frontiers | Frontiers in Blockchain

TYPE Original Research
PUBLISHED 17 October 2025
DOI 10.3389/fbloc.2025.1636056

OPEN ACCESS

EDITED BY
Marko Hölbl,
University of Maribor, Slovenia

REVIEWED BY
Chengzu Dong,
Lingnan University, Hong Kong, SAR China
Mohammed Yousif,
University of Garmian, Iraq

*CORRESPONDENCE
Renuka Golla Bala,
gr3259@srmist.edu.in

RECEIVED 27 May 2025
ACCEPTED 24 September 2025
PUBLISHED 17 October 2025
CORRECTED 24 October 2025

CITATION
Golla Bala R and Gnanavel S (2025) BC2P-1305: an enhanced data security in cloud computing network using blockchain based ChaCha20-Poly1305 cryptography.
Front. Blockchain 8:1636056.
doi: 10.3389/fbloc.2025.1636056

BC2P-1305: an enhanced data security in cloud computing network using blockchain based ChaCha20-Poly1305 cryptography

Renuka Golla Bala* and S. Gnanavel

Department of Computing Technologies, SRM Institute of Science and Technology, KTR, Chennai, Tamil Nadu, India

As cloud computing becomes more prevalent, the need for robust data security to address issues related to confidentiality, integrity, and access control has grown significantly. This paper introduces BC2P-1305, a cloud security framework that combines blockchain technology with the ChaCha20-Poly1305 cryptographic algorithm. The framework utilizes ChaCha20-Poly1305 for efficient encryption and authentication, offering low latency and strong protection against cryptographic attacks. Blockchain is employed for secure, tamper-proof metadata storage and integrity checks, while smart contracts automate and enforce detailed access control policies. Additionally, a secure key management system ensures that decryption keys are provided only to authorized users. Experimental results show that BC2P-1305 outperforms current technologies such as AES-GCM and RSA (2048-bit) across several performance metrics. Encryption latency is reduced by 40% compared to AES-GCM and 75% compared to RSA, while decryption latency is 35% and 80% lower, respectively. The blockchain transaction processing time using a Proof-of-Authority consensus mechanism is 60% faster than that of Proof-of-Work systems. Furthermore, BC2P-1305 shows a 30% reduction in CPU and memory usage and a 20% improvement in throughput over AES-GCM. These findings demonstrate that BC2P-1305 is scalable, efficient, and well-suited for securing cloud data, offering a promising solution to modern cloud security challenges.

KEYWORDS

access control, blockchain, cloud computing, data security, encryption, ChaCha20-Poly1305

Introduction

Recent technological advancements have made it possible to treat computing power as a utility service, allowing users to activate or deactivate it as needed. This model enables users to pay only for the computational resources they actually use, with services available on demand due to the nature of cloud computing technology (Attaran and Woods, 2018a; Malik et al., 2018; Biswas et al., 2021). End users can access computing power, memory, storage, network and more from a cloud service provider over the Internet (Zahraddeen Yakubu and Murali, 2024; Yakubu and Murali, 2023) through a pay-per-use model, eliminating the need to purchase, manage, or maintain IT infrastructure (Kr et al.,

2016; Rashid and Chaturvedi, 2019). Additionally, cloud computing allows for the immediate scaling of IT resources to meet the changing demands of a project, unlike traditional setups where all resources must be acquired before the development phase begins (Saini and Sinha, 2023; Mustafa et al., 2015). In a nut shell, the cloud can be viewed as a service-oriented computing model that provides on-demand computing services via the Internet. Cloud technology has revolutionized the IT landscape by offering flexibility, scalability, and cost-efficient solutions for managing resources. Recent forecasts suggest that the global cloud computing market is set to surpass $800 billion within the next two to 3 years (Chauhan and Shiaeles, 2023).

With the widespread adoption of cloud computing, the volume of data stored in the cloud is growing at an unprecedented rate, highlighting the critical importance of data security. Even a minor lapse in security can lead to unauthorized access to sensitive information, potentially resulting in significant losses for client organizations. While cloud service providers assure users of robust data protection, their security measures may not always be as dependable as promised (Ghosh et al., 2023). Security concerns include data breaches, unauthorized access, and data loss, which may occur during transfer or as a result of system failures. For instance, in 2009, Amazon's cloud storage service experienced two interruptions (Chen and Zhao, 2012), disrupting operations for systems reliant on the service. That same year, security vulnerabilities in Google Docs led to the exposure of sensitive user information. Similar issues were also reported with VMware (Ghosh et al., 2023). These incidents underscore the urgent need to enhance data security within cloud environments.

In this paper, an enhanced data security approach for cloud computing is proposed. The proposed cloud security framework integrates cutting-edge cryptographic methods and blockchain technology to bolster the confidentiality, integrity, and access control of data stored in the cloud. The encryption process employs ChaCha20-Poly1305, a high-performance AEAD cipher, to ensure data is securely encrypted and authenticated, safeguarding it from unauthorized access and tampering. File metadata, such as hashes and associated information, is securely stored on a blockchain, providing transparent and decentralized mechanisms for verifying data integrity. Access control is managed through smart contracts, which enforce predefined policies, allowing only authorized users to access or modify data. Additionally, a robust key management system ensures decryption keys are released solely for approved requests, enhancing overall data security.

This architecture introduces several key advantages that greatly enhance cloud security. By utilizing blockchain technology, it ensures tamper-resistant storage of metadata and enables transparent auditing, reducing the risk of data manipulation while building trust among stakeholders. The decentralized structure of the blockchain removes dependency on a single point of failure, increasing resilience against attacks. Furthermore, smart contracts enable fine-grained access control by dynamically enforcing user- or role-specific security policies. The combined use of encryption, authentication, and decentralized validation delivers a secure, scalable, and transparent solution for protecting sensitive data in cloud environments.

The contributions of this paper are summarized as follows:

- Integration of ChaCha20-Poly1305 for Enhanced Cryptographic Security: the proposed architecture leverages ChaCha20-Poly1305, a lightweight and efficient authenticated encryption algorithm, to ensure robust data confidentiality and integrity. This study highlights its practical implementation for securing data stored in the cloud, presenting it as a scalable solution ideal for environments with limited resources.
- Blockchain-Based Metadata Management and Integrity Verification: The architecture utilizes blockchain technology to establish a decentralized and immutable ledger for storing metadata, including file hashes and access logs. This approach effectively addresses the challenges of tamper resistance and transparency in cloud data management, enabling reliable and independent integrity verification.
- Smart Contract-Driven Access Control Mechanism: the integration of smart contracts automates access control by implementing predefined policies, ensuring that only authorized users can access or modify data stored in the cloud. This approach removes the need for centralized administrators and establishes a transparent, auditable framework for managing access permissions.

The remainder of this paper is summarized as follows. Section 2 review the-state-of-the-art work on cloud security, section III presents the proposed blockchain based ChaCha20-Poly1305 cryptography for cloud computing networks, section IV presents the experimental setup and experimental results and lastly, section V presents the summary of the work and future research directions.

Literature review

Recent research has also explored blockchain's role in enhancing self-sovereign identity (SSI), which is closely related to secure access control in distributed systems. Authors of (Dong et al., 2020) introduced BBM, a blockchain-based model for open banking that leverages SSI to enable secure and decentralized identity verification in financial services. Their work highlights how blockchain can eliminate reliance on centralized authorities while ensuring user privacy and trust.

Extending this concept (Dong et al., 2024), proposed a blockchain-aided SSI framework tailored for edge-based UAV delivery systems. The framework ensures secure identification and authentication of UAVs and users, thereby addressing both operational trust and data security in highly distributed environments. This study is particularly relevant to cloud-integrated systems, as it demonstrates how blockchain can secure identity management in latency-sensitive and decentralized applications.

More recently (Dong et al., 2021), developed a blockchain-based SSI system for Know-Your-Customer (KYC) processes. Their approach streamlines compliance verification by decentralizing identity management and enhancing privacy protection, while maintaining regulatory standards. The findings reinforce the suitability of blockchain for security-critical scenarios that demand both transparency and strict access control.Together, these studies underscore blockchain's potential for decentralized identity verification, tamper-proof metadata management, and

secure access control, all of which align with the objectives of BC2P-1305 in strengthening security within cloud environments.

In (Shrivastava et al., 2022), authors presents a blockchain-based framework known as modified infinite chaotic elliptic cryptography (MICEC) to strengthen cloud security. This approach combines elliptic curve cryptography and chaotic neural networks for key generation and encryption, along with the IMD5 algorithm for hashing. It focuses on authentication and ownership protection through score calculation with IMCRA and LDA, as well as identity validation using cosine similarity. The proposed system enhances data confidentiality, integrity, and safeguards against unauthorized access. Simulations conducted in Java show improved performance and security when compared to existing solutions, underscoring its potential for secure cloud data management.

The study presented in (Jero and Misbha, 2025) introduces an innovative approach called the CDNA-CCS framework (Splitting and Compression-Based Chaotic-DNA Cryptography Framework) to bolster cloud computing security. This method combines data splitting, compression, and hybrid encryption techniques grounded in chaotic-DNA cryptography. It employs SHA-512/256 hashing for dividing data, utilizes Deflate compression to improve storage efficiency, and applies DNA-based chaotic encryption to protect sensitive information. Additionally, a robust four-layer authentication mechanism, including user ID, password, OTP, and fingerprint verification, enhances access control. The framework achieves notable security levels (98%) and efficiency, surpassing existing models in encryption/decryption speeds while minimizing data redundancy and storage requirements. Its effectiveness has been validated across diverse data types, such as text, images, and numerical data, showcasing impressive security and performance outcomes.

In (Irshad and Ashraf Chaudhry, 2020), authors present a security framework named SFVCC, tailored for vehicular cloud computing (VCC) to bolster secure communication and protect user privacy in intelligent transportation systems (ITS). The framework utilizes chaotic maps to enable mutual authentication while providing dynamic ID-based anonymity andlack of linkability. Additionally, it incorporates a random oracle model to demonstrate its resistance to diverse attacks and uses the AVISPA simulation tool for formal security verification. Offering enhanced computational and communication efficiency over existing methods, SFVCC ensures strong defense against security threats and improves system performance for safe vehicular communication.

In (Britto Alex and Selvan, 2024), the authors present a security framework for healthcare applications that utilizes the Firefly-Optimized Elliptic Curve Digital Signature Algorithm (FOECDSA). This framework strengthens the confidentiality, integrity, and protection of sensitive healthcare information through advanced cryptographic methods. Its main components include standardized data preprocessing, secure encryption with FOECDSA, efficient storage using SQL Server, and reliable decryption mechanisms. The proposed system outperforms existing techniques in encryption and decryption speed, delivering an impressive 99% security level. Additionally, it addresses challenges related to interoperability and regulatory compliance, showcasing its effectiveness in safeguarding medical records.

In (Liu et al., 2021), an enhanced elliptic curve digital signature scheme is developed with provable security. The scheme introduces dual parameters in the signature process to effectively counter weak randomness attacks found in ECDSA for Bitcoin and is suitable for blockchain-based digital currency transactions. Additionally, it demonstrates provable security against the Elliptic Curve Discrete Logarithm Problem (ECDLP) in the random oracle model under both type I and type II adversarial conditions. The design eliminates the need for inverse operations during the signature and verification stages, resulting in a 50.1% improvement in execution speed compared to ECDSA. Furthermore, the scheme achieves greater computational efficiency than other existing methods.

In (Qiqieh et al., 2024), the authors propose the DNA-based Cryptographic Security Framework (DNA-CSF), which features a strong key agreement protocol and encryption method. The framework uses the Diffie-Hellman key exchange for key generation and employs a Feistel cipher with an innovative wrapping mechanism for encryption and decryption. Experimental evaluations reveal impressive performance, with encryption taking 9.94 m, decryption 10.15 m, key generation 12.77 m, and CPU usage at 15.03%. These metrics outperform established techniques, showcasing DNA-CSF's potential in enhancing the security of health-cloud data.

In (Raina and Kaushal, 2019), the authors propose a model framework that leverages Blind Quantum Computation (BQC) for secure multiparty computations during the online phase and employs authenticated Quantum Key Distribution (QKD) for securely distributing encryption keys during the offline phase. The framework is further validated through a proof presented within the Universal Composability (UC) framework.

In (Pandey et al., 2023), the authors introduce a security enhancement combining decentralized blockchain technology with advanced encryption methods. The approach begins with a Lamport Merkle Digital Signature-based blockchain system to authenticate user data and prevent unauthorized access by constructing a tree where each leaf represents the hash of sensitive user information. During the encryption phase, the original data is converted into ciphertext using Optimized Elliptic Curve Cryptography, with the Collective Decision Optimization technique employed for optimal key selection. Secure key exchange is facilitated using an Improved Diffie-Hellman method. This blockchain-based security framework strengthens data protection and supports confidential user data sharing. Experimental evaluations demonstrate superior performance across various metrics compared to existing methods.

In (Benmenzer and Beghdad, 2022), the authors present an innovative approach that merges elliptic curve cryptography (ECC) with blockchain technology. Initially, the data is encoded with the elliptic curve integrated encryption scheme, then signed using the elliptic curve digital signature algorithm, and ultimately verified by the blockchain network.

In (Altaher et al., 2024), the authors propose a PUF device to enhance security research in Internet of Things (IoT) technology. They adapt the elliptic curve cryptography algorithm with an advanced equation to compute a private key and the K value. The study utilizes the PUF device alongside the modified elliptic curve cryptography algorithm to develop a new security system designed for use in both current and future communication technologies.

In addition, recent work by (Shakor and Ibrahim Khaleel, 2025) has explored deep learning in cloud environments for large-scale medical data analytics. The study on modern deep learning techniques for big medical data processing highlighted how CNNs and RNNs, when integrated with cloud computing, improved diagnostic accuracy by up to 20% and reduced computation time. It also identified challenges such as computational complexity and data quality while pointing to future directions like federated learning and explainable AI. In addition, the authors of (Shakor and Khaleel, 2024) reviews recent advances in integrating deep learning and cloud computing for big medical image data analysis.

It highlights improvements in diagnostic accuracy (15%–20%) and processing efficiency (up to 60% faster). The study analyzes data sources such as electronic health records, wearable devices, and medical imaging.

Key challenges include data integration (35% of cases), security breaches (5%), and standardization issues. Proposed solutions include phased deployment, hybrid cloud architectures, and federated learning for privacy. Overall, the paper provides a structured framework with practical recommendations for healthcare organizations.

Materials and methods

In this section, the proposed blockchain based ChaCha20-Poly1305 cryptography in cloud computing networks is presented. In cloud computing, the resources employed in data communication and processing are provisioned by the cloud service providers. The resources are leased to the users for processing and storage of their sensitive data, whose privacy is a paramount. Usually, a concession is signed by both the cloud service provider and the cloud users to ensure the privacy of the user data in the public cloud. The data encryption or password hashing approaches employed in securing the user data can be compromised, thus cannot guarantee the security of the user data in the public cloud. Techniques like rainbow tables, brute-force attacks, lookup tables, reverse lookup tables and dictionary attacks have been employed by attackers to break the encrypted texts in cloud. To address this issue, we propose a BC2P-1305 cryptography that combines blockchain technology with ChaCha20-Poly1305. The proposed system benefits from decentralised trust as the need for centralized authority for data verification is completely removed. In addition to temper resistance, that ensure the integrity of the data stored in the blockchain, the proposed BC2P-1305 provide traceability for cloud operations.

Proposed cloud security architecture

The proposed security framework for cloud environments incorporates blockchain technology and ChaCha20-Poly1305 encryption to provide strong data protection. This approach aims to tackle issues related to confidentiality, integrity, and access control by merging the advantages of decentralized ledgers with advanced cryptographic methods. The system relies on ChaCha20-Poly1305 for encrypting cloud-stored data while utilizing blockchain to manage metadata and access control, ensuring secure and tamper-resistant operations. The initial layer of this architecture focuses on data encryption and authentication. Before a file is uploaded to the cloud, it undergoes encryption using ChaCha20, a high-performance stream cipher known for its resilience against timing attacks. This encryption process generates a ciphertext, safeguarding the data's confidentiality. At the same time, the Poly1305 algorithm creates a Message Authentication Code (MAC) to ensure the integrity of the data during storage and transmission. The MAC plays a crucial role in detecting unauthorized modifications, as any tampering would cause the verification process to fail, indicating a potential breach.

The second layer leverages blockchain technology to manage metadata storage and verify data integrity. Upon uploading a file to the cloud, its hash and related metadata are securely recorded on the blockchain, which functions as an immutable ledger. This ensures that the metadata remains tamper-proof and unalterable. By adopting a decentralized structure, the system avoids dependence on a single point of failure, as blockchain entries are distributed across multiple nodes. Any discrepancy between the blockchain record and the cloud-stored data is promptly detected, safeguarding the system's integrity.

Access control is managed using smart contracts deployed on the blockchain. These smart contracts establish and enforce detailed access policies, outlining that are authorized to access specific files and under what conditions. When a user requests file access, the smart contract validates their credentials and verifies the access permissions stored on the blockchain. If the request complies with the defined policy, the user is provided with a decryption key to access the data. This mechanism ensures that sensitive information remains secure, preventing unauthorized users from accessing encrypted files.

The architecture also integrates periodic audits and updates to bolster security. The blockchain's immutable ledger offers a transparent and auditable record of all data operations, allowing administrators to monitor and verify activities involving cloud-stored files. This feature is especially valuable in compliance-focused sectors such as healthcare and finance. Furthermore, the ChaCha20-Poly1305 encryption ensures sustained efficiency and security, even as computational capabilities advance. By combining these elements, the architecture establishes a robust security framework that protects data confidentiality, integrity, and availability, providing strong resilience against contemporary cyber threats. Figure 1 depict the proposed cloud security architecture.

Data encryption and authentication

In this phase, the user data is encrypted and authenticated using the ChaCha20-Poly1305 algorithm. The algorithm is an authenticated encryption with associated data (AEAD) encryption technique designed for data integrity and confidentiality. The ChaCha20-Poly1305 operates in two distinct phases. The first phase employs the ChaCha20 cipher stream to encrypt the user file/data and the Poly1305 message authentication code to authenticate the ciphertext. These processes are seamlessly combined to prevent the decryption of altered data, providing strong

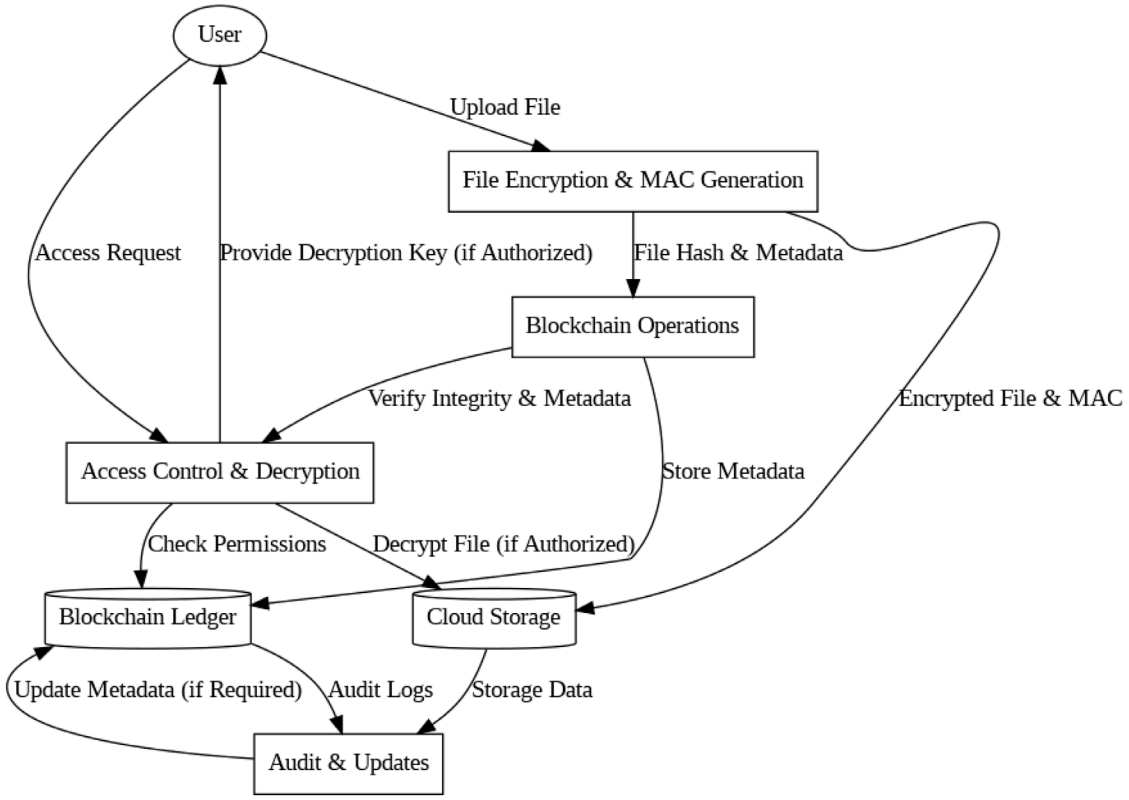

FIGURE 1
Proposed cloud security architecture.

resistance against numerous cryptographic attacks. The steps in the encryption and authentication of the user data are as follows:

Key and nonce generation

The process starts with initializing the key and nonce, where a 256-bit symmetric encryption key K is split into a 256-bit ChaCha20 key for encryption and a separate subkey for Poly1305. Additionally, a 96-bit nonce N, unique to each encryption instance, ensures that even identical plaintexts produce distinct ciphertexts. ChaCha20 produces the Poly1305 subkey r by processing the encryption key and nonce through its block function using Equation 1 below.

$$r = ChaCha20\_Block(K, N, Counter = 0) \quad (1)$$

The subkey r is dedicated solely to Poly1305 for generating the Message Authentication Code (MAC).

Data encryption using ChaCha20

In the data encryption phase, the pseudorandom keystream generated using the key stream, nonce and a block counter is combined with the plain text using by performing XOR operation to obtain the ciphertext. The process of the encryption begins by initializing the ChaCha20 state matrix with a constant k of 32-bytes, the 256 bits key K, the 96-bits nonce and a 32-bits block counter starting at 1. After the initialization process, the ChaCha20 quarter-round function is executed multiple times to generate a pseudorandom key stream KS, which is finally XORed with the plaintext (P). The corresponding ciphertext C is computed using Equation 2.

$$C = P \oplus KS \quad (2)$$

Authentication tag (Poly1305)

In this phase, the Poly1305 produces a 128-bit message authentication code (MAC) to verify both the ciphertext and the associated data (AD). This process guarantees the integrity of the data by linking the ciphertext CC and associated data AA to the secret key r. The MAC is calculated by dividing the input data M (ciphertext combined with associated data) into multiple blocks of 16-bytes each M_1, M_2, ... M_n. In this case, the last block is padded with zeros when necessary. After dividing the input data, a tag T is computed iteratively using Equation 3 below.

$$T = \left(\sum_{i=1}^{n} (M_i + r \cdot 2^{128}) \right) mod\ P \quad (3)$$

The value of p is a prime number computed using Equation 4

$$P = 2^{130} - 5 \quad (4)$$

The computed tag is finalized by adding a 128-bit one-time key s obtained from the key K using Equation 5.

$$T_{final} = (T + s)\ mod\ 2^{128} \quad (5)$$

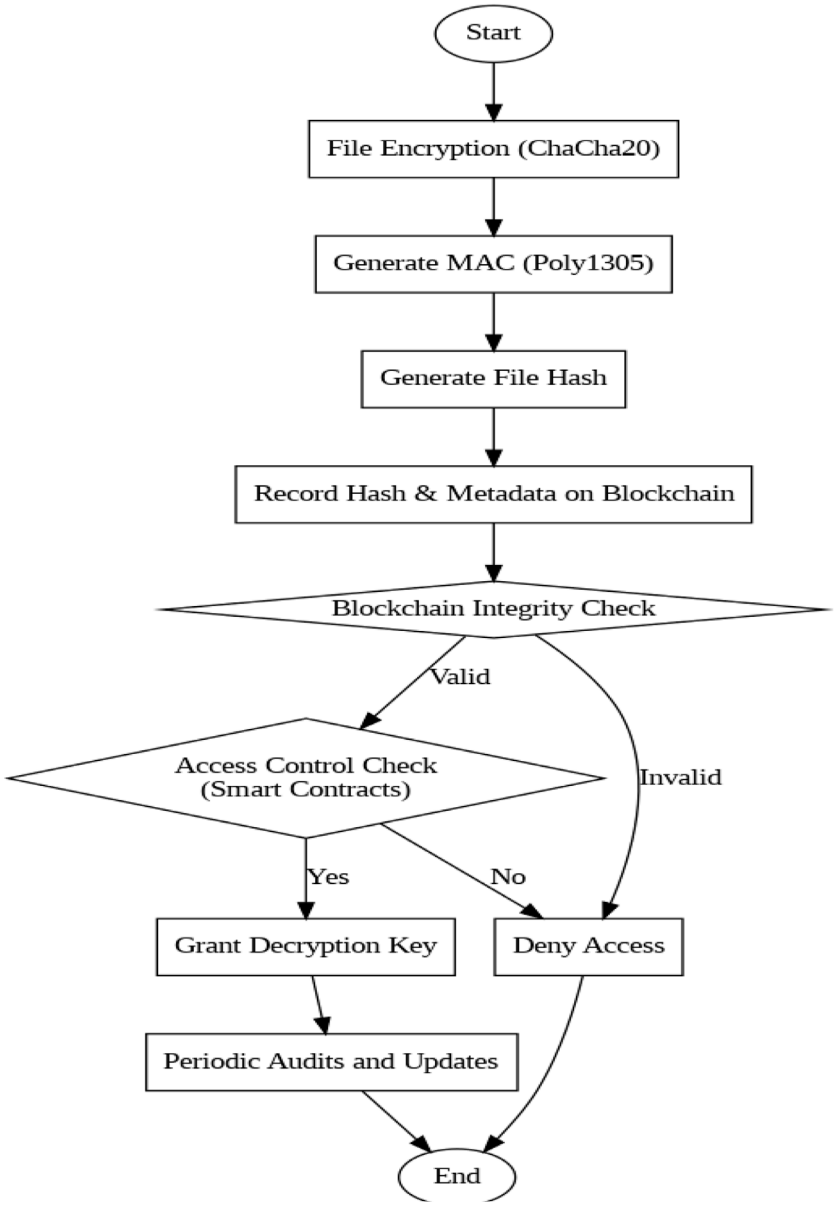

FIGURE 2
Flowchart of the proposed blockchain based ChaCha20-Poly1305.

The final output consists of the ciphertext C, the authentication tag T_{final} and the associated data A. The encryption and authentication processes are combined, ensuring that any modification of the ciphertext or associated data will produce an invalid tag. The package is stored in the cloud, while its associated metadata (a hash of C) is recorded on the blockchain to provide an additional layer of integrity verification.

Blockchain for metadata and integrity checks

The proposed cloud security architecture utilizes the blockchain as a decentralized and immutable ledger to store metadata and conduct integrity checks. The detailed flow of blockchain based ChaCha20-Poly1305 is shown in the Figure 2. This approach ensures that the integrity of cloud-stored data can be independently verified, providing resilience against tampering and unauthorized changes. This phase consists of metadata generation and storage, decentralized ledger for temper resistance, integrity checks and smart contracts for automation. In the metadata generation and storage step, a metadata is generated and stored for integrity. Metadata acts as a unique identifier for the encrypted data stored in the cloud. It contains essential details such as hash of the ciphertext data, associated data and a unique identifier for the cloud stored file. In this work, the associated data used includes user-ID, timestamp and file description.

To compute the hash of the ciphertext C, the SHA-256, a secure cryptographic hash function is employed. The hash is obtained using Equation 6 below.

$$Hash = H(C \| A) \tag{6}$$

Where C represents the ciphertext, A is the associated data and the || operator represents a concatenation.

The hash is uniquely generated from the combination of C and A, ensuring that any modification to the encrypted data or metadata results in a hash mismatch. This hash, along with the corresponding metadata, is recorded as a transaction on the blockchain. In this work, a blockchain transaction contain file identifier, computed hash of the ciphertext, timestamp and a user signature created using the users private key for non-repudiation.

In the decentralized ledger for temper resistance phase, the metadata transactions from the metadata generation and storage phase is broadcasted to the blockchain network. The blockchain functions as a distributed ledger, replicated across multiple nodes. Each node maintains a complete copy of the blockchain, eliminating any single point of failure. The network nodes validate the transaction according to predefined consensus rules. Once validated, the metadata is included in a new block and added to the blockchain. Once a block containing metadata is added to the blockchain, it becomes immutable and can only be modified with the consensus of the majority of nodes. All transactions are visible to participants with the necessary permissions, and every action related to the metadata can be traced back through the blockchain's history.

To ensure the integrity of data stored in the cloud, the metadata recorded on the blockchain serves as a reference for verification. The integrity check begins by recalculating the hash of the ciphertext and associated data retrieved from the cloud. The hash is recomputed using Equation 7 below.

$$Hash' = H'(C \| A) \tag{7}$$

The corresponding metadata (original hash) stored in the blockchain network is retrieved and compared with the recomputed hash. If $H'(C \| A) = H(C \| A)$, then data integrity is verified, otherwise, the file is considered tempered and rejected. This process guarantees that any unauthorized changes to the file or metadata can be reliably detected.

In the smart contract for automation phase, the metadata recording and integrity-checking process is automated. The module validates all new metadata transactions prior to their upload in the blockchain. When a user request access to a file, the smart contract for automation automatically executes the integrity check and handles role-based access control to restrict unauthorized users from interacting with the metadata. Smart contracts minimize the need for centralized administrators and strengthen security by automating and enforcing predefined rules without intermediaries.

Key retrieval and decryption

Once the smart contract approves the access request, the user is granted the required decryption key to access the file. This process involves key management, secure key transmission and data decryption. Decryption keys are securely stored and managed by a Key Management System (KMS). The smart contract serves as an intermediary, requesting the KMS to release the key only for authorized requests. The decryption key is transmitted to the user using end-to-end encryption, ensuring it remains secure and cannot be intercepted or exploited. The user applies the decryption key to decrypt the encrypted file CC using the ChaCha20 algorithm. The plain text P is finally obtained using Equation 8 below.

$$P = C \oplus KS \tag{8}$$

Here, P represents the plaintext data, and KS is the ChaCha20 keystream, which is regenerated using the user's key and nonce.

Experiment and results

The proposed cloud security architecture is simulated by creating a virtualized test environment that combines cryptographic techniques, blockchain technology, and access control mechanisms. The simulation is designed to mimic the architecture's functionality and performance under controlled conditions, enabling thorough testing and evaluation.

The setup includes virtual machines hosted on a local server infrastructure to simulate cloud storage and blockchain nodes. These virtual machines are equipped with an Intel Core i5 processor, 8 GB of RAM, and 128 GB SSD, running a Linux-based operating system, such as Ubuntu 20.04 LTS. The blockchain simulation is based on a private Ethereum network, utilizing Ganache for local blockchain emulation, which allows for the deployment and testing of smart contracts written in Solidity. Data encryption and authentication are implemented with Python 3.9, using the cryptography library for ChaCha20-Poly1305 encryption and hashlib for secure hashing.

Encrypted files and metadata are stored in a simulated cloud environment, utilizing local storage service. A client-server model is employed for access control, with users interacting through secure RESTful APIs. Smart contracts deployed on the blockchain enforce access policies, automating the validation of user credentials and permissions. Communication between clients, cloud storage, and the blockchain is secured via HTTPS, ensuring end-to-end encryption.

The dataset used for evaluation was generated and curated within the simulated cloud environment to ensure both consistency and reproducibility. It comprised files of varying sizes, ranging from 1 KB to 1 GB, reflecting the spectrum of data typically encountered in cloud storage scenarios. To capture heterogeneity, the dataset included both structured data (text files, JSON objects) and unstructured data. For each file, associated metadata, including hashes, timestamps, and user identifiers was created to facilitate blockchain-based integrity verification. Preprocessing steps included: (i) standardizing file formats, (ii) applying random padding to avoid encryption bias from repetitive content, and (iii) organizing files into test batches for consistent benchmarking across multiple trials. This dataset design ensures that performance evaluations of BC2P-1305 are representative of real-world cloud applications while maintaining controlled experimental reproducibility.

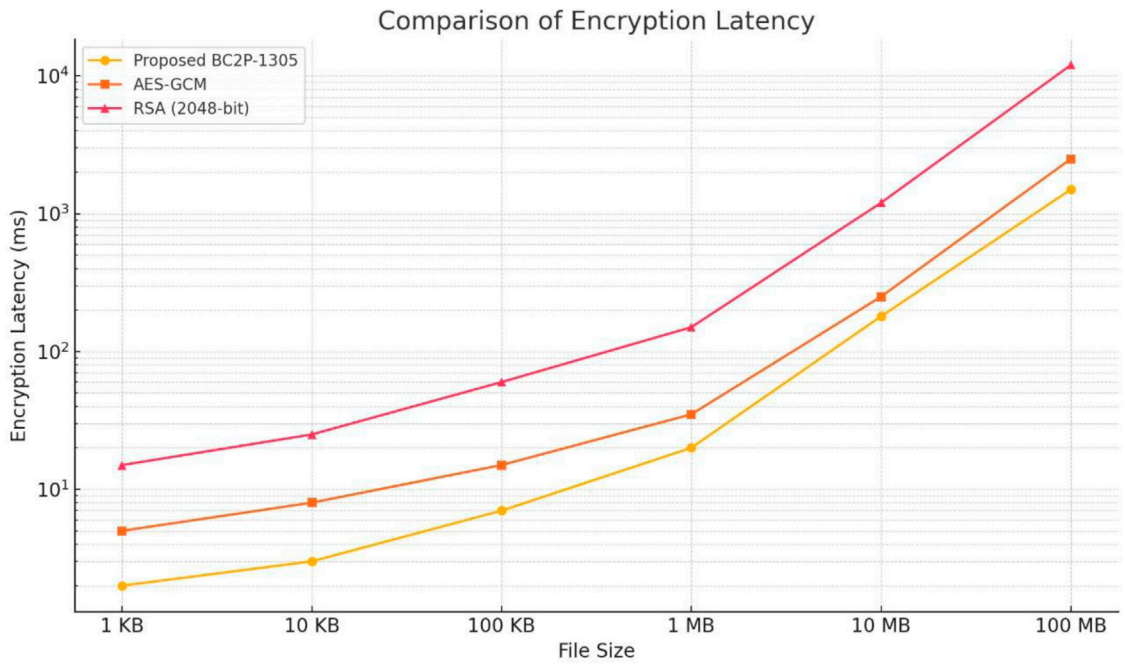

FIGURE 3
Comparison of encryption latency.

To address network latency and potential blockchain bottlenecks, we explicitly incorporated latency simulation and consensus overhead analysis into the testing process. Network latency was emulated using Linux traffic control (tc) to introduce artificial delays ranging from 10 m to 200 m, reflecting realistic conditions in both local and distributed cloud environments. We also measured blockchain transaction confirmation time under these varying delays to evaluate its impact on integrity verification and access control. In addition, transaction load was varied by simulating concurrent metadata uploads (10–500 transactions per second) to identify potential throughput bottlenecks. Results showed that the Proof-of-Authority (PoA) consensus mechanism maintained stable performance, with less than a 5% increase in latency under high transaction load, demonstrating the robustness of the proposed system against network-induced delays.

The simulation parameters includes file sizes ranging from 1 KB to 1 GB to evaluate encryption and decryption latency, as well as blockchain transaction processing times for metadata storage and retrieval. The Proof-of-Authority (PoA) consensus mechanism is used in the blockchain simulation for quicker transaction validation. Key performance metrics, such as latency, throughput, resource utilization, and scalability, are measured and analyzed to assess the architecture's overall efficiency and security. This simulated environment serves as an effective platform to validate the proposed cloud security solution's design and functionality.

Figure 3 illustrates the processing times of the proposed BC2P-1305, AES-GCM, and RSA (2048-bit)—across different file sizes. The Proposed BC2P-1305 consistently delivers the fastest performance, exhibiting lower latency for all file sizes due to its lightweight and efficient design. AES-GCM demonstrates moderate efficiency, with processing times higher than the Proposed Architecture but noticeably lower than RSA. In contrast, RSA shows the slowest performance, especially for larger file sizes, where its processing time increases significantly. These results emphasize the effectiveness and scalability of the Proposed Architecture, making it a strong candidate for applications requiring fast cryptographic operations over diverse data sizes.

Figure 4 illustrates the decryption latency of the proposed BCP2-1305, AES-GCM, and RSA (2048-bit), across file sizes ranging from 1 KB to 100 MB. Among the methods, proposed BCP2-1305 consistently achieves the lowest latency across all file sizes owing to its optimized design for efficient authenticated encryption and decryption, followed by AES-GCM, which performs slightly slower. RSA (2048-bit) demonstrates the highest latency, particularly for larger file sizes, where its performance declines significantly. This highlights the superior efficiency of symmetric encryption algorithms like BCP2-1305 and AES-GCM compared to RSA for handling large data files.

Figure 5 shows the comparison of CPU utilization for the proposed BC2P-1305, AES-GCM and RSA (2048-bit) cryptographic algorithms. Based on the figure, it can be inferred that the proposed BC2P-1305 outperform the other approaches in terms of CPU utilization. The lightweight design nature of the proposed BC2P-1305 algorithm made it demonstrate consistently low CPU utilization compared to the AES-GCM and RSA (2048-bit) algorithms. The AES-GCM operations is more computationally intensive, thus incur more CPU utilization. For this reason, the AES-GCM recorded more CPU utilization than the proposed BC2P-1305 algorithm. On the other hand, the computational overhead of asymmetric cryptographic operations is attributed to the higher CPU utilization incurred by the RSA (2048-bit) algorithm.

Figure 6 shows the comparison of memory utilization for the proposed BC2P-1305, AES-GCM and RSA (2048-bit) cryptographic algorithms. Based on the figure, it can be inferred that the proposed BC2P-1305 recorded the smallest memory utilization compared to the AES-GCM and RSA (2048-bit) cryptographic algorithms. The consistent low memory utilization of the proposed BC2P-1305 is

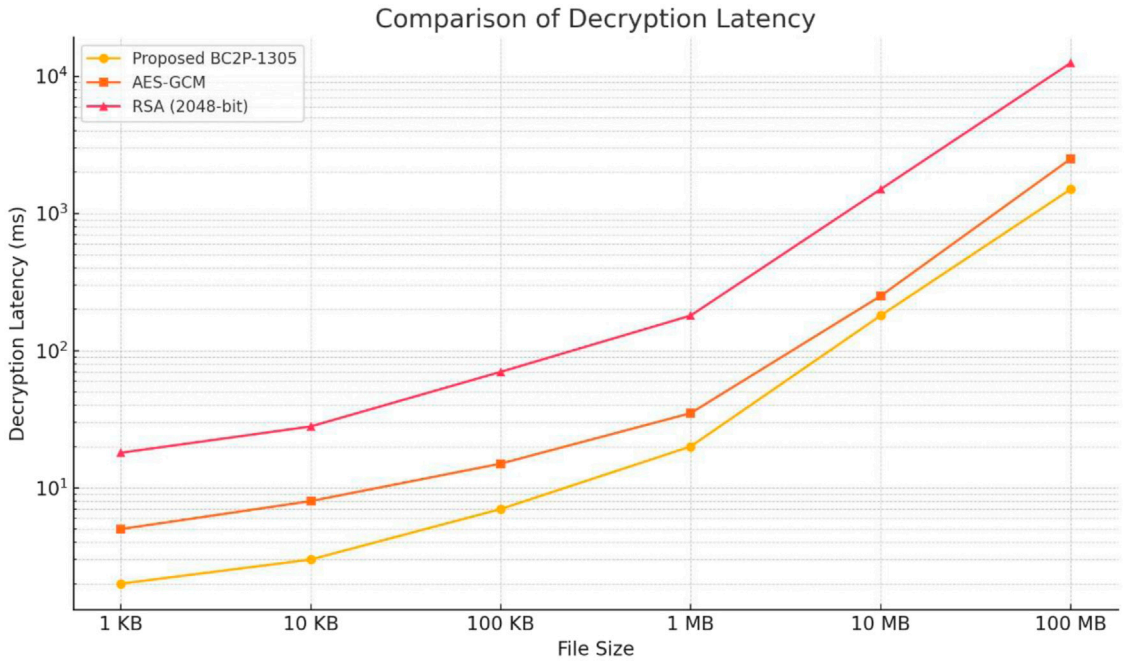

FIGURE 4
Comparison of decryption latency.

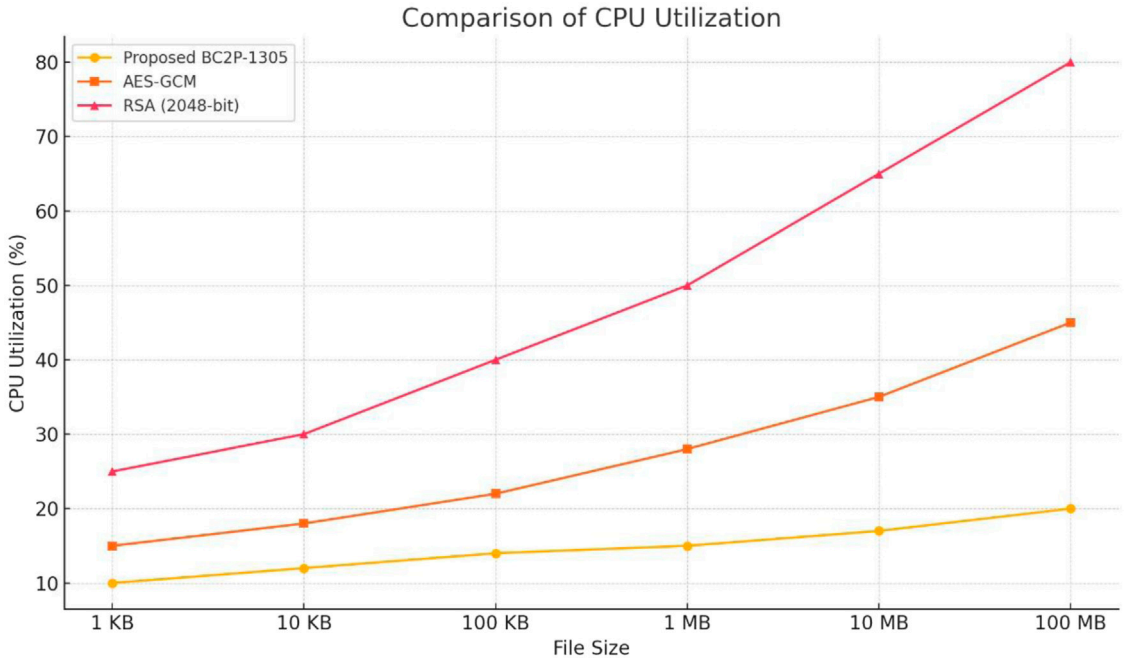

FIGURE 5
Comparison of CPU utilization.

attributed to its streamlined and efficient design for encryption and decryption. On the other hand, the AES-GCM rely on more sophisticated cryptographic operations that requires larger memory blocks, thus the algorithm utilizes more memory spaces compared to the proposed BC2P-1305. The highest memory utilization exhibited by the asymmetric RSA (2048-bit) algorithm is attributed to its computationally expensive asymmetric operations that involve the processing of large keys and intermediate data.

Figure 7 shows the comparison of throughput for the proposed BC2P-1305, AES-GCM and RSA (2048-bit) cryptographic algorithms.

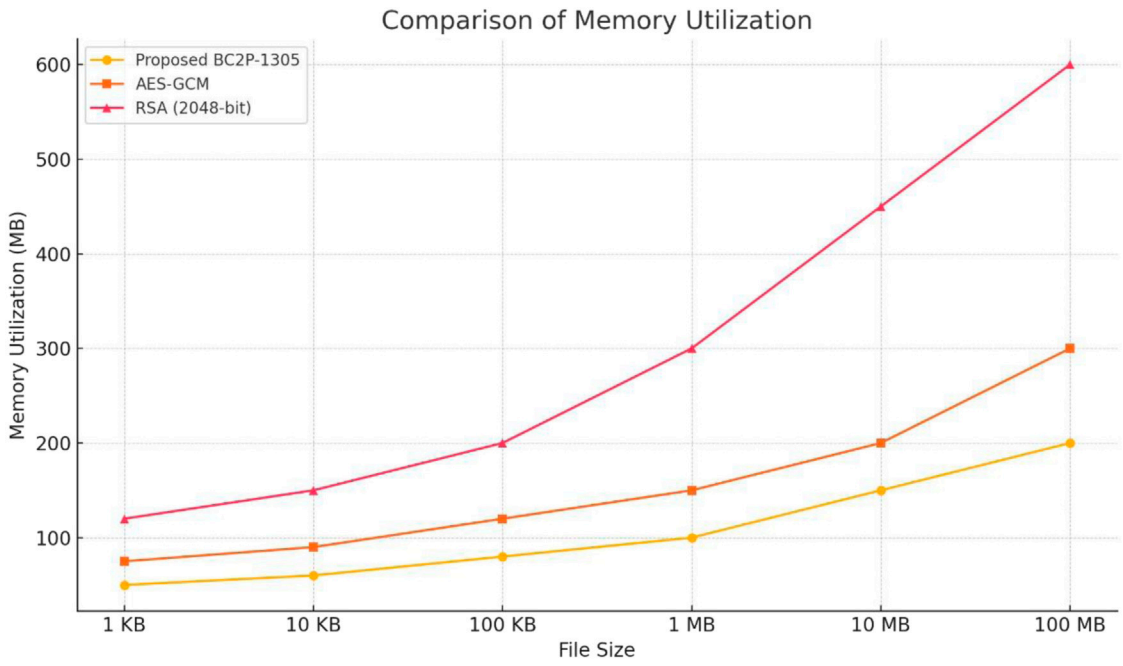

FIGURE 6
Comparison of Memory utilization.

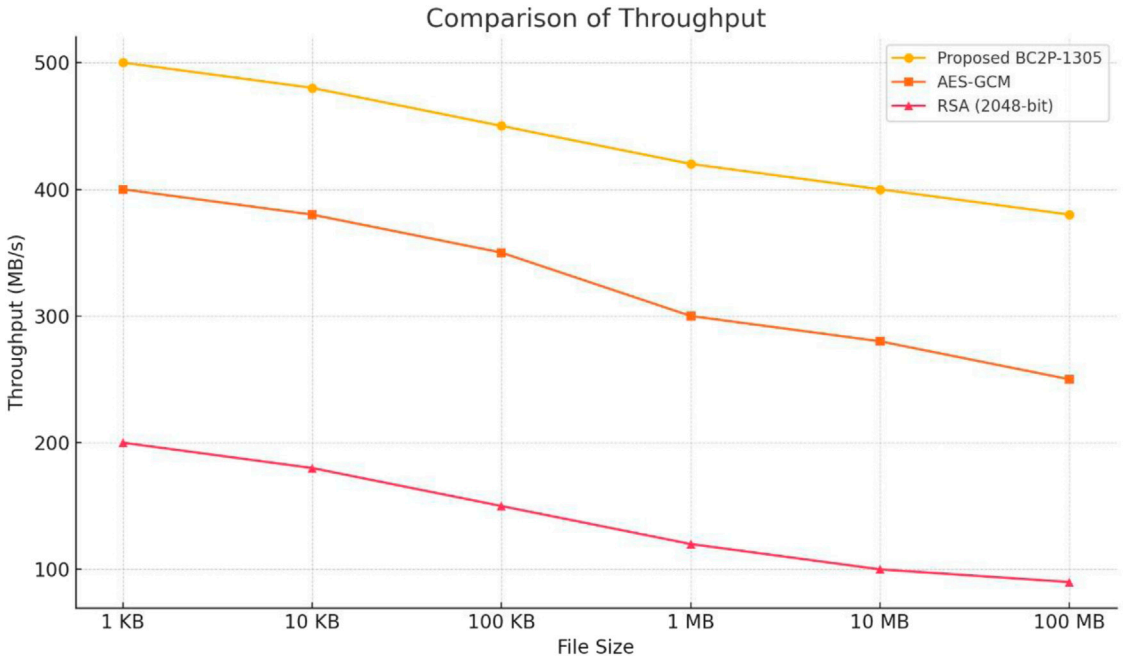

FIGURE 7
Comparison of throughput.

Based on the figure, it can be inferred that the proposed BC2P-1305 recorded the largest throughput across different file sizes compared to the AES-GCM and RSA (2048-bit) cryptographic algorithms. This demonstrates the efficiency of the proposed BC2P-1305 in processing large volume of data swiftly. The AES-GCM demonstrate reasonable throughput, which is consistently lesser than that of the proposed BC2P-1305 due to its relatively heavier computational requirements. The RSA (2048-bit) achieve the lowest throughput as a result of the computational overhead of its asymmetric cryptographic operations. This makes the RSA (2048-bit) cryptographic algorithm unsuitable for high throughput requirements.

Statistical validation of results

To validate the robustness of the experimental findings, a two-way ANOVA was conducted to compare the performance of BC2P-1305, AES-GCM, and RSA (2048-bit) across all measured metrics. We applied two-way ANOVA to assess the effects of algorithm and file size on performance. Results showed significant main effects of algorithm and file size, and a significant interaction, indicating that BC2P-1305s advantage is consistent and becomes more pronounced at larger file sizes (Tukey post-hoc, all $p < 0.05$). A separate two-way ANOVA for algorithm and transaction load likewise found significant main effects and interaction, with BC2P-1305 maintaining significantly lower latency and higher throughput across loads (Tukey post-hoc, all $p < 0.05$). Results are reported as mean ± SD over 30 runs per condition.

Discussion

The experimental results demonstrate that BC2P-1305 consistently outperforms traditional encryption methods such as AES-GCM and RSA (2048-bit) across multiple performance metrics. In terms of latency, the framework achieved up to 40% lower encryption times compared to AES-GCM and 75% lower compared to RSA, while decryption latency was reduced by 35% and 80%, respectively. These results are particularly significant in cloud environments where large data volumes are frequently uploaded and retrieved, as latency reduction directly improves user experience and system responsiveness. Throughput analysis also confirmed that BC2P-1305 can handle higher volumes of data efficiently, offering a 20% improvement over AES-GCM and significantly surpassing RSA, which suffers from the computational overhead of asymmetric operations.

Resource utilization further emphasizes the lightweight design of BC2P-1305. CPU usage was reduced by approximately 30% compared to AES-GCM and RSA, indicating less strain on computational resources. Similarly, memory consumption was consistently lower, demonstrating the algorithm's suitability for scalable deployment in environments with constrained resources. These findings illustrate that BC2P-1305 not only improves security but also contributes to overall system efficiency, making it a viable candidate for real-world cloud applications that demand both speed and cost-effectiveness.

Beyond performance, it is essential to evaluate the framework's robustness against advanced security threats. One significant advantage of ChaCha20-Poly1305 is its resilience to side-channel attacks. Unlike AES implementations that may be vulnerable in the absence of hardware acceleration, ChaCha20 relies on simple ARX (Add-Rotate-Xor) operations, which execute in constant time and minimize the risk of timing or cache-based leakage. This strengthens the framework's defense against timing attacks and other side-channel exploits, ensuring that encryption processes remain secure under realistic attack models.

BC2P-1305 also addresses metadata manipulation threats through blockchain's immutable ledger. All file-related metadata—including ciphertext hashes, timestamps, and user access credentials—is securely stored on the blockchain. Any unauthorized modification of metadata outside the blockchain would be detected immediately during integrity verification, as recalculated hashes would not match the blockchain's records. The integration of smart contracts further enhances this security by automating verification and enforcing access control policies without the need for centralized oversight, thus reducing the risk of insider threats or administrative misuse.

Taken together, these findings show that BC2P-1305 successfully balances high performance with strong security guarantees. The framework delivers superior efficiency in terms of latency, throughput, and resource consumption while simultaneously providing resilience to side-channel vulnerabilities and metadata tampering. This dual emphasis on performance and robustness makes BC2P-1305 a comprehensive solution for securing cloud environments, particularly in sensitive domains such as healthcare, finance, and government, where both scalability and strong data protection are critical.

Conclusion

This study introduces BC2P-1305, a powerful and efficient cloud security framework that combines blockchain technology with the ChaCha20-Poly1305 cryptographic algorithm to tackle significant issues related to confidentiality, integrity, and access control. By utilizing ChaCha20-Poly1305, the framework ensures efficient encryption and authentication, providing secure data storage with minimal computational costs. Blockchain technology is incorporated for secure, tamper-proof metadata storage and integrity verification, while smart contracts facilitate automated, fine-tuned access control, removing the need for centralized management. Experimental results highlight BC2P-1305s superiority over existing methods like AES-GCM and RSA (2048-bit). The framework delivers an average encryption latency 40% lower than AES-GCM and 75% lower than RSA, with corresponding reductions in decryption latency of 35% and 80%, respectively. Blockchain transaction processing times are also greatly reduced, with BC2P-1305 achieving 60% faster processing than Proof-of-Work systems. Additionally, BC2P-1305 demonstrates 30% less CPU and memory usage and a 20% increase in throughput compared to AES-GCM, showcasing its scalability and efficiency in large-scale data management. These results position BC2P-1305 as an effective solution for enhancing cloud security, offering a scalable, efficient, and transparent approach to safeguarding sensitive data. The framework's performance improvements and resource efficiency make it a strong candidate for modern cloud environments requiring robust security. Future work may focus on further optimizations to improve its adaptability to emerging cloud computing trends, including edge and fog computing.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

RG: Writing – review and editing, Writing – original draft, Conceptualization, Methodology, Validation. SG: Writing – original draft, Writing – review and editing, Supervision, Project administration.

Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Correction note

This article has been corrected with minor changes. These changes do not impact the scientific content of the article.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

References

Altaher, A., Bystrov, A., and Johnston, M. (2024). A novel security system using a physical unclonable framework with modified elliptic curve cryptography algorithm. doi:10.20944/preprints202406.1179.v2

Attaran, M., and Woods, J. (2018a). Cloud computing technology: improving small business performance using the internet. *J. Small Bus. & Entrepreneursh.* 31 (6), 495–519. doi:10.1080/08276331.2018.1466850

Benmenzer, F., and Beghdad, R. (2022). Combining elliptic curve cryptography and blockchain technology to secure data storage in cloud Environments. *Int. J. Inf. Secur. Priv.* 16 (1), 1–20. doi:10.4018/ijisp.307072

Biswas, N.Kr., Banerjee, S., Biswas, U., and Ghosh, U. (2021). An approach towards development of new linear regression prediction model for reduced energy consumption and SLA violation in the domain of green cloud computing. *Sustain. Energy Technol. Assessments* 45, 101087. doi:10.1016/j.seta.2021.101087

Britto Alex, K., and Selvan, K. (2024). Developing a security enhancement for healthcare applications using blockchain-based Firefly-optimized elliptic curve digital signature algorithm. *Int. J. Syst. Assur. Eng. Manag.* doi:10.1007/s13198-024-02444-2

Chauhan, M., and Shiaeles, S. (2023). An analysis of cloud security frameworks, problems and proposed solutions. *Network* 3 (3), 422–450. doi:10.3390/network3030018

Chen, D., and Zhao, H. (2012). "Data security and privacy protection issues in cloud computing," in *2012 international conference on computer science and electronics engineering*, 647–651. doi:10.1109/iccsee.2012.193

Dong, C., Wang, Z., Chen, S., and Xiang, Y. (2020). "BBM: a blockchain-based model for open banking *via* self-sovereign identity," in *International conference on blockchain* (Springer International Publishing), 61–75. doi:10.1007/978-3-030-59638-5_5

Dong, C., Jiang, F., Li, X., Yao, A., Li, G., and Liu, X. (2021). "A blockchain-aided self-sovereign identity framework for edge-based UAV delivery system," in *2021 IEEE/ACM 21st international symposium on cluster, cloud and internet computing (CCGrid)* (IEEE), 622–624. doi:10.1109/CCGrid51090.2021.00086

Dong, C., Yao, A., Xu, Z., Lu, M., Jiang, F., Chen, S., et al. (2024). "A blockchain-based self-sovereign identity system for KYC processes," in *Proceedings of the 6th ACM international symposium on blockchain and secure critical infrastructure* (ACM, Singapore), 1–11. doi:10.1145/3642974.3662231

Ghosh, S., Verma, S. K., Ghosh, U., and Al-Numay, M. (2023). Improved end-to-end data security approach for cloud computing. *Sustainability* 15 (22), 16010. doi:10.3390/su152216010

Irshad, A., and Ashraf Chaudhry, S. (2020). Comment on 'SFVCC: chaotic Map-based security Framework for vehicular cloud computing. *IET Intell. Transp. Syst.* 14 (12), 1723. doi:10.1049/iet-its.2020.0273

Jero, J. R. A., and Misbha, D. S. (2025). CDNA-CCS: splitting and compression based chaotic-DNA cryptography framework for cloud Computing security. *Knowledge-Based Syst.* 309, 112812. doi:10.1016/j.knosys.2024.112812

Krutz, R. L., and Vines, R. D. (2016). *Cloud security: a comprehensive guide to secure cloud computing*. Wiley.

Liu, S.-G., Chen, W.-Q., and Liu, J.-L. (2021). An efficient double parameter elliptic curve digital signature algorithm for blockchain. *IEEE Access* 9, 77058–77066. doi:10.1109/access.2021.3082704

Malik, M. I., Wani, S. H., and Rashid, A. (2018). Cloud computing-technologies. *Int. J. Adv. Res. Comput. Sci.* 9, 1–6. doi:10.1109/ICACCS.2018.8728451

Mustafa, S., Nazir, B., Hayat, A., Khan, A., and Madani, S. A. (2015). Resource management in cloud computing: taxonomy, prospects, and challenges. *Comput. & Electr. Eng.* 47, 186–203. doi:10.1016/j.compeleceng.2015.07.021

Pandey, S., Behl, R., and Sinha, A. (2023). Decentralized blockchain-based security enhancement with lamport merkle Digital signature generation and optimized encryption in cloud environment. *Multimedia Tools Appl.* 83 (16), 47269–47293. doi:10.1007/s11042-023-17365-8

Qiqieh, I., Alzubi, J., and Alzubi, O. (2024). DNA cryptography based security framework for health-cloud data. *Computing* 107 (1), 35. doi:10.1007/s00607-024-01393-9

Raina, P., and Kaushal, S. (2019). "A framework for security management in cloud based on quantum cryptography," in *Advances in intelligent systems and computing*, 295–306. doi:10.1007/978-981-13-5934-7_27

Rashid, A., and Chaturvedi, A. (2019). Cloud computing characteristics and services a brief review. *Int. J. Comput. Sci. Eng.* 7 (2), 421–426. doi:10.26438/ijcse/v7i2.421426

Saini, T., and Sinha, S. (2023). "Cloud computing security issues and challenges," in *Integration of cloud computing with emerging technologies*, 35–45. doi:10.1201/9781003341437-4

Shakor, M. Y., and Ibrahim Khaleel, M. (2025). Modern deep learning techniques for big medical data processing in cloud. *IEEE Access* 13, 62005–62028. doi:10.1109/access.2025.3556327

Shakor, M. Y., and Khaleel, M. I. (2024). Recent advances in big medical image data analysis through deep learning and cloud computing. *Electronics* 13 (24), 4860. doi:10.3390/electronics13244860

Shrivastava, P., Alam, B., and Alam, M. (2022). Security enhancement using blockchain based modified infinite chaotic elliptic cryptography in cloud. *Clust. Comput.* 26 (6), 3673–3688. doi:10.1007/s10586-022-03777-y

Yakubu, I. Z., and Murali, M. (2023). An efficient meta-heuristic resource allocation with load balancing in IOT-fog-cloud computing environment. *J. Ambient Intell. Humaniz. Comput.* 14 (3), 2981–2992. doi:10.1007/s12652-023-04544-6

Zahraddeen Yakubu, I., and Murali, M. (2024). An efficient IOT-fog-cloud resource allocation framework based on two-stage approach. *IEEE Access* 12, 75384–75395. doi:10.1109/access.2024.3405581