Augmenting Cloud Performance: An In-Depth Analysis of Algorithms for Data Security

Amandeep Kaur[1], Gagandeep Jagdev[*2]
[1]*Assistant Professor, Guru Gobind Singh Khalsa College, Bhagta Bhai Ka, Bathinda, Punjab, India*
[2*]*Technical Officer, Punjabi University Guru Kashi Campus, Talwandi Sabo, Punjab, India*

Abstract- Cloud computing has emerged as a revolutionary paradigm in the field of information technology, offering scalable, flexible, and cost-effective solutions for data storage and processing. This research paper explores the fundamental concepts, deployment models, and service models of cloud computing, highlighting its transformative impact on businesses and individual users. The paper delves into the technical architecture of cloud services, examining key characteristics such as on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. Furthermore, it investigates the benefits of cloud computing, including enhanced scalability, operational efficiency, disaster recovery capabilities, and collaborative potential. However, the paper also addresses significant challenges such as data security, privacy concerns, regulatory compliance, potential downtime, and vendor lock-in. Through a comprehensive analysis of current literature and case studies, this research aims to provide a holistic understanding of cloud computing, offering insights into its implementation and best practices for leveraging its advantages while mitigating associated risks. The findings of this study are intended to guide organizations in making informed decisions about adopting and optimizing cloud computing solutions to achieve their strategic objectives.

Keywords: AES, Cloud computing, ECC, Homomorphic encryption, RSA.

I. INTRODUCTION

With the revolutionary technology architecture known as cloud computing, users can access a shared pool of reconfigurable computing resources—including servers, storage, databases, networking, software, and more—anytime, anywhere. These resources offer flexible, scalable, and affordable IT solutions since they can be quickly provisioned and released with little management work or service provider contact. Cloud computing is a transformative technology model that enables on-demand access to a shared pool of configurable computing resources such as servers, storage, databases, networking, and software. These resources can be rapidly provisioned and released with minimal management effort, providing flexible, scalable, and cost-effective IT solutions [1]. Key characteristics of cloud computing include on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. Cloud services are typically offered in three forms: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Deployment models include public, private, hybrid, and community clouds, each catering to different business needs and regulatory requirements. The benefits of cloud computing are significant, including cost efficiency, scalability, improved disaster recovery, enhanced collaboration, and automatic updates. However, challenges such as security, compliance, downtime, vendor lock-in, and cost management need to be carefully addressed. Overall, cloud computing has revolutionized the way organizations manage and deploy IT resources, enabling them to innovate and respond quickly to changing demands [2].

Key Characteristics of Cloud Computing

- Instantaneous Self-Service

Users don't need to communicate directly with each service provider to provision computer resources as needed automatically.

- Extensive Network Access:

Using conventional procedures, resources can be accessed by a variety of client platforms, including laptops, tablets, mobile phones, and workstations.

- Pooling of Resources

Using a multi-tenant model, the provider pools its computer resources to serve several customers,

dynamically allocating and reassigning various physical and virtual resources in accordance with demand.

➢ Rapid Elasticity:

Resources can be quickly scaled both inward and outward in accordance with demand by being elastically provided and released, sometimes automatically.

➢ Measured Services:

By utilizing a metering capability at an abstraction level suitable for the type of service (e.g., storage, processing, bandwidth), cloud systems automatically regulate and optimize resource utilization.

Types of Cloud Services

➢ Infrastructure as a Service (IaaS) :

- The internet offers virtualized computer resources. AWS EC2, Microsoft Azure virtual machines, and Google Compute Engine are a few examples.
- Users own control over the operating systems, storage, and installed apps on their virtual machines, as well as the networks and storage they rent [3].

➢ Platform as a Service (PaaS):

- This online resource provides hardware and software tools, usually for application development. AWS Elastic Beanstalk, Google App Engine, and Microsoft Azure App Services are a few examples.
- Users don't need to worry about the underlying infrastructure to create, launch, and maintain apps.

➢ Software as a Service (SaaS):

- Provides software programs via the internet in exchange for a subscription. Salesforce, Microsoft 365, and Google Workspace are a few examples.
- Applications can be accessed directly by users through their web browsers, negating the need for software management, upkeep, and installation.

Models of Deployment

➢ Public Cloud:

- Anybody who wants to acquire services can access them through the public internet. AWS, Microsoft Azure, and Google Cloud Platform are a few examples.
- Scalability, cost-effectiveness, and a decrease in the user's requirement for infrastructure administration are among the advantages.

➢ Private Cloud:

- A single organization's services are kept up to date on a private network. IBM's private cloud services, OpenStack, and VMware are a few examples.
- It is ideal for companies with strict regulatory and security needs because of its benefits, which include increased security, control, and customization [4].

➢ Hybrid Cloud

- A hybrid cloud combines the capabilities of public and private clouds to enable the sharing of apps and data between them. More deployment options and increased flexibility are offered by this paradigm.
- Advantages include increased security and compliance, optimized costs, and better scalability.

➢ Community Cloud

- Shared infrastructure for a certain group of businesses with related issues (security, compliance, jurisdiction, etc.) is known as a community cloud. Governmental organizations sharing resources is one example.
- Shared expenses, enhanced cooperation, and adherence to industry standards are among the advantages.

Benefits of Cloud Computing

➢ Cost Efficiency: Lowers the initial outlay for purchasing software and hardware as well as for establishing and maintaining on-site data centers.

➢ Scalability and Flexibility: Both scalability and flexibility allow for the efficient management of changing workloads by allowing resources to be scaled up or down in response to demand. calamity

➢ Recovery and Business Continuity: These strategies guarantee that data is backed up and recoverable, providing increased resilience and allowing firms to carry on with operations even in the event of a calamity.

➢ Collaboration and Accessibility: This enables several users from various places to access and

work on data simultaneously, which promotes teamwork [5].

- Automatic Updates: To guarantee optimal performance and security, service providers update systems regularly with the newest security patches and technologies.

Challenges and Considerations

- Security and Privacy: Preserving data security and privacy is crucial, especially when handling and storing sensitive data in the cloud.
- Compliance: Businesses need to make sure that the cloud service providers they use abide by all applicable laws and industry standards.
- Reliability and Downtime: Reliance on internet connectivity may lead to outages that impede the use of cloud services.
- Vendor lock-in - The inability to easily transfer data and apps across cloud service providers can reduce flexibility and heighten reliance on one particular provider.
- Cost management: Although cloud computing might save costs, squandering cloud resources can result in unforeseen costs.

Cloud computing, which offers unmatched flexibility, scalability, and cost reductions, has completely changed how businesses manage and use IT resources. Businesses can take advantage of the newest technology developments while concentrating on their core skills by utilizing cloud services. To fully reap the rewards of cloud computing, however, careful consideration of security, compliance, and cost management is necessary.

II. ALGORITHMS OF CLOUD COMPUTING

Because they guarantee high performance, data security, energy efficiency, and the best possible use of resources, algorithms are essential to the smooth running of cloud computing environments. Cloud service providers may address the varied needs of their clientele by providing dependable, scalable, and resilient services by utilizing a blend of these algorithms. Ongoing research and development of sophisticated algorithms will be essential in solving new issues and expanding the capabilities of cloud infrastructures as cloud computing continues to develop [6].

1. *Resource Allocation Algorithms*

- First-Come, First-Served (FCFS):
  - Resource allocation for FCFS jobs is based on arrival order.
  - FCFS is a basic algorithm and may result in inefficiencies.
- Round Robin (RR):
  - Distributes resources to tasks in a cyclical manner to ensure equitable allocation.
  - Ideal for settings with comparable job specifications.
- Priority-Based:
  - Assigns resources to activities according to their importance, giving priority to the most important tasks.
- Genetic Algorithms:
  - By imitating natural selection processes, these algorithms use evolutionary techniques to determine the best way to allocate resources.

2. *Load Balancing Algorithms*

- Randomized: This is a basic but inefficient method of distributing tasks among servers at random.
- Round Robin: This cyclical job distribution method works well for comparable workloads.
- Balanced Round Robin: Distributes duties among servers by allocating weights corresponding to their capacities.
- Least Connections: This dynamic load-balancing technique assigns jobs to the server with the fewest active connections.
- Throttled: Prevents overload by limiting the amount of work allotted to each server.

3. *Data Security Algorithms*

- Advanced Encryption Algorithm (AES): Data security during transit and at rest is ensured by the AES, a symmetric encryption method.
- Rivest, Shamir, Adleman (RSA): The RSA Algorithm is an asymmetric encryption technique that ensures secure data transfer.
- Elliptic Curve Cryptography (ECC): Suitable for cloud environments, it offers robust security with reduced key lengths.
- Homomorphic encryption: It improves privacy by enabling computation on encrypted material without first decrypting it.

4. *Fault Tolerance Algorithms*
- Checkpointing: Allows a process to recover from errors by periodically saving its state. Replication: To guarantee availability in the event of a failure, several copies of the data are created across many nodes.
- Load Rebalancing: To maintain system performance, load balancing redistributes tasks from malfunctioning nodes to functioning ones.
- Heartbeat Mechanism: Consistently monitors node health and quickly identifies malfunctions.

5. *Energy Efficiency Algorithms*
- Dynamic Voltage and Frequency Scaling (DVFS): Lowers energy usage by adjusting CPU voltage and frequency in response to workload.
- Server Consolidation: Reduces the number of servers needed to handle tasks, enabling inactive servers to be turned off.
- Green scheduling: Optimizes for less energy use by scheduling jobs based on trends of energy consumption.
- Thermally-Aware Scheduling: This technique divides up work according to server temperature to avoid overheating and lower cooling expenses.

6. *Scheduling Algorithms*
- Min-Min Algorithm: The Min-Min Algorithm gives the work to the resource that can finish it the quickest after choosing the task with the shortest completion time.
- Max-Min Algorithm: The Max-Min Algorithm designates the work to the quickest resource available based on its maximum completion time.
- Metaheuristic Algorithms: These comprise methods for solving difficult scheduling problems such as Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO).
- Heuristic Algorithms: These algorithms balance computational efficiency and accuracy to provide fast, workable solutions for task scheduling.

7. *Data Management Algorithms*
- MapReduce: Processes large datasets in parallel across a distributed cluster, breaking down tasks into map and reduce functions.
- Hadoop Distributed File System (HDFS): Ensures reliable storage and quick access to large datasets by distributing data across multiple nodes.
- NoSQL Databases: Utilize algorithms for efficient data retrieval and storage in non-relational databases, enhancing scalability.

III. DATA SECURITY ALGORITHMS

Advanced Encryption Standards

Digital data is protected by the commonly used symmetric encryption method known as the Advanced Encryption Standard (AES). Depending on the required level of security, AES allows key sizes of 128, 192, or 256 bits and operates on fixed block sizes of 128 bits. There are different numbers of transformation rounds involved in the encryption process: 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys [7]. Every round has four primary steps: SubBytes modifies each byte of the data block using a substitution table (S-box); ShiftRows cyclically shifts the state array's rows; MixColumns modifies the data within each column (skipped in the last round); and AddRoundKey XORs the data block with a round key that is generated from the original key via a procedure known as key expansion. Together, these cycles improve data security by forming intricate patterns that are challenging for unauthorized users to understand. Due to its reputation for speed and security, AES is the industry standard for encrypting sensitive data in a variety of applications, such as data storage, communications, and financial transactions. The AES encryption process involves a series of well-defined steps that transform plaintext into ciphertext through key-dependent transformations [8]. By performing multiple rounds of SubBytes, ShiftRows, MixColumns, and AddRoundKey operations, AES ensures robust encryption and security. The flowchart depicting the working of AES is shown in Fig. 1 followed by the algorithm.

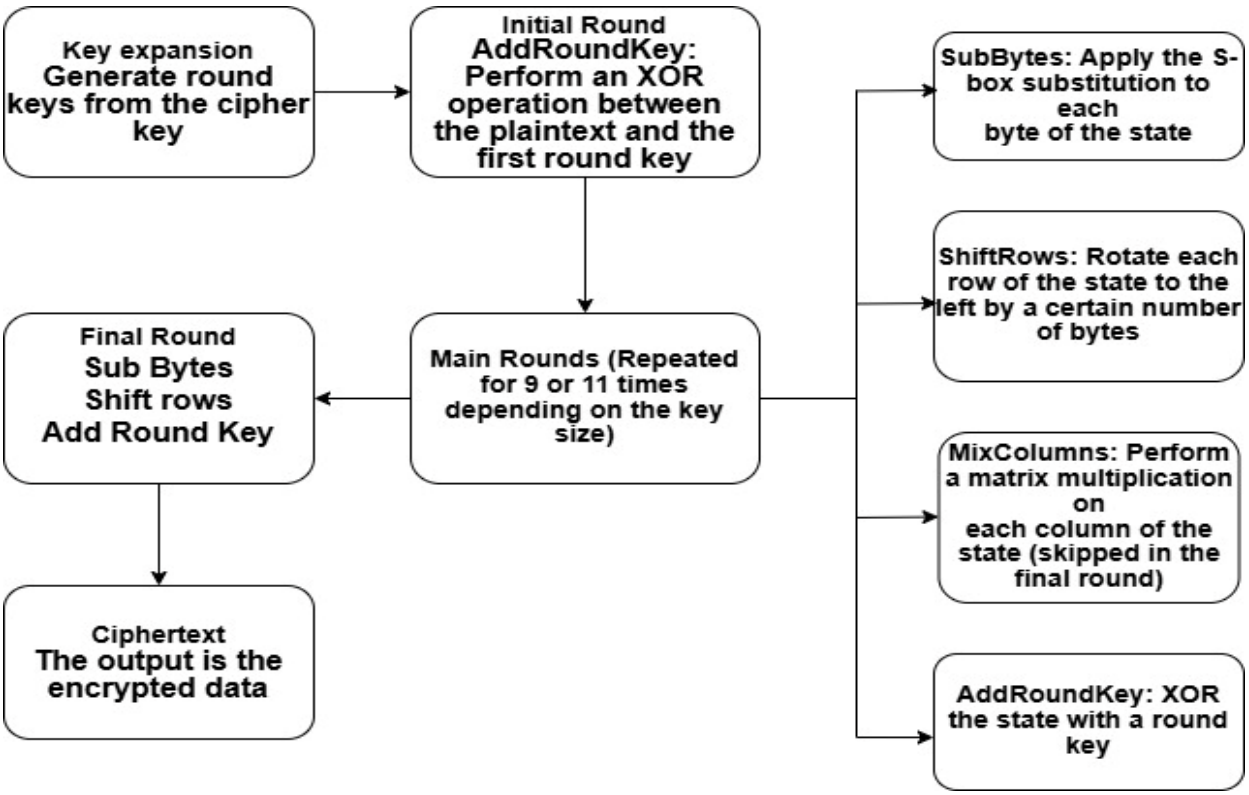

Fig. 1 Working of AES

Algorithm

1. Key Expansion
- Generate Round Keys from the Cipher Key: Expand the initial cipher key into a series of round keys used in each round of encryption.

2. Initial Round
- AddRoundKey: XOR the plaintext block with the first round key to create the initial state.

3. Main Rounds (9, 11, 13 rounds depending on key size)
- SubBytes: Substitute each byte in the state with its corresponding byte from the S-box.
- ShiftRows: Shift the rows of the state array cyclically to the left.
- MixColumns: Mix the data within each column of the state by performing matrix multiplication (skipped in the final round).
- AddRoundKey: XOR the state with the current round key.

4. Final Round
- SubBytes: Substitute bytes using the S-box.
- ShiftRows: Shift rows cyclically.
- AddRoundKey: XOR with the final round key (no MixColumns in this round).

5. Ciphertext
- Output Encrypted Data: The final state after the last round is the ciphertext, the encrypted version of the plaintext.

Rivest, Shamir, Adleman (RSA)

The RSA algorithm is a widely used asymmetric cryptographic technique that ensures secure data transmission through the use of two keys: a public key for encryption and a private key for decryption. The algorithm begins with key generation, where two large prime numbers, (p) and (q), are selected and multiplied to produce (n), the modulus for both keys. The totient function *(phi(n) = (p-1)(q-1))* is calculated, and an integer (e) is chosen such that *1 < (e) < (phi(n))* and (e) is coprime with (phi(n)). The public key is formed by the pair (e, n). The private key (d) is then determined as the modular multiplicative inverse of (e) modulo (phi(n)). To encrypt a message, the sender converts the plaintext message (M) into an integer (m) within the range of (0<m< n). The ciphertext (c) is computed using the formula *(c=m^e mod n)* and is sent to the recipient. Upon receiving the ciphertext, the recipient decrypts it using their private key (d) with the formula *(m=c^d mod n)*. Finally, (m) is converted back to the original plaintext message (M). The security of

RSA relies on the practical difficulty of factoring the large number (n) into its prime factors (p) and (q), which makes the decryption infeasible without the private key [9]. The RSA algorithm ensures secure data transmission through asymmetric encryption and decryption processes, relying on the computational difficulty of factoring large prime numbers. This flowchart shown in Fig. 2 followed by the algorithm encapsulates the key steps involved, from key generation to encryption and decryption, highlighting the simplicity and robustness of the RSA encryption method.

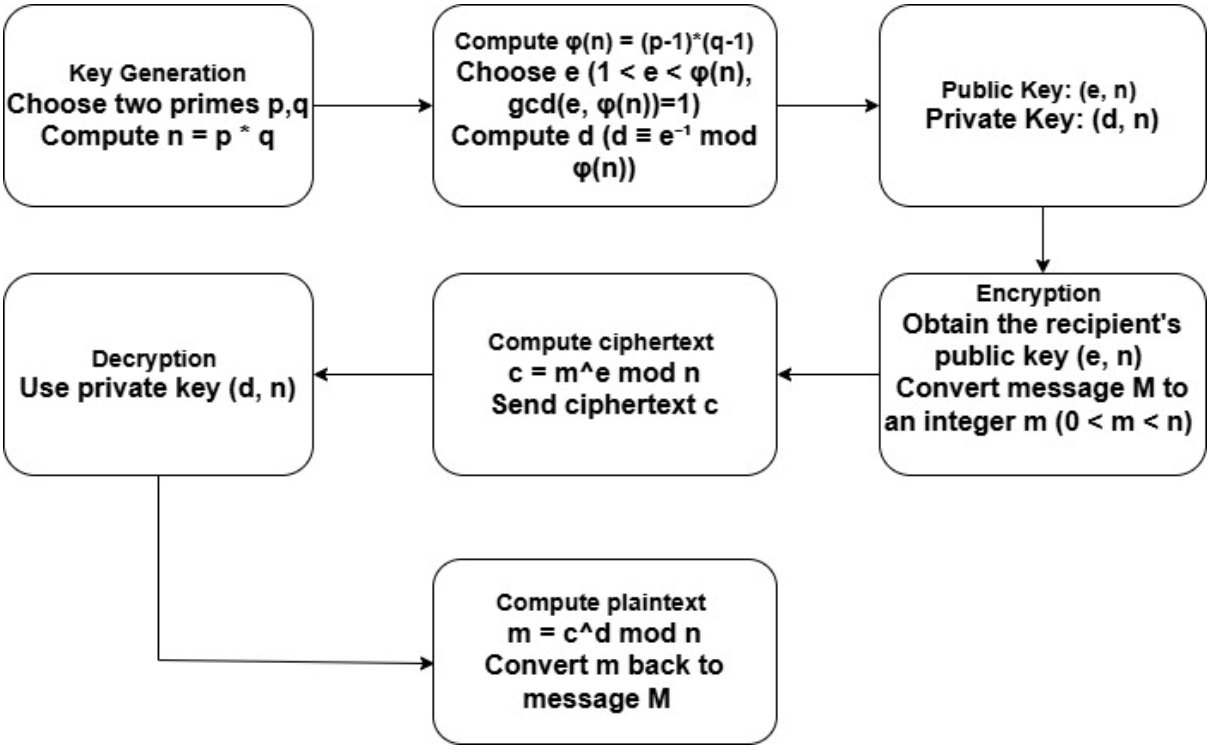

Fig. 2 Working of RSA algorithm

Algorithm

1. Key Generation

- Choose Two Primes p and q**: Select two distinct large prime numbers.
- Compute n = p * q**: Multiply the primes to get n, used as the modulus for both public and private keys.
- Compute φ(n) = (p-1) * (q-1)**: Calculate the totient function φ(n), which is used in the key generation process.
- Choose e (1 < e < φ(n), gcd(e, φ(n))=1)**: Select a public exponent e such that it is coprime with φ(n).
- Compute d (d ≡ e^{-1} mod φ(n)): Calculate the private exponent d, which is the modular multiplicative inverse of e modulo φ(n).
- Public Key (e, n): The public key consists of the exponent e and the modulus n.
- Private Key (d, n): The private key consists of the exponent d and the modulus n.

2. Encryption

- Obtain Recipient's Public Key (e, n): Retrieve the public key of the recipient.
- Convert Message M to Integer m (0 < m < n): Convert the plaintext message M into an integer m within the range of 0 to n-1.
- Compute Ciphertext c = m^e mod n: Encrypt the message by raising m to the power of e and taking the modulus n.
- Send Ciphertext c: Transmit the ciphertext to the recipient.

3. Decryption

- Use Private Key (d, n): Use the recipient's private key for decryption.
- Compute Plaintext m = c^d mod n: Decrypt the ciphertext by raising c to the power of d and taking the modulus n.

- Convert m back to Message M: Convert the integer m back to the original plaintext message M.

Elliptic Curve Cryptography

Using the mathematical characteristics of elliptic curves over finite fields, Elliptic Curve Cryptography (ECC) is a contemporary public-key encryption method that offers strong security with comparatively small key sizes. ECC is predicated on the challenging task of calculating discrete logarithms of elliptic curve points, in contrast to conventional techniques like RSA, which depend on the difficulty of factoring huge integers. This method increases the efficiency of ECC's use of memory and processing power by allowing it to offer comparable levels of security with significantly lower key lengths [10]. Therefore, situations with limited resources, such as those seen in mobile devices, Internet of Things applications, and other systems where security and performance are crucial, are especially well-suited for ECC. ECC is essential to contemporary cryptographic protocols because it offers robust encryption, digital signatures, and key exchange methods that guarantee data integrity, confidentiality, and authenticity in a range of digital communications. Elliptic Curve Cryptography (ECC) is a strong and effective cryptographic technique that offers a high level of security with comparatively small key sizes [11]. The flowchart in Fig. 3 followed by the algorithm highlights how the technique uses finite field arithmetic and elliptic curve points to secure data. It summarizes the important processes in the ECC algorithm, from key generation to encryption and decryption [12].

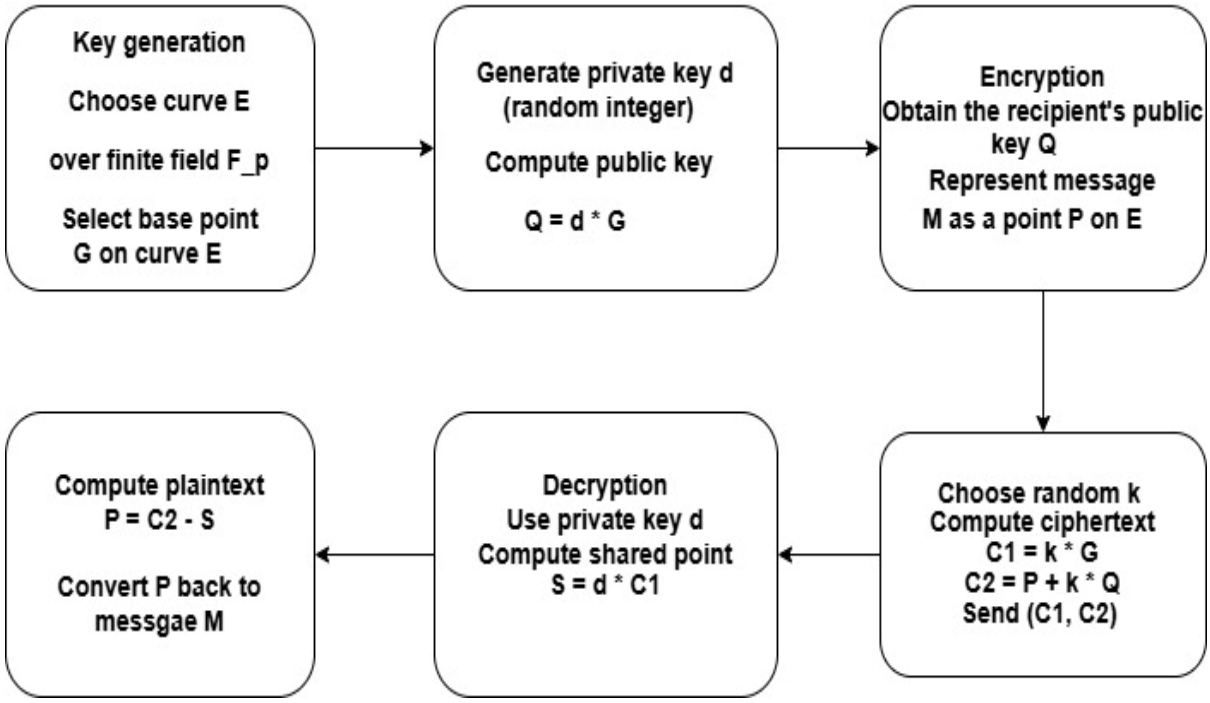

Fig. 3 Working of Elliptic Curve Cryptography

Algorithm

1. Key Generation

- Choose Curve E over Finite Field F_p: Select an elliptic curve E defined over a finite field F_p.
- Select Base Point G on Curve E: Choose a point G on the elliptic curve that will be used as a generator.
- Generate Private Key d (Random Integer): Select a random integer d, which serves as the private key.
- Compute Public Key Q = d * G: Calculate the public key Q by multiplying the base point G by the private key d.

2. Encryption

- Obtain Recipient's Public Key Q: Retrieve the recipient's public key.
- Represent Message M as a Point P on E: Convert the plaintext message M into a point P on the elliptic curve E.
- Choose Random k: Select a random integer k.

- Compute Ciphertext C1 = k * G and C2 = P + k * Q: Calculate the ciphertext components C1 and C2. C1 is the product of k and the base point G, while C2 is the sum of P and the product of k and the recipient's public key Q.
- Send (C1, C2): Transmit the pair (C1, C2) as the encrypted message.

3. Decryption

- Use Private Key d: Use the recipient's private key for decryption.
- Compute Shared Point S = d * C1: Calculate the shared point S by multiplying C1 by the private key d.
- Compute Plaintext P = C2 - S: Retrieve the original point P by subtracting the shared point S from C2.
- Convert P back to Message M: Convert the point P back to the original plaintext message M.

Homomorphic encryption

A strong cryptographic method called homomorphic encryption enables calculations to be done directly on encrypted data, producing results that, when decoded, match those of operations carried out on the plaintext. This feature is especially helpful in situations where data security and privacy are critical, such as cloud computing and privacy-preserving data analysis. A public key for encryption and a private key for decryption are the two keys that are generated during key generation in a standard homomorphic encryption system [13]. The public key is used to encrypt plaintext data, creating ciphertext. This ciphertext can then be used to perform desired computations, like addition and multiplication, without the requirement to first decrypt it. These processes produce another ciphertext, which the private key is used to decrypt to reveal the final plaintext result. Critical issues in data security and privacy are addressed by homomorphic encryption, which permits safe and private data processing and guarantees that sensitive information is safeguarded even during computation [14]. Homomorphic encryption ensures privacy and security even during processing by enabling safe calculations on encrypted data. The flowchart in Fig. 4 followed by the algorithm highlights how this encryption technique allows meaningful computations without exposing sensitive data by illuminating the major steps required, from key generation through encryption and homomorphic operations to decryption [15].

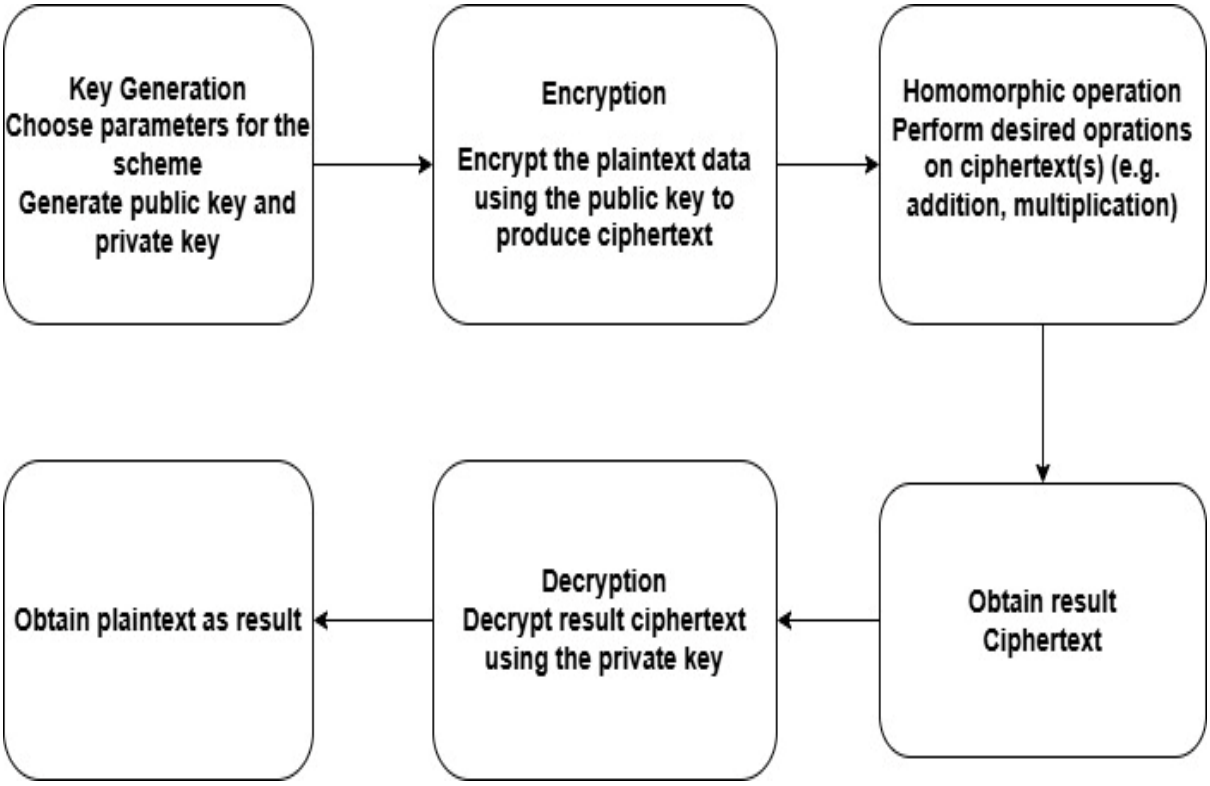

Fig. 4 Working of Homomorphic Encryption

Algorithm

1. Key Generation

- Choose Parameters for the Scheme: Select the necessary parameters that define the specific homomorphic encryption scheme.
- Generate Public Key and Private Key: Use the chosen parameters to generate a pair of keys: a public key for encryption and a private key for decryption.

2. Encryption

- Encrypt Plaintext Data Using the Public Key to Produce Ciphertext**: Convert the plaintext data into ciphertext using the public key. This ensures the data is securely encrypted and can only be decrypted with the private key.

3. Homomorphic Operation

- Perform Desired Operation on Ciphertext(s)**: Execute the required computational operation directly on the encrypted data. This can include mathematical operations like addition or multiplication, depending on the homomorphic encryption scheme.
- Obtain Result Ciphertext: The result of the operation on the ciphertext(s) is another ciphertext, which represents the encrypted form of the operation's outcome.

4. Decryption

- Decrypt Result Ciphertext Using the Private Key**: Use the private key to decrypt the resulting ciphertext, converting it back to plaintext.
- Obtain Plaintext Result: The decrypted result corresponds to the operation performed on the original plaintext data.

IV. CONCLUSION

To sum up, the examination and evaluation of cloud computing algorithms demonstrate their vital function in enhancing the efficiency, expandability, and safety of cloud services. These algorithms are crucial for handling the complexity and needs of contemporary applications as cloud computing develops further. The study covers several different algorithm categories, such as load balancing, data security, fault tolerance, and resource allocation. Every area tackles distinct obstacles related to cloud settings, ranging from effectively allocating resources to guaranteeing strong security protocols and upholding optimal availability. The broad use of cloud computing may be attributed in large part to the developments in these algorithms, which allow enterprises to take advantage of flexible and scalable infrastructure without sacrificing security or performance.

To keep up with the exponential growth of data and the ever-increasing complexity of cloud settings, future research should concentrate on improving the efficacy and efficiency of these algorithms. Advancements in edge computing, quantum computing, and blockchain technology offer fresh chances to create sophisticated algorithms that enhance the functionality and reliability of cloud computing. Overall, cloud computing algorithms' continuous development and improvement will play a critical role in determining how cloud services will develop in the future and how reliable, effective, and safe they will be able to fulfill the changing needs of users all over the world.

V. REFERENCES

[1]. Lim, S.Y.; Kiah, M.M.; Ang, T.F. Security Issues and Future Challenges of Cloud Service Authentication. Polytech. Hung. 2017, 14, 69–89.

[2]. Borylo, P.; Tornatore, M.; Jaglarz, P.; Shahriar, N.; Cholda, P.; Boutaba, R. Latency and energy-aware provisioning of network slices in cloud networks. Comput. Commun. 2020, 157, 1–19. [CrossRef]

[3]. Carmo, M.; Dantas Silva, F.S.; Neto, A.V.; Corujo, D.; Aguiar, R. Network-Cloud Slicing Definitions for Wi-Fi Sharing Systems to Enhance 5G Ultra-Dense Network Capabilities. Wirel. Commun. Mob. Comput. 2019, 2019, 1–17. [CrossRef]

[4]. Dang, L.M.; Piran, M.; Han, D.; Min, K.; Moon, H. A Survey on Internet of Things and Cloud Computing for healthcare. Electronics 2019, 8, 768. [CrossRef]

[5]. Srinivasamurthy, S.; Liu, D. Survey on Cloud Computing Security. 2020. Available online: https://www. semanticscholar.org/ (accessed on 19 July 2020).

[6]. Mathkunti, N. Cloud Computing: Security Issues. Int. J. Comput. Commun. Eng. 2014, 3, 259–263

[7]. Stefan, H.; Liakat, M. Cloud Computing Security Threats And Solutions. J. Cloud Comput. 2015, 4, 1. [CrossRef] Electronics 2020, 9, 1379 22 of 25
[8]. Fauzi, C.; Azila, A.; Noraziah, A.; Tutut, H.; Noriyani, Z. On Cloud Computing Security Issues. Intell. Inf. Database Syst. Lect. Notes Comput. Sci. 2012, 7197, 560–569.
[9]. Palumbo, F.; Aceto, G.; Botta, A.; Ciuonzo, D.; Persico, V.; Pescapé, A. Characterizing Cloud-to-user Latency as perceived by AWS and Azure Users spread over the Globe. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Taipei, Taiwan, 7–11 December 2019; pp. 1–6.
[10]. Hussein, N.H.; Khalid, A. A survey of Cloud Computing Security challenges and solutions. Int. J. Comput. Sci. Inf. Secur. 2017, 1, 52–56.
[11]. Le Duc, T.; Leiva, R.G.; Casari, P.; Östberg, P.O. Machine Learning Methods for Reliable Resource Provisioning in Edge-Cloud Computing: A Survey. ACM Comput. Surv. 2019, 52, 1–39. [CrossRef]
[12]. Li, K.; Gibson, C.; Ho, D.; Zhou, Q.; Kim, J.; Buhisi, O.; Gerber, M. Assessment of machine learning algorithms in cloud computing frameworks. In Proceedings of the IEEE Systems and Information Engineering Design Symposium, Charlottesville, VA, USA, 26 April 2013; pp. 98–103.
[13]. Callara, M.; Wira, P. User Behavior Analysis with Machine Learning Techniques in Cloud Computing Architectures. In Proceedings of the 2018 International Conference on Applied Smart Systems, Médéa, Algeria, 24–25 November 2018; pp. 1–6.
[14]. Singh, S.; Jeong, Y.-S.; Park, J. A Survey on Cloud Computing Security: Issues, Threats, and Solutions. J. Netw. Comput. Appl. 2016, 75, 200–222.
[15]. Khan, A.N.; Fan, M.Y.; Malik, A.; Memon, R.A. Learning from Privacy Preserved Encrypted Data on Cloud Through Supervised and Unsupervised Machine Learning. In Proceedings of the International Conference on Computing, Mathematics and Engineering Technologies, Sindh, Pakistan, 29–30 January 2019; pp. 1–5.

About the Authors

Dr. Amandeep Kaur is working in the capacity of Assistant Professor at Guru Gobind Singh Khalsa College, Bhagta Bhai Ka, Bathinda, Punjab since 2019. Earlier she served as an Assistant Professor at Sant Baba Bhag Singh Memorial College, Sukhanand, Punjab. She has a vast teaching experience to her credit. Her areas of research interest are Big Data, Data Mining, Cloud Computing, and Neural Networks.

Dr. Gagandeep Jagdev is currently serving in the capacity of Technical Officer at Punjabi University Guru Kashi Campus, Damdama Sahib (PB). His total teaching and research experience is more than 13 years and has more than 125 International and National publications in reputed journals and conferences to his credit. He is also a member of the editorial board of several reputed International Journals indexed in ESCI, Scopus, ACM, WoS, and Pubmeds and has been an active Technical Program Committee member of several International and National conferences conducted by renowned universities and academic institutions. He has been bestowed with Best Research Paper awards multiple times by reputed Government Institutions. He has actively participated in more than 100 Webinars and FDPs. His field of expertise is Image Processing, Big Data Analytics, Data Science, Cloud Computing, Cloud Security, Cryptography, and WANETs.