International Journal on Emerging Technologies **16(2): 113-119(2025)**

ISSN No. (Print): 0975-8364
ISSN No. (Online): 2249-3255

Performance Evaluation of the Secure and Integrated Encryption Model (SHIELD)

A.R. Atuluku* and F.B. Osang
Department of Computer Science, Faculty of Computing, National Open University of Nigeria.

(Corresponding author: A.R. Atuluku)*

(Received 06 June 2025, Revised 03 August 2025, Accepted 22 August 2025)
(Published by Research Trend, Website: www.researchtrend.net)

DOI: https://doi.org/10.65041/IJET.2025.16.2.17

ABSTRACT: The rise of cloud computing has highlighted the need for strong data security and privacy. This study evaluates the Secure and Integrated Encryption Model (SHIELD), designed to enhance public cloud security. SHIELD uses optimized hybrid encryption (256-bit AES and 2048-bit RSA), real-time OTP authentication, Role-Based Access Control (RBAC), and auditing tools. Developed in Python with Django, it incorporates multithreading for AES and PKCS1_OAEP for RSA to boost performance. Results show encryption and decryption time improved by 28.8% and 76.2%, respectively, with an Avalanche effect over 50%, indicating strong encryption. SHIELD outperformed existing models in all areas, offering a robust solution for securing public cloud data. Future research will explore its use in private and hybrid clouds. The study's findings indicated that by combining symmetric and asymmetric key techniques, concerns about the integrity, security, privacy, and confidentiality of data stored in the cloud are addressed by the new security system.

Keywords: Performance, Cloud Computing, AES, RSA and Cryptography.

INTRODUCTION

Cloud security is a major concern in the field of computer science. The cloud is not only responsible for storing data but also offers services such as Infrastructure as a Service (IaaS) and Platform as a Service (PaaS), which provide both platforms and complete infrastructure solutions. Additionally, cloud services enable organizations to access resources both internally and externally through private, public, and hybrid cloud models (Pandey and Sharma 2017).

Over the past few years, there has been a significant increase in the volume of cloud computing utilization, particularly in the use of public cloud. This growth has increased the potentially vulnerable amount of sensitive data. More and more security risks and issues arise, particularly in public cloud environments due to its multi-tenancy and virtualization features (Wang *et al.*, 2018).

One of the main security problems the adoption of cloud computing raises is protecting sensitive data in public cloud infrastructure. The issue of secure data storage in cloud computing has become a significant concern as enterprises find it difficult to ensure the availability, confidentiality, and integrity of their information assets when entrusting them to outside cloud service providers (Stergiou *et al.*, 2020). Cryptography is a crucial element in achieving this goal (Wang *et al.*, 2020). Using cryptographic technology, which involves the encryption and decryption process, it is possible to ensure the confidentiality of data transmitted over a network (Hossain *et al.*, 2019). This paper presents the improved, secure, and integrated encryption model, SHIELD to address the intricate security needs of public cloud computing environments regarding sensitive data storage.

MATERIAL AND METHODS

Riyaz Fathima Abdul and Saravanan Arumugam (2025) address the challenge of strengthening data security against cryptanalytic attacks while preserving data integrity by proposing a hybrid encryption scheme combining RSA, ChaCha20-Poly1305, and a shuffling mechanism. Although effective in enhancing robustness, the model's high computational overhead results in slow execution and low throughput, making it less suitable for real-time or resource-constrained environments, and it lacks post-quantum protection as well as extensive real-world testing.

Shukla *et al.* (2024) address the need for enhanced security in Wireless Sensor Networks (WSNs) by proposing the Three Phase Hybrid Cryptographic (TPHC) algorithm, which combines AES, DES, and a modified RSA to provide confidentiality and authentication, alongside a custom hashing algorithm to ensure data integrity through one-wayness, collision resistance, second pre-image resistance, and a strong avalanche effect. The TPHC outperforms existing WSN algorithms in computation time, encrypted and digested data size, and energy usage. However, its use of three encryption algorithms introduces computational complexity, the modified RSA remains resource-intensive for low-power nodes, and the model lacks evaluation for scalability, diverse real-world deployments, and post-quantum readiness.

Almalawi *et al.* (2024) address data security in decentralized Edge AI networks with a hybrid mechanism combining Ant Lion Optimization (ALO) and Diffie–Hellman-based Twofish cryptography (DHT), enhanced by an Autoencoder for malicious data detection. Achieving 99.45% accuracy, 2 s execution time, 0.8 s delay, and 3.2 mJ energy use, the method offers strong security but suffers from computational overhead, high cost, and limited scalability, requiring optimization for resource-constrained and dynamic environments.

Anjana (2024) addresses the challenge of securing data confidentiality and integrity in cloud computing by proposing an enhanced three-layer cryptographic algorithm that combines AES and ChaCha20 for hybrid encryption with SHA-3 hashing for message integrity verification. The study evaluates the scheme's encryption strength, computational efficiency, and resistance to attacks, showing its suitability for mitigating common cloud security threats while offering insights to inform best practices in cloud data protection. However, it does not address performance trade-offs in large-scale deployments, lacks real-world implementation testing, introduces potential computational complexity from its multi-layer design, and omits consideration of post-quantum readiness.

Rath *et al.* (2024) presented a hybrid model for file encryption that combined a modified AES with masking and RSA algorithms. This dynamic AES-RSA algorithm, featuring masked key management, offered enhanced security and resistance against cryptographic attacks.

In the proposed encryption system, Sabitha *et al.* (2023) used a hybrid methodology that combined elliptical and parametric cryptography techniques. Authorization, key generation, and data encryption are the three separate security checkpoints that make up the system.

Chatterjee *et al.* (2023) used steganography and cryptography techniques to solve cloud storage problems. Block-per-block encryption was done using AES, RC6, Blowfish, and BRA. Sensitive data was safeguarded using the LSB method. The SHA1 hash algorithm was used to validate data security.

Ranganatha and Sujatha (2023), in their proposed method, created keys by utilising a lightweight Edwards curve. The created private keys were changed using the user's Identity-Based Encryption. The Advanced Encryption Standard (AES) encryption process was accelerated by using the suggested key reduction technique to further shorten the keys. Diffie-Hellman key exchange was used to exchange the public keys.

Using the hybrid approach, William *et al.* (2022) combined a hash function, an asymmetric algorithm (ECC), and a symmetric algorithm (AES) (SHA256). In this instance, digital data was encrypted using the mathematical formula SHA-256. It was employed to guarantee the integrity of the data. The proposed hybrid solution claimed to be comparable to an existing method that encrypts text and images using the AES algorithm. In comparison with the previously discussed approaches, the proposed strategy was more efficient since it was more effective at text encryption. Future advances could result in a reduction of the time needed to encrypt and decode a photograph.

A multilevel approach with three encryption levels was suggested by Binita and Blessy (2022). Data Encryption Standard (DES) was used at the first level, mixed transposition was used at the second level, and Blowfish was used at the third level to achieve this. Since deciphering an algorithm took time, data security increased with the number of encryption levels.

A. Algorithms Used in our Experiment.

(i) Advanced Encryption Standard (AES). The Advanced Encryption Standard (AES) algorithm, created in 1998 by Joan Daemen and Vincent Rijmen, is a symmetric key block cipher. It supports data and key lengths of 128, 192, and 256 bits. AES operates on a 128-bit data length, organized into four operational blocks as an array of bytes arranged in a 4×4 matrix known as the state. These blocks undergo various transformations during rounds. The number of rounds (N) for full encryption varies: 10 rounds for a key length of 128, 12 rounds for 192, and 14 rounds for 256. Each AES round employs permutation and substitution networks and is adaptable for hardware and software implementations (Priyadarshini *et al.*, 2016).

(ii) RSA Algorithm. Founded in 1977, RSA is a public key cryptosystem, an asymmetric cryptographic algorithm named after its founders Rivest, Shamir, and Adelman. It generates two keys: a public key for encryption and a private key for decryption. The RSA algorithm consists of three steps: key generation, encryption, and decryption. Key generation creates the keys used for data encryption and decryption, encryption transforms plaintext into ciphertext, and decryption reverses this process. RSA relies on the factoring problem, finding the product of two large prime numbers. Key sizes range from 1024 to 4096 bits (Priyadarshini *et al.*, 2016).

METHODS

This study adopted the mixed methodology that comprises SSADM and the experimental method. The Structured Systems Analysis and Design Methodology was used to define, ideate, and prototype our solution, while the experimental method involved the investigation and comparison of the variable's impacts based on various parameters to obtain solutions and analyse findings based on experiment outcomes.

A. Evaluation Parameters

It's critical to assess the encryption algorithms to make sure they can handle a range of data kinds, workloads, and usage scenarios with efficiency, scalability, and practicality. The behaviour and results of the algorithm were measured in this study using empirical approaches that involved real-world experiments or simulations, giving researchers direct feedback and insights into the functioning and performance of the system. Python 3.12 was utilized for implementation, enabling the testing of different data types and sizes for encryption and decryption techniques. The following are a few metrics that are used to evaluate performance:

(i) Encryption Time. Encryption time is the time that an encryption algorithm takes to produce a cipher text from a plain text. We measured the encryption time in seconds in our experiment. The system's performance is affected by the encryption time. For the system to be fast and responsive, the encryption time must be less.

(ii) Decryption Time. The time taken to produce plain text from cipher text is called decryption time. To improve system responsiveness and speed, the decryption time should be less. System performance is impacted by the decryption time. We measured the decryption time in seconds in our experiment.

(iii) Avalanche Effect. This is a desirable feature of cryptographic algorithms, where a change in a single bit of ciphertext should lead to changes in multiple bits of the plaintext message, or vice versa. To calculate it: Avalanche Effect = (Number of Changed Bits in Ciphertext) / (Number of Bits in Ciphertext). A robust cipher should always achieve an avalanche effect greater than 50%.

RESULTS AND DISCUSSION

In this section, we discussed the results obtained based on three parameters.

Experiment/Case Study 1. The encryption time for each of the different techniques is shown in Table 1.

Table 1: Encryption time result (seconds).

File	File Size	AES Algorithm	RSA Algorithm	SHIELD Model
File 1	23kb	0.126	0.263	0.115
File 2	27kb	0.14	0.315	0.116
File 3	40kb	0.2	0.441	0.176
File 4	62kb	0.367	0.65	0.315
File 5	100kb	0.511	1.081	0.502
File 6	117kb	0.625	1.25	0.549
File 7	120kb	0.615	1.355	0.582
File 8	128kb	0.643	1.315	0.581
File 9	138kb	0.683	1.464	0.616
File 10	160kb	0.782	1.719	0.715
File 11	200kb	1.523	2.049	0.965
File 12	300kb	1.5	3.099	1.388
File 13	500kb	2.348	5.182	2.332
File 14	893kb	4.565	9.232	4.318
File 15	1MB	5.016	10.381	4.966
File 16	1.17MB	5.965	12.781	5.699
File 17	1.5MB	7.894	17.53	7.532
File 18	1.75MB	9.238	18.348	8.599
File 19	2MB	10.783	21.266	10.45
File 20	5MB	26.966	52.205	25.97
File 21	6.8MB	34.383	74.847	33.967
File 22	10MB	68.602	124.797	52.335
File 23	50MB	285.11	510.284	282.932
File 24	100MB	638.726	1009.974	618.437

Experiment / Case Study 2.

Table 2 displays the decryption time for all the various techniques.

Table 2: Decryption time result (seconds).

File	File Size	AES Algorithm	RSA Algorithm	SHIELD Model
File 1	23kb	0.198	0.786	0.115
File 2	27kb	0.269	0.796	0.129
File 3	40kb	0.197	1.247	0.181
File 4	62kb	0.321	1.797	0.286
File 5	100kb	0.514	2.798	0.276
File 6	117kb	0.58	3.283	0.289
File 7	120kb	0.844	3.365	0.306
File 8	128kb	0.59	3.547	0.306
File 9	138kb	0.636	3.829	0.298
File 10	160kb	0.756	4.433	0.345
File 11	200kb	0.916	5.481	0.332
File 12	300kb	1.389	8.448	0.332
File 13	500kb	2.525	13.616	0.649
File 14	893kb	4.801	24.332	1.656
File 15	1MB	4.982	28.55	1.415
File 16	1.17MB	6.282	32.655	2.899
File 17	1.5MB	8.582	43.514	2.849
File 18	1.75MB	9.382	49.398	4.683
File 19	2MB	10.833	55.365	5.399
File 20	5MB	31.098	144.951	17.172
File 21	6.8MB	36.764	186.415	22.565
File 22	10MB	60.589	308.599	46.381
File 23	50MB	284.066	1659.753	216.999
File 24	100MB	608.262	2806.451	443.166

Experiment / Case Study 3.

Table 3: Avalanche Effect (%).

File	File Size	AES Algorithm		RSA Algorithm		SHIELD Model	
		En%	De%	En%	De%	En%	De%
File 1	23kb	99.28	99.91	99.26	99.5	99.98	99.98
File 2	27kb	99.58	99.92	99.64	99.44	99.97	99.58
File 3	40kb	99.76	99.01	99.42	99.86	99.96	99.76
File 4	62kb	99.91	99.11	99.62	99.34	99.41	99.41
File 5	100kb	99.43	99.06	99.15	99.33	99.78	99.78
File 6	117kb	99.55	99.91	99.91	99.39	99.59	99.55
File 7	120kb	99.7	99.99	99.55	99.36	99.44	99.7
File 8	128kb	99.98	99.01	99.09	99.72	99.17	99.98
File 9	138kb	99.96	99.05	99.83	99.47	99.13	99.96
File 10	160kb	99.11	99.04	99.04	99.49	99.67	99.91
File 11	200kb	99.99	99.02	99.86	99.88	99.69	99.99
File 12	300kb	99.13	99.02	99.12	99	99.74	99.13
File 13	500kb	99.92	99.99	99.63	99.91	99.4	99.92
File 14	893kb	99.85	99.01	99.14	100	99.02	99.02
File 15	1MB	99.91	99.02	99.12	99.03	99.97	99.91
File 16	1.17MB	99.08	99.99	99.76	99.99	99.98	99.08
File 17	1.5MB	99.85	99.01	99.99	99.73	99.98	99.85
File 18	1.75MB	99.97	99.99	99.04	99.83	99.81	99.97
File 19	2MB	99.77	100	99.01	99.99	99.94	99.77
File 20	5MB	99.07	100	99.07	99.84	99.02	99.02
File 21	6.8MB	99.98	99.02	99.99	99.9	99.93	99.93
File 22	10MB	99	100	99.91	99.87	99.9	99
File 23	50MB	99.02	100	99.93	99.84	99.02	99.02
File 24	100MB	99	99	99.82	99.73	99.98	99.9

Analysis of the Results. Table 4 presents a comparison of the performance evaluation among all the encryption methods used in the experiments. The result analysis is based on the average encryption time, average decryption time and average avalanche effect. The files used in the tests vary from 23kb to 100mb. Graphs of both the algorithms along with their tables are shown below.

Table 4: Performance Comparison among all the encryption methods for encryption and decryption time.

Algorithms	Average Encryption time (seconds)	Average Decryption time (seconds)	Average Avalanche effect (%).	
			Enc	Dec
AES	46.138	44.807	99.574	99.46
RSA	78.409	224.725	99.495	99.64
SHIELD	44.339	32.043	99.646	99.68

(i) Analysis and evaluation based on Encryption Time. The time taken to execute SHIELD encryption is shown in a graphic form in Fig. 1. The encryption time is displayed on the Y axis in seconds, and the X axis displays algorithms. The encryption time for AES was 46.138, that of RSA was 78.409, and SHIELD has an encryption time of 44.339. SHIELD has a lower encryption time compared to both AES and RSA. So, in terms of encryption time, SHIELD shows approximately a 3.90% improvement compared to AES and approximately a 43.45% improvement compared to RSA. Therefore, SHIELD shows an average encryption time improvement of approximately 28.8% over the combined average encryption times of AES and RSA. This single percentage represents the overall improvement in encryption time achieved by SHIELD compared to using AES and RSA separately.

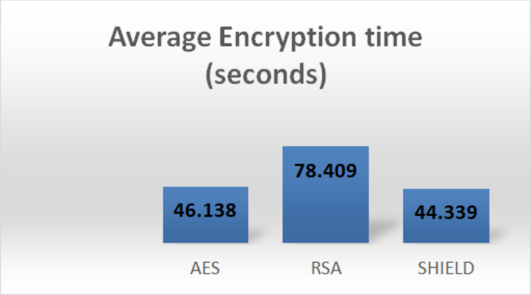

Fig. 1. Encryption time of existing methods and SHIELD model.

(ii) Analysis and evaluation based on Decryption Time. The time taken to execute SHIELD decryption is shown in a graphic form in Figure 2. The decryption time is displayed on the Y axis in seconds, and the X axis displays the algorithms. The decryption time for AES was 44.807, that of RSA was 224.725, and SHIELD has a decryption time of 32.043 indicating that SHIELD has a significantly lower decryption time compared to both AES and RSA. So, in terms of decryption time, SHIELD shows approximately a 28.49% improvement compared to AES and approximately an 85.76% improvement compared to RSA. Therefore, SHIELD shows an average decryption time improvement of approximately 76.2% over the combined average decryption times of AES and RSA. This single percentage represents the overall improvement in decryption time achieved by SHIELD compared to using AES and RSA separately.

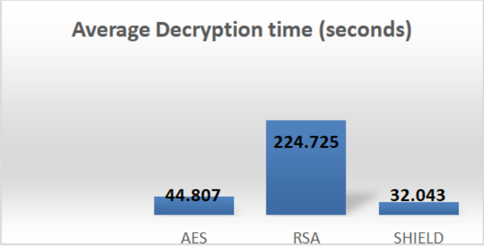

Fig. 2. Decryption time of existing methods and SHIELD model.

(iii) Analysis and Evaluation Based on Average Avalanche effect (%). A minimum of 50% avalanche effect is required to classify the algorithm as very secure. The avalanche effect as shown in a graphical representation in figure 3 indicates that all the three methods (AES, RSA, SHIELD) show very similar avalanche effects for encryption, with SHIELD slightly higher than AES and RSA, indicating a strong and comparable level of security in terms of the avalanche effect. The model achieved an Avalanche effect exceeding 50% which shows it achieved the desired level of unpredictability in the output ciphertext to provide strong encryption as compared to AES and RSA.

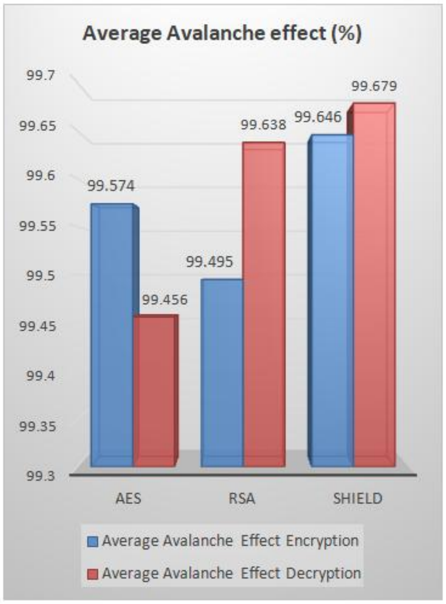

Fig. 3. Average Avalanche effect (%) of existing methods and SHIELD model.

CONCLUSIONS

The experiment was conducted by running files ranging in sizes from 23kb to 100MB using three algorithms: AES, RSA and SHEILD. The overall analysis of these algorithms was assessed using parameters like encryption time, and decryption time. A key feature of this model was its ease of use and stability with large amounts of data; for instance, a larger-sized .txt file (100 MB) used for dump files was accurately encrypted and decrypted. However, it was observed that encrypting larger files took a considerable amount of time due to the size and complexity of the data. Consequently, smaller file sizes resulted in shorter encryption and decryption times, while larger file sizes (100 MB and above) led to longer times. Encryption and decryption times have all been assessed for the SHIELD model using various data file sizes and formats. It proved efficient in encrypting all file formats, including text, images, and videos. Our solution made use of the shortest time in seconds for encryption and decryption. Hence, compared to AES and RSA, the SHIELD model offers confidentiality and better results in terms of security, encryption and decryption times. The study's findings indicated that by combining symmetric and asymmetric key techniques, concerns about the integrity, security, privacy, and confidentiality of data stored in the cloud are addressed by the new security system.

The SHIELD model demonstrated strong performance by integrating AES for fast and secure data encryption with RSA for robust key management, further enhanced by PKCS1_OAEP padding and multithreading. It excelled in throughput, decryption time, memory usage, and CPU/network efficiency, proving stable and effective across diverse file formats and sizes up to 500 MB. The model ensured high confidentiality, integrity, and resistance to cryptanalytic attacks, while offering ease of use and reduced processing costs. However, its performance was influenced by the underlying hardware, with optimal results achieved on higher-end processors such as Core i7. Large files, especially video data, introduced longer encryption/decryption times and potential memory limitations, making the approach less suited for resource-constrained environments or extremely large datasets.

FUTURE SCOPE

Future research should focus on testing the SHIELD model with major public cloud service providers such as AWS, Azure, and Google Cloud Platform to assess compatibility, performance, and cross-platform effectiveness. Security and privacy enhancements should be pursued through penetration testing, vulnerability analysis, and compliance checks. Performance evaluation should extend to real-world user scenarios, larger datasets, and advanced simulations on higher system configurations to determine scalability and efficiency. Comparative testing against established benchmarks and across different hardware environments will provide deeper insights into its practical applicability and potential for optimization.

Acknowledgement. We extend our gratitude to the Faculty of Computing, National Open University of Nigeria, for their invaluable support and contributions to this study. Their expertise and assistance were instrumental in the completion of this research.

Conflict of Interest. None.

REFERENCES

Almalawi, A., Hassan, S., Fahad, A. & Khan, A. (2024). A hybrid cryptographic mechanism for secure data transmission in edge AI networks. *International Journal of Computational Intelligence Systems, 17,* Article 44196-024-00417-8.

Anjana (2024). An enhanced three layer cryptographic algorithm for cloud information security. *International Journal of Intelligent Systems and Applications in Engineering, 12*(17s), 615–627.

Binita, T. & Blessy, T. (2022). An approach for enhancing the security of data over cloud using the multilevel algorithm. In *Congress on Intelligent Systems* (pp. 305–318). Springer.

Chatterjee, P., Bose, R., Banerjee, S. & Roy, S. (2023). Enhancing data security of cloud-based LMS. *Wireless Personal Communications.*

Hossain, M. A., Ullah, A., Khan, N. I. & Alam, M. F. (2019). Design and development of a novel symmetric algorithm for enhancing data security in cloud computing. *Journal of Information Security, 10*(4), 199–236.

Pandey, A. & Sharma, S**.** (2017). Hybrid Encryption Technique for Security of Cloud Data. *International Journal of Theoretical & Applied Sciences*, (Research Trend) *9*(2), 283–287.

Priyadarshini, P., Prashant, N. B., Narayan, D. G. & Meena, S. M**.** (2016). A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish. *Procedia Computer Science*, *78,* 617–624.

Ranganatha, B. R. & Sujatha, B. (2023). A hybrid elliptic curve cryptography (HECC) technique for fast encryption of data for public cloud security. *Measurement: Sensors*, *29*, 100870.

Rath, S. B., Priyadarshini, S. B., Patel, D. K., Sahu, P. K., Jagadev, N., Panda, M., Patra, N. & Sahoo, S. (2024). AES-RSA: An innovative hybrid security framework for file authentication, integrity, and data secrecy model. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(18s), 303–312.

Riyaz Fathima Abdul & Saravanan, A. (2025). A novel data transmission model using hybrid encryption scheme for preserving data integrity. *Advances in Technology Innovation, 10*(1), 15–28.

Sabitha, R., Shaik, J. S., Karthik, S. & Kavitha, M. S. (2023). Secure data storage on cloud using hybrid cryptography methods. *Easy Chair Preprint,* 10126.

Shukla, D. K., Khalaf, O., Vallabhaneni, R., Srivastava, S. K. & Algburi, S. (2024). A three-phase hybrid cryptography algorithm: Utilized in public sensor network for data security with an enhancement of hashing algorithm. *International Journal of Computing and Digital Systems, 15*(1), 1–13.

Stergiou, C. L., Plageras, A. P., Psannis, K. E. & Gupta, B. B. (2020). Secure machine learning scenario from big data in cloud computing via Internet of Things network. In B. Gupta, G. Perez, D. Agrawal, & D. Gupta (Eds.), *Handbook of computer networks and cyber security* (pp. 593–621). Springer.

Wang, S., Zhang, Y. & Zhang, Y. (2018). A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. *IEEE Access, 6,* 38437–38450.

Wang, Z., Wang, N., Su, X. & Ge, S. (2020). An empirical study on business analytics affordances enhancing the management of cloud computing data security. *International Journal of Information Management, 50,* 387–394.

William, P., Choubey, A., Chhabra, G. S., Bhattacharya, R., Vengatesan, K. & Choubey, S. (2022). Assessment of hybrid cryptographic algorithm for secure sharing of textual and pictorial content. In *2022 International Conference on Electronics and Renewable Systems (ICEARS)* (pp. 918–922). IEEE.

How to cite this article: A.R. Atuluku and F.B. Osang (2025). Performance Evaluation of the Secure and Integrated Encryption Model (SHIELD). *International Journal on Emerging Technologies, 16*(2): 113–119.