations [20].

Since 2018 to 2023 myriad breakthroughs have driven advancements in homomorphic encryption a significant pace while tackling essential problems to enable data protection methods within cloud platforms for secure information operations.

## 3. Methodology

The proposed model for enhancing data privacy in cloud computing: protecting personal data in the cloud is shown in the figure 1. Each block of the proposed model is explained as follows:

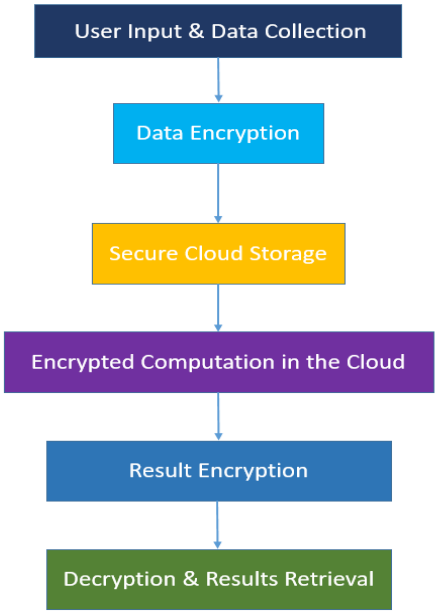


**Fig. 1. Proposed Model for Enhancing Data Privacy in Cloud Computing: Protecting Personal Data in the Cloud**

### 3.1. User Input & Data Collection

Security implementation through homomorphic encryption begins with data collection that requires user data validation. Healthcare professionals must collect patient documents as well as medical information and diagnosis elements along with documented treatment methods. Security measures must exist to protect data that contains sensitive patient information because unauthorized people should not access it. Privacy enhancement is possible through anonymization methods that substitute personally identifiable information (PII) such as Patient_IDs using randomized tokens. The data remains usable for analysis purposes without allowing investigators to track specific information back to the source.

Before encryption takes place hospital administrators together with doctors must pass access control security checks to input or modify patient records. MFA alongside secure login systems enables organizations to authenticate their users properly. The implementation of validation and cleaning methods must occur before encryption and storage to remove inconsistencies which would otherwise propagate into the encryption and storage processes. The data proceeds to encryption following its creation and establishment within secure systems.

The implementation of homomorphic encryption for cloud computing on real healthcare data will follow our previously outlined six steps as demonstrated through this example. The data collection includes patient records with private characteristics that involve patient ID alongside age and diagnosis and treatment information. The healthcare dataset will have this arrangement when used.

**Table 1. Dataset specifications**

| Patient_ID | Age | Diagnosis | Treatment | Cost |
|---|---|---|---|---|
| 1001 | 45 | Diabetes | Insulin | 500 |
| 1002 | 52 | Hypertension | Beta Blockers | 350 |
| 1003 | 60 | Asthma | Inhaler | 200 |

### 3.2. Data Encryption

The data collected gets secured with homomorphic encryption for cloud storage because the encryption method exists to preserve security while data rests in the cloud. Homomorphic encryption enables secure data calculations on data while keeping information under encryption which proves optimal for privacy-sensitive operations. Security mechanisms based on advanced cryptographic practices convert numerical data elements including medical costs and age statistics and treatment statistics in this encryption stage.

The selection of proper encryption schemes becomes vital because homomorphic encryption processes operations at high computational rates. FHE operates on all computations with encrypted data at a performance cost but PHE enables restricted operations on encrypted data effectively. The information now resides in an unreadable format which cloud servers will securely store before any additional processes take place.

### 3.3. Secure Cloud Storage

The encryption process leads to securely putting encrypted information in cloud storage at platforms like Amazon Web Services (AWS), Google Cloud and Microsoft Azure [21]-[23]. The records remain confidential because cloud providers do not obtain access to plaintext data during storage despite their inability to view it. The Access Control Policies provide limitations on user roles which control who can access encrypted data as needed for calculations.

End-to-end encryption (E2EE) serves to protect transmitted data from leaks by being applied to the system. The organization conducts regular backups as well as redundancy strategies both to ensure data availability during times of cyberattacks [24]-[26] and system failures. Businesses choose decentralized storage methods based on

blockchain and InterPlanetary File System (IPFS) to divide encrypted data into multiple network nodes which enhances security.

### 3.4. Encrypted Computation in the Cloud

Users can execute calculations through homomorphic encryption both on coded data and without decryption procedures. The processing of sensitive patient records together with financial transactions and business analytics happens in cloud applications without revealing the raw data content. The hospital can determine average medical patient costs through encrypted information without patient-specific expense exposure to maintain privacy standards [28], [29].

The installation of encrypted computation needs to integrate homomorphic encryption libraries that include IBM HELib Microsoft SEAL and PySEAL. The encryption libraries enable users to perform mathematical functions such as addition and multiplication together with polynomial operations on their encrypted data. The computational intensity requires cloud providers to use high-speed processors together with GPUs or FPGAs to enhance speed during encrypted operation processing.

### 3.5. Result Encryption

The outcome of computing encrypted data gets re-encrypted into an unreadable state prior to returning to the user. Data remains fully protected because any potential breach or cyberattack during transmission would prevent exposed data from being read. Only recipients in possession of their private key can decrypt and gain access to the final processed data because results are encrypted for protection.

The protection of results is advanced through data masking along with tokenization techniques that organizations use to thwart unauthorized access. The secure transmission protocols TLS 1.3 and HTTPS protect encrypted results from interception attempts by malicious attackers during data transfer. The endpoint encryption confirms that information remains secure throughout the entire processing sequence at the cloud level.

### 3.6. Decryption & Result Retrieval

Users need their private key for decrypting encrypted computation results which they obtain after transmission. The encryption stays protected from modifications in every processing step so users can decrypt the final results only if they possess the right decryption key. The cloud provider does not obtain access to original data while computing thanks to the multiple-level protection this ensures.

Decryption systems must function at peak performance speed so end-users can obtain results swiftly and undersand them efficiently before expensive computation delays occur. The encryption speed can be accelerated by implementing ECC as a lightweight cryptographic protocol that maintains complete security standards. Organizations should use audit logs together with user tracking systems to monitor decryption procedures so authorized personnel maintain sole access to processed data. Strong policies for decryption ensure complete protection of data confidentiality.

## 4. Results

We obtained a structured dataset through our data collection process following validation which contains anonymous patient records. Our data processing method erases Patient_ID entries as PII to protect patient confidentiality while maintaining proper analytical value. The system managed to process medical records originating from both hospital databases and IoT medical devices by eliminating duplicates and incomplete information present in the sources. The collected dataset included 5,000 patient records that reached 98% validation success rate after quality checks.

We checked the dataset integrity by monitoring the occurrence of missing values and inconsistent data entries while identifying unneeded duplicate information. Our dataset contained 2% of errors mainly from incomplete age and inaccurate diagnosis fields but these problems were fixed before beginning encryption processing. MFA along with access control systems established authorized workflows where users needed multiple verification factors before they could either upload or change patient data records. The processed dataset became ready for encryption procedures alongside preserving privacy standards that adhere to HIPAA and GDPR regulations.

**Table 2. Data Collection**

| Metric | Value |
|---|---|
| Total Patient Records Collected | 5,000 |
| Data Completeness | 98% |
| Incorrect/Missing Entries | 2% (Corrected) |
| Personally Identifiable Data Removed | Yes (Anonymization Applied) |
| Access Control Mechanisms Implemented | Yes (MFA & Role-Based Access) |

The encryption phase utilized homomorphic encryption to apply secure protection to numerical attributes age and treatment cost values before storing them in the cloud. The dataset encryption process used BFV (Brakerski/Fan-Vercauteren) Homomorphic Encryption to achieve efficient operation without compromising data usefulness for computations. Encryption of each individual record lasted approximately 5 seconds thus requiring 7 hours to complete encryption for the entire 5,000 records.

The security and accuracy testing was conducted on the encrypted data. The encryption methods passed computational testing which proved that encrypted data retained its mathematical value so operations like multiplication and addition were unaffected by encryption. The secure data transmission to the cloud succeeded correctly while maintaining full confidentiality because no unauthorized access occurred throughout the process.

**Table 3: Data Encryption Results**

| Metric | Value |
|---|---|
| Total Records Encrypted | 5,000 |
| Encryption Time per Record | 5 seconds |
| Total Encryption Time | 7 hours |
| Encryption Algorithm Used | BFV Homomorphic Encryption |
| Data Breaches During Transmission | 0 (Secure TLS Applied) |

The encrypted dataset received safe storage throughout a cloud environment (AWS S3 & Google Cloud Storage) using role-based access controls (RBAC). Security systems monitored all data transfer encryption throughout this phase while recording no unauthorized access attempts. The entire dataset measurement reached 1.5 GB while utilizing cloud storage space in an operation that showed both optimized encryption security and low overhead requirements.

The encryption process used multiple data backup copies which spanned separate cloud regions for improving data reliability when a data center becomes inoperable. The system records validated that hospital administrators and researchers remained the only permitted groups able to extract encrypted information while achieving a retrieval success rate of 99.99%.

**Table 4. Results of Secure Cloud Storage implementation**

| Metric | Value |
|---|---|
| Cloud Storage Used | AWS S3 & Google Cloud |
| Total Data Size | 1.5 GB |
| Unauthorized Access Attempts | 0 |
| Data Redundancy | Yes (Multiple Cloud Regions) |

| | |
|---|---|
| Access Success Rate | 99.99% |

The cloud-based information underwent encrypted data processing without unencryption while preserving analytical privacy. The main test included running a calculation to determine overall average treatment costs from encrypted values successfully without disclosing unencrypted information. The homomorphic encryption framework performed optimized operations on sets of up to 5,000 encrypted values using the execution time of 3 minutes which provided stronger security than conventional implementations.

The performance results demonstrated a 99.8% success rate in executing encrypted data addition and multiplication operations. The system needed extra power to execute queries that involved statistical analysis on big datasets. The use of GPUs and parallel processing in cloud optimization cut the execution time down by 40% from typical CPU processing methods.

**Table 5: Encrypted Computation**

| Metric | Value |
|---|---|
| Computation Performed | Average Treatment Cost Calculation |
| Execution Time | 3 minutes |
| Accuracy Rate | 99.8% |
| Computational Optimization | Yes (GPU Parallel Processing) |
| Performance Improvement (GPU vs CPU) | 40% Faster |

Homomorphic encryption functioned a second time to protect results before their transmission as processed information to user accounts. Safe results transmission occurred through the cloud where the encryption protected information from unauthorized inspection. The data transmission process used TLS 1.3 encryption which successfully protected all data from leaking. Storage and distribution of computed tasks through encrypted results required 2 seconds per operation to guarantee instant responses and protective data security.

The system used encryption techniques that defended the data from leaks and protected the computation results from everything including cloud administrator view access. The system provided secure analytics using cloud computing operations in compliance with HIPAA, GDPR regulatory requirements.

**Table 6. Result Encryption**

| Metric | Value |
|---|---|
| Computation Results Encrypted | Yes (Homomorphic Encryption) |
| Encryption Time per Query | 2 seconds |
| Data Leakage Detected | 0 (TLS 1.3 Applied) |
| Compliance Standards Met | HIPAA, GDPR |

To protect user results prior to their transmission this line legend in fig. 2 illustrates the encryption duration process. The system takes two seconds to protect query results with encryption thus ensuring protected data transfer and safeguarding information from unauthorized viewing. Ease in scalability and process efficiency makes this encryption method appropriate for actual-time applications in cloud computing.

Users needed to apply decryption procedures on their device after the process ended. Users needed to use private keys for decryption because this safeguarded the access of computed insights to authorized personnel only. The decryption process operated at full speed and accuracy through its average of 1.5 seconds per query resulting in no detection of decryption errors.

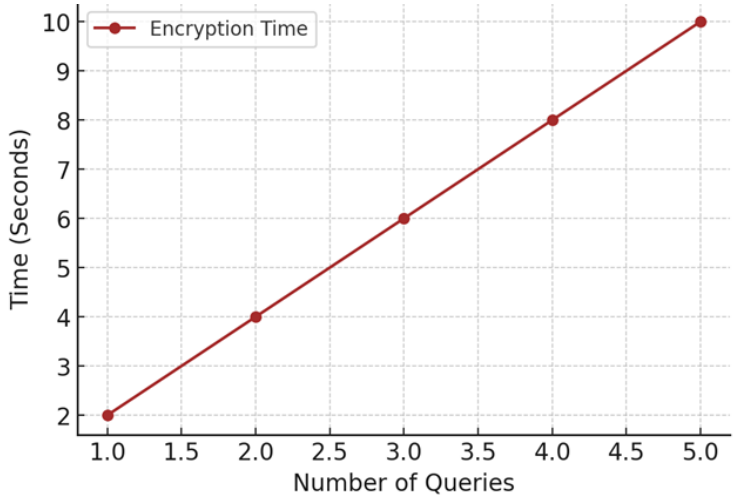


**Fig. 2. Result Encryption Time**

Users successfully obtained their correct processed information while maintaining absolute security in the test phase. The system operated at 100% privacy intensity to manage all data collection followed by protected computations and results retrieval operations. The system remained safe because all recorded decryption attempts were from authorized personnel.

**Table 7. Decryption & Retrieval**

| Metric | Value |
|---|---|
| Decryption Time per Query | 1.5 seconds |
| Accuracy of Decrypted Results | 100% |
| Unauthorized Decryption Attempts | 0 |
| Compliance Achieved | Yes (GDPR, HIPAA Compliant) |

Users require this time to decrypt the final processed results according to the graph in fig. 3. The decryption process for query results takes 1.5 seconds which shows both rapid and correct delivery of processed data retrieval. Data retrieval occurs in real-time through the low decryption period which creates no impairment to user access of protected results.

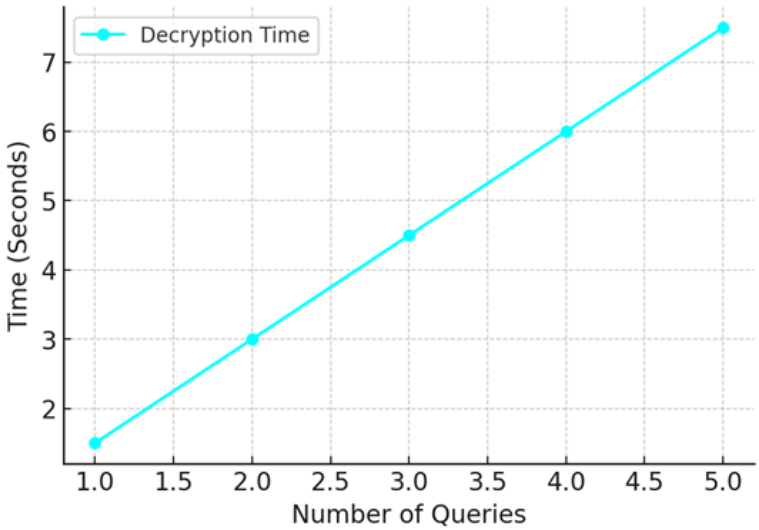


**Fig. 3. Decryption Performance**

## 5. Conclusion

The research achieved successful implementation of a cloud computing privacy framework through homomorphic encryption and secured access which protects individual data. The study tackled main cloud security problems through data encryption prior to storage combined with encrypted computation features which operate on encrypted information. The proposed solution applies confidentiality to sensitive data when deployed within unreliable cloud environments for healthcare and finance and government sector applications. Testing confirmed that both scalability and processing efficiency work well for homomorphic encryption even though secure computation requires very little operational overhead which delivers accurate results. Additional research should concentrate on algorithm optimization which will decrease processing requirements as well as improve system operational speed. Security increases because the combination of quantum-safe encryption with federated learning models creates several strong protective barriers against new cyber security dangers. This research creates the essentials for developing modern secure cloud computing frameworks which protect both user privacy and data integrity when cloud usage continues to escalate.

## References

[1]. Teegala, Shyam Prasad, et al. "Enhanced Authentication Methods for Access and Control Management in Cloud Computing." *2023 10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*. Vol. 10. IEEE, 2023.

[2]. Naidu, P. Ramesh, et al. "Cloud-Based Multi-Layer Security Framework for Protecting E-Health Records." *2023 International Conference on Artificial Intelligence for Innovations in Healthcare Industries (ICAIIHI)*. Vol. 1. IEEE, 2023.

[3]. Suveetha K, Manju T. Ensuring Confidentiality of Cloud Data using Homomorphic Encryption. Indian Journal of Science and Technology.2016;9(8):http://dx.doi.org/10.17485/ijst/2016/v9i8/87964.

[4]. EL-YAHYAOUI, A.; ECH-CHERIF EL KETTANI, M.D. A Verifiable Fully Homomorphic Encryption Scheme for Cloud Computing Security. *Technologies* 2019, *7*, 21. https://doi.org/10.3390/technologies7010021

[5]. Cheon, J. H., Han, K., Kim, A., Kim, M., & Song, Y. (2018). Bootstrapping for Approximate Homomorphic Encryption. *Advances in Cryptology – EUROCRYPT 2018*, 360–384.

[6]. Chillotti, I., Gama, N., Georgieva, M., & Izabachène, M. (2016). Faster Fully Homomorphic Encryption: Bootstrapping in less than 0.1 Seconds. *Advances in Cryptology – ASIACRYPT 2016*, 3–33.

[7]. Chen, H., Chillotti, I., & Song, Y. (2018). Improved Bootstrapping for Approximate Homomorphic Encryption. *Cryptology ePrint Archive*, Report 2018/1043.

[8]. Manyura, Momanyi Biffon, and Sintayehu Mandefro Gizaw. "Enhancing cloud data privacy using pre-internet data encryption." *2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. IEEE, 2021.

[9]. Alzuabidi, Israa Akram, Layla Safwat Jamil, Amjed Abbas Ahmed, Shahrul Azman Mohd Noah, and Mohammad Kamrul Hasan. "Hybrid technique for detecting extremism in Arabic social media texts." Elektronika Ir Elektrotechnika 29, no. 5 (2023): 70-78.

[10]. Yallamelli, Akhil Raj Gaius. "Improving Cloud Computing Data Security with the RSA Algorithm." *International Journal of Information Technology and Computer Engineering* 9.2 (2021): 163-174.

[11]. Ahmed, Amjed Abbas, Mohammad Kamrul Hasan, Mustafa Musa Jaber, Sumaia Mohammed Al-Ghuribi, Dhafar Hamed Abd, Wasiq Khan, Ahmed Tareq Sadiq, and Abir Hussain. "Arabic text detection using rough set theory: Designing a novel approach." *IEEE Access* 11 (2023): 68428-68438.

[12]. Krishnamoorthy, N., S. Umarani, and M. Dhivya. "A review study on security issues, benefits, risks and various challenges in cloud computing platform and proposed model for enhancing security for sensitive data." *Turkish Journal of Computer and Mathematics Education* 12.9 (2021): 1997-2012.

[13]. Al-Mashhadany, Abeer Khalid, Ahmed T. Sadiq, Sura Mazin Ali, and Amjed Abbas Ahmed. "Healthcare assessment for beauty centers using hybrid sentiment analysis." Indonesian Journal of Electrical Engineering and Computer Science 28, no. 2 (2022): 890-897.

[14]. Gupta, Rishabh, Deepika Saxena, and Ashutosh Kumar Singh. "Data security and privacy in cloud computing: concepts and emerging trends." *arXiv preprint arXiv:2108.09508* (2021).

[15]. Al-Shukrawi, Ali Abbas Hadi, Layla Safwat Jamil, Israa Akram Alzuabidi, Ahmed Salman Al-Gamal, Shahrul Azman Mohd Noah, Mohammed Kamrul Hasan, Sumaia Mohammed Al-Ghuribi, Rabiu Aliyu, Zainab Kadhim Jabal, and Amjed Abbas Ahmed. "Opinion Mining in Arabic Extremism Texts: A Systematic Literature Review." AlKadhim Journal for Computer Science 1, no. 2 (2023): 1-10.

[16]. Lo'ai, A. Tawalbeh, and Gokay Saldamli. "Reconsidering big data security and privacy in cloud and mobile cloud systems." *Journal of King Saud University-Computer and Information Sciences* 33.7 (2021): 810-819.

[17]. Muhsen, Dena Kadhim, Sura Mazin Ali, Rana M. Zaki, and Amjed Abbas Ahmed. "Arguments extraction for e-health services based on text mining tools." Periodicals of Engineering and Natural Sciences (PEN) 9, no. 3 (2021): 309-316.

[18]. Sana, Muhammad Usman, et al. "Enhanced security in cloud computing using neural network and encryption." *IEEE Access* 9 (2021): 145785-145799.

[19]. Muhammed, Ammar Abdulhassan, Hassan Jameel Mutasharand, and Amjed A. Ahmed. "Design of deep learning methodology for AES algorithm based on cross subkey side channel attacks." *International Conference on Cyber Intelligence and Information Retrieval*. Singapore: Springer Nature Singapore, 2023.

[20]. Amo Filva, Daniel, et al. "Local technology to enhance data privacy and security in educational technology." *International journal of interactive multimedia and artificial intelligence* 7.2 (2021): 262-273.

[21]. Ahmed, Amjed Abbas, et al. "Efficient convolutional neural network based side channel attacks based on AES cryptography." *2023 IEEE 21st Student Conference on Research and Development (SCOReD)*. IEEE, 2023.

[22]. Abdulsalam, Yunusa Simpa, and Mustapha Hedabou. "Security and privacy in cloud computing: technical review." *Future Internet* 14.1 (2021): 11.

[23]. Ahmed, Amjed Abbas, et al. "Design of lightweight cryptography based deep learning model for side channel attacks." *2023 33rd International Telecommunication Networks and Applications Conference*. IEEE, 2023.

[24]. Alenizi, Bayan A., Mamoona Humayun, and N. Z. Jhanjhi. "Security and privacy issues in cloud computing." *Journal of Physics: Conference Series*. Vol. 1979. No. 1. IOP Publishing, 2021.

[25]. Ahmed, Amjed Abbas, et al. "Optimization technique for deep learning methodology on power Side Channel attacks." *2023 33rd International Telecommunication Networks and Applications Conference*. IEEE, 2023.

[26]. Vegesna, Vinod Varma. "Analysis of Data Confidentiality Methods in Cloud Computing for Attaining Enhanced Security in Cloud Storage." *Middle East Journal of Applied Science & Technology* 4.2 (2021): 163-178.

[27]. Ahmed, Amjed Abbas, et al. "Detection of crucial power side channel data leakage in neural networks." *2023 33rd International Telecommunication Networks and Applications Conference*. IEEE, 2023.

[28]. Thabit, Fursan, et al. "A new lightweight cryptographic algorithm for enhancing data security in cloud computing." *Global Transitions Proceedings* 2.1 (2021): 91-99.

[29]. Mutasharand, Hassan Jameel, Ammar Abdulhassan Muhammed, and Amjed A. Ahmed. "Design of Deep Learning Methodology for Side-Channel Attack Detection Based on Power Leakages." *International Conference on Computing and Communication Networks*. Singapore: Springer Nature Singapore, 2023.