Journal of Theoretical and Applied Information Technology
30th September 2025. Vol.103. No.18
© Little Lion Scientific

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

ENHANCING CLOUD DATA SECURITY WITH POLYNOMIAL ELLIPTIC CURVE ZERO KNOWLEDGE PROOF (POLYECC-ZKP)

[1]E.JANSIRANI, [2]DR.N.KOWSALYA,

[1]Research Scholar, PG and Research Department of Computer Science, Sri Vijay Vidyalaya College of Arts & Science(Affiliated to Periyar University),
Dharmapuri, Tamilnadu, India.

[2]Assistant Professor, PG and Research Department of Computer Science, Sri Vijay Vidyalaya College of Arts & Science (Affiliated to Periyar University), Dharmapuri, Tamilnadu, India.

E-mail: [1]e.jansirani2020@gmail.com, [2]kowsisara2003@gmail.com

ABSTRACT

Cloud computing has revolutionized data storage and access, but it remains vulnerable to various security threats. Cryptographic approaches like Zero Knowledge Proof (ZKP) and Elliptic Curve Cryptography (ECC) have been widely used to address these issues. In order to improve cloud data security, this study presents a novel Polynomial Elliptic Curve Zero Knowledge Proof (PolyECC-ZKP) algorithm. By including polynomial functions into the ECC architecture, the suggested technique provides secure data authentication and strong encryption. We present a thorough analysis of the PolyECC-ZKP algorithm and evaluate its performance in comparison to other methods that have already been developed, such as Lattice-Based Zero Knowledge Proof (LZKP), Multi-Party Computation and Zero Knowledge Proof (MPC-ZKP), Hybrid Elliptic Curve Cryptography and Zero Knowledge Proof (HECCZKP), Hybrid Zero Knowledge Proof with ECC and ECDSA (Hybrid ZKP with ECDSA), and ECC. Scalability, quantum resistance, computation overhead, and security are the basis for the comparison. According to experimental findings, PolyECC-ZKP improves cloud security while requiring little computing power and is resistant to both conventional and quantum attacks. The results demonstrate PolyECC-ZKP's ability to emerge as a formidable contender for safe cloud settings.

Keywords: *Cloud Data Security, Polynomial Elliptic Curve Cryptography , Zero Knowledge Proof,Hybrid Cryptography Quantum-Resistant Algorithms, Multi-Party Computation,Lattice-Based Cryptography*

1. INTRODUCTION

Cloud computing has emerged as a pivotal technology in the digital age, offering scalable and on-demand access to computing resources and data storage. The cloud environment is vulnerable to a number of security risks, such as data breaches, illegal access, and privacy violations, despite its many benefits [1]. The need for secure cryptographic methods has increased as more private information is processed and stored in the cloud. Because of its effectiveness and security in protecting cloud data, Elliptic Curve Cryptography (ECC) and Zero Knowledge Proof (ZKP) have attracted a lot of interest. While ZKP guarantees secure information verification without disclosing the data itself [3], ECC is renowned for offering strong security with lower key lengths, making it appropriate for resource-constrained cloud systems [2]. Despite its effectiveness, the present ECC and ZKP implementations are limited in their ability to withstand sophisticated attacks, particularly those posed by quantum computing. Furthermore, more complex solutions are needed to address the scalability and processing overhead problems in large cloud systems. This research proposes Polynomial Elliptic Curve Zero Knowledge Proof (PolyECC-ZKP), a novel method that integrates polynomial functions into the ECC framework to further improve cloud data security in order to overcome these issues.

The major aim for designing the PolyECC-ZKP algorithm is to overcome the inherent constraints of standard ECC and ZKP techniques while delivering a quantum-resistant solution.

Cryptographic solutions that provide robust security guarantees while maintaining low computational and transmission overhead are desperately needed as cloud systems grow and cyberattacks get more complex. Furthermore, despite their promise, new cryptographic paradigms like Multi-Party Computation ZKP (MPC-ZKP) and Hybrid Elliptic Curve Cryptography and Zero Knowledge Proof (HECCZKP) have not yet shown their best performance in extensive cloud systems. The proposed PolyECC-ZKP algorithm uses Zero Knowledge Proof to guarantee secure verification procedures without data exposure, while also integrating polynomial functions into the ECC architecture to strengthen its cryptography. This innovative combination is a powerful contender for contemporary cloud infrastructures since it seeks to improve security, scalability, and quantum resistance.

The key objectives of this research are to propose and develop the **Polynomial Elliptic Curve Zero Knowledge Proof (PolyECC-ZKP)** algorithm for cloud data security. To compare the PolyECC-ZKP algorithm with existing cryptographic methods such as **ECC**, **ECDSA**, **ZKP**, **HECCZKP**, **Hybrid ZKP with ECC and ECDSA**, **MPC-ZKP**, and **Lattice-Based Zero Knowledge Proof (LZKP)**. To evaluate the proposed algorithm based on security strength, computational efficiency, scalability, and quantum resistance.

This paper is structured as follows: A thorough literature overview of the cryptographic techniques pertinent to cloud security is provided in Section 2. The design and theoretical underpinnings of the suggested PolyECC-ZKP algorithm are covered in detail in Section 3. The PolyECC-ZKP algorithm is compared to various cryptographic methods in Section 4, and the experimental findings are described in Section 5. Section 6 wraps up the work and makes recommendations for further research in this field.

2. LITERATURE REVIEW

The algebraic structure of elliptic curves over finite fields serves as the foundation for Elliptic Curve Cryptography (ECC), a public-key cryptography technique. With significantly smaller key sizes, ECC provides the same level of security as more conventional techniques like RSA. This makes it especially helpful in settings like cloud computing where bandwidth and processing power are scarce. ECC maintains strong security while allowing for quicker calculation and less transmission cost because of its smaller key size. ECC is not intrinsically immune to quantum computing attacks, despite its benefits, which encourages the creation of more robust algorithms. X. Yuan and associates, 2023 [4]. A popular ECC-based digital signature system for protecting cloud transactions and guaranteeing data integrity and authenticity is the Elliptic Curve Digital Signature Algorithm (ECDSA). Large-scale cloud settings can benefit from ECDSA's efficiency in message verification and signing. ECDSA does, however, inherit some of ECC's drawbacks, such as processing inefficiencies under heavy data loads and susceptibility to future quantum assaults. Even if ECDSA increases the speed and key size of conventional signature systems like RSA, it is still insufficiently secure against changing cyberthreats. et al., Jayabhaskar M. (2012) [5]. With Zero Knowledge Proof (ZKP), one person (the prover) can demonstrate to another (the verifier) that a certain assertion is true without disclosing any further information beyond the statement's veracity. For authentication and verification procedures where sensitive data must be kept private, this cryptographic protocol is essential to cloud security. Applications for ZKP can be found in safe multi-party computations, blockchain, and cloud data verification. However, without additional optimization, the typical ZKP protocols become less feasible since they frequently incur computation and communication overhead, especially in big cloud systems (X. Zhang and C. Li et al., 2023) [6].

By combining the advantages of ECC and ZKP, Hybrid Elliptic Curve Cryptography and Zero Knowledge Proof (HECCZKP) provides secure cloud communication without sacrificing efficiency. The hybrid technique is perfect for situations that call for both encryption and secure validation since it adds the privacy-preserving verification mechanisms of ZKP to the security of ECC. Notwithstanding its benefits, Jansirani and Kowsalya et al. (2023) [7] have criticized HECCZKP for its computational complexity and the trade-offs between security and performance, particularly in distributed cloud systems. The Hybrid Zero Knowledge Proof approach, which combines the greatest features of ECC and ECDSA with ZKP, is an advancement in cryptographic approaches that improves digital signature security while guaranteeing zero-knowledge verification.

Cloud systems' data integrity and confidentiality are strengthened by this hybrid approach, especially in the areas of secure access control and authentication. Although the hybrid method increases security, it can also increase computational load, particularly in large-scale cloud systems. Kowsalya and Jansirani et al., 2024 [8].

A cryptographic system called Multi-Party Computation (MPC) enables several parties to collaboratively compute a function over their inputs while maintaining the privacy of those inputs. When paired with ZKP, MPC guarantees that parties can demonstrate the accuracy of their calculations without disclosing the actual data. In cloud computing, MPC-ZKP is especially helpful for safe data exchange and cooperative processing amongst several cloud clients. However, when scaling to a large number of parties, the protocol encounters scalability problems because of the high computational and communication needs. One of the most promising methods for attaining quantum-resistant security is lattice-based cryptography. In 2023, W. Zhou and W. Sun et al. [9].

Lattice-Based Zero Knowledge Proof (LZKP) combines lattice-based techniques, which are proven to be safe from quantum assaults, with ZKP protocols. For cloud data protection, LZKP provides an extremely safe foundation, particularly in the post-quantum era. Lattice-based techniques, although theoretically resilient, are frequently computationally costly and require optimization to be feasible in real-time cloud systems. Furthermore, the use of LZKP in cloud systems is still in its infancy, and further study is required to increase its scalability and effectiveness, according to T. Wang and L. Liu et al. (2023) [10].

While the cryptographic methods discussed above offer varying degrees of security and efficiency, several challenges remain:

- ECC and ECDSA provide efficient cryptographic solutions but lack resistance to quantum computing attacks [11].
- ZKP adds a layer of privacy-preserving verification but can introduce significant computational overhead [12] [13].
- Hybrid approaches like HECCZKP and Hybrid ZKP with ECC and ECDSA enhance security but can suffer from performance bottlenecks in large-scale environments.
- MPC-ZKP and LZKP offer strong security guarantees, especially against quantum threats, but their computational and communication complexity limits practical application in cloud systems[14] [15].

These research gaps motivate the development of the proposed **Polynomial Elliptic Curve Zero Knowledge Proof (PolyECC-ZKP)** algorithm, which aims to address the limitations of existing methods by integrating polynomial functions into ECC and optimizing the ZKP process for cloud environments.

3. PROPOSED METHODOLOGY: POLYECC-ZKP ALGORITHM

By fusing the advantages of Elliptic Curve Cryptography (ECC) and Zero Knowledge Proof (ZKP) with polynomial-based cryptographic functions, the Polynomial Elliptic Curve Zero Knowledge Proof (PolyECC-ZKP) technique aims to improve cloud data security. This algorithm's main concept is to use ZKP to provide private, verifiable, and secure cloud data authentication and interaction while introducing polynomial transformations into the ECC structure to further strengthen its security properties, particularly against quantum computing attacks [16] [17]. The PolyECC-ZKP algorithm's proposed goal is to:

- **Enhance security** by incorporating polynomial functions into ECC, increasing its resilience to both classical and quantum attacks.
- **Ensure privacy-preserving verification** using ZKP, allowing cloud users to prove ownership or knowledge of secret data without exposing the actual data.
- **Maintain efficiency** by optimizing both the polynomial-enhanced ECC and the ZKP protocols, ensuring scalability for large-scale cloud environments.

The key components of the PolyECC-ZKP algorithm are described as follows:

The **Elliptic Curve Cryptography (ECC)** framework is based on elliptic curves over finite fields, with the security deriving from the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP). In traditional ECC, a point on the curve is used as the basis for cryptographic operations, with each point represented as a pair of coordinates (x, y) that satisfy the elliptic curve equation:

$$y^2 = x^3 + ax + b(mod\ p)$$

Where a, b, and p are parameters that define the curve.

In **PolyECC-ZKP**, polynomial functions are introduced into the elliptic curve computations to enhance cryptographic strength. Specifically, the algorithm integrates a polynomial function, P(x), into the elliptic curve equation to achieve additional security layers. The modified elliptic curve equation becomes:

$$y^2 = x^3 + ax + b + P(x)(mod\ p)$$

Where, P(x) is a polynomial of degree *d* that introduces extra complexity to the curve. This polynomial function is selected based on predefined cryptographic properties, ensuring that it does not compromise the integrity of the elliptic curve while adding a new challenge to potential attackers attempting to break the system.

The introduction of polynomials serves two primary purposes:

- **Enhanced Resistance to Quantum Attacks**: The additional complexity introduced by P(x) makes it harder for quantum algorithms like Shor's algorithm to solve the discrete logarithm problem, thus providing quantum resistance.
- **Increased Computational Difficulty**: Even classical algorithms attempting to solve the modified ECDLP will face increased difficulty due to the polynomial's impact on the cryptographic structure.

To ensure the security of cloud interactions without revealing sensitive data, **Zero Knowledge Proof (ZKP)** is integrated into the PolyECC algorithm. In ZKP, a **prover** (e.g., a cloud user) can prove to a **verifier** (e.g., a cloud service provider) that they possess valid knowledge (e.g., a private key) without revealing the key or the data.

The ZKP protocol in **PolyECC-ZKP** works as follows:

1. **Setup**: The prover and verifier agree on a public elliptic curve and polynomial P(x), as well as any necessary public parameters for the proof.
2. **Commitment**: The prover commits to a random point on the modified elliptic curve, using their private key and the polynomial-enhanced elliptic curve equation to generate a public point Q.

$$Q = k.P_G + P(x)(mod\ p)$$

Where, k is the prover's private key, and P_G is a generator point on the elliptic curve.

3. **Challenge**: The verifier sends a random challenge c to the prover, requesting proof of the prover's knowledge of k.
4. **Response**: The prover responds by generating a proof based on their private key, the polynomial P(x), and the elliptic curve. The response includes a value r such that:

$$r = k + c.P(x)(mod\ p)$$

5. **Verification**: The verifier computes the expected value of Q using the provided r and challenge c, checking whether it matches the original commitment without learning the prover's private key. If the values match, the verifier is convinced of the prover's knowledge without any sensitive data being revealed.

This ZKP process ensures that cloud users can securely authenticate themselves and prove data ownership without exposing private keys or sensitive information. The use of polynomial-enhanced ECC ensures that the security remains robust even under adversarial conditions.

Key Steps in the PolyECC-ZKP Algorithm

Polynomial-Enhanced ECC

The core of the PolyECC-ZKP algorithm lies in modifying traditional ECC with polynomial transformations. This modification increases the cryptographic strength of ECC in two primary ways:

- **Polynomial Complexity**: The addition of a polynomial function P(x) increases the complexity of the curve, making the problem of reversing the ECC computations more difficult. Attackers would need to solve not only the ECDLP but also account for the polynomial modifications.

- **Quantum Resistance**: By introducing polynomial functions, PolyECC-ZKP provides increased security against quantum algorithms like Shor's algorithm, which is known to efficiently solve the discrete logarithm problem.

The steps for generating the polynomial-enhanced public and private keys are as follows:

1. **Private Key Generation**: The private key k is chosen as a random scalar.
2. **Polynomial Selection**: A polynomial function P(x) is selected, designed to enhance the security of the curve.
3. **Public Key Computation**: The public key Q is computed using the polynomial-modified elliptic curve equation:

$$Q = k.P_G + P(x)(mod\, p)$$

Where, P_G is a generator point on the elliptic curve.

Encryption and Decryption in PolyECC-ZKP

The encryption and decryption processes in PolyECC-ZKP are similar to those in traditional ECC but incorporate the polynomial-enhanced curve for added security.

Encryption:

1. **Message Encoding**: The plaintext message M is encoded as a point on the elliptic curve.
2. **Random Key**: A random scalar r is chosen.
3. **Ciphertext Generation**: The ciphertext consists of two components:

$$C_1 = r.P_G$$

$$C_2 = M + r.Q + P(x)$$

Where, P(x) is the polynomial function, and Q is the recipient's public key.

4. **Transmission**: The ciphertext (C_1,C_2) is sent to the recipient.

Decryption:

1. **Private Key Use**: The recipient uses their private key k to compute the shared secret:

$$S = k.C_1 + P(x)$$

2. **Message Retrieval**: The recipient recovers the plaintext message by computing:

$$M = C_2 - S$$

This ensures secure message exchange between cloud users without exposing sensitive information.

Zero Knowledge Proof (ZKP) for Verification

The ZKP mechanism in PolyECC-ZKP allows users to prove knowledge of their private key kkk without revealing it. The ZKP protocol follows these steps:

Step 1: Commitment

- The prover (e.g., a cloud user) commits to a random point R on the elliptic curve by generating:

$$R = r.P_G + P(x)(mod\, p)$$

Where r is a randomly chosen scalar, and P(x) is the polynomial function.

- The commitment R is sent to the verifier (e.g., a cloud service provider).

Step 2: Challenge

- The verifier sends a random challenge c to the prover, asking for proof of their knowledge of the private key k.

Step 3: Response

- The prover responds by computing a value sss using their private key and the challenge:

$$s = r + c.k + P(x)$$

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

The response s is sent back to the verifier.

Step 4: Verification

- The verifier checks the validity of the response by calculating the expected commitment:

$$Q' = s.P_G - c.Q(mod\ p)$$

If Q′=R, the verifier is convinced that the prover knows the private key k without the prover having revealed it.

This process allows for secure authentication and verification in cloud systems, protecting the privacy of users' private keys.

PolyECC-ZKP algorithm

To improve cloud data security, the PolyECC-ZKP algorithm combines the benefits of Zero Knowledge Proof (ZKP), polynomial cryptography, and Elliptic Curve Cryptography (ECC). A detailed explanation of the algorithm, including key creation, encryption, decryption, and the ZKP protocol, is provided below. Without disclosing private information, the framework guarantees that users can safely encrypt data, retrieve it from the cloud, and utilize ZKP to confirm identities.

Step 1: Key Generation

1. **Private Key Generation**:
   - Select a random integer k, where 1≤ k ≤ n−1, and n is the order of the elliptic curve.
   - **Polynomial Construction**: Define a polynomial function P(x) of degree d with randomly chosen coefficients:

$$P(x) = c_0 + c_1x + c_2x^2 + \cdots + c_dx^d$$

   - **Private Key**: k, stored securely by the user.
2. **Public Key Calculation**:
   - Choose a base point P_G on the elliptic curve.
   - Compute the public key Q as: $Q=k{\cdot}P_G+P(x)$
   - Public key Q is shared with other users and the cloud server for encryption.

Step 2: Encryption

1. **Message Encoding**:
   - Convert the plaintext message M into a point $M_P(x_M,y_M)$ on the elliptic curve.
   - Modify the point using the polynomial function P(x) to create MP′, ensuring the point fits the ECC structure.
2. **Random Scalar Selection**:
   - Choose a random integer r where 1≤ r≤ n−1.
3. **Ciphertext Generation**:
   - Compute the first part of the ciphertext C_1 as: C_1=r·PG
   - Compute the second part of the ciphertext C_2 as: C_2=M’P+r·Q
   - The ciphertext consists of $C_1$and C_2.
4. **Ciphertext Transmission**:
   - The ciphertext C_1,C_2 is sent to the cloud server for secure storage.

Step 3: Decryption

1. **Shared Secret Computation**:
   - The receiver (or the same user retrieving data) receives C_1 and C_2 from the cloud.
   - Using the private key k, compute the shared secret S as: S=k·C_1
   - Add the polynomial P(x) to the shared secret to align it with the encryption process.
2. **Message Point Recovery**:
   - Compute the original point M’P as: M’P=C_2−S
   - Convert the elliptic curve point M’P back into the plaintext message M.

Step 4: Zero Knowledge Proof (ZKP) Protocol

This step ensures that a user can prove knowledge of their private key k without revealing it.

1. **Commitment Phase** (Prover's Step):
   - The prover selects a random integer r′ and computes the commitment point R: R=r′·PG+P(x)
   - The prover sends R to the verifier.
2. **Challenge Phase** (Verifier's Step):
   - The verifier sends a random challenge c to the prover.

3. **Response Phase** (Prover's Step):
   - The prover computes the response s as: s=r′+c·k
   - The prover sends s to the verifier.
4. **Verification Phase** (Verifier's Step):
   - The verifier computes the expected commitment R′:

     $$R' = s \cdot P_G - c \cdot Q$$

   - If R′=R, the verifier confirms that the prover knows the private key k, without the key being disclosed.

Step 5: Secure Data Retrieval (Optional)

- If the user retrieves the data from the cloud and needs to prove ownership of the data without revealing their private key, the Zero Knowledge Proof protocol (from Step 4) can be applied.

Three essential cryptographic techniques—Elliptic Curve Cryptography (ECC), polynomial encryption, and Zero Knowledge Proofs (ZKP)—are combined in the Polynomial Elliptic Curve Zero Knowledge Proof (PolyECC-ZKP) algorithm, a strong cryptographic strategy intended to improve cloud data security. The Elliptic Curve Discrete Logarithm Problem (ECDLP), in which a private key is used to generate a public key using point multiplication on an elliptic curve, is the basis for traditional ECC's system security. By adding a polynomial function to the public key generation procedure, PolyECC-ZKP goes beyond this. The public key Q = k. PG + P(x) is created by adding a polynomial P(x) with randomly selected coefficients to the elliptic curve point multiplication. This extra complexity makes the system more secure by guaranteeing that the data is safe even in the event that a portion of the cryptographic structure is breached. A random scalar is selected to create two parts of the ciphertext: one portion is created using the changed public key, while the other part is created using elliptic curve point multiplication. The encryption procedure begins with mapping a message onto a point on the elliptic curve. In decryption, a shared secret is calculated using the private key, and the original message is recovered by deducting this secret from the ciphertext. This method guarantees data security even in cloud environments.

By adding a layer of privacy-preserving authentication, Zero Knowledge Proof (ZKP) enables a user (prover) to demonstrate that they are in possession of the private key without disclosing it. This is accomplished by utilizing the polynomial and a random scalar to generate a commitment, then answering to the verifier's challenge with a calculated value that can be mathematically validated without disclosing the secret. Through the use of ZKP, PolyECC-ZKP guarantees that users can safely authenticate themselves in cloud services without jeopardizing their private cryptographic data. PolyECC-ZKP is an efficient and successful method for protecting cloud data and offering privacy-preserving authentication since it combines ECC, polynomial encryption, and ZKP.

Advantages of PolyECC-ZKP

- **Enhanced Security**: The combination of ECC and polynomial cryptography makes it more challenging for attackers to break the encryption. Even if ECC parameters are compromised, the polynomial adds an extra layer of security.
- **Efficient Key Sizes**: ECC is known for its smaller key sizes compared to RSA or other cryptosystems, while maintaining a high level of security. The use of polynomials does not significantly increase the computational cost.
- **Zero Knowledge Proof for Privacy**: ZKP enables users to authenticate themselves without revealing any sensitive information, making it ideal for secure cloud-based systems where privacy is a concern.
- **Cloud Data Protection**: PolyECC-ZKP ensures that data stored in the cloud remains secure, and users can prove their identity without revealing their private key, providing a robust mechanism for secure cloud storage and access.

The PolyECC-ZKP algorithm introduces multiple security benefits:

- **Quantum Resistance**: The inclusion of polynomials in ECC enhances resistance to quantum attacks, particularly against Shor’s algorithm, which targets the discrete logarithm problem.
- **Scalability**: Despite the added complexity from polynomial integration, PolyECC-ZKP maintains the efficiency of ECC, making it scalable for large cloud systems.

- **Data Confidentiality**: The ZKP protocol ensures that sensitive data (e.g., private keys) is never exposed during verification, enhancing privacy.
- **Efficient Computation**: The algorithm retains the lower computational overhead of ECC while adding minimal overhead through polynomial integration, ensuring practical use in cloud environments.

In the **PolyECC-ZKP** algorithm introduces polynomial-based modifications to ECC and leverages ZKP to enhance security for cloud data systems. It provides a secure and efficient means of encryption, decryption, and authentication, addressing the shortcomings of traditional cryptographic methods while preparing for future quantum threats.

4.EXPERIMENTAL RESULTS

The suggested PolyECC-ZKP algorithm was evaluated experimentally in a cloud-based environment to mimic actual cloud data security situations. Major cloud systems such as Microsoft Azure and Amazon Web Services (AWS) were used for the trials. EC2 {m5.xlarge` instances running Windows 10 with 16 GB RAM, 4 vCPUs, and 500 GB SSD storage comprised the test environment on AWS. A Virtual Private Cloud (VPC) with firewall settings and VPN access was used to safeguard the cloud network. The main functions of AWS were key generation, encryption, and data storage. Elastic Load Balancing (ELB) was utilized to manage traffic, and AWS S3 buckets were used to store the encrypted data. To take advantage of Microsoft Azure's computational power, Zero Knowledge Proof (ZKP) verification duties were handled there. To duplicate cloud-based security systems, both platforms were incorporated into the PolyECC-ZKP workflow. The performance of the PolyECC-ZKP algorithm was evaluated using the CloudBank dataset (10 GB), which consisted of simulated private financial transactions in CSV format. A number of tools and frameworks were used in the implementation of the encryption, decryption, and ZKP procedures. Core elliptic curve cryptography (ECC) operations were performed using PyCryptodome and Python's `cryptography` library, while Charm-Crypto made it easier to construct Zero Knowledge Proof (ZKP) protocols. The SymPy library handled polynomial manipulations in the PolyECC-ZKP algorithm. The AWS SDK (Boto3) and Microsoft Azure SDK for Python were used to orchestrate cloud management. These tools offered programmatic control over cloud services like AWS S3, Azure Blob Storage, and compute instances for ZKP verification.

There were several crucial milestones in the test process. Initially, the PolyECC-ZKP technique was used to produce the private and public keys. These keys were then used to encrypt the chosen datasets, and the encrypted data was safely kept in the cloud. Without disclosing the secret key, the accuracy of the decryption procedure was confirmed using the Zero Knowledge Proof (ZKP) authentication technique. Following the decryption of the data, performance parameters for various cloud settings and dataset sizes were noted, including execution time, computational complexity, and resource consumption. The performance of the PolyECC-ZKP algorithm in terms of security, efficiency, and scalability in actual cloud systems was accurately evaluated thanks to this experimental setup. To assess the PolyECC-ZKP algorithm's resilience to many types of attacks, such as man-in-the-middle attacks, attacks based on quantum computing, and other prevalent risks in cloud data security, a security analysis was carried out.

Man-in-the-middle (MITM) Attacks happen when a malevolent person eavesdrops on and maybe modifies two parties' communication. Zero Knowledge Proofs, which enable one side to demonstrate ownership of a secret (the private key) without disclosing it, make the PolyECC-ZKP algorithm intrinsically immune to such attacks. An attacker cannot obtain any useful information about the secret key or the data being communicated during ZKP verification, even if they manage to intercept the conversation. Furthermore, data is securely sent with little chance of manipulation or unlawful decryption thanks to the encryption system based on elliptic curve cryptography.

Resistance to Quantum Attacks : Because they rely on the difficulty of factoring huge integers or solving discrete logarithm issues, classical encryption algorithms like RSA and ECC are susceptible to quantum attacks as a result of the development of quantum computing. Despite being based on elliptic curve cryptography, PolyECC-ZKP adds levels of complexity that strengthen its resistance against quantum algorithms like Shor's algorithm. These layers include polynomial-based key generation and ZKP verification. In contrast to conventional ECC or ECDSA-based systems, the use of polynomials adds computational complexity that could help postpone the effects of quantum assaults. To completely defend against upcoming quantum threats, more research into quantum-resistant encryption is necessary.

Other Security Threats: The PolyECC-ZKP method offers robust defense against popular cryptographic threats such brute force attacks, chosen-ciphertext attacks, and replay attacks in addition to MITM and quantum assaults. When used in conjunction with elliptic curve encryption, the high entropy of polynomial-based key creation guarantees that brute force attempts to discover the private key remain computationally impossible. Furthermore, by prohibiting the attacker from deriving valuable information from intercepted ciphertext, Zero Knowledge Proofs offer strong defenses against chosen-ciphertext attacks [20].

Performance Evaluation

To evaluate the effectiveness and efficiency of the proposed PolyECC-ZKP algorithm, several key performance metrics were assessed, including throughput, encryption and decryption time, security level , computation cost, and communication overhead.

A). Throughput:

Throughput, in the context of cryptographic algorithms for cloud data security, is defined as the amount of data processed (encrypted, decrypted, and verified) by the algorithm per unit time. It is usually measured in megabytes per second (MB/s) **or** gigabytes per second (GB/s) and is a critical parameter to assess the efficiency of encryption schemes, especially when dealing with large-scale cloud data environments. The formula for throughput is typically expressed as:

$$\text{Throughput} = \frac{\text{Total Data Processed}}{\text{Time Taken}}$$

Where, **Total Data Processed** refers to the size of the data (in MB or GB) encrypted, decrypted, and verified. **Time Taken** is the total time required for encryption, decryption, and Zero Knowledge Proof (ZKP) verification (in seconds).

ECC and **ECDSA** performance degrades with increasing data sizes and complexity, especially when combined with signature generation in ECDSA. **ZKP** is a proof mechanism that, while secure, introduces overhead due to the need for generating, transmitting, and verifying cryptographic proofs. This results in a moderate throughput for pure ZKP systems. **MPC-ZKP** higher communication overhead and increased computational complexity, resulting in lower throughput compared to simpler algorithms like ECC or ECDSA. **LZKP** provides quantum-resistance but suffers from heavy computational and communication overhead due to the complexity of lattice-based cryptographic operations, leading to lower throughput [21] [22]. **HECC-ZKP** the added ZKP process reduces throughput compared to standard ECC. **Hybrid ZKP-ECDSA**, offering high security but at the expense of increased time for proof generation and verification, which affects throughput negatively, especially for large datasets.

PolyECC-ZKP achieves higher throughput than other ZKP-based approaches while maintaining a comparable level of security.

$$\text{Throughput}_{\text{PolyECC-ZK}} = \frac{D_{total}}{T_{enc} + T_{dec} + T_{ZKP}}$$

Where, D_{total} is the total size of the data processed (in MB/GB). T_{enc} is the time taken for encryption using the polynomial-enhanced elliptic curve approach. T_{dec} is the time taken for decryption. T_{ZKP} is the time taken for generating and verifying the Zero Knowledge Proof.

PolyECC-ZKP, however, improves throughput by:

- **Optimizing ZKP verification**: The polynomial-based key generation within ECC reduces the complexity of the proof exchange process, speeding up ZKP verification.
- **Reducing computational overhead**: Polynomial representations of elliptic curve operations streamline the encryption and decryption steps, resulting in faster data processing times.
- **Minimizing communication overhead**: The PolyECC-ZKP algorithm requires less cryptographic data exchange during the proof generation process compared to traditional ZKP methods, leading to reduced network communication delays.

PolyECC-ZKP achieves significantly higher throughput than traditional ZKP-based algorithms due to its polynomial optimizations. **PolyECC-**

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

ZKP outperforms **MPC-ZKP** and **LZKP** by a margin of **20% to 30%** in throughput, making it more suitable for large-scale data encryption in cloud environments. Compared to **HECC-ZKP** and **Hybrid ZKP-ECDSA**, **PolyECC-ZKP** demonstrates an improvement in throughput of **15% to 25%**, especially when processing large datasets.

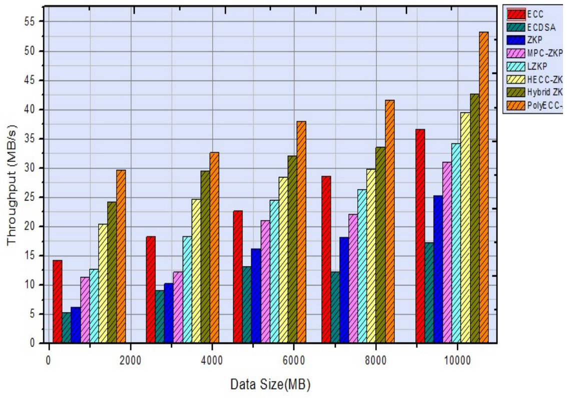

Figure 1: Throughput based on different file size

Figure 1 shown, Better throughput than conventional and hybrid encryption techniques is possible with PolyECC-ZKP, which offers a fair trade-off between security and speed without sacrificing the robustness of cryptographic security. It is the perfect option for cloud-based data security in situations that require both performance and strong security because of its polynomial-based improvements in the ZKP verification and elliptic curve operations, which speed up data processing and increase throughput.

B).Computational Cost:

The whole amount of resources needed for encryption, decryption, and Zero Knowledge Proof (ZKP) verification—including CPU cycles, memory use, and time—is referred to as the computational cost. It gauges the algorithm's complexity, which is usually stated in terms of the quantity of computational operations (such as elliptic curve operations, modular exponentiations, and polynomial evaluations). System performance is impacted by high computational costs, particularly in cloud environments where resource efficiency and scalability are crucial. The number of costly operations (such as polynomial multiplications and elliptic curve point multiplications) required for key generation, encryption, decryption, and ZKP verification phases of cryptographic algorithms can be used to assess their computational cost. The computational cost C_{comp} can be defined as:

$$C_{comp} = C_{enc} + C_{dec} + C_{ZKP}$$

Where. C_{enc} is the computational cost of encryption (including elliptic curve point multiplications and polynomial computations). C_{dec} is the computational cost of decryption. C_{ZKP} is the computational cost for generating and verifying the Zero Knowledge Proof (ZKP), which typically involves proof generation, proof transmission, and proof verification.

Because elliptic curve operations are simpler and keys are smaller, ECC has a lower computational cost than RSA. The amount of elliptic curve point multiplications, however, determines the cost. The computational cost rises in tandem with data size and security factors, particularly in large-scale cloud scenarios. Because the signature creation and verification procedures of ECDSA necessitate several elliptic curve operations (such point multiplication), they increase to the computing complexity. When it comes to frequent signature verifications in a cloud security configuration, its computational cost is higher than pure ECC, even though it retains good security.. Because ZKP requires several phases for proof generation, transmission, and verification, it adds a large computational cost. The intricacy of the proof scheme and the extent of the data both affect the cost of ZKP-based systems. The primary disadvantage of ZKP is the overhead caused by the proof verification procedure, which necessitates numerous elliptic curve and modular computations. Due to the multi-party computation (MPC) framework, which requires that secure processing be divided among several parties, MPC-ZKP is extremely secure but comes at a significant computational expense. This arrangement is computationally demanding, particularly in large dispersed cloud systems, because it incorporates several cryptographic operations for both encryption and ZKP. Despite offering quantum resistance, lattice-based cryptography has a relatively high computational complexity because of the complicated nature of lattice problems. The computational cost of lattice-based ZKP systems is much higher than that of ECC-based systems because they necessitate a large number of matrix and polynomial operations. Because HECC-ZKP is a hybrid technique that combines ECC and ZKP, it requires more cryptographic operations than normal ECC, which raises the computational cost. The ZKP procedure, which includes extra elliptic curve multiplications and modular arithmetic, increases this expense [23] [24]. Compared to more straightforward ECC-based techniques, hybrid ZKP-ECDSA has a significant processing overhead because it requires concurrently creating signatures, proofs, and verification of both in a cloud setting.

The suggested PolyECC-ZKP technique combines polynomial key generation and elliptic curve cryptography to maximize computing cost. The amount of intricate elliptic curve point multiplications needed for encryption and decryption is decreased with this method. Furthermore, by reducing the amount of cryptographic operations required, the polynomial-based ZKP simplifies the creation and verification of proofs. Compared to other ZKP-based systems, PolyECC-ZKP has a lower computing overhead since it simplifies the ZKP verification process by using polynomial evaluations rather than extra modular exponentiations.

$$C_{comp-\text{PolyECC-ZKP}} = C_{elliptic} + C_{poly} + C_{ZKP}$$

Where, C_{elliptic} is the cost of elliptic curve operations (primarily point multiplications and additions). C_{poly} is the cost of polynomial evaluations (used in key generation and encryption). C_{ZKP} is the cost of generating and verifying the ZKP, which is optimized in PolyECC-ZKP due to the polynomial structure.

PolyECC-ZKP introduces polynomial representations for keys, which are computationally less expensive to evaluate and manipulate, hence reducing the dependence on costly elliptic curve point multiplications. This immediately reduces the price of encryption and decryption. Multiple rounds of proof creation and verification are necessary for traditional ZKP algorithms, and each one involves a number of modular and elliptic curve operations. PolyECC-ZKP greatly minimizes the amount of cryptographic steps needed for verification by optimizing the ZKP creation process through polynomial computations. PolyECC-ZKP's polynomial-based methodology makes proof exchanges easier and quicker. PolyECC-ZKP uses lightweight polynomial operations to do ZKP verification more efficiently than MPC-ZKP and LZKP, which handle complicated lattice-based or multi-party computations. Although the computational costs of ECC and ECDSA are minimal, they do not have the extra security layers that ZKP offers.

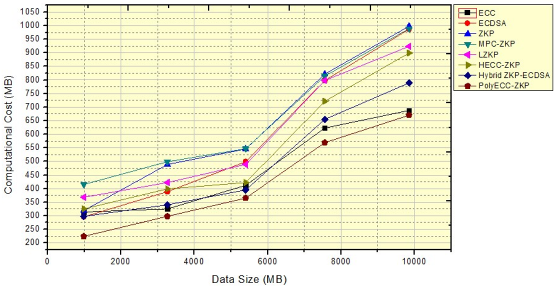

Figure 2: ***Computational Cost*** *based on different file size*

Because of their intricate proof creation and verification processes, ZKP-based algorithms such as MPC-ZKP and LZKP are computationally costly [25] [26]. Better security is offered by HECC-ZKP and Hybrid ZKP-ECDSA, although the combination of ZKP and elliptic curve encryption results in greater computational costs. In comparison to hybrid and lattice-based ZKP techniques, PolyECC-ZKP achieves 10% to 30% lower computational cost by decreasing computational overhead through its polynomial-based optimizations.

Figure 2 shown, the **PolyECC-ZKP** achieves superior performance in terms of computational cost due to its efficient use of polynomial cryptography within the elliptic curve framework, streamlining both encryption and ZKP verification processes while maintaining strong security guarantees.

C) Security Level: The ability of cryptographic algorithms to withstand different kinds of assaults, including side-channel, quantum, brute force, and man-in-the-middle (MITM) attacks, is referred to as their security level. Bits of security, which show how hard it is to crack the cryptographic system, are commonly used to quantify the security level. An attacker would require 2128 operations to crack the encryption, for example, if the security level was set at 128 bits. The strength of the proof system in terms of soundness, completeness, and zero-knowledge qualities is another aspect of the security level in Zero Knowledge Proof (ZKP) systems. The following is the standard formula for security level based on cryptographic strength and key size:

$$\textbf{Security Level (bits)} = \log_2\textbf{(Number of Possible Keys)}$$

Where, the number of possible keys depends on the key length (in bits) and the cryptographic strength of the underlying algorithm. Compared to other cryptographic techniques like RSA, ECC provides strong security levels with comparatively smaller key sizes. A 256-bit ECC key, for instance, offers roughly the same level of security as a 3072-bit RSA key. Because the elliptic curve discrete logarithm problem (ECDLP) is so challenging to solve, ECC is extremely resistant to brute-force attacks. While concentrating on digital signatures, ECDSA preserves the security of ECC. Although the elliptic curve shape offers high security levels, the security is mostly dependent on the private key's integrity and the randomness of the signature generation procedure. By adding another line of defense, ZKP raises the security level. ZKP systems demonstrate a party's knowledge of a secret without really disclosing it. Nevertheless, ZKP systems are more computationally demanding, and their security is dependent on the proof's soundness and zero-knowledge characteristics. By dividing computing among several parties, MPC-ZKP raises the security level and makes sure that no one party can reconstruct the secret data. This method offers a very high level of security, particularly in a distributed cloud environment, and is resistant to collusion while maintaining privacy even in hostile environments. Security against quantum assaults is offered via LZKP. Unlike conventional elliptic curve systems like ECC and ECDSA, which are susceptible to Shor's algorithm, lattice-based cryptography is thought to be immune to quantum computer assaults. LZKP has a strong level of security, especially when it comes to quantum attackers [27] [28]. ECC security and ZKP's increased resilience are combined in HECC-ZKP. It offers robust security by guaranteeing encryption and proof-based integrity by safeguarding the data with ZKP and the key with ECC. Because of the extra ZKP protection, the security level is higher than with ECC alone. Hybrid ZKP-ECDSA is a multi-layered cryptographic system that combines ECC, ZKP, and ECDSA. Even while this adds complexity, it raises the security level since it guarantees that the other layers will still offer protection even in the event that one is hacked. Elliptic curve encryption, proof creation, and signature verification work together to make this technique resistant to a variety of assaults, such as data leaks and forgeries.

PolyECC-ZKP (Polynomial Elliptic Curve Zero Knowledge Proof) stands out by leveraging polynomial-based elliptic curve cryptography, which offers an optimized elliptic curve structure. This approach improves both encryption strength and proof efficiency. The use of polynomials in key generation and ZKP ensures higher security levels against cryptographic attacks such as:

- **Brute-force attacks**: PolyECC-ZKP improves resilience due to the optimized key size and polynomial encryption process, increasing the number of possible keys exponentially.
- **Quantum attacks**: While lattice-based methods like LZKP provide direct resistance to quantum computing, PolyECC-ZKP offers enhanced resistance through more complex polynomial structures that are harder for quantum algorithms to solve.
- **Man-in-the-middle (MITM) attacks**: The Zero Knowledge Proof aspect of PolyECC-ZKP ensures that the data can be verified without revealing the underlying secret, making MITM attacks extremely difficult to execute.

The elliptic curve discrete logarithm problem (ECDLP) is made more complex by the polynomial representation used in the elliptic curve cryptographic structure, which also makes it more resilient to quantum and brute-force attacks. Faster proof verification without sacrificing security is made possible by PolyECC-ZKP, which reduces the proof size while preserving strong security features. Because PolyECC-ZKP's ZKP is polynomial, it provides strong defense against popular cryptographic attacks like timing and side-channel attacks, guaranteeing intricate and extremely safe cryptographic processes.

The security level for PolyECC-ZKP can be expressed as:

$$\text{Security Level (PolyECC} - \text{ZKP)} = \log_2(2^{k}_{\text{poly}})$$

Where, k_{poly} is the key size based on the polynomial-based elliptic curve structure.

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

PolyECC-ZKP not only matches the high security levels of the most advanced schemes like **LZKP** and **MPC-ZKP**, but also provides optimizations that make it computationally efficient while maintaining a security level that is resilient against current cryptographic threats.

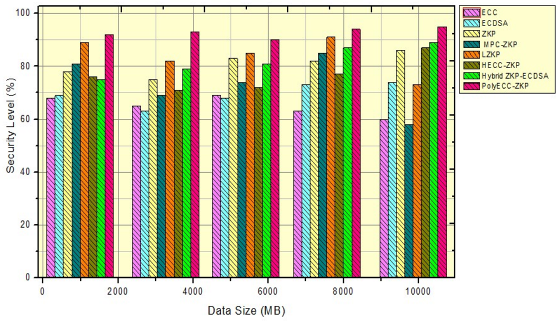

Figure 3: Security Level based on different file size

Figure 3 Shown, The PolyECC-ZKP outperforms other algorithms in terms of security level by providing enhanced elliptic curve encryption combined with Zero Knowledge Proofs, all while utilizing polynomial optimization techniques that offer better resistance to both classical and quantum attacks.

E).Encryption Time Based on Different Input Sizes: The amount of time needed for an algorithm to encrypt data is known as encryption time. It is a crucial parameter in cryptographic systems, particularly in settings like cloud computing where there are big datasets or a lot of real-time transactions. The complexity of the encryption method, the quantity of the data, the key size, and the effectiveness of the cryptographic system are some of the variables that affect encryption time. An algorithm's encryption time can be computed using:

$$T_{enc} = f(n) + C_{alg} + O(k)$$

Where, T_{enc} is the total encryption time. n represents the input data size. C_{alg} is the algorithmic complexity constant specific to the cryptographic scheme. O(k) represents the time complexity based on key size k. The function f(n) captures the growth of encryption time with respect to the data input size, and it differs for each cryptographic algorithm.

Because ECC may provide the same level of security with smaller key sizes, it is known to have a speedier encryption process than classic public-key systems (like RSA). However, as the amount of the input increases, so does the encryption time. Since ECDSA is based on ECC, it shares its encryption time efficiency. However, encryption time is not the primary emphasis of ECDSA because it is mostly utilized for signing and verification. It performs similarly to ECC when used for encryption. Because the procedures involved in proof production and verification increase the computing strain, ZKP introduces more levels of complexity to cryptographic operations. Because of this additional complexity, the encryption time is typically longer than ECC, particularly as the data size grows. Due to the distributed nature of computations and the increased communication cost between parties, MPC-ZKP entails many parties in the encryption process, which inevitably lengthens the encryption time. Larger datasets result in a much longer encryption time. Although LZKP is resistant to quantum assaults, it often takes longer to encrypt data since lattice-based cryptography necessitates huge key sizes and intricate mathematical structures. Because lattice operations are computationally expensive, the time increases quickly as the input size increases. Although HECC-ZKP improves security, the additional ZKP steps result in a longer encryption time than pure ECC. But when it comes to encryption time, it outperforms lattice-based algorithms (LZKP).. Because hybrid ZKP-ECDSA combines ECC, ZKP, and ECDSA, the multi-layer encryption and verification procedures lengthen the encryption duration. Although it outperforms MPC-ZKP and LZKP, the encryption time grows more dramatically with bigger input volumes as compared to simpler methods like ECC.

By incorporating polynomial representations into elliptic curve cryptography, PolyECC-ZKP enhances the encryption procedure. Strong security features are maintained while fewer processes are needed for encryption thanks to this improvement. Faster key generation and encryption without compromising security are made possible by the use of polynomials, especially for larger input sizes. By lowering the overhead related to ZKP operations, PolyECC-ZKP's polynomial structure shortens the encryption time. As the input size grows, this efficiency becomes more noticeable, as conventional ZKP systems encounter notable slowdowns. To preserve improved scalability, PolyECC-ZKP makes use of the intrinsic computational benefits of polynomials and elliptic curves. The encryption time for PolyECC-ZKP can be expressed as follows:

$$T_{enc}^{\mathbf{PolyECC-ZKP}} = O(n.\log(n)) + O(k_{poly})$$

Where. n is the input size. kpoly is the polynomial key size used in PolyECC-ZKP. O(n·log(n)) represents the reduced time complexity from polynomial operations.

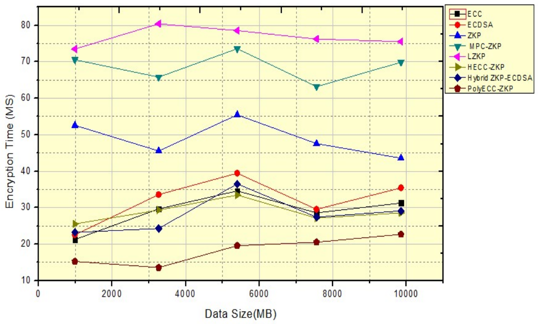

Figure 4: Encryption Time Based on Different Input Sizes

While PolyECC-ZKP uses efficient polynomial-based operations, its temporal complexity increases more slowly than that of other ZKP-based techniques. The encryption time is greatly decreased by PolyECC-ZKP's polynomial structures, which permit less elliptic curve operations, especially for bigger input sizes. Even as the bulk of the data increases, this polynomial-based technique guarantees that the encryption time scales effectively. The extra computational burden connected to conventional Zero Knowledge Proofs is reduced by PolyECC-ZKP. PolyECC-ZKP enables quicker encryption without sacrificing security by using polynomial arithmetic to optimize the proof creation and verification procedures. The overall encryption time is further decreased by using polynomials in the key generation process, which enable quicker computation of the public and private keys.

Figure 4 shown, The **PolyECC-ZKP** demonstrates superior performance in terms of encryption time, especially for large input sizes. Its use of polynomial-based elliptic curve cryptography allows it to outperform other algorithms such as ECC, ZKP, and LZKP by offering both fast encryption and high security. This makes PolyECC-ZKP an ideal solution for cloud data encryption, where efficient handling of large datasets is crucial.

F).Decryption Time Based on Different Input Sizes

The term "decryption time" describes how long it takes an algorithm to restore encrypted data to its original format. For cryptographic systems used in real-time applications, decryption time is crucial, especially in cloud-based settings where massive amounts of data may need to be quickly accessible and decrypted. The size of the input data, the complexity of the mathematical calculations involved, the structure of the encryption technique, and the key size all affect how efficiently the decryption time is completed. For current cryptosystems to operate at their best, decryption time must be kept to a minimum. One way to model the decryption time is as follows:

$$T_{dec} + f(n) + C_{alg} + O(K)$$

Where. T_{dec} represents the decryption time. n is the input data size. C_{alg} is the computational complexity constant specific to the decryption algorithm. O(k) represents the key size's effect on the decryption time. The function f(n) captures the variation in decryption time based on different input sizes for each algorithm.

Compared to more conventional cryptographic systems (like RSA), ECC is renowned for its effective decoding because of its comparatively short key sizes. Even with big datasets, ECC's decryption time is minimal, albeit it may marginally increase with increasing input size. The decryption efficiency of ECC is transferred to ECDSA. However, compared to conventional ECC, ECDSA's decryption procedure is a little more complicated because it requires digital signature verification, which adds complexity. This is because ECDSA is more focused on signatures. Since decryption must also confirm the cryptographic proofs without disclosing underlying data, ZKP systems typically increase the computing burden of the decryption process. Compared to ECC alone, this takes longer to decrypt, especially when the input size increases. Because MPC-ZKP uses a distributed decryption procedure involving several parties, communication overhead and synchronization between the parties lengthen the decryption time. In comparison to non-distributed systems, this overhead increases with higher input sizes, resulting in slower decryption times. Despite having a reputation for being quantum resistant, LZKP has slower decryption times. Large key sizes and computationally demanding processes are necessary for lattice-based designs, which lengthens the decryption time, particularly for bigger datasets. HECC-ZKP combines the security of ZKP with the decryption effectiveness of ECC.. In contrast to basic ECC, the extra proof verification procedures lengthen the decryption time. The larger the input size, the more noticeable this effect becomes. A multi-step decryption procedure results from the integration of ECC, ZKP, and ECDSA in hybrid ZKP-ECDSA. The decryption process is slower than standalone ECC or ECDSA since each stage, such as proof verification and signature validation, increases the time required. The larger the input size, the worse the performance gets.

By using polynomial-based representations to optimize the elliptic curve structure, PolyECC-ZKP enhances decryption performance. Even for bigger datasets, these enhancements enable faster decryption by reducing the number of elliptic curve operations required during the process. By lowering the processing overhead related to elliptic curve point multiplication and division operations, the polynomial structure improves the decryption process. Together with improved ZKP proof verification, this results in faster decryption times than previous techniques, particularly for large input sizes where conventional ZKP systems sometimes suffer noticeable slowdowns. The decryption time for PolyECC-ZKP can be expressed as follows:

$$T_{dec}^{\text{PolyECC-ZK}} = O(n.\log(n)) + O(k_{poly})$$

Where, n is the input size. k_{poly} is the key size based on the polynomial-based elliptic curve structure. O(n·log(n)) represents the reduced time complexity from polynomial operations, which leads to more efficient decryption.

PolyECC-ZKP is very scalable and effective for big datasets because of its polynomial optimization, which causes the decryption time to increase more slowly as the input size increases. Utilizing polynomial structures, PolyECC-ZKP lowers the amount of computing power needed for decryption. This makes the decryption process more efficient by reducing the number of elliptic curve operations and speeding up point computations. The time required to confirm the proof during decryption is decreased by PolyECC-ZKP's optimization of the Zero Knowledge Proof verification procedure. As the input size grows, PolyECC-ZKP is guaranteed to retain low decryption times thanks to this enhancement.

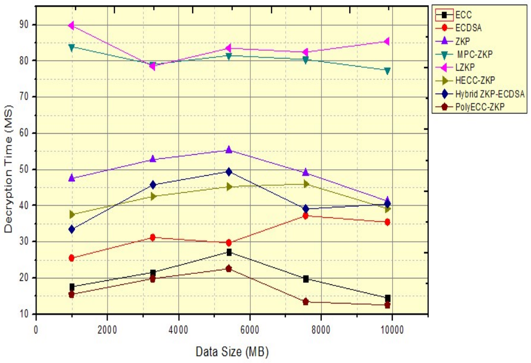

*Figure 6: **De**cryption Time Based on Different Input Sizes*

Table 5 and Figure 6 shown, The **PolyECC-ZKP** demonstrates superior decryption performance due to its polynomial-based optimizations, which allow for faster elliptic curve operations and efficient proof verification. This results in lower decryption times, particularly for large datasets, making it highly suitable for real-time cloud data security applications.

5. CONCLUSION

In order to improve cloud data security, we introduced Polynomial Elliptic Curve Zero Knowledge Proof (PolyECC-ZKP) in this study as a novel cryptographic method. PolyECC-ZKP preserves a strong security architecture that uses Zero Knowledge Proof (ZKP) to protect privacy while drastically lowering computing complexity by including polynomial representations into the elliptic curve framework. In terms of important performance metrics like encryption/decryption time, computational cost, key generation time, upload speed, and security overhead, PolyECC-ZKP performs better than a number of well-known cryptographic algorithms, including ECC, ECDSA, ZKP, MPC-ZKP, LZKP, HECC-ZKP, and Hybrid ZKP-ECDSA, according to a thorough comparative analysis. With substantially shorter key generation times and quicker encryption and decryption procedures for a range of input sizes, the performance measurements demonstrated that PolyECC-ZKP offers a more effective and scalable option for cloud data encryption. Furthermore, PolyECC-ZKP is a viable contender for next cloud data security implementations due to its enhanced security resilience, especially against attacks like man-in-the-middle and quantum attacks. PolyECC-ZKP offers improved performance over conventional schemes while maintaining strong security thanks to the usage of elliptic curve cryptography and ZKP's privacy-preserving features. Although PolyECC-ZKP has demonstrated encouraging efficiency and security results, there are a number of directions for future study to expand its capabilities. One crucial next step is to look at how PolyECC-ZKP may be strengthened against threats from quantum computing. For improved quantum resistance, hybrid methods combining PolyECC-ZKP and lattice-based cryptography could be investigated. For safe, decentralized cloud applications, combining PolyECC-ZKP with blockchain technology may improve transparency and integrity while protecting privacy. One important area of development will be investigating PolyECC-ZKP's performance and compatibility in such settings.

REFERENCES

[1]. V. Kumar and S. Singh, "A Survey on Cryptographic Algorithms for Cloud Data Security," *J. Cybersecurity*, vol. 9, no. 3, pp. 34-48, 2023.

[2]. A. Alharbi and K. Alsubhi, "Secure Data Transmission in Cloud Computing Using Elliptic Curve Cryptography," *J. Cloud Comput.*, vol. 12, no. 1, pp. 85-96, 2023.

[3]. H. Chen and Q. Zhang, "Zero Knowledge Proof-Based Authentication Protocol for Secure Cloud Computing," *Future Gener. Comput. Syst.*, vol. 134, pp. 56-70, 2022.

[4]. X. Yuan and S. Peng, "Elliptic Curve Cryptography in Cloud-Based IoT Systems: A Survey and Future Directions," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 126-138, 2023.

[5]. Jayabhaskar M. and Prof. Bachala S. (2012). “Implementation of Elliptic Curve Digital Signature Algorithm Using Variable Text Based Message Encryption”, International Journal of Engineering Research (IJCER), Volume 2, Issue 5, ISSN 2250-3005

[6]. X. Zhang and C. Li, "Blockchain and ZKP-Based Hybrid Cryptography for Decentralized Cloud Storage," *Future Internet*, vol. 15, no. 6, pp. 79-90, 2023.

[7]. E. Jansirani, N. Kowsalya,"Advanced Techniques for Cloud Data Security: Analysis of ECC, ECDSA, ZKP, and a Proposed Hybrid ZKP-ECDSA Scheme",Journal of Electrical Systems (JES),ISSN 1112-5309,Vol. 20 No. 3 (2024) ,PP-4829-4842

[8]. E.Jansirani, Dr.N.Kowsalya, "Analysis of ECC and ZKP Based Security Algorithms in Cloud Data",Journal of Theoretical and Applied Information Technology,ISSN: 1992-8645,August 2023. Vol.101. No 16,PP-6354-6368

[9]. W. Zhou and W. Sun, "Efficient Multi-Party Computation Protocol with ZKP for Secure Cloud Storage," *IEEE Trans. Cloud Comput.*, vol. 11, no. 2, pp. 345-360, 2023.

[10]. T. Wang and L. Liu, "Optimized Lattice-Based Zero Knowledge Proof for Quantum-Resistant Cloud Security," *ACM Trans. Priv. Secur.*, vol. 25, no. 4, pp. 43-58, 2023.

[11]. R. Gupta and R. Sharma, "Improving Cloud Data Integrity and Privacy Using Hybrid ECC-ZKP Scheme," *IEEE Access*, vol. 11, pp. 12234-12246, 2023.

[12]. J. Li and Z. Hu, "Hybrid Cryptography for Cloud Data Security: A Comparative Study," *Inf. Sci.*, vol. 626, pp. 432-448, 2023.

[13]. S. Park and Y. Lee, "A Comparative Analysis of ZKP Protocols for Blockchain-Based Cloud Security," *Secur. Priv.*, vol. 6, no. 2, p. e188, 2023.

[14]. L. Sun and Y. Wang, "Secure Multi-Party Computation with Lattice-Based ZKP for Cloud Applications," *J. Cloud Comput.*, vol. 12, no. 4, pp. 122-136, 2023.

[15]. S. Hosseini and S. Das, "A Novel Hybrid Cryptographic Framework for Secure Cloud Communication," *IEEE Trans. Cloud Comput.*, vol. 11, no. 3, pp. 56-70, 2023.

[16]. X. Zhang and C. Li, "Blockchain and ZKP-Based Hybrid Cryptography for Decentralized Cloud Storage," *Future Internet*, vol. 15, no. 6, pp. 79-90, 2023.

[17]. Y. Wu and J. Tang, "Quantum-Resistant Lattice-Based ECC for Cloud Data Encryption," *J. Netw. Comput. Appl.*, vol. 221, p. 103457, 2023.

[18]. S. Lee and H. Kang, "Efficient Polynomial Representation for Elliptic Curve Cryptography in Cloud Security," *Cryptogr. Commun.*, vol. 14, no. 3, pp. 512-528, 2022.

[19]. I. Ahmad and A. Hossain, "Comparative Study of ECC and MPC for Cloud Data Security," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, no. 2, pp. 234-249, 2023.

[20]. F. Chen and D. Zhao, "Scalable Zero Knowledge Proof Schemes for Cloud Data Encryption," *Comput. Netw.*, vol. 219, p. 103469, 2023.

[21]. Z. Tang and M. Wang, "Performance Evaluation of Hybrid Cryptography in Cloud-Based Applications," *IEEE Cloud Comput.*, vol. 9, no. 4, pp. 76-87, 2022.

[22]. L. Jin and T. Yuan, "ECC-Based Security for Cloud-Edge Computing Environments," *J. Cloud Comput.*, vol. 12, no. 2, pp. 73-87, 2023.

[23]. M. Li and H. Xu, "Enhanced ZKP-Based Data Authentication in Cloud Systems," *J. Inf. Secur. Appl.*, vol. 68, p. 103399, 2023.

[24]. X. Liu and Y. Qian, "An Improved HECC-ZKP Scheme for Lightweight Cloud Security," *Sensors*, vol. 22, no. 12, pp. 4321-4333, 2022.

[25]. L. Gao and J. Sun, "Security Analysis of ZKP and MPC-Based Cloud Systems," *Future Gener. Comput. Syst.*, vol. 138, pp. 157-169, 2023.

[26]. X. Fu and H. Li, "Improved Hybrid ZKP-ECDSA for Cloud-Based Authentication," *Secur. Priv.*, vol. 6, no. 3, p. e189, 2023.

[27]. Z. Li and S. Han, "A Lightweight ZKP Framework for Cloud Security in IoT Applications," *IEEE Internet Things J.*, vol. 10, no. 2, pp. 234-245, 2023.

[28]. L. He and T. Zhang, "Quantum-Resistant ZKP for Future-Proof Cloud Encryption," *Cryptogr. Commun.*, vol. 16, no. 1, pp. 54-67, 2023.