International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 13 Issue 12, December-2024

A comparative analysis on RSA and Blowfish cryptography algorithm, based on their computational performance, encryption scalability, and avalanche effect

Parv Sagar Jain[1], Somyak Agarwal[2]

[1,2]Grade 12 IBDP Students Jayshree Periwal International School, Jaipur Rajasthan India

ABSTRACT

The performance of the RSA and Blowfish encryption algorithms is examined and contrasted in this paper using three important metrics: the Avalanche Effect, encryption scalability, and computing performance. While encryption scalability was tested using datasets of increasing sizes to assess how well the algorithms handle large-scale data, computational performance was examined based on encryption and decryption times and memory utilization. The Avalanche Effect evaluated the diffusion capabilities of each algorithm by measuring how sensitive it was to slight modifications in the plaintext. The results demonstrate that Blowfish is the best option for real-time encryption of big datasets due to its exceptional speed and scalability. RSA, on the other hand, is better in diffusion and security, which makes it appropriate for important, smaller-scale encryption jobs. These results highlight how crucial it is to use encryption algorithms [1] that are suited to the demands of certain applications. Future research might concentrate on hybrid techniques and modifications for cryptography that is immune to quantum errors.

INTRODUCTION:

"Without strong encryption, you will be spied on systematically by lots of people," remarked Whitfield Diffie, a pioneer of modern cryptography. Cryptography has become an essential part of a fast-moving world, an era marked by the growth of digital communications [1]. The increasing digital data exchange and the rising cyber threats have urged a growing need for cyber security provided by encryption techniques.

Cryptography is built upon several fundamental concepts essential for understanding its mechanisms and applications. A text or message refers to the data that needs to be transmitted over an insecure channel and is often called plaintext. It consists of symbols arranged in a particular order. When this data is encrypted, it is transformed into ciphertext, a secure representation of the message that can be transmitted over untrusted channels. The ciphertext is designed so that even if intercepted by an adversary, it cannot be deciphered into its original plaintext without the appropriate decryption key.

Encryption is the process of converting readable data, referred to as plaintext, into an unreadable format known as ciphertext. This transformation ensures that the data remains secure and inaccessible to unauthorized entities during transmission or storage. The process involves the use of a cryptographic key, which is a specific value used by the encryption algorithm to encode the plaintext. Decryption is the reverse operation, where the ciphertext is converted back into plaintext using the appropriate key, allowing the original information to be retrieved. These processes form the basis of secure communication and data protection. Encryption algorithms are broadly classified into two types: symmetric encryption, where the same key is used for both encryption and decryption, and asymmetric encryption, which uses a pair of keys—a public key for encryption and a private key for decryption.

Cryptography is indispensable in securing information transferred across untrusted communication channels. These channels, prone to interception and unauthorized access, necessitate robust encryption mechanisms to ensure the confidentiality and integrity of transmitted messages. Understanding these fundamental concepts is crucial for exploring the advanced methods and applications of cryptographic systems.

For this research, the selected algorithms are RSA and Blowfish which would be evaluated on metrics such as computational performance, encryption scalability, and the avalanche effect.

LITERATURE REVIEW

Numerous research have thoroughly examined Blowfish's computational efficiency and performance. The high encryption and decryption speeds of Blowfish are highlighted in the publication A Comparative Study of Twofish, Blowfish, and Advanced Encryption Standard, underscoring its appropriateness for lightweight applications that demand rapid data processing. The study highlights the notable time and resource advantages of symmetric algorithms like Blowfish over asymmetric ones like RSA [1]. Blowfish's efficiency is indirectly supported by the publication Performance Analysis of AES and Twofish Encryption Schemes, which emphasizes the symmetric encryption's intrinsic speed. It supports Blowfish's use in computationally limited circumstances by attributing RSA's slower performance to the computational difficulty of its asymmetric design [3].

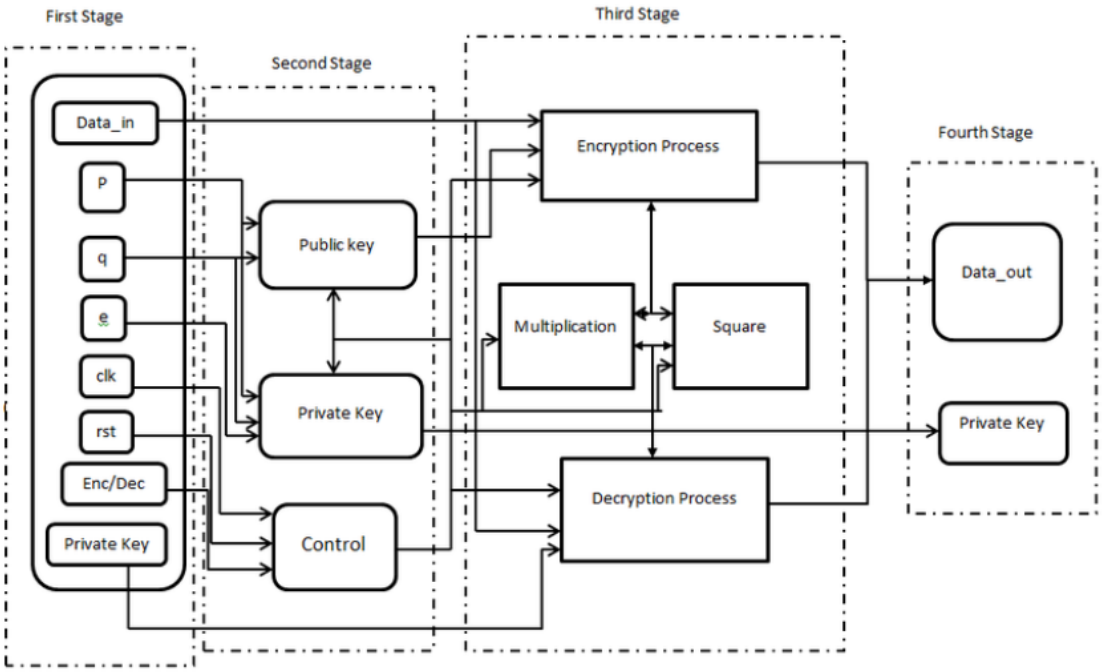
Evolution of AES, Blowfish, and Twofish Encryption Algorithms delves deeper into the evolution and flexibility of Blowfish, emphasizing its small S-boxes and Feistel structure, which let it run quickly and use resources efficiently. In contrast to RSA's computationally demanding nature, the article highlights Blowfish's scalability with various key sizes, making it a flexible option for real-time encryption jobs [9]. Furthermore, Blowfish's lightweight architecture complements the results of Implementation of AES Cryptography and Twofish Hybrid Algorithms for Cloud, which meet the need for scalability and efficiency in cloud contexts. Although this study focuses on AES and Twofish, the findings highlight the advantages of Blowfish over RSA for cloud encryption [7].

Lastly, a direct comparison of Blowfish with AES and Twofish may be found in Comparison of Encryption Algorithms: AES, Blowfish, and Twofish for Security of Wireless Networks. According to the study, Blowfish is perfect for resource-constrained applications like wireless networks because of its exceptional speed and minimal resource use. The study also highlights Blowfish's potent Avalanche Effect, which improves security without sacrificing speed and gives it a definite edge over RSA in terms of scalability and performance [8].

Background:

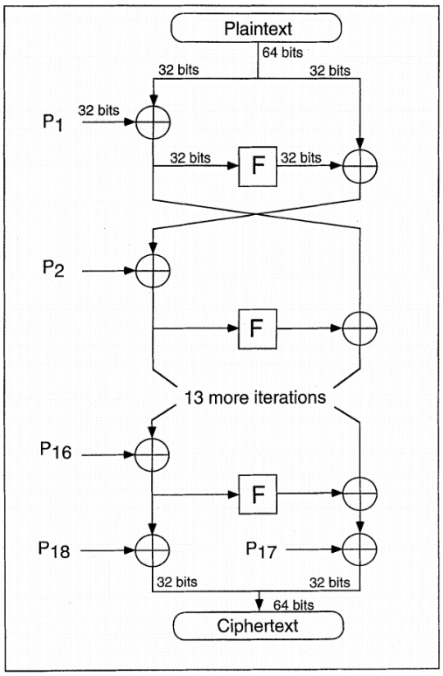
RSA:

The RSA algorithm was invented in 1977 and named after the surnames of its three founders– Ron Riverest, Adi Shamir, and Leonard Adleman at MIT [2]. Addressing the problem of encryption and security of communication in the digital age [5], the RSA algorithm became familiar to multiple users across the globe, making it the first widely adopted public-key encryption. RSA is based on the mathematical challenge of factoring large prime numbers, that remain computationally infeasible for sufficient large keys, making the technique of RSA the cornerstone of modern cryptography. The RSA technique is rather complex and, therefore, used in places where security is most important. It is famous for its robustness and, therefore, used for secure digital signatures and key exchanges.



The process in RSA begins with key generation, where two different keys—r and p—are created. The foundation of RSA stands on two different keys: The public key—used for encryption, and the private key—used for decryption. The prime keys—p and q—are multiplied to create a module n to serve as a commute for both keys. After the creation of the keys, the totients–all the numbers that are not a factor of the number but rather are smaller than the number–are calculated. The totient $\varphi(n) = (p-1)(q-1)$ is calculated next. For the public key, an integer 'e' is selected such that it satisfies $1 < e < \varphi(n)\ and\ GDC\big(e, \varphi(n)\big) = 1$. On the other hand, the private key is determined by the integer 'd', the modular multiplicative reciprocal of e modulo $\varphi(n)$. For the encryption process, the initial message 'm' is transformed into ciphertext 'c' using the public key using the formula $c = m^e \times |n|$. Decryption, however, is accomplished by computing the original message 'v' using the ciphertext 'c' using the private key: $v = c^d \times |n|$. This structure helps RSA to transmit the messages securely over insecure channels by employing encryption and decryption keys.

Blowfish:
Bruce Schneier, a well-known cryptographer, developed the blowfish approach in 1993 as a quick and safe symmetric encryption substitute for the outdated Data Encryption Standard (DES). Blowfish's simplicity, speed, and adaptability to many hardware and software platforms quickly made it well-known among other cryptographers [3]. The Blowfish approach is mostly utilized in resource-constrained settings, such mobile applications and Internet of Things devices. It is widely used and the greatest option for applications needing quick encryption, such as wireless networks, because it is significantly quicker and easier to use [8].



Blowfish operates on fixed-size blocks of data, specifically 64 bits. This algorithm begins with the key expansion, accepting all the variables ranging from 32 bits to 448 bits. The key in the Blowfish algorithm is used to create 18 more subkeys that are further stored in an array called 'P-array,' along with four 'S-boxes,' each containing 256 entries of 32 bits. During the encryption, the original data is split into two halves of 32 bits each, referred to as left(L) and right. The encryption process consists of 16 rounds where in each round, the left half L is XORed with a subkey from the P-array, while the right half R transforms a function F(L), utilizing the S-boxes. After processing through all rounds, a final transformation combines the two halves to produce the ciphertext. Decryption mirrors the encryption process but uses the P-array in reverse order, allowing Blowfish to decrypt data with the same keys used for encryption efficiently.

METHODOLOGY:

This study employs a structured approach to evaluate and compare the performance of RSA and Blowfish encryption algorithms across three critical metrics: computational performance, encryption scalability, and avalanche effect.

Dataset:
To equalize the influence of computational performance, a dataset varying from 1 KB to 5120 KB (5 MB) was used to test the encryption and decryption times, memory usage, and scalability. The plaintext dataset consists of randomly generated text files. The test of the avalanche effect involved creating ten small plaintexts by making small, incremental changes to an original text, such as altering a single letter.

Algorithm implementation:
RSA and Blowfish were implemented using the Python library," PyCryptodome." RSA uses 2048-bit keys, and Blowfish uses an 8-byte symmetric key.

Metric Evaluation:

1. **Computational Performance:** Time for encryption and decryption was measured using Python's time library, memory usage with trace malloc , and scalability was assessed by observing time variations as data size increased [5].

2. **Avalanche Effect:** Ciphertexts generated from original and modified plaintexts were compared bit-by-bit to calculate the percentage of differing bits, indicating the sensitivity of each algorithm to input changes.

3. **Security Resilience:** The Avalanche Effect was used as a proxy for evaluating security, with higher values indicating better diffusion.

Case 1 Computational Performance:
RSA:

Table 1

Encryption times (seconds)	Decryption times (seconds)
0.006997585296630	0.038727521896362
0.006511926651000	0.045619010925292
0.006998538970947	0.043354511260986
0.006611824035644	0.045294523239135
0.008998394012451	0.045454263687133
0.007869720458984	0.036142110824584
0.006000757217407	0.036463737487792
0.006503582000732	0.036893844604492
0.007002592086791	0.040827989578247
0.006440162658691	0.036343812942504

Table 2

Memory usage (KB)
22.9501953125
8.5966796875
8.5966796875
8.5927734375
8.5966796875
8.5966796875
8.5966796875
8.5966796875
8.5966796875
8.5966796875

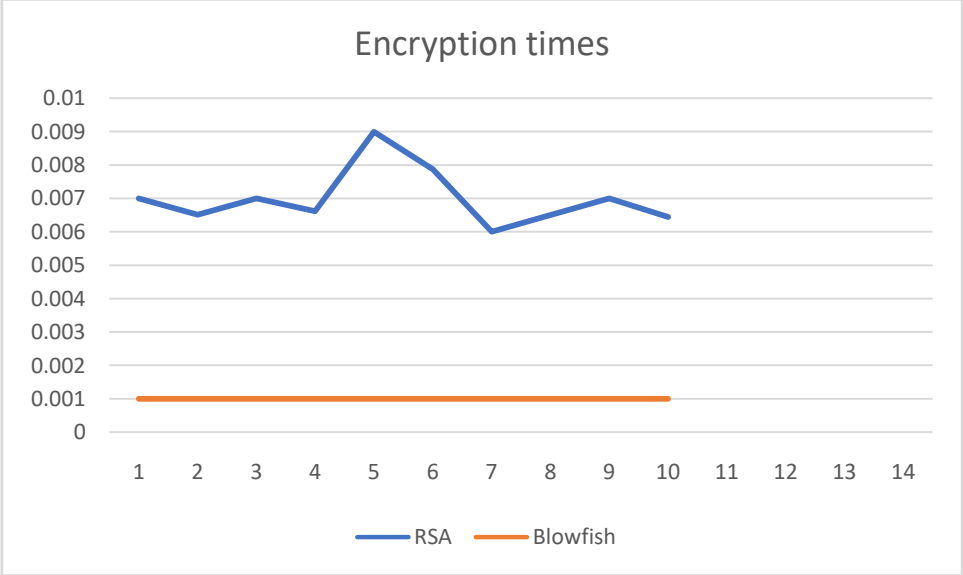
Table 3

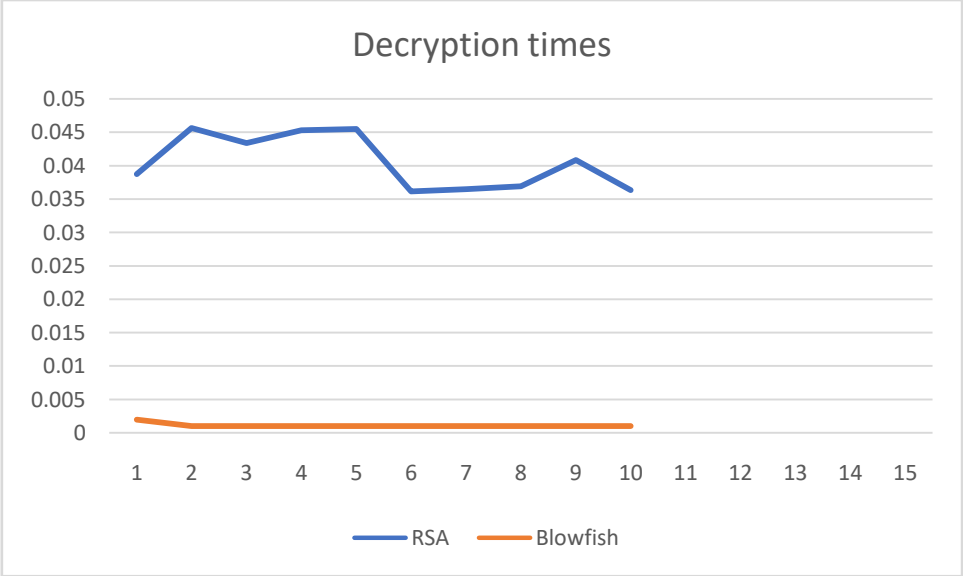
Blowfish:

Encryption times (seconds)	Decryption times (seconds)
0.001000165939331	0.001999855041562
0.001000165939331	0.001000404357910
0.000999689102172	0.000999927520751
0.000999927520751	0.001000165939331
0.000999927520751	0.001003980636596
0.000999927520751	0.000999927520751
0.000999927520751	0.000999927520751
0.001000165939331	0.001000165939331
0.000999689102172	0.000999927520751
0.001000404357910	0.000999927520751

Table 3

Memory usage (KB)
2.060546875
1.0244140625
1.0244140625
1.0244140625
1.0244140625
1.0244140625
1.0244140625
1.0244140625
1.0244140625
1.0244140625

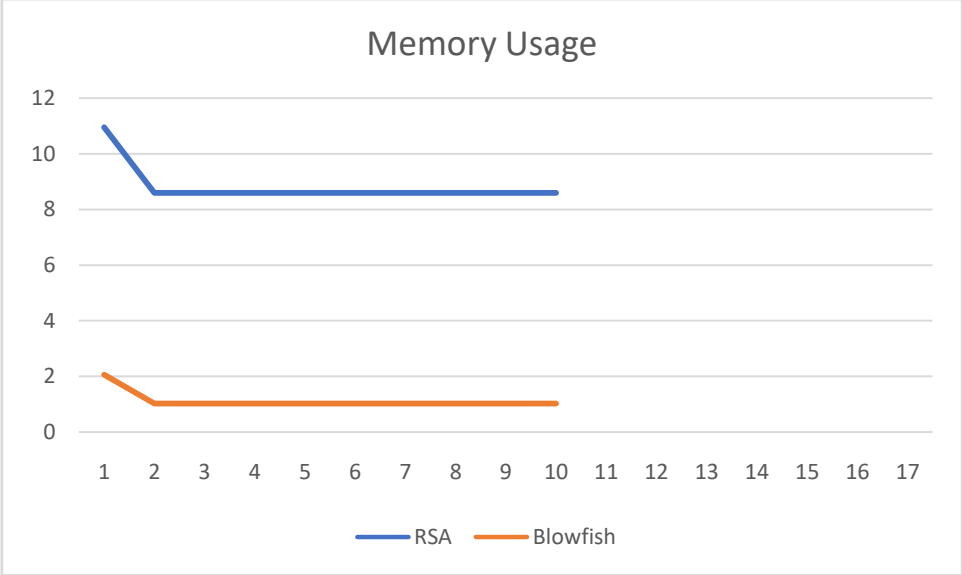




The differences in encryption and decryption times between RSA and Blowfish is due to the differences in their structures, highlighting their computational power and operational efficiency [7]. RSA is an asymmetric algorithm that relies on public private key pair and computationally intensive operations like modular exponential. Hence this results in RSA's high encryption time average of 0.006994 seconds, slower than Blowfish. Similarly, RSA decryption averages 0.040512 seconds, reflecting the higher computational complexity which is required to achieve strong security. RSA's slower performance is a trade-off for its suitability in securing small pieces of important data, such as encrypting digital signatures, rather than large datasets.

On the other hand, Blowfish is a symmetric algorithm designed for speed and efficiency. Its lightweight Feistel structure make it exceptionally fast. Leading to average encryption time of 0.001000 seconds, and almost identical average decryption time of 0.001100 seconds due to the symmetric nature of the algorithm. These results reflect Blowfish's suitability for high-throughput applications where large datasets need to be encrypted and decrypted with minimum computational resources.

The disparity in performance between RSA and Blowfish stems from their differing purposes. RSA prioritizes security over speed, making it ideal for securing small data or key exchanges, while Blowfish optimizes for fast bulk encryption with lower computational requirements. This demonstrates that Blowfish is significantly faster, particularly for applications requiring rapid processing, whereas RSA is better suited for scenarios demanding robust cryptographic security at the expense of speed.



The memory usage results show a significant difference between RSA and Blowfish, reflecting their difference between their structures. RSA has an average memory usage of 10.03 KB, consuming more memory as compared to Blowfish, which uses an average of 1.13 KB. This differenceisdue to RSA's reliance on complex mathematical operations and the need to handle large key sizes, typically 2048 bits or more. RSA operations likethe modular exponential and key management, require significant memory, especially for storing the large key pairs and calculation results during encryption and decryption.

In contrast, Blowfish's memory efficiency is due to its symmetric nature and streamlined design. Blowfish operates on smaller block sizes and uses predefined subkeys and compact S-boxes, which significantly reduce memory requirements during encryption and decryption. Additionally, Blowfish's fixed key sizes and easy operations like XOR and substitution require minimal resources, making it well-suited for memory-constrained environments.

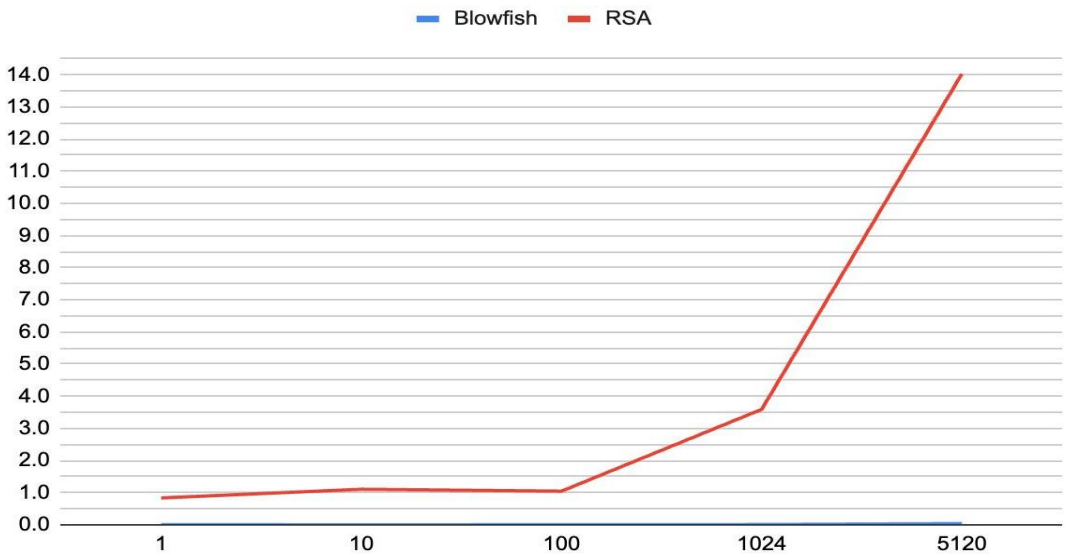
Case 2 Encryption Scalability:

RSA:

Data size (KB)	Time (seconds)
1 KB	0.8356 seconds
10 KB	1.1061 seconds
100 KB	1.0467 seconds
1024 KB	3.5932 seconds
5120 KB	14.0332 seconds

Blowfish:

Data size (KB)	Time (seconds)
1 KB	0.0014 seconds
10 KB	0.0001 seconds
100 KB	0.0010 seconds
1024 KB	0.0075 seconds
5120 KB	0.0367 seconds



The encryption scalability results highlight the algorithms' efficiency with increasing data size. For RSA, the time required for encryption increases significantly as data size increases, showing a steep rise in the graph. This is due to RSA's computationally intensive nature, which involves complex mathematical operations such as modular exponential. These operations scale poorly with larger data sizes, making RSA less suitable for bulk data encryption [7].

In contrast, Blowfish demonstrates exceptional scalability, with minimal increases in encryption time even as data sizes increases. This efficiency is due to its simple block cipher design, which processes fixed-size blocks of data in a streamlined manner, maintaining consistent performance regardless of data size. This characteristic makes Blowfish highly suitable for applications requiring the encryption of large datasets [8].

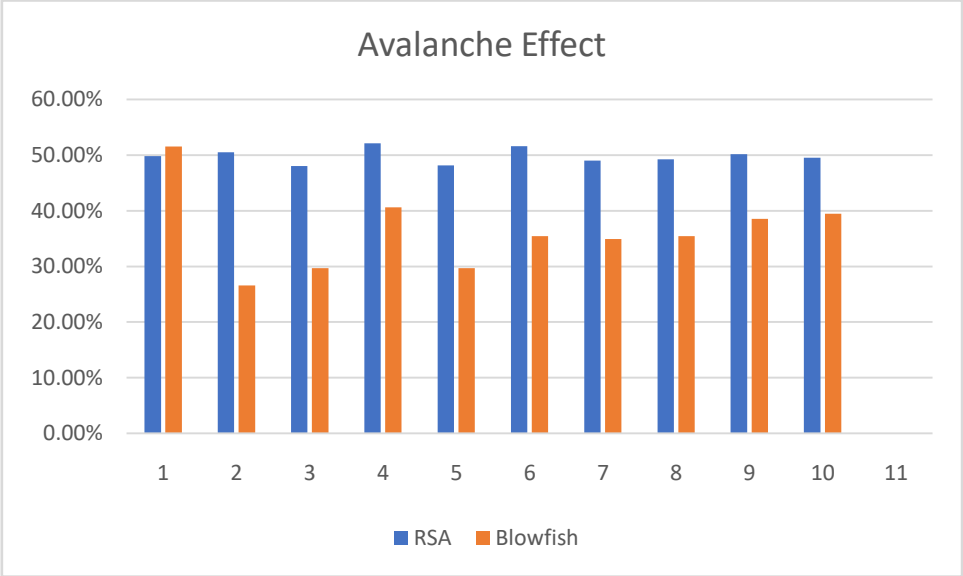
Case 3 Avalanche Effect:
RSA:

	Original Plaintext	RSA Ciphertext	Blowfish Avalanche effect (%)
1.	Computer Science is fun	4fe226e1618d388069c5b5acdfc8b8fa1b0a4b5f362b9	49.80%
2.	Compoter Sciense is fun	ba470df4bb7058a85b13337f2c0fdb3c0367f99e6b894	50.49%
3.	Computer Sciense is fin	49ef0d29dcf41bf1b7c436e906c7d729cdf8f62d900302	48.04%
4.	Compyter Science is fun	211853a1f3a879ffecdc874acae816d9aeaf7b20f82bca	52.10%
5.	Computer Sciemce is fun	118e6797f43e4042d29048d29f641cc97785747fff6583	48.14%
6.	Computer Science is fyn	36d9ff259021a6728fa2e4d06361fa8cc142c902aefbce	51.61%
7.	Comvuter Science is fun	243baabf1968bf4b95d45bd14610d4d77e3aab7f9d66d	49.02%
8.	Computer Scyence is fun	0c7eabda9357e41fcec8ddf0bc084c30329d9abe1a065d	49.22%
9.	Computer Science is fun!	7b4dea2d7bdcab42c356ff8d7a4d723fa0f6062d5d6837	50.19%
10.	Compuer Science is fun	bfc621927f2c887af8ab69e97283719eec1eecad6b977c	49.51%

RSA Average Avalanche Effect: 49.81%
Blowfish:

	Original Plaintext	Blowfish Ciphertext	Blowfish Avalanche effect (%)
1.	Computer Science is fun	fdb564929d9e3ef45fc88b1ae0c8872bc5858bf6de0c71	51.56%
2.	Compoter Sciense is fun	7c3e14ea6516dfddbff4cff2eecc164dc5858bf6de0c714	26.56%
3.	Computer Sciense is fin	fdb564929d9e3ef4bff4cff2eecc164d9bbf81ed1dcfa187	29.68%
4.	Compyter Science is fun	f0277c119090bc285fc88b1ae0c8872bc5858bf6de0c71	40.63%
5.	Computer Sciemce is fun	fdb564929d9e3ef410997d3d1095f52ec5858bf6de0c71	29.68%
6.	Computer Science is fyn	fdb564929d9e3ef45fc88b1ae0c8872bfed8e394fdb9dca	35.41%
7.	Comvuter Science is fun	fd1cfe1a31e351ab5fc88b1ae0c8872bc5858bf6de0c714	34.89%
8.	Computer Scyence is fun	fdb564929d9e3ef4acbe972563b122e8c5858bf6de0c71	35.41%
9.	Computer Science is fun!	fdb564929d9e3ef45fc88b1ae0c8872b6b3ac10bbbbe5d	38.54%
10.	Compuer Science is fun	292a278e81db97629d217eb5094d0ec5c914affe2f280a	39.45%

Blowfish Average Avalanche Effect: 36.18%



The Avalanche Effect results indicate a difference between RSA and Blowfish, highlighting the varying levels of sensitivity to minor changes in the plaintext. RSA demonstrates a higher average Avalanche Effect of 49.81%, while Blowfish shows a lower average of 36.18%.

RSA has a high Avalanche effect because of its complex operations, suggesting that even a small change in the plaintext significantly alters the ciphertext which is important for security [4]. This property makes RSA highly secure, as even the smallest change in the plaintext produces a completely different ciphertext which ensures robust cryptographic diffusion.

Blowfish's lower Avalanche Effect, averaging 36.18%, is due to its simple block cipher design and focus on speed. Although lower than RSA, Blowfish exhibits some degree of diffusion through its Feistel structure and lightweight operations, resulting in changes in less variation in the ciphertext as the plaintext changes [5].

Due to RSA's asymmetric nature, even a small change in the plaintext leads to a significant variation in the ciphertext, resulting in a higher Avalanche Effect. On the other hand, Blowfish is a symmetric algorithm which is optimised for speed and efficiency, explaining its lower Avalanche Effect. RSA is typically used for securing important data, where high diffusion and security are crucial, making its high Avalanche Effect advantageous. On the other hand, Blowfish is better suited for scenarios where rapid encryption and decryption is required, hence, a trade-off in diffusion is acceptable to achieve better performance.

EVALUATION

Critical insights into the unique features and applications of RSA and Blowfish are provided by this comparison study. Blowfish routinely beats RSA in terms of computational performance, demonstrating far faster encryption and decryption speeds as well as lower memory use. Because of its straightforward symmetric structure, Blowfish can handle data quickly and efficiently, which makes it ideal for applications that need to encrypt or decrypt data in real time. RSA is appropriate for situations where computational resources are not constrained, including secure key exchanges and digital signatures, but its computational complexity and dependence on larger keys lead to longer processing times and higher memory consumption [9].

Blowfish's merits are further shown by the encryption scalability results. Its scalability and versatility for handling big datasets are demonstrated by its capacity to manage growing data sizes with no effect on encryption time. However, when data size grows, RSA's encryption time increases sharply, indicating the limitations of its asymmetric architecture for encrypting large amounts of data. According to these findings, RSA is still the best option for encrypting little but extremely sensitive data, whereas Blowfish is the recommended option for applications involving large volumes of data.

The results from Avalanche effect highlight the security aspect. RSA demonstrates a higher Avalanche Effect, with an average of 49.81%, compared to Blowfish's 36.18%. This indicates that RSA is more sensitive to small changes in plaintext, resulting in greater diffusion in the ciphertext which is an important property for securing critical data.

Blowfish, does not show the same level of diffusion, but balances security with performance which makes it suitable for applications where speed and efficiency are prioritized over extreme sensitivity to plaintext changes.

CONCLUSION

In conclusion, the particular needs of the application will determine whether to use RSA or Blowfish [4]. For situations where efficiency is crucial, such encrypting big datasets or guaranteeing real-time processing, Blowfish is incredibly effective. On the other hand, RSA is better at protecting smaller but extremely sensitive data, such cryptographic keys and secure conversations, because to its superior diffusion and robust security.

FUTURE SCOPE

Future studies can focus on optimizing RSA to handle massive data more effectively and enhancing Blowfish to make it more secure without sacrificing speed. It would be advantageous to test these algorithms on various data kinds, such as multimedia or real-time data, and get them ready for upcoming technologies like quantum computing. These innovations demonstrate how encryption techniques may be enhanced to satisfy growing cybersecurity requirements in domains such as cloud computing, secure messaging, and the Internet of Things, guaranteeing safer data in the digital realm.

REFERENCES

[1]. M. A. Al-Shabi, "(PDF) a survey on symmetric and asymmetric cryptography algorithms ininformationsecurity,"ResearchGate,https://www.researchgate.net/publication/332176079_A_Survey_on_Symmetric_and_Asymmetric_Cryptography_Algorithms_in_information_Security.

[2]. D. Mahto and D. K. YADAV, "(PDF) RSA and ECC: A comparative analysis," Research Gate,https://www.researchgate.net/publication/322558426_RSA_and_ECC_A_comparative_analysis .

[3]. S. Zeeshan, "Performance analysis of AES and Twofish Encryption Schemes | Request PDF,"ResearchGate,https://www.researchgate.net/publication/224250326_Performance_Analysis_of_AES_and_TwoFish_Encryption_Schemes.

[4]. V. P. Bansal and S. Singh, "A hybrid data encryption technique using RSA and Blowfish for Cloud Computing on fpgas | IEEE conference publication | IEEE xplore," Research Gate, https://ieeexplore.ieee.org/document/7453367/ .

[5]. P. Patil, P. Narayankar , N. D.G., and M. S.M., "A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish," Procedia Computer Science, https://www.sciencedirect.com/science/article/pii/S1877050916001101 .

[6]. K. A. Agyei, "[PDF] A Comparative Study of twofish, Blowfish, and Advanced Encryption Standard for secured data transmission semantic scholar," semantic scholar, https://www.semanticscholar.org/paper/A-Comparative-Study-of-Twofish,-Blowfish,-and-for-Assa-Agyei-Olajide/e63b9d87591e284a3a8007c74646acc2eb115396 .

[7]. K. Santoso, "(PDF) implementation of AES cryptography and Twofish Hybrid algorithms forcloud,"ResearchGate,https://www.researchgate.net/publication/341729844_Implementation_of_AES_cryptography_and_twofish_hybrid_algorithms_for_cloud .

[8]. A. Ghosh, "(PDF) comparison of encryption algorithms: AES, Blowfish and Twofish for SecurityWirelessNetworks,"ResearchGate,https://www.researchgate.net/publication/342764235_Comparison_of_Encryption_Algorithms_AES_Blowfish_and_Twofish_for_Security_of_Wireless_Networks.

[9]. E. Jeevalatha and S. S. Murugan, "Evolution of AES, Blowfish and two fish encryption algorithm," IJSER, https://www.ijser.org/researchpaper/Evolution-of-AES-Blowfish-and-Two-fish-Encryption-Algorithm.pdf .