Journal of Theoretical and Applied Information Technology
15th March 2026. Vol.104. No.5

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REAL-TIME CONTEXT AWARE VARIABLE STRENGTH CRYPTOGRAPHY FOR CLOUD STORAGE

[1]Dr.**A.BARAKATH BEGAM,** [2]Dr.A.**BHUVANESHWARI,** [3]Dr.**J.JENIFER,** [4]**Dr.S.VEERAPANDI,** [5]**Dr.S.MURUGANANDAM,** [6]**PRAJWALASIMHA S N,** [7]**POTHUMARTHI SRIDEVI,** [8*]Dr.**L.THENMOZHI,** [9]**DR.T.VENGATESH**

[1]Assistant Professor, PG Department of Computer Science, Nehru Memorial College (Autonomous) (Affiliated to Bharathidasan University), Puthanampatti, Trichy – 621 007, Tamil Nadu, India.

[2]Assistant Professor, Department of Computer Science and Applications (BCA-Data Science), SRM Institute of Science and Technology, Ramapuram, Chennai - 600 089, Tamil Nadu, India.

[3]Assistant professor, Department of Information Technology, St. Joseph's College (Autonomous), Trichy, Tamil Nadu, India.

[4]Assistant Professor, Department of Master of Computer Applications, Easwari Engineering College, Ramapuram, Chennai, Tamilnadu,India-600089.

[5]Associate Professor, Department ofComputer science and Business systems Panimalar Engineering College , Chennai, Tamil Nadu, India.

[6]Associate Professor, Dept. of Computer Science and Engineering (Cyber Security), School of Engineering (SoE), Dayananda Sagar University, Bangalore, India.

[7]Assistant Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur Dist., Andhra Pradesh - 522302,India.

[8*(**Corresponding Author**)]Assistant Professor, Department of Computer Science and Applications (AI & ML), FSH, SRM Institute of Science and Technology, Ramapuram campus, Chennai - 600 089, Tamil Nadu, India.

[9]Assistant Professor, Department of Computer Science, Govt.Arts& Science College, Theni, Affiliated to Madurai Kamaraj University, Madurai, Tamilnadu ,India

Email ID: [1]barakathbegam@nmc.ac.in, [2]Buvana.abj@gmail.com,[3] jenifer.john3@gmail.com, [4]viruyamini@gmail.com, [5]smuruganandam.csbs@panimalar.ac.in, [6]prajwalasimha.sn1@gmail.com, [7]psridevi@kluniversity.in, [8*]Thenmozhilakshmanan@Gmail.Com, [9]venkibiotinix@gmail.com

ABSTRACT

In response to the escalating dependence of individuals and enterprises on cloud storage solutions, the demand for robust data protection mechanisms has become increasingly imperative. Conventional cryptographic techniques typically use a fixed level of encryption strength, which can either waste resources or fail to provide adequate protection by not adjusting to different levels of data sensitivity or evolving security threats. This paper introduces an innovative solution called Real-time Context-Aware Variable Strength Cryptography (RTCAVSC) for cloud storage. RTCAVSC automatically modifies encryption strength in real time based on various contextual factors, such as how sensitive the data is, patterns in user access and the prevailing security conditions. Fuzzy Real-time Sensitivity Tracer, Dynamic Cloud Environment Interpreter and Variable Strength Cryptography Manager are the novel contributed modules assembled unison in RTCAVSC work. A dedicated cloud server is leased to evaluate the performance of RTCAVSC method in terms of Encryption Time, Decryption Time, Throughput, Latency, Resource Utilization and Data Security parameters in real-time environment.

Keywords: *Context Aware, Cloud Data Security, Real-Time Variable Strength Cryptography, Variable Strength Cryptography*

.

1. INTRODUCTION

Cloud storage refers to a model of data storage in which digital data is stored in logical clusters and the physical storage travers to multiple servers in several locations), managed and owned by a hosting company [1]. Rather than storing data on a local hard drive or other physical storage devices, cloud storage allows users to store their data online, making it accessible from any device with an internet connection. Cloud storage offers several key features, including easy accessibility from any internet-connected device, making it highly convenient for users. It is scalable, allowing users to adjust storage capacity as needed without physical hardware. The cost efficiency of cloud storage eliminates the need for expensive on-premises infrastructure, with a pay-as-you-go model based on usage [2]. It provides robust data redundancy and backup options by storing data across multiple locations, ensuring data reliability and disaster recovery [3]. Cloud storage also emphasizes security through encryption and access controls, protecting data from unauthorized access [4]. Additionally, it supports collaboration, enabling file sharing and real-time collaboration among users [5].

Cloud storage comes in several types, catering to different needs. Public cloud storage is hosted by third-party providers and shared among multiple customers, offering scalability and cost-efficiency; examples include services like Amazon S3 and Google Cloud Storage. Private cloud storage is dedicated to a single organization, providing enhanced security and control, often used for sensitive data or specific compliance needs [6]. Hybrid cloud storage combines both public and private cloud elements, allowing data to be stored based on sensitivity or accessibility requirements, making it a flexible choice [7]. Lastly, community cloud storage is shared by multiple organizations with common concerns, such as security or compliance, offering a tailored approach to storage while maintaining shared infrastructure [8].

Cloud storage operates through a network of data centers that utilize virtualization to provide scalable storage. Users transfer files to the cloud using web interfaces, APIs, or specialized software. Once uploaded, data is stored across multiple servers to ensure redundancy and is often distributed geographically for increased durability and faster access [9]. Users can retrieve their data via the internet using various platforms, such as web apps, mobile apps, or desktop clients. Cloud storage providers also offer management tools for organizing, indexing and monitoring data usage, as well as automated features like file synchronization and regular backups [10].

Though cloud storage offers many benefits, it also comes with security risks. One major concern is data breaches, where sensitive information may be exposed due to vulnerabilities in the cloud provider's security systems or due to human error, such as weak passwords. Unauthorized access is another risk, as cybercriminals may exploit security flaws to gain entry to cloud-stored data [11]. Data loss or corruption can occur due to system failures, cyber-attacks, or inadequate backup measures. Insider threats, where employees or former employees misuse their access privileges, pose additional security challenges [12].

Storing sensitive data in cloud storage presents several risks. Data breaches are a significant concern, as cybercriminals could exploit security vulnerabilities to access confidential information, leading to identity theft, financial loss, or exposure of proprietary business information. Unauthorized access might occur due to inadequate access controls or compromised credentials, allowing attackers to infiltrate sensitive data [13]. Data loss is another risk, as technical failures, cyber-attacks, or natural disasters could result in the permanent loss of critical information. Insider threats pose risks where employees or partners misuse their access to sensitive data for malicious purposes. Legal and compliance issues may arise if cloud storage practices do not align with regulatory requirements, such as GDPR or HIPAA, leading to potential fines and legal consequences [14]. Lack of control and visibility over where and how data is stored can also be problematic, as it may hinder an organization's ability to protect sensitive information effectively. Lastly, cloud storage services might face service outages or interruptions, temporarily making sensitive data inaccessible, which could disrupt business operations [15].

The usage of context-sensitive data security is crucial in cloud storage for protecting sensitive information [16]. Context-sensitive data security involves adapting security measures based on the specific context in which the data is accessed or used. This approach ensures that security protocols are dynamically adjusted to address varying levels of risk associated with different scenarios [17]. For instance, if data is accessed from an unfamiliar device or location, additional authentication steps or encryption methods might be triggered. By tailoring security measures to the context of data access, organizations can better safeguard against unauthorized access, mitigate potential threats and

ensure that sensitive information remains protected throughout its lifecycle in the cloud [18] [19]. RTCAVSC is an attempt to enhance the overall data security by responding to real-time risks and providing a more nuanced and effective protection mechanism [20].

2. EXISTING METHODS

A hybrid elliptic curve cryptography (HECC) technique for fast encryption of data for public cloud security [21], Data security in a cloud environment using cryptographic mechanisms [22], A novel hybrid cryptographic framework for secure data storage in cloud computing by integrating AES-OTP and RSA with adaptive key management and time-limited access control [23] and Intrusion detection model using an optimized quantum neural network and elliptic curve cryptography for data security [24] are found to be comparable methods to the proposed RTCAVSC method. A detailed study of these existing methods is carried out to understand their functionalities, advantages and limitations, which are provided in this section.

2.1. A hybrid elliptic curve cryptography technique for fast encryption of data for public cloud security (HECC)

In 2023, B. Ranganatha Rao et.al., proposed HECC work for the purpose of offering clients a centralized system for data storage and various enterprise solutions, cloud computing requires robust security to manage each and every devices, applications and data connected to the cloud. On account of these strong security measures, only authorized individuals can access the data and applications. Public cloud security provides a dependable strategy for customers to access applications and data, enabling service providers to quickly address any emerging security concerns. HECC work provides a public cloud security technique using Hybrid Elliptic Curve Cryptography in which Keys are generated using a lightweight Edwards curve and Identity-Based Encryption is applied to modify the private keys. A key reduction method is unveiled to further minimize key length, enhancing the speed of the Advanced Encryption Standard (AES) encryption process. Public keys are exchanged by means of Diffie-Hellman key exchange. The performance of the proposed model is evaluated by measuring throughput and key generation, encryption and decryption times.In HECC, a public key ecosystem, integral to identity-based encryption (IBE), allows any string, such as email addresses or dates, to function as a valid public key. Identity-based encryption simplifies key management by enabling the generation of public keys from known identity values without prior key distribution. A trusted third party is essential for deriving the corresponding private key, which enhances security while reducing encryption complexity. The proposed model utilizes mail IDs and secret numbers to generate IBE keys, eliminating the need for additional software or advanced preparation for message recipients.

Efficient encryption, encryption speed and decryption speed are the observed advantage of HECC method whereas, dependency of larger key sizes has an impact over the security – which is identified as the limitation.

2.2. Data security in a cloud environment using cryptographic mechanisms (DSCECM)

Abdul Azis Fairosebanu et.al., introduced DSCECM work in 2022 to introduce an innovative approach for securing data in a hybrid cloud environment, where virtual computing resources are intelligently allocated based on user requests. A hybrid cloud, combining both public and private clouds, is identified as optimal for data storage and access, though maintaining security in such environments is challenging. The proposed strategy employs cryptographic methods to safeguard user data across both types of clouds, leveraging various encryption techniques. Three symmetric encryption methods are offered as cloud services and are evaluated for their performance and security. The proposed approach is found to be more efficient than alternative methods and is adaptable for relational data, with potential for further enhancement to handle multimedia data.

DSCECM technique prioritizes cloud storage security by allowing data to be stored in a hybrid cloud based on user preferences, with choices between private or public clouds depending on data sensitivity. The system ensures that encryption, keys and storage are managed separately to prevent any single cloud provider from accessing all aspects of the data. The approach employs symmetric encryption for data in both public and private clouds, with encryption keys managed by a cloud-based service provider.

Symmetrical encryption as a service (SEaaS) offers tailored encryption solutions, while key generation and management are handled by the Key Provisioning and Management as a Service (KPMaaS). This separation ensures that SEaaS doesn't know the encryption keys or data storage locations, while KPMaaS manages the keys without

knowing where the encrypted data resides. This architecture enhances data security by ensuring no single provider has full access to sensitive information, making the system robust against potential breaches.

Improved Security and Versatile encryption methodologies are the stated advantages of the DSCECM work, yet the higher computational complexity increases the encryption and decryption time, which is observed as the limitation.

2.3. A novel hybrid cryptographic framework for secure data storage in cloud computing by integrating AES-OTP and RSA with adaptive key management and time-limited access control

AES-OTP work is proposed in 2023 by D. Shivaramakrishna et.al., to bring forward an innovative hybrid cryptographic framework designed for secure cloud data storage, integrating time-limited access control, adaptive key management and two robust encryption methods: RSA and AES-OTP. The framework combines symmetric (AES-OTP) and asymmetric (RSA) encryption to enhance data confidentiality and integrity. The adaptive key management component strengthens cryptographic security by intelligently managing key creation, distribution and rotation, while time-limited access control further safeguards data privacy by enforcing strict access time constraints. The framework's effectiveness is demonstrated through performance assessments, revealing high accuracy, precision, recall and F1-score.

The extensive healthcare dataset used in this study encompasses critical patient data, including medical histories, diagnoses and treatment plans, serving as a foundation for exploring cryptographic methods to enhance data privacy and security. Researchers evaluate various encryption techniques, such as AES, RSA and ECC, to assess their effectiveness in protecting against unauthorized access and breaches. The study integrates AES-OTP and RSA to address confidentiality, integrity and authenticity in secure communications, with AES-OTP providing robust encryption for large data volumes using a one-time pad for added security. The combined RSA and AES-OTP approach employs multi-layered encryption: RSA secures key exchange, while AES-OTP encrypts the data, mitigating risks associated with single-method vulnerabilities. The proposed architecture emphasizes the importance of a strong adaptive key management system to continually safeguard sensitive data against evolving threats.

Enhanced security, Time-Limited access control and improved accuracy performance are the perceived advantages of AES-OTP method. Key management complexity and escalated performance overhead are having a negative impact in encryption and decryption speeds, discovered as the limitation of AES-OTP work.

2.4. Intrusion detection model using an optimized quantum neural network and elliptic curve cryptography for data security (QNN-WOA)

Heba Kadry et.al., introduced QNN-WOA work in 2023 to integrate Whale Optimization Algorithm (WOA) with a quantum neural network (QNN) and Elliptic Curve Cryptography (ECC) to enhance attack detection and data protection. WOA is employed to select key features from network data, improving the accuracy of intrusion detection. The optimized quantum network combines WOA with feedforward and backpropagation algorithms for precise attack identification. ECC ensures the secure encryption of sensitive data stored on servers, with the best key determined through ECC optimization to protect the documentation effectively. The proposed intelligent framework leverages a Quantum Neural Network to detect service attacks, combining neural networks, quantum computing and machine learning algorithms.

The QNN is designed for high-level data processing and optimization tasks, utilizing qubits encoded into specific states and manipulated through parameterized rotation and entangling gates. Outcomes are decoded and optimized using techniques like the Adam optimizer, with variational quantum circuits (VQC) managing various tasks. The framework represents an advanced system that operates on quantum computing principles, integrating quantum neural networks to enhance computational efficiency and precision in problem-solving. Elliptic Curve Cryptography is utilized to encrypt files by generating robust personal and public cryptographic keys. A key advantage of ECC is its use of smaller key sizes compared to non-elliptic curve methods, which increases efficiency while maintaining strong security. However, accurate key generation is crucial, as improper random selection can lead to decryption issues, necessitating optimization techniques like Cuckoo search to ensure reliable private key selection.

High reliability and Robust data security are the noted advantages of QNN-WOA work. Even so,

increased processing time affects the overall performance efficiency of system, which is observed as the limitation of QNN-WWOA work.
A detailed summary of the key points regarding these existing methods are provided in Table1. It also includes a comparative analysis of the applied techniques, highlighting their strengths and limitations.

Author	Work	Year	Methodology	Advantages	Limitations
B. Ranganatha Rao et.al.,	A hybrid elliptic curve cryptography technique for fast encryption of data for public cloud security	2023	Hybrid elliptic curve cryptography	High Encryption Speed	Low Security
Abdul Azis Fairosebanu et.al.,	Data security in a cloud environment using cryptographic mechanisms	2022	Multiple Symmetric Encryption	High Security	High processing time
D. Shivaramakrishnaet.al.,	A novel hybrid cryptographic framework for secure data storage in cloud computing by integrating AES-OTP and RSA with adaptive key management and time-limited access control	2023	AES One Time Password	High Security	High processing time
Heba Kadry et.al.,	Intrusion detection model using an optimized quantum neural network and elliptic curve cryptography for data security	2023	Whale Optimization Algorithm, Quantum neural network	High Security	Low performance efficiency

Table 1: Key Features and Performance of existing methods

Despite the availability of various cryptographic solutions for cloud data security, a review of the literature reveals persistent gaps. Most prior research emphasizes either enhancing encryption strength or improving performance, yet it typically depends on fixed or hybrid cryptographic configurations that do not adapt to variations in data sensitivity or cloud conditions. For instance, HECC is optimized for faster encryption, whereas DSCECM and AES-OTP prioritize robust security through complex key management, often resulting in increased computational overhead. Advanced models such as QNN-WOA can enhance detection and protection capabilities, but introduce substantial processing complexity, reducing their practicality in dynamic cloud environments. A recurring limitation in these studies is the absence of systems that simultaneously consider data context, user behavior and cloud resource availability when determining encryption strength. Consequently, existing solutions may apply excessive protection to low-risk data or insufficient safeguards to high-risk scenarios, leading to inefficiencies or potential vulnerabilities. Addressing this imbalance between security and performance motivates the development of the RTCAVSC framework, which seeks to provide a more adaptive, context-aware and efficient cryptographic approach for cloud storage.

3. BACKGROUND

It is imperative that delivering a brief description about Context awareness in cloud storage security and Variable Strength Cryptography to provide a clearer explanation of the proposed method. This section provides a thorough explanation of the most important details regarding these technical backgrounds.

3.1. Context Awareness in cloud storage security

Context sensitivity in cloud storage security refers to the adaptive approach of applying security measures based on the specific needs, context and environment in which the data is stored and accessed. In cloud environments, where data resides across various platformspublic, private, or hybrid cloudssecurity demands differ according to the data type, user roles, access patterns and external threats. Cloud storage solutions must be intelligent enough to recognize these diverse conditions and apply security protocols accordingly. Recent research emphasizes the need for tailored security strategies to address data sensitivity dynamically. Sensitive data, such as healthcare records or financial information, requires stricter encryption, enhanced access controls and

real-time monitoring compared to less critical data types. This concept is further supported by advancements in adaptive encryption, where techniques like attribute-based encryption (ABE) adjust the encryption strength based on the context, such as user location or device integrity [25]. Additionally, context-aware systems can ensure that users accessing the cloud from unknown or insecure locations are subject to more stringent security measures, while routine operations under trusted environments maintain optimal performance [26].

Context sensitivity not only enhances data security but also helps optimize the balance between security and performance in cloud storage systems. For example, hybrid cloud models often benefit from context-aware encryption methods where less-sensitive data is stored in public clouds and critical data in private clouds [27]. Furthermore, the advent of AI-powered threat detection systems in cloud environments allows for real-time adjustments to security protocols based on current risk assessments [28].

This adaptive mechanism is crucial in addressing the evolving landscape of cybersecurity threats, compliance mandates and the heterogeneity of cloud deployments. By incorporating context sensitivity into their security frameworks, organizations can safeguard their cloud-stored data, ensuring security measures that dynamically adjust based on the user's role, the data's sensitivity and potential threats.

3.2. Variable Strength Cryptography

Variable Strength Cryptography in cloud storage refers to the use of cryptographic techniques that can dynamically adjust the strength of encryption based on various factors such as the sensitivity of the data, user requirements, or system performance [29]. This approach allows for a more flexible and efficient use of cryptography in cloud environments, where different types of data may require different levels of security. Adaptive encryption Strength, Resource Optimization, Data-Sensitivity based Encryption, User application-driven Flexibility and Improved cloud security are the primary benefits of applying variable strength cryptography in cloud storage environment.

Adaptive Encryption Strength allows for encryption levels to adjust based on the specific context of the data [30]. For example, highly sensitive information such as financial records would require stronger encryption, like AES-256, to ensure enhanced security. In contrast, less critical data, such as temporary files, can utilize lighter encryption, such as AES-128, to conserve computational resources while still maintaining adequate protection. This flexibility optimizes both security and efficiency based on the nature of the data being handled.

Resource Optimization is crucial as stronger encryption algorithms generally demand greater computational power and memory. In large-scale cloud storage operations, this can result in performance bottlenecks [31]. Variable Strength Cryptography (VSC) addresses this by allowing the system to allocate resources more efficiently, applying only the required encryption strength for each specific task. This approach ensures optimal performance without compromising security.

Data Sensitivity-based Encryption allows encryption levels to be tailored according to the sensitivity of the data. Using Variable Strength Cryptography [32], highly sensitive information, such as healthcare or personal data, can be secured with stronger encryption algorithms. In contrast, less sensitive public data can be protected with encryption methods that are less computationally demanding. This approach ensures an appropriate balance between security and resource efficiency, based on the data's classification.

User or Application-driven Flexibility in variable strength cryptography allows encryption levels to be customized by either users or applications [33]. Users can opt for higher encryption levels for critical tasks, while the system can automatically adjust encryption strength based on observed usage patterns. This dynamic approach enhances both security and performance, ensuring that resources are efficiently allocated while maintaining the necessary protection for different operations.

Variable Strength Cryptography significantly improves cloud security by enabling customized encryption strategies tailored to the sensitivity and importance of the data being stored. This approach introduces multiple layers of protection, as different levels of encryption are applied depending on the data's criticality [34]. For instance, highly sensitive information, such as financial or personal data, can be protected with the strongest available encryption algorithms, while less critical data uses lighter encryption. This variability not only optimizes performance but also creates additional barriers for potential attackers. By facing different encryption standards, attackers

are forced to overcome multiple cryptographic challenges, making it much harder to infiltrate the system. As a result, VSC fortifies the cloud storage infrastructure by dynamically adapting its security measures, ensuring that data remains well-protected without unnecessarily straining computational resources.

4. PROPOSED METHOD

The RTCAVSC framework integrates three innovative modules, namely the Fuzzy Real-time Sensitivity Tracer, the Dynamic Cloud Environment Interpreter and the Variable Strength Cryptography Manager, which are designed to work together seamlessly, providing a cohesive solution for adaptive and context-aware cloud security. This section provides a thorough and detailed explanation of each of these modules, offering an in-depth understanding of their functionalities and how they contribute collectively to the overall system's effectiveness in managing cloud security and data protection.

4.1. Fuzzy Real-time Sensitivity Tracer (FRST)

FRST handles the sensitivity of incoming data based on four sensitivity labels namely OPEN, LOW, MEDIUM and HIGH. These labels are assigned based on several factors such as Data type, User category, Access frequency, Geographical location, Volume of data and Data file permissions.A sensitivity coefficient δ is introduced in FRST module through which a collective sensitivity of the above-mentioned factors is computed as follows.

Factor 1[The Data type]: The input data type is categorized into 5 different categories specifically Low sensitivity, Moderate sensitivity, High sensitivity, Very high sensitivity, Critical Sensitivity and Extremely critical sensitivity.

Log data, derived data, Meta data, geospatial data, Time-series data and transactional data are categorized under the Low sensitivity category. Semi-structured data, structured data and Multimedia data are come under the Moderate sensitivity category. Real-time data, encrypted dataand blockchain data are the examples for High sensitivity data category. Personal data and Financial related data are organized under the Very high sensitivity data category. Personal medical data and Electronic health records are the examples for Critical data group. National defense forces and Confidential research related data are grouped under the Extremely critical sensitivity classification. Let D be the input data, the data type-based sensitivity coefficient δ_D is assigned as in Equation 1.

$$\delta_D = \begin{cases} \frac{1}{6} \; if \; D \in Low \; Sensitive \\ \frac{2}{6} \; if \; D \in Medium \; Sensitive \\ \frac{3}{6} \; if \; D \in High \; Sensitive \\ \frac{4}{6} \; if \; D \in Very \; high \; Sensitive \\ \frac{5}{6} \; if \; D \in Critical \; Sensitive \\ 1 \; otherwise \end{cases}$$

Equation (1)

Factor 2 [User category]: There are 4 sensitivity categories are provided for Users in FRST model such as Low, Medium, High and Very high. Guest users, Regular Users and Third-party external users are belonging to under Low category. Data analyst, System operator, Backup/Recovery manager, Developer andPower User are assigned with Medium sensitivity category. Data owner, Auditor, Compliance officer, Data steward and Security officer are set to the High sensitivity category. Service account, Administrator and Super Administrator are grouped into Very high sensitivity category. Since there are 4 different categories exist in User Category factor$U$, the used category-based sensitivity coefficient δ_U is allocated with a value as given in Equation 2.

$$\delta_U = \begin{cases} \frac{1}{4} \; if \; U \in Low \\ \frac{1}{2} \; if \; U \in Medium \\ \frac{3}{4} \; if \; U \in High \\ 1 \; otherwise \end{cases}$$

Equation (2)

Factor 3 [Access Frequency]: There are 5 different access types are prevailing in a typical cloud environment. They are Rare access, Occasional access, Moderate access, Frequent access and Continuous access. From the list, Rare and Occasional accesses are set to be in Low sensitivity category, Moderate access is set to be in Moderate sensitivity category, Frequent and Continuous access are set to be in High sensitivity category. Let A be the access type of the data, then the value for access type-based sensitivity coefficient δ_A will be assigned as in Equation 3.

$$\delta_A = \begin{cases} \frac{1}{3} \; if \; A \in Low \\ \frac{2}{3} \; if \; A \in Moderate \\ 1 \; otherwise \end{cases}$$

Equation (3)

Factor 4 [Geographical Location]: Local, Regional, International, Restricted Zones and Untrusted Zones are the geographical location categories take place in cloud data accessing. If the data accessed within the organization, it comes under the Local category. If the data is accessed in the same country or from a trusted zone, then the geographical location is treated as Regional. If the data access crosses the border of the country, then is the classified in the International group. Some geographical locations are marked as Restricted zones by the Regularity authorities or by Government agencies. Access from these Restricted locations requires strict regularity and security concerns. Untrusted zones are usually declared by Government and National Security Agencies or by Cybersecurity agencies. Any access request from these zones should be properly logged and a highest security protocol should be followed to provide the access. Let$G$ be the geographical location and$\delta_G$be the geographical location-based sensitivity coefficient, then the value of δ_G is determined as in Equation 4.

$$\delta_G = \begin{cases} \frac{1}{5} \; if \; G \in Local \\ \frac{2}{5} \; if \; G \in Regional \\ \frac{3}{5} \; if \; G \in International \\ \frac{4}{5} \; if \; G \in Restricted \\ 1 \; otherwise \end{cases}$$

Equation (4)

Factor 5 [Volume of data]: In FRST module, volume of data is treated into 4 different categories namely Small, Medium, Large and Very large. Data volumes can be ranked by sensitivity, starting with small volumes, which have low sensitivity due to limited data exposure. Medium volumes pose moderate sensitivity, with risks associated with moderate-sized data leaks. Large volumes are more sensitive, as they often involve critical or personal information. At the highest level, very large data volumes carry the greatest sensitivity, typically containing enterprise-level or high-value data, making them prime targets for security breaches. Let V be the volume of data andδ_V be the volume-based sensitivity coefficient, then the value is determined as in Equation 5.

$$\delta_V = \begin{cases} \frac{1}{4} \; if \; V \in Small \\ \frac{1}{2} \; if \; V \in Medium \\ \frac{3}{4} \; if \; V \in Large \\ 1 \; otherwise \end{cases}$$

Equation (5)

Factor 6 [Data File Permissions]: File access permissions are essential in cloud environments because they control who can view, modify, or delete data, thereby ensuring the confidentiality, integrity and availability of sensitive information. Proper management of these permissions helps prevent unauthorized access and potential data breaches, safeguarding against both internal and external threats in a cloud-based infrastructure. Read, Write, Modify and Delete permissions are the common file operational in a cloud environment. There are three categories allocated for file permissions in the FRST module, namely: Low, Medium and High. Read permission is assigned with Low sensitivity since it is the most common permission used in the cloud environments. Write permission is assigned with Medium sensitivity label. Modify and Delete permissions are enumerated under the High sensitivity category. Let P be the assigned file permission andδ_P be the permission-based sensitivity coefficient. Equation 6 is used to designate the value for δ_P.

$$\delta_P = \begin{cases} \frac{1}{3} \; if \; P \in Low \\ \frac{2}{3} \; if \; P \in Medium \\ 1 \; otherwise \end{cases}$$

Equation (6)

The overall FRST sensitivity coefficient δ is calculated using Equation 9.

$$\delta = \frac{1}{n_\delta} \sum_{x=D,U,A,G,V,P} \delta_x$$

Equation (9)

where n_δ is the number of considered factors.

Based on the value of δ, the FRST sensitivity label Δ is determined as in Equation 10.

$$\Delta = \begin{cases} OPEN \; if \; \delta < \frac{1}{4} \\ LOW \; if \; \frac{1}{4} \leq \delta < \frac{1}{2} \\ MEDIUM \; if \; \frac{1}{2} \leq \delta < \frac{3}{4} \\ HIGH \; otherwise \end{cases}$$

Equation (10)

The live sensitivity label Δ will be used by the succeeding modules of RTCAVSC.

4.2. Dynamic Cloud Environment Interpreter (DCEI)

There are 5 key factors namely CPU utilization, Memory usage, Storage usage, Network resource utilization and Latency are speculated in

DCEI module to interpret the status of cloud environment. Based on these criteria, DCEI module fixes the cloud environment state label Γ from the set {IDLE, LOW, MODERATE, HIGH, OVERLOAD}. These factors are metered in percent units (%) by several tools such as CloudWatch, Prometheus and Grafana. The utilization percentages are converted to a singular coefficient in DCEI module to determine the cloud environment load label.

The CPU usage U_{CPU} directly impacts cloud load by affecting resource allocation, auto-scaling and performance. High CPU usage can lead to performance degradation, increased costs and resource contention, requiring more cloud instances to manage the demand. Cloud providers monitor CPU trends to optimize load balancing, prevent throttling and ensure efficient infrastructure utilization.

Memory usage U_{MEM} affects cloud load by influencing the allocation and efficiency of virtual machines or containers. High memory usage can lead to performance issues, such as slower application response times and increased need for additional instances, which in turn raises the overall cloud load. Providers often monitor memory usage to adjust resources and optimize performance, ensuring that demand is met without overloading the infrastructure.

Storage usage U_{STO} impacts cloud load by affecting data management and resource allocation. High storage demands can lead to increased I/O operations and potentially slower performance, as well as the need for additional storage capacity, which raises the overall cloud load. Cloud providers monitor storage usage to manage capacity and optimize performance, ensuring that storage resources are efficiently utilized and preventing potential bottlenecks.

Network resource usage U_{NET} affects cloud load by influencing data transfer rates and bandwidth consumption. High network usage can lead to congestion, increased latency and slower response times, which may necessitate additional resources to handle the traffic, thereby increasing the overall cloud load. Cloud providers monitor network usage to balance traffic, optimize performance and prevent bottlenecks, ensuring efficient data flow and resource utilization.

Latency U_{LAT} affects cloud load by impacting the speed at which data is transferred between clients and cloud services. High latency can slow down application performance and increase the time required for processing requests, leading to a higher load on cloud resources as they work harder to manage delays and retries. Cloud providers monitor latency to ensure efficient operation, optimize response times and balance loads to maintain performance and reliability.

The overall cloud load coefficient U is calculated using the following formula

$$U = \frac{\frac{1}{5}(U_{CPU}+U_{MEM}+U_{STO}+U_{NET}+U_{LAT})}{10^2}$$

Equation (11)

Based on the computed value of U, the cloud environment load label Γ is mapped by the following equation.

$$\Gamma = \begin{cases} IDLE\ if\ U < \frac{1}{5} \\ LOW\ if\ \frac{1}{5} \le U < \frac{2}{5} \\ MODERATE\ if\ \frac{2}{5} \le U < \frac{3}{5} \\ HIGH\ if\ \frac{3}{5} \le U < \frac{4}{5} \\ OVERLOAD\ otherwise \end{cases}$$

Equation (12)

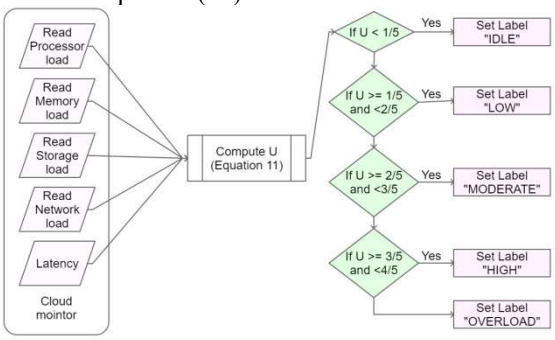

Figure 1: Dynamic Cloud Environment Interpreter

4.3. Variable Strength Cryptography Manager (VSCM)

VSCM uses Rijndael - a symmetric key block cipher designed by Joan Daemen and Vincent Rijmen, known for its flexibility in block and key sizes. Unlike AES, which has a fixed 128-bit block size, Rijndael allows different block sizes of 128, 192, or 256 bits and key sizes of 128, 192, or 256 bits. The number of encryption rounds in Rijndael could be 10, 12, or 14 roundsbased on size of the key. It uses a substitution-permutation network (SPN) for strong security, ensuring robust diffusion and confusion. VSCM takes advantage of this high configurable property of Rijndael cryptographic procedure to fit in the RTCAVSC work. There are 4 different operational modes in specific Electronic Codebook (ECB), Cipher Block Chaining (CBC), Counter (CTR) and Galois Counter Mode (GCM) are used in VSCM module. These modes inherit

different computational complexity and different levels of security. The block sizes are also taken care by the VSCM algorithm to appoint the more appropriate cryptography configuration based on the inputs Δ and Γ.

Algorithm: VSCM

Input: Δ&Γ
Output: Block size β, Key Size κ, Operation Mode Ω

Step 1: Fetch Δ from FRST
Step 2: Fetch Γ from DCEI
Step 3: If Δ = OPEN
Step 4: If $\Gamma = IDLE$, Set $\beta = 192$, $\kappa = 192, and \Omega = GCM$
Step 5: else if $\Gamma = LOW$, Set $\beta = 192$, $\kappa = 192, and \Omega = CTR$
Step 6: else if $\Gamma = MODERATE$, Set $\beta = 192$, $\kappa = 192, and \Omega = CBC$
Step 7: else if $\Gamma = HIGH$, Set $\beta = 192$, $\kappa = 192, and \Omega = ECB$
Step 8: else Set $\beta = 128$, $\kappa = 128, and \Omega = ECB$
Step 9: else if $\Delta = LOW$
Step 10: If $\Gamma = IDLE$, Set $\beta = 256$, $\kappa = 256, and \Omega = GCM$
Step 11: else if $\Gamma = LOW$, Set $\beta = 192$, $\kappa = 192, and \Omega = CTR$
Step 12: else if $\Gamma = MODERATE$, Set $\beta = 128$, $\kappa = 192, and \Omega = CBC$
Step 13: else if $\Gamma = HIGH$, Set $\beta = 192$, $\kappa = 128, and \Omega = ECB$
Step 14: else Set $\beta = 128$, $\kappa = 128, and \Omega = ECB$
Step 15: else if $\Delta = MEDIUM$
Step 16: If $\Gamma = IDLE$, Set $\beta = 256$, $\kappa = 256, and \Omega = GCM$
Step 17: else if $\Gamma = LOW$, Set $\beta = 192$, $\kappa = 256, and \Omega = GCM$
Step 18: else if $\Gamma = MODERATE$, Set $\beta = 192$, $\kappa = 256, and \Omega = CTR$
Step 19: else if $\Gamma = HIGH$, Set $\beta = 192$, $\kappa = 192, and \Omega = CTR$
Step 20: else Set $\beta = 128$, $\kappa = 128, and \Omega = CBC$
Step 21: else if $\Delta = HIGH$
Step 22: If $\Gamma = IDLE$, Set $\beta = 256$, $\kappa = 256, and \Omega = GCM$
Step 23: else if $\Gamma = LOW$, Set $\beta = 192$, $\kappa = 256, and \Omega = GCM$
Step 24: else if $\Gamma = MODERATE$, Set $\beta = 256$, $\kappa = 192, and \Omega = GCM$
Step 25: else if $\Gamma = HIGH$, Set $\beta = 192$, $\kappa = 192, and \Omega = GCM$
Step 26: else Set $\beta = 192$, $\kappa = 192, and \Omega = CTR$
Step 27: end if // Δ
Step 28: Return β, κ, Ω

As per the VSCM algorithm, a harmoniously aligned cryptography model is selected based context sensitivity and the cloud environmental load to secure the data with greater effectivity.

5. EXPERIMENTAL SETUP

A computer with i7 4GHz 8-core processor backed by 16GHz RAM, 1TB NVMe storage is used to develop the source codes related to this RTCAVSC work. A cloud server is leased from i2k2 [35] to test the performance of the compared methods in real-time. A tailored user interface (UI) is created using Visual Studio IDE [36] to perform the evaluation in easier way. Advanced C++ 20.0 [37] programming language is used to write the algorithm scripts used in the proposed modules of RTCAVSC work. The cloud resource utilizations are fetched from CloudWatch [38], Prometheus [39] and Grafana [40] tools and the averages are taken in proposed DCEI module. The compiled codes of proposed method are deployed into the cloud environment using Common Gateway Interface [41] desktop client software and the results are fetched from the dedicated cloud server hypervisor interface. The UI is designed in a way to generate the human interpretable Report file and the Graph presentations.

6. RESULTS AND ANALYSIS

Essential performance benchmark parameters such as Encryption Time, Decryption time, Throughput, Latency, Resource Utilization and Security level are measured during the evaluation process. These parameters are interconnected and collectively influence the performance, reliability and security of cloud storage solutions. Balancing them is essential to ensure a cloud storage system meets the needs of users and businesses effectively. These metrics are metered for every 10MB of data in range form 10MB to 100MB. The performance scores of the compared works are presented in this section for vivid comprehension.

6.1. Encryption time

This measures how long it takes to encrypt data before it is stored in the cloud. It's important because slower encryption can delay data storage and retrieval, affecting overall performance and

user experience. Efficient encryption ensures data is secured without compromising speed. Measured encryption time values for considered methods for different data sizes are provided in Table 2. A comparison graph is provided in Figure 2.

Encryption Time (mS)					
Data (MB)	**HECC**	**DSCECM**	**AESOTP**	**QNNWOA**	**RTCAVSC**
10	342	476	452	386	356
20	450	608	578	534	448
30	522	683	679	602	485
40	579	744	751	678	507
50	586	778	776	719	564
60	630	807	831	723	581
70	622	838	856	782	587
80	659	878	901	788	599
90	662	897	936	838	631
100	686	939	955	838	657

Table 2: Encryption Time

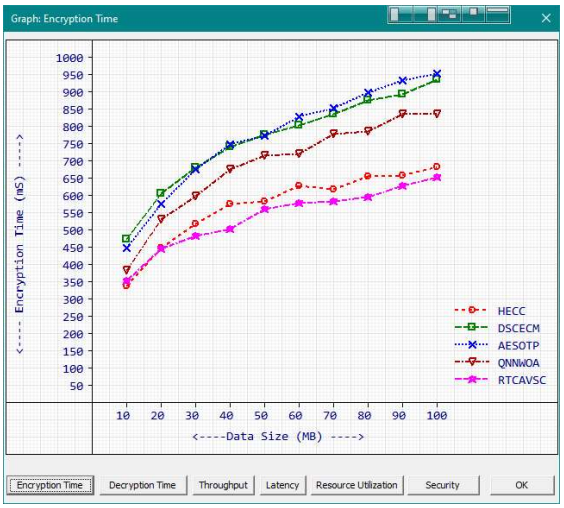

Figure 2: Encryption Time

The results indicate that as the data size increases, encryption times also rise for all algorithms. Based on the average encryption times, RTCAVSC is the best method, with an average encryption time of 523.5 mS, followed closely by HECC at 570.8 mS. While HECC has the fastest encryption times for individual data sizes, RTCAVSC offers the best average performance across all tested sizes, making it the most efficient choice overall when considering the average encryption time. RTCAVSC exhibits notable performance improvements in average encryption time compared to other algorithms. It achieves an approximately 8.25% improvement over HECC, a substantial 29.86% enhancement over DSCECM, a remarkable 31.99% improvement compared to AESOTP and a 23.98% boost over QNNWOA. These percentages indicate that RTCAVSC not only performs better than the slower algorithms but also maintains competitive efficiency against faster methods, reinforcing its effectiveness as a reliable encryption choice in cloud storage applications.

6.2. Decryption Time

Similar to encryption time, this is the time required to decrypt data when it is accessed. Quick decryption is essential for ensuring smooth and timely data retrieval, especially for applications that require real-time or near-real-time access to sensitive data. The decryption times of compared methods for different data sizes are listed in Table 3.

Decryption Time (mS)					
Data (MB)	**HECC**	**DSCECM**	**AESOTP**	**QNNWOA**	**RTCAVSC**
10	283	423	387	318	296
20	382	536	516	484	381
30	467	609	623	546	417
40	524	684	690	622	439
50	526	718	721	661	514
60	558	741	776	655	521
70	568	770	795	711	534
80	592	815	841	717	535
90	611	837	873	775	570
100	634	881	886	770	607

Table 3: Decryption Time

The decryption time performance of the algorithms varies with increasing data sizes. For data sizes from 10 MB to 100 MB, the average decryption times are as follows: HECC shows a relatively consistent increase, starting at 283 mS for 10 MB and reaching 634 mS at 100 MB. DSCECM has the longest times overall, beginning at 423 mS and ending at 881 mS. AESOTP performs comparably to HECC, with times ranging from 387 mS to 886 mS. QNNWOA exhibits moderate performance, starting at 318 mS and peaking at 770 mS. Lastly, RTCAVSC demonstrates the best average performance, with decryption times increasing from 296 mS to 607 ms. Overall, TCAVSC emerges as the best-performing algorithm, demonstrating consistently lower decryption times across all tested data sizes. With times ranging from 296 mS at 10 MB to 607 mS at 100 MB, it proves to be the most efficient choice for applications requiring fast decryption. A comparison graph for decryption time is plotted in Figure 3.

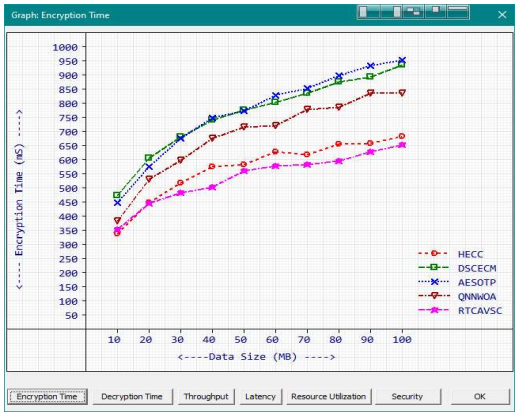

Figure 3: Decryption Time

6.3. Throughput

In cloud storage and data management scenarios, high throughput is crucial because it directly impacts the system's ability to handle large volumes of data efficiently.

Throughput (kbps)					
Data (MB)	**HECC**	**DSCECM**	**AESOTP**	**QNNWOA**	**RTCAVSC**
10	379975	312287	324281	357067	372682
20	325093	246511	261689	283242	326121
30	289151	208694	210787	249281	308130
40	261061	178009	174836	211543	297045
50	257550	161184	162455	191451	268925
60	235975	147175	134528	188819	260461
70	239545	131475	122994	159876	256653
80	220879	111008	99512	156925	250641
90	219592	102349	82584	131112	234964
100	207851	80783	73203	131218	222176

Table 4: Throughput

Throughput refers to the rate at which data is successfully processed over a specific period, typically measured in bits per second (bps), or in bytes per second (Bps). Systems with optimized throughput can better manage spikes in demand, ensuring that even during peak usage, performance remains stable and reliable. This is particularly important in scenarios like online streaming, e-commerce transactions, or large-scale data backups, where delays can lead to user dissatisfaction or data loss. Throughput maximization in cloud storage environments is vital for maintaining efficiency, performance and user satisfaction while dealing with the complexities of large-scale data processing. Table 5 provides the obtained throughput values of compared methods for 10MB to 100MB data sizes.

The observation results show that the proposed RTCAVSC method secured more throughput average score during the evaluation process. At the 100 MB data size, the RTCAVSC algorithm achieves the highest throughput of 222,176 kbps, surpassing all other algorithms. HECC follows with 207,851 kbps, while DSCECM, AESOTP and QNNWOA show lower throughput values of 80,783 kbps, 73,203 kbps and 131,218 kbps, respectively. This performance indicates that RTCAVSC is the most effective method for handling large data sizes, making it the preferred choice for applications requiring high throughput.

A comparison graph for Throughput is provided in Figure 4

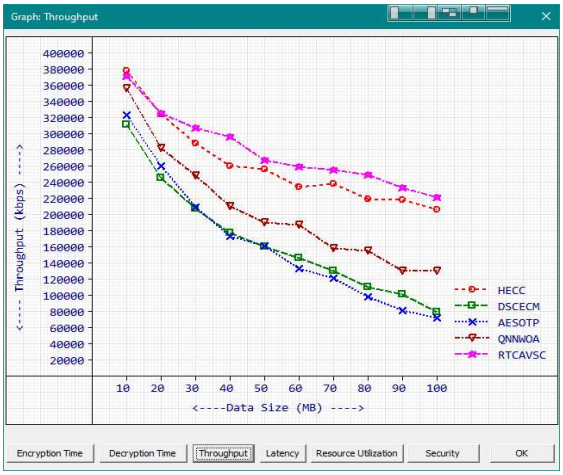

Figure 4: Throughput

6.4. Latency

Low latency is critical for responsive interactions with cloud-stored data. High latency can lead to delays in data access, which can frustrate users and negatively impact applications that rely on real-time data. Computed Latency values are given in Table 5 and the graph is provided as Figure 5.

Journal of Theoretical and Applied Information Technology
15th March 2026. Vol.104. No.5

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Latency (mS)					
Data (MB)	**HECC**	**DSCECM**	**AESOTP**	**QNNWOA**	**RTCAVSC**
10	50	62	61	55	53
20	60	75	71	69	60
30	66	81	81	74	65
40	71	88	88	82	67
50	74	89	88	84	70
60	78	91	93	86	72
70	75	95	98	89	73
80	79	100	101	90	74
90	80	101	104	96	77
100	82	105	104	94	78

Table 5: Latency

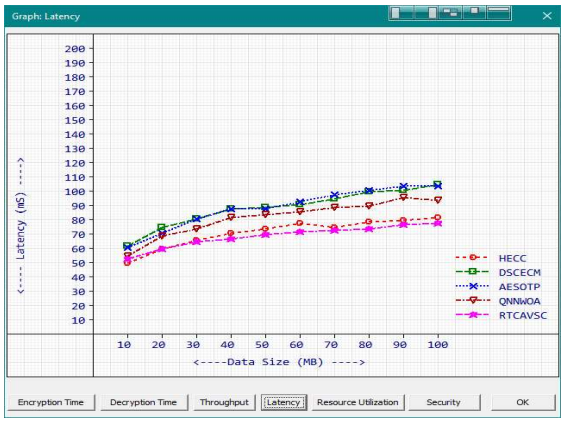

Figure 5: Latency

The latency formula is $Latency = RTT - Processing\ Time$, where RTT refers the Round Trip Time that refers to the total time it takes for a request to travel from the client to the cloud storage and back, while Processing Time accounts for the duration the cloud storage system takes to process the request, such as reading or writing data.

The average latency (measured in milliseconds) across all data sizes for the five algorithmsHECC, DSCECM, AESOTP, QNNWOA and RTCAVSCindicates that RTCAVSC performs the best overall. RTCAVSC achieves the lowest average latency, demonstrating its efficiency in processing requests, while DSCECM consistently records the highest latency, showcasing its struggles with larger datasets. HECC, although effective, does not outperform RTCAVSC, which maintains superior performance throughout the data sizes. The results highlight RTCAVSC as the most reliable option for minimizing latency in cloud storage scenarios, particularly for applications that require swift response times. At the 100 MB data size, RTCAVSC demonstrates a notable improvement in latency compared to HECC, achieving a latency of 78 ms versus HECC's 82 ms. This results in approximately a 4.88% improvement, highlighting RTCAVSC's superior efficiency in processing requests. This advantage underscores RTCAVSC's effectiveness in minimizing latency, making it an excellent choice for applications that require quick response times when handling large datasets.

6.5. Resource Utilization

Resource utilization refers to the efficiency with which a cloud storage system uses its available computing resources, such as CPU, memory and storage bandwidth. High resource utilization means that the system can handle more data and transactions without requiring additional hardware, which helps to keep operational costs down. However, if resources are overutilized, it can lead to performance bottlenecks, increased latency and slower response times, ultimately degrading the user experience. Conversely, underutilization may indicate wasted resources and increased costs without corresponding benefits, which is not ideal for efficient cloud management. Therefore, optimizing resource utilization is essential for balancing performance, cost-effectiveness and scalability in cloud storage solutions, ensuring that they can meet the demands of varying workloads while maintaining high service levels. The metered resource utilization values are enumerated in Table 6.

Resource Utilization (%)					
Data (MB)	**HECC**	**DSCECM**	**AESOTP**	**QNNWOA**	**RTCAVSC**
10	33	45	42	37	33
20	42	55	52	49	41
30	49	61	60	55	44
40	54	66	67	62	45
50	52	71	69	64	52
60	56	71	76	66	54
70	55	75	78	70	53
80	61	80	82	72	55
90	60	80	83	76	57
100	61	83	85	76	60

Table 6: Resource Utilization

Efficient use of resources is vital for optimizing performance and cost in cloud environments. High resource utilization can lead to increased costs and slower performance, making it important to strike a balance.

The data on resource utilization (%) for five algorithmsHECC, DSCECM, AESOTP, QNNWOA and RTCAVSC across various data sizes reveals varying levels of efficiency. For 100

MB data size, HECC and RTCAVSC exhibit lower resource utilization at 61% and 60%, respectively, compared to DSCECM (83%), AESOTP (85%) and QNNWOA (76%). This indicates that while HECC and RTCAVSC are more efficient in resource usage, DSCECM and AESOTP utilize resources more intensively.

Notably, RTCAVSC shows an improvement of approximately 9% in resource utilization compared to its performance at 10 MB, where it recorded 33%. This trend highlights RTCAVSC's balanced approach in managing resource usage while still performing effectively, making it a suitable choice for applications that require both efficiency and performance in handling large datasets.

The graphical visualization chart is for Resource utilization is illustrated as Figure 6.

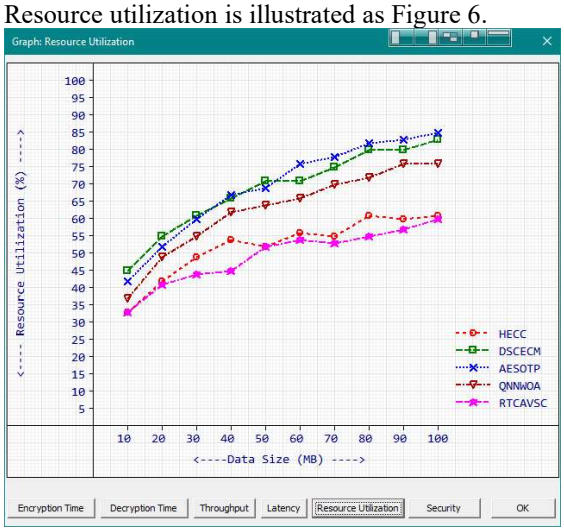

Figure 6: Resource Utilization

6.6. Security Level

Encryption strength is primarily determined by several critical factors. Key length is a fundamental aspect, with longer keys providing exponentially more security against brute-force attacks. The choice of encryption algorithm also plays a significant role, with well-established algorithms like AES offering better resistance to known cryptographic attacks. Additionally, the mode of operation, such as GCM or CBC, can impact security, with some modes providing enhanced protection against pattern leakage and ensuring data integrity. Effective key management is crucial, as secure key generation, storage and distribution are essential to maintaining encryption strength. The randomness and quality of entropy used in key and initialization vector generation further contribute to encryption robustness. Finally, the scheme's resistance to various cryptanalysis methods, including brute-force and side-channel attacks, ensures comprehensive security for protecting sensitive data.

Strong encryption protects sensitive information such as personal data, financial records and intellectual property from being read or altered by malicious actors. It also provides a safeguard against data breaches, reducing the risk of costly legal consequences, reputational damage and loss of trust. Furthermore, robust encryption helps meet regulatory compliance requirements for data protection, such as GDPR, HIPAA and CCPA, which mandate stringent security measures. Finally, strong encryption allows users and organizations to maintain control over their data, ensuring that only authorized parties can decrypt and access it, even in multi-tenant cloud environments. This layer of protection is essential for preserving the privacy and integrity of data stored in the cloud. Measured security strength of compared methods for different data sizes are provided in Table 7.

Security (%)					
Data (MB)	**HECC**	**DSCECM**	**AESOTP**	**QNNWOA**	**RTCAVSC**
10	96.19	97.53	98.71	98.07	99.080002
20	96	97.74	98.6	98.04	98.940002
30	96.26	97.71	98.51	98.16	99.019997
40	96.15	97.62	98.68	98.18	98.839996
50	96.1	97.69	98.76	98.28	98.839996
60	96.11	97.72	98.55	98.15	99.059998
70	96.17	97.53	98.73	98.2	99.050003
80	96.23	97.69	98.72	98.07	98.959999
90	96.04	97.76	98.73	98.04	98.830002
100	96.09	97.76	98.59	98.27	98.93

Table 7: Security level

The experimental results indicate that RTCAVSC consistently achieves the strongest security performance among the compared methods. It peaks at 99.08% for 10 MB of data and maintains similarly high values across larger file sizes, highlighting both stability and reliability. AESOTP appears second, ranging from 98.51% to 98.76%, with RTCAVSC offering a relative improvement of about 0.37% at 10 MB and 0.34% at 100 MB. These reported percentages are

heuristic measures derived from entropy estimation, key length, operational mode and basic brute-force resistance and therefore should be viewed as indicative rather than definitive proofs of cryptographic strength. More rigorous evaluations such as advanced cryptanalysis, side-channel resilience testing, penetration testing and formal verification were beyond the present scope. Nevertheless, the results directly reinforce the concern outlined in the abstract that fixed-strength encryption may either waste computational resources or fail to ensure sufficient protection. By dynamically adjusting encryption strength to context, RTCAVSC provides stronger and more efficient security, adapting to both data sensitivity and evolving threat conditions.A graph chart for security level of the compared methods is illustrated in Figure 7.

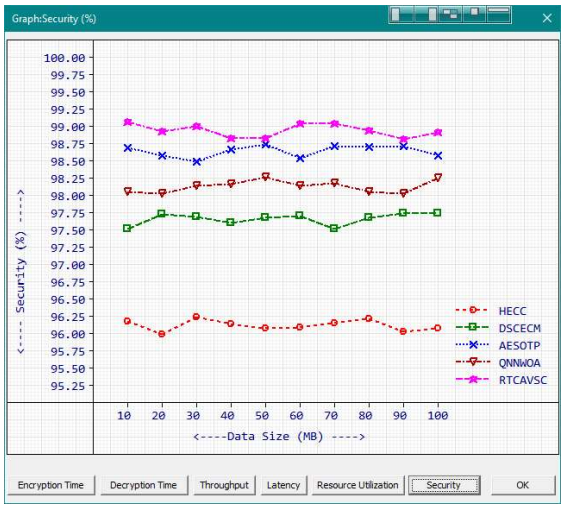

Figure 7: Securitylevel

The experimental findings show that the RTCAVSC framework consistently outperforms other methods in terms of efficiency and security. This improvement comes from its ability to adjust encryption strength based on data sensitivity and real-time cloud conditions, instead of using the same security level for all data. In comparison to earlier methods like HECC, DSCECM, AES-OTP and QNN-WOA, which use fixed or heavy computational methods, RTCAVSC provides a more balanced solution by cutting down unnecessary overhead while still offering strong protection. The results also suggest that the framework's effectiveness relies on predefined sensitivity rules and operational thresholds, which may need to change in highly dynamic or large-scale cloud settings. Additionally, the current assessment is limited to a controlled experimental setup. Expanding validation under different workloads and conducting deeper security analyses is an important area for future research. These observations highlight both the strengths of RTCAVSC and the areas that require further improvement to ensure it can work effectively in real-world cloud systems.

8. CONCLUSION

The proposed Real-time Context-Aware Variable Strength Cryptography (RTCAVSC) framework demonstrates the ability to adapt encryption strength dynamically based on contextual information and prevailing cloud conditions. This adaptability has shown clear improvements in both security and performance within cloud storage environments. Experimental results indicate that RTCAVSC consistently achieves higher efficiency, reduced latency and more balanced resource utilization when compared with existing fixed or hybrid cryptographic approaches. From the authors' perspective, the primary strength of this work lies in its practical and flexible design, as it moves away from a one-size-fits-all encryption strategy and instead responds intelligently to varying data sensitivity and operational demands. At the same time, the study emphasizes that accurate context definition and appropriate threshold selection are critical factors influencing effectiveness and careful tuning will be essential for real-world deployment. Overall, RTCAVSC represents a promising step toward more adaptive and resilient cloud security solutions, while also highlighting avenues for further refinement and validation in larger and more dynamic cloud environments.

Future Work: Application of Machine learning based optimization towards context sensitivity detection and predicting emergency security threats in a cloud environment could be the augmentation of RTCAVSC work.

REFERENCES

[1] Akbar, M., Ahmad, I., Mirza, M. et al. Enhanced authentication for de-duplication of big data on cloud storage system using machine learning approach. Cluster Comput 27, 3683–3702 (2024). https://doi.org/10.1007/s10586-023-04171-y

[2] Zhao, S., Miao, J., Zhao, J. et al. A comprehensive and systematic review of the banking systems based on pay-as-you-go payment fashion and cloud computing in the pandemic era. Inf Syst E-Bus Manage

(2023). https://doi.org/10.1007/s10257-022-00617-9

[3] Hussien ZA, Abdulmalik HA, Hussain MA, Nyangaresi VO, Ma J, Abduljabbar ZA, Abduljaleel IQ. Lightweight Integrity Preserving Scheme for Secure Data Exchange in Cloud-Based IoT Systems. Applied Sciences. 2023; 13(2):691. https://doi.org/10.3390/app13020691

[4] S. M. Shaffi, S. Vengathattil, J. N. Sidhick and R. Vijayan, "AI-Driven Security in Cloud Computing: Enhancing Threat Detection, Automated Response and Cyber Resilience," arXiv preprint, May 6, 2025. arXiv

[5] Chuting Wang, Ruifeng Guo, Haoyu Yu, Yi Hu, Chao Liu, Changyi Deng, Task offloading in cloud-edge collaboration-based cyber physical machine tool, Robotics and Computer-Integrated Manufacturing, Volume 79, 2023, 102439, ISSN 0736-5845, https://doi.org/10.1016/j.rcim.2022.102439

[6] Budati AK, Vulapula SR, Shah SBH, Al-Tirawi A, Carie A. Secure Multi-Level Privacy-Protection Scheme for Securing Private Data over 5G-Enabled Hybrid Cloud IoT Networks. Electronics. 2023; 12(7):1638. https://doi.org/10.3390/electronics12071638

[7] R. Kalaria, A. S. M. Kayes, W. Rahayu et al., "Adaptive context-aware access control for IoT environments leveraging fog computing," International Journal of Information Security, vol. 23, pp. 3089–3107, Jul 2024. SpringerLink

[8] A. Sasikumar, L. Ravi, K. Kotecha, A. Abraham, M. Devarajan and S. Vairavasundaram, "A Secure Big Data Storage Framework Based on Blockchain Consensus Mechanism With Flexible Finality," in IEEE Access, vol. 11, pp. 56712-56725, 2023, https://doi.org/10.1109/ACCESS.2023.3282322

[9] Li F, Guo T, Li X, Wang J, Xia Y, Ma Y. Transportation of Service Enhancement Based on Virtualization Cloud Desktop. Electronics. 2023; 12(7):1572. https://doi.org/10.3390/electronics12071572

[10] Wu, S., Tu, Z., Zhou, Y., Wang, Z., Shen, Z., Chen, W., ... & Mao, B. (2023). FASTSync: a FAST delta sync scheme for encrypted cloud storage in high-bandwidth network environments. ACM Transactions on Storage, 19(4), 1-22.

[11] Fargana Abdullayeva, Cyber resilience and cyber security issues of intelligent cloud computing systems, Results in Control and Optimization, Volume 12, 2023, 100268, ISSN 2666-7207, https://doi.org/10.1016/j.rico.2023.100268

[12] Pandey, N.K., Kumar, K., Saini, G. et al. Security issues and challenges in cloud of things-based applications for industrial automation. Ann Oper Res (2023). https://doi.org/10.1007/s10479-023-05285-7

[13] Dayanikli, D., Lehmann, A. (2024). Password-Based Credentials with Security Against Server Compromise. In: Tsudik, G., Conti, M., Liang, K., Smaragdakis, G. (eds) Computer Security – ESORICS 2023. ESORICS 2023. Lecture Notes in Computer Science, vol 14344. Springer, Cham. https://doi.org/10.1007/978-3-031-50594-2_8

[14] Divadari, S., Surya Prasad, J., Honnavalli, P. (2023). Managing Data Protection and Privacy on Cloud. In: Gunjan, V.K., Zurada, J.M. (eds) Proceedings of 3rd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications. Lecture Notes in Networks and Systems, vol 540. Springer, Singapore. https://doi.org/10.1007/978-981-19-6088-8_33

[15] Yadav, M., Mishra, A. An enhanced ordinal optimization with lower scheduling overhead based novel approach for task scheduling in cloud computing environment. J Cloud Comp 12, 8 (2023). https://doi.org/10.1186/s13677-023-00392-z

[16] Z. Aref, S. Wei and N. B. Mandayam, "Human-AI Collaboration in Cloud Security: Cognitive Hierarchy-Driven Deep Reinforcement Learning," arXiv preprint, Feb 22, 2025. arXiv

[17] Paolo Bellavista, Nicola Bicocchi, Mattia Fogli, Carlo Giannelli, Marco Mamei, Marco Picone, Requirements and design patterns for adaptive, autonomous and context-aware digital twins in industry 4.0 digital factories, Computers in Industry, Volume 149, 2023, 103918, ISSN 0166-3615, https://doi.org/10.1016/j.compind.2023.103918

[18] Ahmad, A., Malik, A.W., Alreshidi, A. et al. Adaptive Security for Self-Protection of Mobile Computing Devices. Mobile Netw Appl 28, 653–672 (2023). https://doi.org/10.1007/s11036-019-01355-y

[19] Y. Wang and X. Yang, "Research on Enhancing Cloud Computing Network Security using Artificial Intelligence Algorithms," arXiv preprint, Feb 25, 2025. arXiv

[20] S. A. Kawalkar and D. B. Bhoyar, "Design of an Efficient Cloud Security Model through Federated Learning, Blockchain, AI-Driven Policies and Zero Trust Frameworks," International Journal of Intelligent Systems and Applications in Engineering, vol. 12, no. 10s, pp. 378–388, Jan 2024. IJISAE

[21] B. Ranganatha Rao, B. Sujatha, A hybrid elliptic curve cryptography (HECC) technique for fast encryption of data for public cloud security, Measurement: Sensors, Volume 29, 2023, 100870, ISSN 2665-9174, https://doi.org/10.1016/j.measen.2023.100870

[22] Abdul Azis Fairosebanu, Antony Cruz Nisha Jebaseeli PG and Research Department of Computer Science, Government Arts and Science College, Affiliated to Bharathidasan University, Trichy, India Bulletin of Electrical ngineering and Informatics Vol. 12, No. 1, February 2023, pp. 462~471 ISSN: 2302-9285, DOI: 10.11591/eei.v12i1.4590

[23] D. Shivaramakrishna, M. Nagaratna, A novel hybrid cryptographic framework for secure data storage in cloud computing: Integrating AES-OTP and RSA with adaptive key management and Time-Limited access control, Alexandria Engineering Journal, Volume 84, 2023, Pages 275-284, ISSN 1110-0168, https://doi.org/10.1016/j.aej.2023.10.054

[24] Heba Kadry, Ahmed Farouk, Elnomery A. Zanaty, Omar Reyad, Intrusion detection model using optimized quantum neural network and elliptical curve cryptography for data security, Alexandria Engineering Journal, Volume 71, 2023, Pages 491-500, ISSN 1110-0168, https://doi.org/10.1016/j.aej.2023.03.072

[25] Inshi S, Chowdhury R, Ould-Slimane H, Talhi C. Secure Adaptive Context-Aware ABE for Smart Environments. IoT. 2023; 4(2):112-130. https://doi.org/10.3390/iot4020007

[26] Singh, I., Lee, SW. Self-adaptive and secure mechanism for IoT based multimedia services: a survey. Multimed Tools Appl 81, 26685–26720 (2022). https://doi.org/10.1007/s11042-020-10493-5

[27] Garg D, Rani S, Herencsar N, Verma S, Wozniak M, Ijaz MF. Hybrid Technique for Cyber-Physical Security in Cloud-Based Smart Industries. Sensors. 2022; 22(12):4630. https://doi.org/10.3390/s22124630

[28] I. Bibi, A. Akhunzada and N. Kumar, "Deep AI-Powered Cyber Threat Analysis in IIoT," in IEEE Internet of Things Journal, vol. 10, no. 9, pp. 7749-7760, 1 May1, 2023, https://doi.org/10.1109/JIOT.2022.3229722

[29] Fursan Thabit, Ozgu Can, Sharaf Alhomdy, Ghaleb H. Al-Gaphari, Sudhir Jagtap, A Novel Effective Lightweight Homomorphic Cryptographic Algorithm for data security in cloud computing, International Journal of Intelligent Networks, Volume 3, 2022, Pages 16-30, ISSN 2666-6030, https://doi.org/10.1016/j.ijin.2022.04.001t

[30] Mohiyuddin, A., Javed, A.R., Chakraborty, C. et al. Secure Cloud Storage for Medical IoT Data using Adaptive Neuro-Fuzzy Inference System. Int. J. Fuzzy Syst. 24, 1203–1215 (2022). https://doi.org/10.1007/s40815-021-01104-y

[31] Malik S, Tahir M, Sardaraz M, Alourani A. A Resource Utilization Prediction Model for Cloud Data Centers Using Evolutionary Algorithms and Machine Learning Techniques. Applied Sciences. 2022; 12(4):2160. https://doi.org/10.3390/app12042160

[32] Anand, K., Vijayaraj, A. & Vijay Anand, M. An enhanced bacterial foraging optimization algorithm for secure data storage and privacy-preserving in cloud. Peer-to-Peer Netw. Appl. 15, 2007–2020 (2022). https://doi.org/10.1007/s12083-022-01322-7

[33] Backendal, M., Günther, F., Paterson, K.G. (2022). Puncturable Key Wrapping and Its Applications. In: Agrawal, S., Lin, D. (eds) Advances in Cryptology – ASIACRYPT 2022. ASIACRYPT 2022. Lecture Notes in Computer Science, vol 13792. Springer, Cham. https://doi.org/10.1007/978-3-031-22966-4_22

[34] Sijjad Ali, Shuaib Ahmed Wadho, Aun Yichiet, Ming Lee Gan, Chen Kang Lee, Advancing cloud security: Unveiling the protective potential of homomorphic secret sharing in secure cloud computing, Egyptian Informatics Journal, Volume 27, 2024,

100519, ISSN 1110-8665, https://doi.org/10.1016/j.eij.2024.100519
[35] https://www.i2k2.com/hybrid-cloud-solutions/
[36] https://visualstudio.microsoft.com/
[37] https://www.geeksforgeeks.org/features-of-c-20/
[38] https://aws.amazon.com/cloudwatch/
[39] https://prometheus.io/
[40] https://grafana.com/grafana/dashboards/
[41] https://www.ibm.com/docs/en/i/7.4?topic=functionality-cgi