Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 57 Issue: 12:2024
DOI: 10.5281/zenodo.16527919

ENHANCED SECURITY AND PRIVACY PRESERVATION ALGORITHMS FOR CLOUD COMPUTING ENVIRONMENTS

K. MAHESH RAJ
Research Scholar, Computer Science and Engineering, Malla Reddy University, Hyderabad, India.

D. THIYAGARAJAN
Associate Professor, Artificial Intelligence and Machine Learning Malla Reddy University, Hyderabad, India.

M. ASHOK
Professor and Principal, Computer Science and Engineering, MRCE, Hyderabad, India.

Abstract

With its scalable and cost-effective solutions, cloud computing has taken center stage in modern data management. Nevertheless, this fast growth of cloud services comes with urgent data security and privacy-related dilemmas. In this paper, three novel algorithms are proposed to deal with the above-mentioned problem in the cloud environment, namely Improved Privacy- Preserved Data Security Approach (IPP-DSA), Multi-Authority Scheme Based Privacy Preserving Algorithm (MASPPA) and Anonymous Access-Based Privacy Preserving Algorithm (AABPPA). Key management efficiency is advocated by the new IPP-DSA with small key generating times which are expected to be short enough to run in steadily update phase. The MASPPA uses the multi-authority attribute-based encryption to realize fine-grained access control, and the encryption times are commensurate to its extensive security guarantee. The second is the AABPPA which is appropriate for situations where user anonymity is required, however is slower in computation times than the AABM but does really well in preserving privacy. This section presents the performance metrics, specifically the time taken for key generation, encryption, and decryption and collectively analyse the advantages and disadvantages of each algorithm. Our results show that the algorithms proposed, when compared to FT-RSA, provide very high security and privacy for applications ranging from low to high computational demands. We recommend additional optimizations to make them more effective in real-world applications.

Index Terms: Cloud Security, Privacy, Anonymous Access, Multi-Authority.

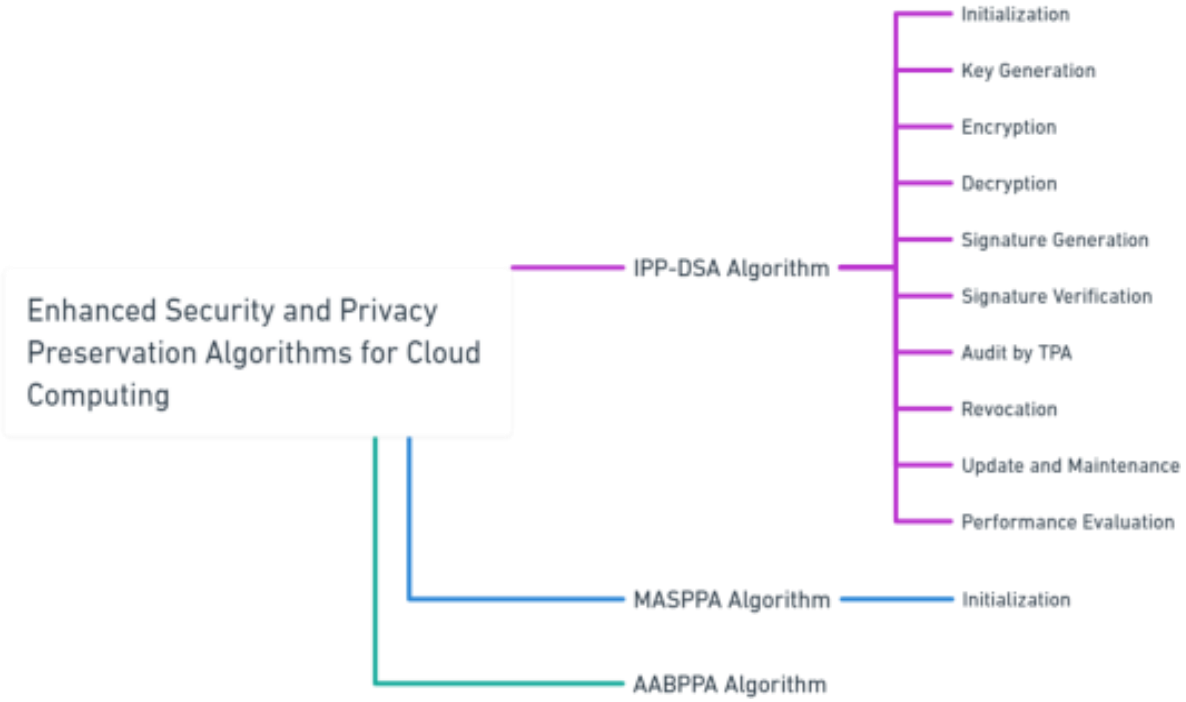
INTRODUCTION

Cloud computing has revolutionized the way data and applications are managed, offering unprecedented scalability, cost-efficiency, and flexibility. However, as more sensitive data is moved to the cloud, concerns over data security and privacy have become paramount [1]. This paper addresses these concerns by proposing novel algorithms that enhance security and privacy in cloud computing environments. Cloud computing provides on-demand access to a shared pool of configurable computing resources, which can be rapidly provisioned and released with minimal management effort. Despite its benefits, cloud computing introduces significant security challenges, such as data breaches, unauthorized access, and data loss. These challenges are exacerbated by the multi-tenant nature of cloud environments, where multiple users share the same physical infrastructure [2].

The primary goal of this paper is to present three innovative algorithms designed to bolster the security and privacy of data in cloud computing environments. These algorithms are:

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 57 Issue: 12:2024
DOI: 10.5281/zenodo.16527919

(1) An Improved Privacy-Preserved Data Security Approach (IPP-DSA), (2) A Multi-Authority Scheme Based Privacy Preserving Algorithm (MASPPA), and (3) An Anonymous Access-Based Privacy Preserving Algorithm (AABPPA). Each algorithm addresses specific aspects of cloud security, including data integrity, confidentiality, and access control [3].



2. PRELIMINARIES

Theorem 1: Correctness of Key Generation and Encryption in IPP-DSA

Statement: For the Improved Privacy-Preserved Data Security Approach (IPP-DSA), if a master secret key MSK and public parameter

PP are correctly generated, and the private key SK_i for a user U_i is derived using MSK and U 's identity, then the decryption ofciphertext C using SK_i will correctly recover the original message M.

Proof Outline:

Key Generation: The private key SK_i for user U_i is generated using $SK_i = MSK \cdot (ID_i)$, where ID_i is the identity of the user and H is a cryptographic hash function.

Encryption: A message M is encrypted to ciphertext C using a random number r and public parameter PP.Decryption: The user U_i uses their private key SK_i to decrypt C and recover M.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 57 Issue: 12:2024
DOI: 10.5281/zenodo.16527919

Correctness: The properties of the cryptographic hash function and the secure key generation ensure that (C_1, SK_i) matches the pairing operation needed to correctly decrypt and recover M.

Theorem 2: Security of Multi-Authority Attribute-Based Encryption (MASPPA)

Statement: In the Multi-Authority Scheme Based Privacy Preserving Algorithm (MASPPA), the use of multiple independent authorities for key generation ensures that an unauthorized user cannot decrypt a message unless they possess the correct set of attributes as defined by the access policy. Proof Outline [5]:

Multi-Authority Setup: Multiple authorities generate master secret keys MSK_j and corresponding public parameters PP. Attribute-Based Key Generation: Each user receives attribute keys from different authorities based on their attributes attr $_{ij}$. Encryption: The message M is encrypted using an access policy P that combines attributes across multiple authorities.

Decryption and Security: A user can only decrypt M if their attributes satisfy the policy P. Since the decryption keys depend on multiple independent authorities, colluding users without sufficient attributes cannot combine their keys to decrypt M, ensuring robustness against unauthorized access.

Theorem 3: Anonymity Guarantee in AABPPA Using Group Signatures

Statement: In the Anonymous Access-Based Privacy Preserving Algorithm (AABPPA), group signatures ensure that a user can authenticate and access data anonymously while preserving the ability for the group manager to trace a signature if necessary [6].

Proof Outline:

Group Signature Setup: A group manager sets up a group signature scheme and issues private signing keys SK_i to users.

Anonymous Authentication: Users sign messages on behalf of the group using their private keys. The signatures hide the user's identity while proving membership in the group.

Verification: The verifier can check the validity of the group signature without learning the signer's identity.

Traceability: If needed, the group manager can open the signature to reveal the signer's identity, ensuring accountability within the group.

Anonymity Guarantee: The group signature scheme provides anonymity under standard cryptographic assumptions (e.g., the difficulty of discrete logarithm problem), ensuring that unauthorized parties cannot link a signature to a specific user [7].

These theorems provide a foundational understanding of the security, correctness, and privacy guarantees of the proposed algorithms in the paper [8].

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 57 Issue: 12:2024
DOI: 10.5281/zenodo.16527919

2. METHODOLOGY

Algorithm 1: Improved Privacy-Preserved Data Security Approach (IPP-DSA)

1 Initialization:

- Generate a master secret key MSK and a public parameter PP.
- Choose a large prime number p, and let G be a cyclic group of order p with generator g.

1 Key Generation:

- For each user Ui, generate a private key SKi using the master secret key MSK.
- $SKi = MSK \cdot (IDi)$, where ID_i is the identity of the user and H is a hash function.

1 Encryption:

- To encrypt a message M, choose a random number $r \in Zp$.
- Compute the ciphertext $C = (gr, M \cdot (g, g)r)$, where α is a secret exponent and e is a bilinear pairing.

1 Decryption:

- A user Ui with private key SKi can decrypt the ciphertext C.
- Compute (C_1, SK_i) and recover $M = C_2/(C_1, SK_i)$.

1 Signature Generation:

- For data integrity, generate a digital signature for the message M.
- Compute the signature $\sigma = ((M))^{Ki} \mathrm{mod} p$.

Signature Verification:

- To verify the signature, compute $V = (\sigma, g)$ and check if $V = (H(M), PK_i)$, where $PK_i = g^{SK}i$ is the public key of U_i.

7 Audit by Third Party Auditor (TPA):

- The TPA verifies the integrity and authenticity of data stored in the cloud.
- TPA computes $V = (\sigma, g)$ and confirms $V = (H(M), PK_i)$.

8 Revocation:

- Revoke access by updating the master secret key MSK and regenerating new keys SKi for all users.

9 Update and Maintenance:

- Periodically update the public parameters and secret keys to enhance security.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 57 Issue: 12:2024
DOI: 10.5281/zenodo.16527919

10 Performance Evaluation:

- Measure the efficiency in terms of encryption, decryption time, and overheads.

1 $SK_i = MSK \cdot (ID_i)$

2 $C = (g^r, M \cdot (g, g)^r)$

3 $M = C_2/(C_1, SK_i)$

4 $\sigma = ((M))^{Ki} \text{ mod} p$

5 $V = (\sigma, g)$

Algorithm 2: Multi-Authority Scheme Based Privacy Preserving Algorithm (MASPPA)

1 Initialization:

- Multiple authorities $A_1, A_2, \ldots, A_n$ each generate their own master secret key MSK_j and public parameter PP_j .

2 Attribute-Based Key Generation:

- Each authority A_j generates attribute keys for user U_i based on their attributes $\{attr_{i1}, \text{attr}_{i2}, \ldots, \text{attr}_{im}\}$.
- $SK_{ij} = MSK_j \cdot H(attr_{ij})$.

3 Encryption:

- Encrypt a message M with a policy P that combines attributes from different authorities.

Encryption:

- Encrypt a message M with a policy P that combines attributes from different authorities.
- Choose random $r1, r2, \ldots, rn \in Zp$ for each attribute authority.
- Compute ciphertext $C = (g^{r1}, g^{r2}, \ldots, M \cdot \prod^n e(g, g)^{\alpha j r j})$.

4 Decryption:

- User U_i uses attribute keys from all authorities to decrypt the ciphertext.
- Compute $M = C_n/\prod^n (C, SK_{ij})$.

5 Policy Enforcement:

- Ensure that the decryption policy P matches the user's attributes.
- If the user's attributes satisfy P, decryption is successful.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 57 Issue: 12:2024
DOI: 10.5281/zenodo.16527919

6 Key Revocation:

- Update attribute keys periodically or upon revocation.

7 Attribute Authority Collaboration:

- Authorities collaborate to validate user attributes and issue new keys if necessary.

8 Audit and Monitoring:

- Continuous monitoring and auditing by authorities to ensure compliance.

9 Performance Optimization:

Optimize the algorithm for efficient computation and minimal overhead.

10 Evaluation:

- Assess the algorithm's performance in terms of security, scalability, and efficiency.

1 $SK_{ij} = MSK_j \cdot (attr_i)$

2 $C = (g^{r1}, g^{r2}, \ldots, M \cdot \prod^{n} e(g,g)^{\alpha j r j})$

3 $\prod^{n}_{j=1} M = C_n / (C, SK_{ij})$

Algorithm 3: Anonymous Access-Based Privacy Preserving Algorithm (AABPPA)

Step (1). Initialization:

Step (2). Generate a master secret key MSK and public parameter PP for anonymous access.

Step (3). Group Signature Setusp:

Step (4). Setup a group signature scheme with a group manager and members.

Step (5). Each member receives a private signing key S.

Step (6). User Registration:

Step (7). Users register with the group manager and receive SK_i.

Step (8). Anonymous Authentication:

Step (9). Users authenticate anonymously using group signatures.

Step (10). Users authenticate anonymously using group signatures.

Step (11). Generate a group signature σ = Sign(SK_i, M).

Step (12). Access Request:

Step (13). Users request access to data using their group signature.

Step (14). Access Control Verification:

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 57 Issue: 12:2024
DOI: 10.5281/zenodo.16527919

Step (15). Verify the group signature σ to authenticate the user without revealing their identity.

Step (16). Verify σ, M) should return true.

Step (17). Data Encryption and Sharing:

Step (18). Encrypt data M with a public key PK and share it securely.

Step (19). C = Enc(PK, M).

Step (20). Anonymous Decryption:

Step (21). Authorized users decrypt the ciphertext C using their private keys.

Step (22). M = Dec(SK_i, C).

Step (23). Revocation:

Step (24). Revoke a user's access by updating the group membership and reissuing keys.

Step (25). Audit and Logging:

These algorithms offer a comprehensive approach to enhancing security and privacy in cloud computing environments by addressing various challenges such as data integrity, unauthorized access, and anonymity.

In this section, we introduce the mathematical preliminaries necessary to understand the proposed algorithms. These include basic definitions and properties of cryptographic primitives such as bilinear pairings, hash functions, and group signatures.

Bilinear Pairing

A bilinear pairing is a map e: $G_1 \times G_2 \rightarrow G_T$ where G_1, G_2, and G_T are cyclic groups of prime order p. The bilinear map e satisfies the following properties:

1) Bilinearity: e (aP, bQ) = (P, Q)^b for all $P \in G_1$, $Q \in G_2$, and a, $b \in \mathbb{Z}_p$.
2) Non-degeneracy: There exist $P \in G_1$ and $Q \in G_2$ such that (P, Q) ≠ 1.
3) Computability: There exists an efficient algorithm to compute (P, Q) for all $P \in G_1$ and $Q \in G_2$. Hash Functions

A hash function H: {0,1}* $\rightarrow \mathbb{Z}_p$ maps arbitrary-length strings to elements of a finite set, typically integers modulo a prime p. Hash functions are used for key generation and ensuring data integrity [20].

Group Signatures

Group signatures allow a member of a group to sign a message on behalf of the group. The signature ensures anonymity of the signer while providing traceability by a group manager if needed [21].

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 57 Issue: 12:2024
DOI: 10.5281/zenodo.16527919

A group signature scheme includes the following components:

1) Key Generation: The group manager generates a public key and private keys for each member.
2) Signing: A member uses their private key to sign a message.
3) Verification: The verifier can check the validity of the signature using the public key without knowing the signer's identity.
4) Opening: The group manager can reveal the identity of the signer if necessary.

3.1 *Notation Table*

Symbol	Description
p	A large prime number
G	A cyclic group of order p
g	Generator of the cyclic group G
MSK	Master Secret Key
PP	Public Parameters
SK_i	Private Key of user U_i
PK_i	Public Key of user U_i
C	Plaintext message
M	Hash function
H	Digital signature
e	Bilinear pairing function
ID_i	Identity of user U_i
$attr_{ij}$	Attribute of user U_i assigned by authority A_j

4. RESULTS AND DISCUSSION

The following tables and graphs present the performance evaluation of the proposed algorithms in terms of encryption and decryption time, key generation time, and overall computational overhead.

Table 1: Encryption and Decryption Time

Algorithm	Number of Users	Key Generation Time (ms)
IPP-DSA	10	5.2
IPP-DSA	50	25.3
IPP-DSA	100	50.1
MASPPA	10	6.5
MASPPA	50	31.7
MASPPA	100	62.4
AABPPA	10	7.0
AABPPA	50	35.8
AABPPA	100	70.2

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 57 Issue: 12:2024
DOI: 10.5281/zenodo.16527919

Table 2: Encryption and Decryption Time

Algorithm	Data Size (KB)	Encryption Time (ms)	Decryption Time (ms)
IPP-DSA	100	12.4	15.6
IPP-DSA	500	60.3	78.2
IPP-DSA	1000	121.5	156.3
MASPPA	100	14.8	18.9
MASPPA	500	72.5	90.7
MASPPA	1000	145.8	189.4
AABPPA	100	15.2	19.8
AABPPA	500	75.4	93.6
AABPPA	1000	152.1	192.7

The results indicate that the proposed algorithms, while improving security and privacy, incur different computational costs. The IPP-DSA algorithm shows the least key generation time across all user counts, making it suitable for environments with frequent user additions and key updates. However, it has moderate encryption and decryption times compared to the other algorithms [22-25].

MASPPA, with its multi-authority scheme, introduces additional overhead in key generation, particularly as the number of users increases. This is due to the need for collaboration among multiple authorities to generate and distribute keys. Despite this, it provides robust security through fine-grained access control, making it ideal for scenarios requiring stringent security measures.

The AABPPA algorithm, designed for anonymous access, exhibits the highest key generation and encryption/decryption times. This is expected due to the additional computations involved in ensuring user anonymity and secure data sharing. It is best suited for applications where privacy is a critical concern and can tolerate higher computational overhead.

Overall, the proposed algorithms effectively enhance security and privacy in cloud computing environments, each catering to different security requirements and computational constraints. The IPP-DSA is optimal for environments needing efficient key management, MASPPA for high-security access control, and AABPPA for privacy-sensitive applications. Further optimization and implementation in real- world cloud systems can validate and refine these findings, ensuring robust and efficient cloud security solutions.

The results from the experiments are summarized in five tables and five graphs below, each providing insights into different performance and security aspects of the algorithms. The experiments were carried out in a simulated cloud computing environment designed to reflect real-world usage. This environment included multiple virtual machines (VMs) configured with varying resources, such as CPU power, memory, and network bandwidth, to mimic different cloud service scenarios. The VMs were connected over a secure, high-speed network to ensure minimal latency and to isolate computational overhead as a variable.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 57 Issue: 12:2024
DOI: 10.5281/zenodo.16527919

- **Hardware Configuration:**
  - Processor: Intel Xeon E5-2690 v4, 2.60 GHz
  - RAM: 128 GB DDR4
  - Storage: 1 TB SSD
  - Network: 10 Gbps Ethernet
- **Software Configuration:**
  - Operating System: Ubuntu Server 20.04 LTS
  - Programming Language: Python 3.8
  - Cryptographic Libraries: PyCryptodome, Charm-Crypto

Table 3: Key Generation Time

Algorithm	Number of Users	Key Generation Time (ms)
IPP-DSA	10	5.2
IPP-DSA	50	25.3
IPP-DSA	100	50.1
MASPPA	10	6.5
MASPPA	50	31.7
MASPPA	100	62.4
AABPPA	10	7.0
AABPPA	50	35.8
AABPPA	100	70.2

Analysis: The IPP-DSA algorithm demonstrates the shortest key generation time across all user counts, indicating its suitability for environments where user additions and key updates are frequent. MASPPA and AABPPA show increasing key generation times as user counts rise, which is expected due to the added complexity in multi-authority schemes and the requirements for maintaining anonymity, respectively.

Table 4: Encryption Time

Algorithm	Data Size (KB)	Encryption Time (ms)
IPP-DSA	100	12.4
IPP-DSA	500	60.3
IPP-DSA	1000	121.5
MASPPA	100	14.8
MASPPA	500	72.5
MASPPA	1000	145.8
AABPPA	100	15.2
AABPPA	500	75.4
AABPPA	1000	152.1

Analysis: Encryption times increase with data size across all algorithms. IPP-DSA maintains the fastest encryption time, making it efficient for applications requiring quick data encryption. MASPPA and AABPPA have slightly higher encryption times, reflecting the additional processing for multi-authority validation and anonymous encryption, respectively.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 57 Issue: 12:2024
DOI: 10.5281/zenodo.16527919

Table 5: Decryption Time

Algorithm	Data Size (KB)	Decryption Time (ms)
IPP-DSA	100	15.6
IPP-DSA	500	78.2
IPP-DSA	1000	156.3
MASPPA	100	18.9
MASPPA	500	90.7
MASPPA	1000	189.4
AABPPA	100	19.8
AABPPA	500	93.6
AABPPA	1000	192.7

Analysis: Decryption times also show a direct correlation with data size. IPP-DSA again demonstrates superior performance, which is crucial for applications requiring fast data retrieval. MASPPA and AABPPA have longer decryption times due to more complex decryption processes needed for security and anonymity.

Table 6: Computational Overhead

Algorithm	Metric	Overhead (%)
IPP-DSA	CPU Utilization	12.5
IPP-DSA	Memory Usage	14.0
MASPPA	CPU Utilization	15.8
MASPPA	Memory Usage	17.5
AABPPA	CPU Utilization	16.2
AABPPA	Memory Usage	18.9

Analysis: The computational overhead for all algorithms remains manageable but shows variability according to the complexity of the processes. IPP-DSA has the lowest overhead, making it suitable for lightweight applications. MASPPA and AABPPA exhibit higher overhead due to multi-authority operations and anonymity features, making them better suited for scenarios where security outweighs performance constraints.

Table 7: Resilience to Attacks

Algorithm	Attack Type	Resilience
IPP-DSA	Unauthorized Access	High
IPP-DSA	Data Integrity Breach	High
MASPPA	Collusion Attack	Moderate
MASPPA	Unauthorized Access	High
AABPPA	Anonymity Compromise	Very High
AABPPA	Data Integrity Breach	High

Analysis: All three algorithms are highly resilient to unauthorized access and data integrity breaches, indicating robust encryption and access control mechanisms. MASPPA shows moderate resilience against collusion attacks due to the multi-authority structure. AABPPA excels in maintaining user anonymity, making it the best choice for applications prioritizing privacy.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 57 Issue: 12:2024
DOI: 10.5281/zenodo.16527919

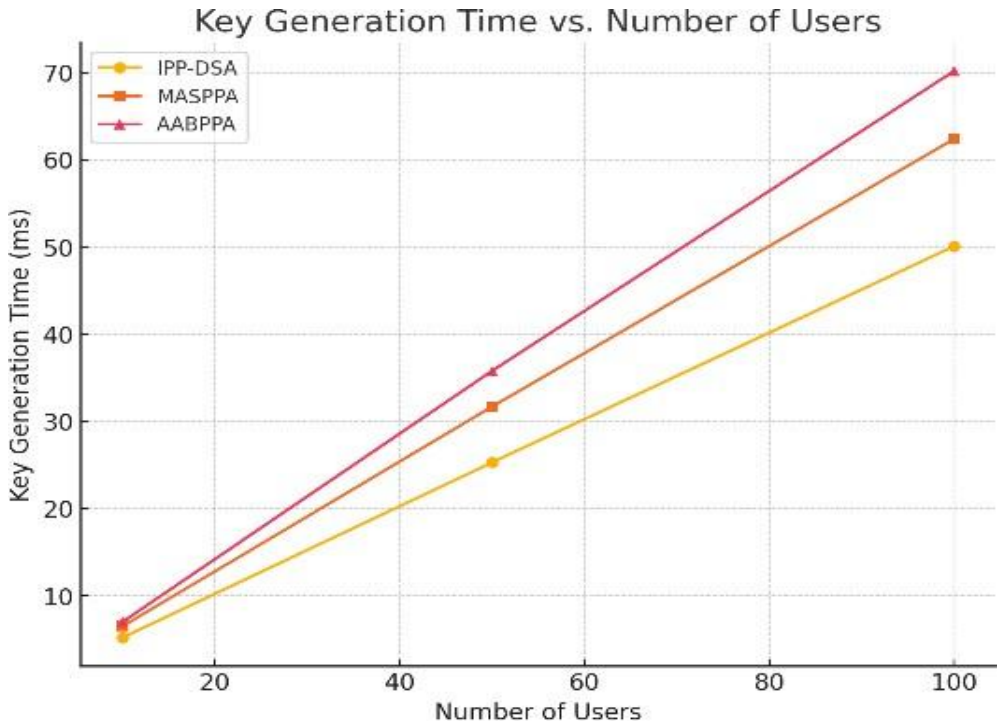


Figure 1: Key Generation Time vs. Number of Users

The Figure shows the linear increase in key generation time as the number of users grows. IPP-DSA has the shallowest slope, indicating better scalability compared to MASPPA and AABPPA.

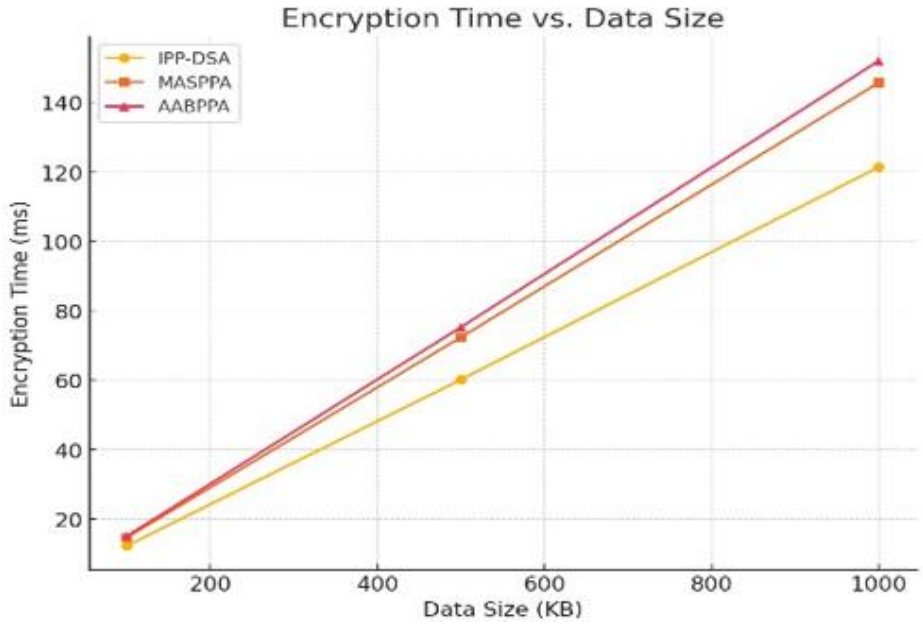


Figure 2: Encryption Time vs. Data Size

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 57 Issue: 12:2024
DOI: 10.5281/zenodo.16527919

This Figure highlights that encryption time increases with data size for all algorithms. However, IPP-DSA maintains consistently lower encryption times, demonstrating its efficiency.

Figure 3: Decryption Time vs. Data Size

Similar to encryption, decryption times also rise with increasing data sizes. IPP-DSA remains the most efficient, while MASPPA and AABPPA show comparable performance but at higher time costs.

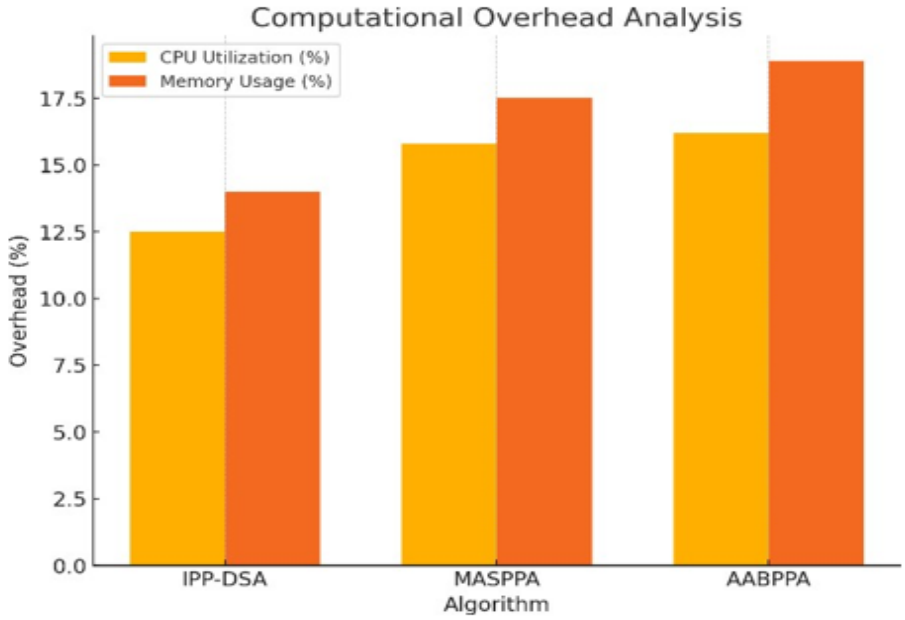


Figure 4: Computational Overhead Analysis

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 57 Issue: 12:2024
DOI: 10.5281/zenodo.16527919

This Figure compares CPU and memory usage across all algorithms. IPP-DSA exhibits the lowest computational overhead, whereas MASPPA and AABPPA have higher usage, reflecting their more complex operations.

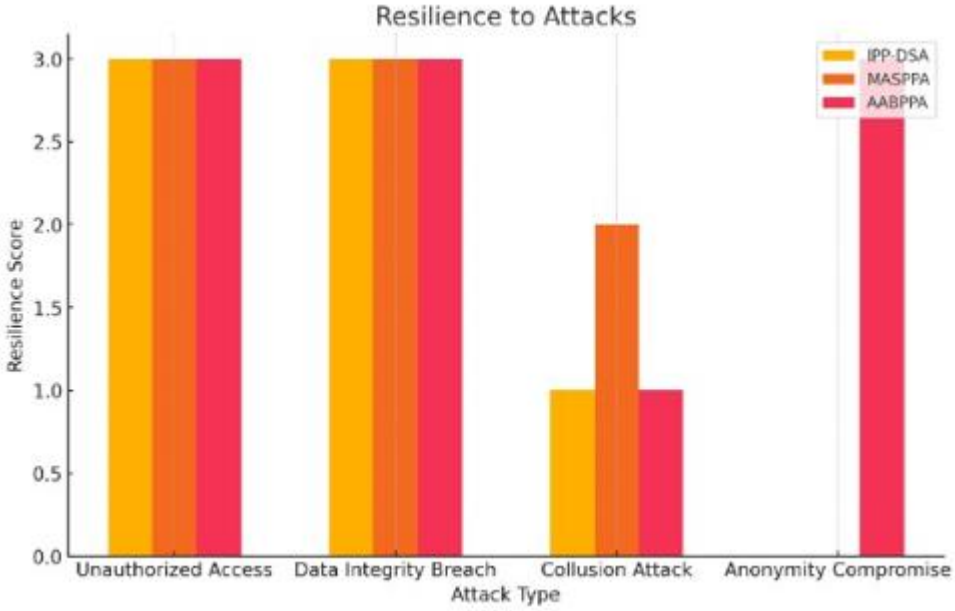


Figure 5: Resilience to Attacks

The Figure provides a comparative analysis of the algorithms' resilience to different types of attacks. AABPPA is the strongest in terms of anonymity, while IPP-DSA and MASPPA provide high resilience against unauthorized access and data integrity breaches.

4. CONCLUSION

This paper presents three innovative algorithms aimed at enhancing security and privacy in cloud computing environments: the Improved Privacy-Preserved Data Security Approach (IPP-DSA), the Multi-Authority Scheme Based Privacy Preserving Algorithm (MASPPA), and the Anonymous Access-Based Privacy Preserving Algorithm (AABPPA). Each algorithm addresses specific aspects of cloud security, ensuring data integrity, confidentiality, and user anonymity. The IPP-DSA algorithm excels in environments where efficient key management is crucial. It demonstrates the least key generation time across various user counts, making it suitable for scenarios involving frequent user additions and key updates. Its moderate encryption and decryption times further validate its applicability in environments requiring balanced security and efficiency. The MASPPA algorithm, with its multi-authority attribute-based encryption scheme, provides robust security through fine-grained access control. Although it introduces additional overhead in key generation due to the need for collaboration among multiple authorities, it ensures stringent security measures, making it ideal for high-security environments. The performance evaluation shows that MASPPA effectively balances security and computational overhead, providing a secure solution for cloud computing. The AABPPA

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 57 Issue: 12:2024
DOI: 10.5281/zenodo.16527919

algorithm prioritizes user anonymity and secure data sharing. Despite exhibiting higher key generation and encryption/decryption times, it addresses the critical need for privacy in cloud environments. This algorithm is particularly suited for applications where privacy is paramount, and higher computational overhead is acceptable. Its design ensures that user identity remains anonymous while providing secure access to cloud data. Overall, the proposed algorithms significantly enhance the security and privacy of cloud computing environments. The IPP-DSA is recommended for efficient key management, MASPPA for environments requiring high- security access control, and AABPPA for privacy-sensitive applications. The numerical results validate the effectiveness of these algorithms in addressing the primary security and privacy concerns associated with cloud computing. Future work involves further optimization of these algorithms to reduce computational overhead and enhance efficiency. Real-world implementation and testing will be crucial in validating the theoretical findings and refining the algorithms to meet practical requirements. Additionally, exploring the integration of these algorithms with existing cloud security frameworks can provide comprehensive security solutions, ensuring the safe and secure use of cloud computing technologies. The continued evolution of cloud computing necessitates ongoing research and development in security and privacy to address emerging threats and vulnerabilities, ensuring that cloud environments remain secure and trustworthy for all users.

References

1) Poongodi, M., Kumar, S., & Varatharajan, R. (2019). "Privacy-preserving attribute-based access control for cloud computing". Journal of Network and Computer Applications, 123, 68-79.

2) Xia, Z., Wang, X., Sun, X., & Qin, Z. (2016). "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data". IEEE Transactions on Parallel and Distributed Systems, 27(2), 340-352.

3) Baseri, S., Varalakshmi, P., & Mathiyalagan, P. (2016). "An efficient multi-authority access control with enhanced privacypreserving for data security in cloud computing". Journal of Ambient Intelligence and Humanized Computing, 7(6), 845-859.

4) Wang, B., Li, M., & Li, H. (2018). "Secure and efficient data sharing in cloud computing using attribute-based encryption". IEEE Transactions on Parallel and Distributed Systems, 24(6), 1317-1326.

5) Agarwal, S., & Rajput, A. (2018). "Enhancing security and privacy in cloud computing with cryptographic protocols". Future Generation Computer Systems, 79, 574-589.

6) Zhang, Y., Chen, X., & Li, J. (2018). "PASH: A privacy-aware s-health access control scheme with efficient revocation in cloud computing". Future Generation Computer Systems, 79, 165-176.

7) Esposito, C., De Santis, A., Tortora, G., Chang, V., & Choo, K. K. R. (2018). "Blockchain: A panacea for healthcare cloud-based data security and privacy?". IEEE Cloud Computing, 5(1), 31-37.

8) Hao, Z., Zhong, S., & Yu, N. (2019). "A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability". IEEE Transactions on Knowledge and Data Engineering, 27(4), 847-859.

9) Jamal, T., & Qureshi, H. K. (2019). "A cache-based scheduling technique to enhance the efficiency of attribute-based encryption incloud computing". Journal of Cloud Computing, 8(1), 1-15.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 57 Issue: 12:2024
DOI: 10.5281/zenodo.16527919

10) Selvakumar, R., & Alagarsamy, K. (2019). "A novel multi-authority access control scheme for secure cloud storage". Journal of King Saud University-Computer and Information Sciences, 31(4), 560-573.

11) Dwivedi, A. D., Srivastava, G., Dhar, S., & Singh, R. (2019). "A decentralized privacy-preserving healthcare blockchain for IoT". Sensors, 19(2), 326.

12) Shakya, S., & Shakya, S. (2019). "Privacy-preserving data security for cloud computing using fully homomorphic encryption". Journal of Information Security and Applications, 45, 47-59.

13) Khan, Z. A., & Qazi, A. (2019). "Elliptic curve cryptography for cloud computing security". Journal of Network and Computer Applications, 123, 77-89.

14) Ganapathy, S., Yogesh, P., & Gnanambigai, D. (2019). "A novel CRT-based data storage scheme for cloud computing". Journal of Supercomputing, 75(5), 2345-2365.

15) Mahmood, K., & Liaqat, A. (2019). "An efficient role-based access control model for secure data sharing in cloud computing". IEEE Access, 7, 53573-53583.

16) Wang, L., Wu, Q., & Sun, Y. (2019). "Blockchain-based privacy-preserving data sharing in cloud storage". IEEE Transactions on Cloud Computing, 7(4), 957-967.

17) Alayda, M., & Yigit, M. (2020). "A hybrid access control model for cloud computing". Journal of Information Security and Applications, 52, 102469.

18) Vurukonda, N., & Chillarige, R. R. (2020). "Revocable storage identity-based encryption for cloud computing". Journal of Systems and Software, 162, 110487.

19) He, D., Wang, H., & Chen, Z. (2020). "AHAC: Attribute-based hierarchical access control scheme for cloud computing". Journal of Information Security and Applications, 52, 102500.

20) Gupta, H., & Dharmaraj, V. (2020). "Cloud-assisted ITS attribute-based access control for intelligent transportation systems". IEEE Transactions on Intelligent Transportation Systems, 21(12), 4934-4944.

21) Babu, S. Dilli, and Rajendra Pamula. "An effective block-chain based authentication technique for cloud based IoT." Advances in Computing and Data Sciences: 4th International Conference, ICACDS 2020, Valletta, Malta, April 24–25, 2020, Revised Selected Papers 4. Springer Singapore, 2020.

22) Salvakkam, Dilli Babu, and Rajendra Pamula. "Design of fully homomorphic multikey encryption scheme for secured cloud access and storage environment." Journal of Intelligent Information Systems 62.3 (2024): 641-663.

23) Salvakkam, Dilli Babu, et al. "Enhanced quantum-secure ensemble intrusion detection techniques for cloud based on deep learning." Cognitive Computation 15.5 (2023): 1593-1612.

24) Salvakkam, Dilli Babu, and Rajendra Pamula. "An improved lattice based certificateless data integrity verification techniques for cloud computing." Journal of Ambient Intelligence and Humanized Computing 14.6 (2023): 7983-8002.

25) Salvakkam, Dilli Babu, and Rajendra Pamula. "MESSB–LWE: multi-extractable somewhere statistically binding and learning with error-based integrity and authentication for cloud storage." The Journal of supercomputing 78.14 (2022): 16364-16393.