INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

ORIGINAL ARTICLE

Intelligent Key Cryptography (IKC) using Tamil Unicode and Temporal Time: A Research Perspective Analysis

V Rojasree[1]*, J Gnana Jayanthi[1]

1 Department of Computer Science, Rajah Serfoji Govt. College (A), (Affiliated to Bharathidasan University), Thanjavur-613005, Tamil Nadu, India

OPEN ACCESS

Received: 20/02/2025

Accepted: 28/04/2025

Published: 29/06/2025

Citation: Rojasree V, Gnana Jayanthi J (2025) Intelligent Key Cryptography (IKC) using Tamil Unicode and Temporal Time: A Research Perspective Analysis. Indian Journal of Science and Technology 18(24): 1953-1960. https://doi.org/10.17485/IJST/v18i24.339

* **Corresponding author**.

rojasree.v@gmail.com

Funding: None

Competing Interests: None

Copyright: © 2025 Rojasree & Gnana Jayanthi. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: To prove the qualitative competency of using Temporal Time and Tamil Unicode as the strength of Intelligent Key Cryptography (IKC) in the environment of quantum cryptographic threats by mathematically comparing the efficacy of IKC with the currently used cryptographic algorithms. **Methods:** This paper is focussed on adopting methods that prove the strengths of IKC by evaluating the literature for the various quantum threats and then proving mathematically how IKC withstands those threats. The methods include the mathematical evaluation for (i) dynamical key management, (ii) multi-algorithm encryption, (iii) robustness of security with key entropy, (iv) resistance to quantum attacks raised by Shor's and Groover's algorithm, (v) comparing the performance and scalability of IKC with the currently used cryptographic algorithms like DES, Blowfish, and AES in terms of encryption time, memory used and avalanche effect. **Findings:** This study's core contribution lies in the rigorous mathematical validation of IKC's performance competencies. It highlights IKC's methodologies, results, and discussions, which prove IKC to be a novel finding with the strength of dynamicity introduced by Tamil Unicode and Tamil in IKC to prove a strong cryptographic algorithm in the quantum cryptographic environment. **Novelty:** This study's key novelty lies in the findings substantiating that IKC meets the essential criteria for resilient cryptographic architecture. It can withstand quantum computational attacks and establish itself as a robust, quantum-attack-resistant architecture.

Keywords: Post-Quantum Cryptography (PQC); Quantum Threats; Intelligent Key Cryptography (IKC); Cryptographic Algorithms; Quantum-Resistant Algorithms

1 Introduction

Quantum Computing algorithms have severely threatened the currently widely used cryptographic algorithms [(1)]. The presently used cryptographic algorithms, namely RSA and ECC, are considered strong with their mathematical procedures used in encryption and decryption, which were eventually broken by the invention of Shor's and Grover's algorithms [(2)]. These algorithms led to the invention of quantum computing, which

supported swift calculations, thus making difficult mathematical calculations easier. This threatened the world by preventing sensitive data from being secured. The potential eavesdropping and tampering are controlled by Quantum Key Distribution (QKD) used in Post-Quantum Cryptographic (PQC) algorithms[3]. *However, transitioning to these technologies will require a widespread update to the current cryptographic infrastructure.*

The global realization of quantum-based threats urges industries and governments to adopt quantum-safe solutions, which involves researching better encryption techniques by revising the protocols to incorporate long-term information security[4]. *Organizations should be vigilant about the progression of quantum technology and invest in solutions that safeguard against vulnerabilities.* The evolution of quantum computing stresses industries to rely on more secure communication and follow a quantum-resistant cryptographic algorithm to withstand quantum attacks. Despite potential solutions like QKD offered by quantum cryptography, real-world scalability is still challenging. *A quantum-powered future could be achieved only by global cooperation and tailoring a robust solution to such challenges*[5].

The challenges mentioned above are efficiently addressed by introducing the Tamil Unicode and temporal Time for randomizing the keys generated in the proposed Intelligent Key Cryptography (IKC). Section 2 presents a comprehensive review of recent literature that underscores the efficacy of the proposed approach. To further substantiate this, Section 3 provides a detailed mathematical analysis of IKC's computational strength and benchmarks IKC by evaluating IKC in the context of threats that challenge traditional cryptographic algorithms. The paper is concluded in section 4.

2 Quantum Cryptographic Threats

Quantum cryptography confirms secure communication by employing the principles of quantum computing to resist all computational attacks. The introduction of quantum computing induced cryptographic threats, which are analyzed deeply and briefed below.

1. **Shor's Algorithm:** Shor's algorithm efficiently solves problems with integer factorization and discrete logarithms. Classical cryptographic algorithms such as RSA, DSA, and ECC are vulnerable to this algorithm[6] because q*uantum computers that could run Shor's algorithm could easily decrypt messages secured by these algorithms.*
2. **Grover's Algorithm:** Grovers' algorithm threatens symmetric cryptographic systems by quadratic speed up in brute-force attacks[7]. *Even though this cannot break the symmetric encryption, increasing the key size is required. For example, the AES-128 would need to move to AES-256 to maintain the security guarantees.*
3. **Lattice-Based Attacks:** The strong resistance to quantum attacks of Lattice-based cryptography is a choice of PQC[8]. *However, recognizing its toughness to non-quantum attacks and enhancing its performance remains challenging for widespread implementation.*
4. **Hash-Based Cryptography:** The large vital sizes limit hash-based signatures to limited usage scenarios[9]. *Though hashing provides a robust solution, the practical application in general-purpose cryptography is minimal.*

The PQC is now a leading defence strategy, thus demanding new cryptographic schemes to offer secure and scalable solutions. Hence, to provide PQC and withstand the threats briefed in the literature review, the new IKC architecture was proposed, designed, developed, and well-accepted by the research forum. To substantiate this, the strength of IKC is evaluated in the context of systematic methodologies of IKC resilience against quantum threats with security highlights, as in section 3 below.

3 Results and Discussion - Systematic Methodologies of IKC Resilience against Quantum Threats

Research findings have proved that all classical cryptographic systems are vulnerable[1]. However, based on plenty of research studies, no remedies have been evaluated for overcoming the vulnerabilities of public key cryptosystems. Completely different and well-advanced techniques must be adopted to understand the complex operations of such complicated problems. IKC uses an entirely different technique of Tamil Grammar, which is implemented using Tamil Unicode and the temporal Time constant to achieve strength in overcoming quantum cryptographic threats[10]. The process was explained in detail in another publication.

IKC stands out in cryptographic research with its noteworthy features such as adaptability in (i) Dynamic Key Generation Management, (ii) Multi-Algorithm Encryption, and (iii) Robust Security Evaluation. The mathematical foundations behind IKC give it significant advantages over traditional cryptography and even PQC. A breakdown of some key mathematical principles that contribute to the excellence of IKC is presented with the systematic methodologies in the following subsections. The results and discussions are also briefed.

3.1 Dynamic Key Generation and Management

All classical cryptography methods use static keys, which makes them vulnerable to attacks over time. IKC introduces the Dynamic Key Generation based on real-time security metrics that reduce vulnerability by frequently changing keys. If k represents the cryptographic key, the E(m,k) is the typical encryption scheme. Where m is the message conveyed, and k is the static key. In IKC, k is not static. Instead, it is dynamically generated using an Intelligent Function, f(t,C), where t is temporal time and C is the content conveyed. In traditional cryptography, keys are often static, which makes them vulnerable to attacks over time. IKC introduces *Dynamic Key Generation* based on real-time security metrics, which can reduce vulnerability by frequently updating keys. Let k represent the cryptographic key used in the system. In a typical encryption scheme, E(m, k), where m is the message, and E is the encryption function, k is fixed. In IKC, however, k is not static but is dynamically generated using an *Intelligent Function*, say f(t, C), where t is temporal time and C represents the current security context. Thus, k evolves over time:

$$\mathrm{k_t} = \mathrm{f(t,\, C)}$$

In IKC, this dynamic nature means that, even if an attacker gains access to the key k_t at time t, it becomes useless after $t + \delta t$ when the key is updated. In a new instance of time $t + \delta t$, the key $\mathrm{k}_{(t+\delta\ t)}$ becomes as follows,

$$\mathrm{k_t} + \delta\mathrm{t} = \mathrm{f(t} + \delta\mathrm{t, C)}$$

In IKC,dthis adds a layer of complexity for attackers, as the cryptographic key continuously *changes.*

3.2 Multi-Algorithm Encryption

IKC uses a dynamic approach to choose cryptographic algorithms based on the current computational environment and security needs. The current computational environment includes the choice of a Unicode language that is selected during IKC user registration. In IKC, the chakras are used based on the environment's security level requirement. The cryptographic algorithm varies based on the security needs of implementing the chakras. Thus, the combination of the computational algorithms is implemented in the IKC algorithm. Mathematically, this is asserted through multi-algorithm encryption: Let A be a set of cryptographic algorithms such that

$$\mathrm{A} = \{\mathrm{A}_1, \mathrm{A}_2, \mathrm{A}_3 \ldots, \mathrm{An}\}$$

Each A_i is a cryptographic algorithm based on the combination of the alphabet set and chakras chosen by the user during the implementation. IKC evaluates the current security context C and chooses an algorithm $\mathrm{A}_C$from the set A:

$$\mathrm{E_C(m)} = \mathrm{A_C(m, K)}$$

The decision to switch between algorithms is driven by three parameters such as (i) Computational Load (If the system resources are limited, IKC may choose a lighter algorithm with only the basic two chakras), (ii) Security Threats (IKC switches to a more robust (even if computationally heavier) algorithm in high-risk scenarios), and (iii) Environmental Context (IKC algorithm can also vary depending on the user's location or time zone, making the system harder to predict or attack).

In IKC, this adaptive switching based on the implementation context makes IKC the most efficient algorithm by optimizing performance and security.

3.3 Robust Security Evaluation

The security evaluation of a cryptographic algorithm can be assessed by its mathematical complexity and key entropy, which refers to the *randomness* of the cryptographic key, a crucial factor in determining the security of any cryptographic system. In conventional cryptographic algorithms, the entropy of a key H(k) is a fixed value determined by the key length and the randomness of the key generation process and defined as,

$$H(k) = -\sum p_i \log p_i$$

where p_i is the probability distribution of key values[11]. In IKC, because keys are dynamically generated, their entropy evolves. If k_t is the key at time t, the entropy $H\,(k_t)$ is a function of temporal time and context and is defined as,

$$H\,(k_t) = \ H\,(f(t, C))$$

By dynamically generating keys based on real-time inputs, IKC maximizes the key entropy over time, reducing the likelihood of an attacker being able to predict the key at any given moment. This contrasts with static cryptographic systems, where key entropy remains constant after the key generation phase.

3.4 Resistance to Quantum Attacks

One of the key challenges of post-quantum cryptography is defending against quantum algorithms like Shor's and Grover's algorithms.

Shor's algorithm dramatically reduces the time needed to break asymmetric cryptographic systems (e.g., RSA) by solving integer factorization in polynomial time. *In IKC, no mathematical calculations are involved, so Shor's and Grover's algorithms are not used to break the IKC cryptographic logic. The frequent key changes and adaptive algorithm switching offer increased 'quantum resistance'.*

Grover's algorithm provides a quadratic speedup for brute-force attacks. The time to brute-force a key in symmetric cryptography drops from $O(2^n)$ to $O(2^{n/2})$, where n is the bit-length of the key [12]. In IKC, since keys are constantly updated, the time δt restricts the search space available to an attacker using Grover's algorithm, over which the key remains valid. Even with a quadratic speedup, the quantum attacker's window is significantly shortened and is represented by the following function.

$$T_{Grover}(k) = O(2^{n/2}) \qquad \text{but} \qquad \delta t \ll T_{Grover}(k)$$

This frequent key regeneration and algorithm switching in IKC reduce the effective time window quantum computers must break the cryptographic key.

3.5 Performance and Scalability

This section inspects the performance of the IKC by taking two files of sizes 25kb and 100kb and comparing them with the performance of the most used algorithms such as DES, Blowfish, and AES algorithms.

3.5.1. Encryption Time Comparison

The Encryption Time (ET) is calculated as the time taken by the encryption algorithm from the input of plaintext to produce the cipher text. ET is compared in terms of file size and variable number of characters. Two different file sizes of 25kb and 100kb are taken. In each 25kb and 100kb files, characters ranging from 10000 to 50000 are taken and compared.

In Figure 1, the Encryption Time results of IKC are compared with DES, Blowfish, and AES algorithms on a 25 kb file and a 100kb file. The results show that the IKC algorithm ensures minimal ET over other algorithms.

Figure 1 depicts the ET results of IKC with widely used algorithms under 25kb and 100kb files. The results imply that for the 25kb file, IKC ensures a minimal ET of 3.097s, whereas DES, Blowfish, and AES obtain increased ET of 5.947s, 3.844s, and 3.872s, respectively. Similarly, for the 100kb file, the results imply that the IKC reaches a minimal ET of 4.597s, whereas the AES, Blowfish, and DES models obtain increased ET of 5.392s, 5.504s, and 7.497s, respectively.

The comparative ET results under various numbers of characters offered by IKC with chosen widely used algorithms are given in Figure 2.

Figure 2 portrays the ET comparison of IKC under 25kb and 100kb files with various numbers of characters. In Figure 2, for 25 kb file on 10000 characters, IKC offers a decreased ET of 10.97ms, whereas DES, Blowfish, and AES obtain increased ET of 102.82ms, 62.35ms, and 32.77ms, respectively. Also, based on 50000 characters, IKC offers decreased ET of 81.03ms, whereas DES, Blowfish, and AES obtain increased ET of 344.12ms, 227.36ms, and 112.16ms, respectively.

Similarly, in the 100kb file with 10000 characters, IKC offers a decreased ET of 12.59ms, whereas DES, Blowfish, and AES obtain increased ET of 104.33ms, 63.97ms, and 34.53ms, respectively. Also, based on 50000 characters, IKC offers a decreased ET of 82.79ms, whereas DES, Blowfish, and AES obtain increased ET of 345.85ms, 229.16ms, and 113.66ms, respectively.

3.5.2. Memory Requirement Comparison

The Memory Requirement (MR) is the internal memory required during the encryption process. The MR is calculated by implementing a CPP code for the Encryption algorithm, which calculates the data type size, number of variables used, size of the data structure, and calculation of dynamic memory allocation by considering the memory leaks and memory fragmentation. The MR is analysed for files of size 25kb and 100kb.

In Figure 3, the MR results of IKC are compared with a 25 kb file. The results imply that IKC reaches a lesser MR of 13.89 s, whereas the DES, Blowfish, and AES models obtain increased MR of 65.71 MB, 36.05 MB, and 21.40 MB, respectively. Under

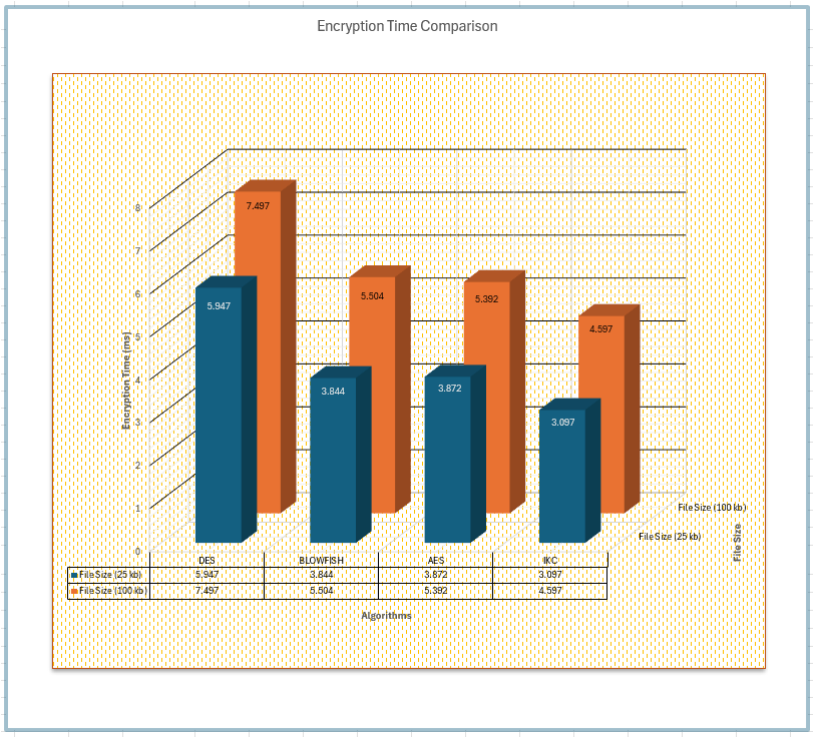

Fig 1. Encryption Time Comparison of IKC with DES, Blowfish, and AES

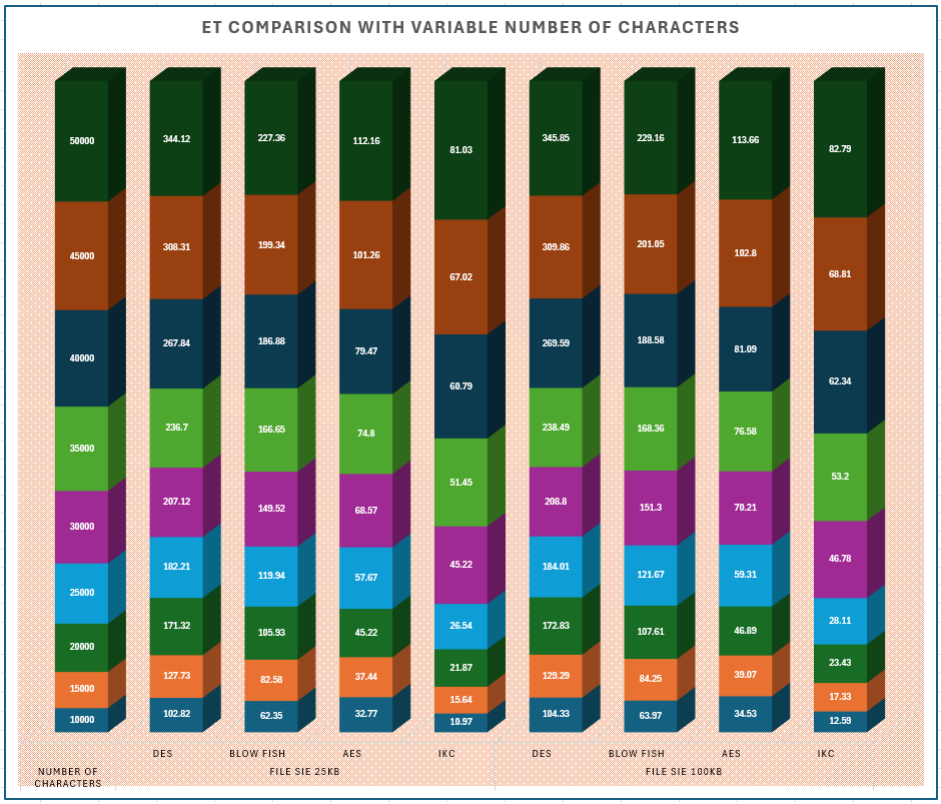

Fig 2. ET Comparison with Variable Number of Characters

100 kb file, the outcome inferred that IKC gains a lesser MR of 15.67 MB, whereas Blowfish, DES, and AES algorithms obtain higher MR of 67.49 MB, 37.81 MB, and 22.98 MB, respectively.

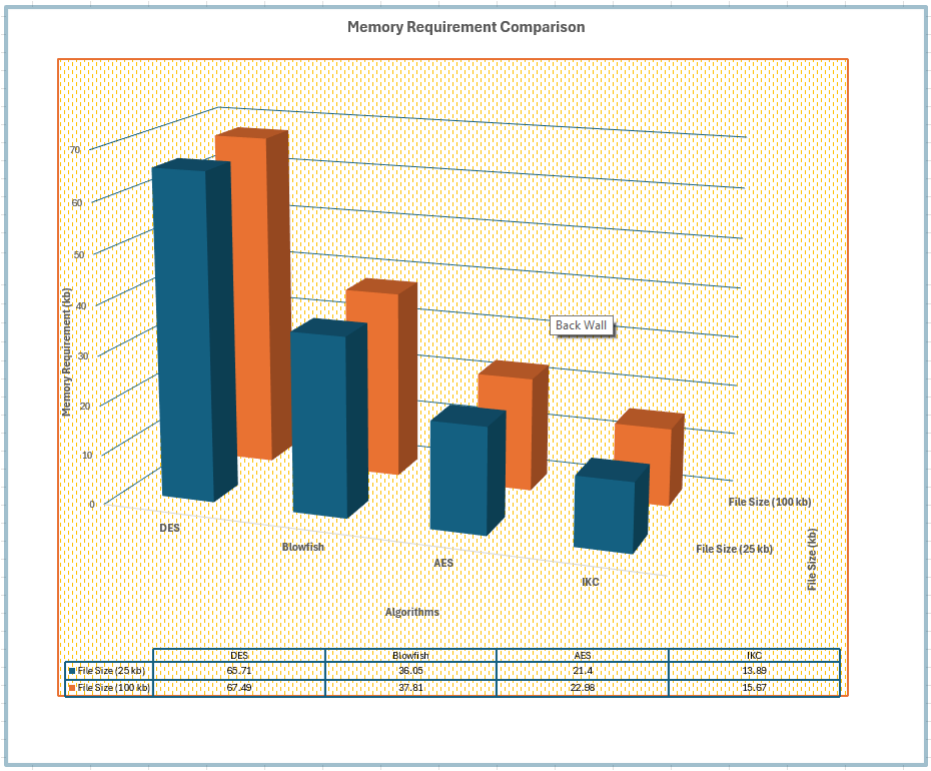

Fig 3. Memory Requirement Comparison of IKC with DES, Blowfish, and AES

3.5.3. Avalanche Effect

The Avalanche effect in Cryptography is the property by which the algorithms produce a drastically varied output (ciphertext) for every small change in the input (plaintext). The minor changes in the input produce hugely varied output, making the relationship between the input and output more random and confirming that the encrypted data is susceptible to changes. If x is the length of the plain text, Ci and Pi are the i^{th} bits of the ciphertext and the plaintext, respectively, then the Avalanche Effect (AE) is given by the following formula,

$$AE = \frac{1}{x}\sum_{i=1}^{x}|c_i - p_i|$$

$$\text{Percentage of bits changed} = AE\ x\ 100$$

The Avalanche Effect of IKC will be empirically tested with a precondition of using the same algorithm, same key, same mode, same padding, and same plaintext and analysed against DES, Blowfish, and AES by implementing it on various categories of changed plain text as (i) single word single character change, (ii) single word multiple character change, (iii) Multiple word single character change, (iv) multiple word multiple character change, and (v) single letter repeated. When tested, *it is astonishing that the fifth category of repeated single-letter characters gave an exciting output. DES, Blowfish and AES showed the same ciphertext, whereas IKC produced a varied ciphertext at every repeated character*. This is because the encryption for a single character in the plain text gets manipulated with a set of rotating chakras. So, every time the character is encrypted, it produces an entirely different ciphertext. Figure 4 represents the bit change for each of the five categories of tests, and the Avalanche Effect is projected as a radar graph to best fit the percentage comparison. The Avalanche effect of IKC is very high, even if there is no change in the plaintext, and that is the highlighted strength of IKC, because when the exact text is given continuously, every time more

than 99% of the ciphertext changes. This is because the ciphertext in IKC is time dependent. None of the currently available algorithms achieves this.

Figure 4 shows that IKC has a strong Avalanche Effect compared to DES, Blowfish, and AES. Further, IKC guarantees an Avalanche Effect even for the same plaintext, as the encryption is done based on *Temporal Time* as a key factor. *Thus, IKC is tougher for the attacker to guess the plaintext from the ciphertext than any other widely used cryptographic algorithm.*

Algorithms	Avalanche Effect (%)				
Changes in Plain Text	Single word single character change	single word multiple character change	multiple word single character change	multiple word multiple character change	single letter repeated
DES	56	79	40	89	0
Blowfish	54	80	55	79	0
AES	63	88	78	89	0
IKC	99	98	99	100	100

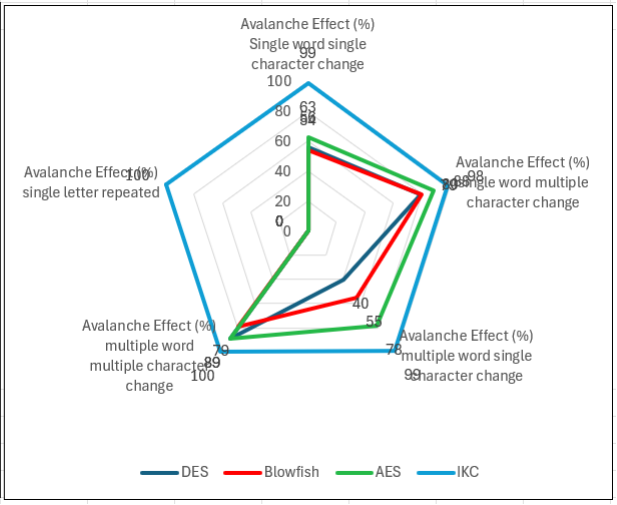

Fig 4. Avalanche Effect Comparison of IKC with DES, Blowfish, and AES

4 Conclusion

IKC is a revolution in the cryptographic environment, where ciphering and deciphering use Tamil grammar by representing the alphabet of Tamil in Unicode and using temporal time as the key element. This unique method implemented in IKC makes it a promising solution to secure sensitive communications in a world preparing for large-scale quantum computing. IKC assures flexibility, intelligence, and adaptability in cryptographic systems. The only known limitation is the sharing of the temporal Time between the sender and receiver. The use of the quantum entanglement method can overcome this limitation. This could be a future perspective of my research to transmit the temporal Time between the sender and the receiver. This paper demonstrates IKC's dynamic approach, which allows real-time intelligent key management and represents a compelling alternative to PQC, thus proving to be a post-quantum cryptographic solution. Future research can explore IKC with other Languages. The challenges surrounding IKC with other languages will be implementing the grammar of the chosen language to the changing chakras of IKC.

References

1) Tom JJ, Anebo NP, Onyekwelu BA, Wilfred A, Eyo RE. Quantum Computers and Algorithms: A Threat to Classical Cryptographic Systems. *International Journal of Engineering and Advanced Technology (IJEAT)*. 2023;12(5). https://doi.org/10.35940/ijeat.E4153.0612523.
2) Nokia. A clear road to the quantum internet. 2024. Available from: https://www.nokia.com/thought-leadership/articles/clear-road-to-quantum-internet/?gad_source=1&gclid=CjwKCAjwl6-3BhBWEiwApN6_kq5wf6-BCtK3JpfObGXLckzJcIoX9hIAh5pdcohpriYexuoURM4DDhoCJDcQAvD_BwE.
3) Gillis TS. Quantum key distribution (QKD). 2024. Available from: https://www.techtarget.com/searchsecurity/definition/quantum-key-distribution-QKD.
4) Shukla A. Future of Current Encryption Algorithm in Quantum Computing Age and their Future. *International Journal of Science and Research (IJSR)*. 2023;12(10):2111–2115. https://doi.org/10.21275/SR231116135144.
5) NIST. Post-Quantum Cryptography PQC. 2023. Available from: https://csrc.nist.gov/projects/post-quantum-cryptography.
6) Wong HY. Introduction to Quantum Computing. Cham. Springer. 2023. https://doi.org/10.1007/978-3-031-36985-8_29.
7) Qiu D, Luo L, Xiao L. Distributed Grover's algorithm. *Theoretical Computer Science*. 2024;993:114461. https://doi.org/10.1016/j.tcs.2024.114461.
8) Ravi et al . Side-channel and Fault-injection attacks over Lattice-based Post-quantum Schemes (Kyber, Di lithium): Survey and New Results. *ACM Transactions on Embedded Computing Systems*. 2024;23(2):35–54. https://doi.org/10.1145/3603170.
9) Rojasree V, Jayanthi JG, Sujatha DC. Research Intuitions of Hashing Crypto System. *International Journal of Engineering Research & Technology (IJERT)*. 2020;9(12):402–406. https://doi.org/10.17577/IJERTV9IS120185.
10) Rojasree V, Jayanthi JG. Design and Implementation of Intelligent key Cryptography using Time and Tamil Unicode. *Indian Journal of Science and Technology*. 2023;16(2):133–145. https://doi.org/10.17485/IJST/v16i2.1917.

11) Dodis Y, Smith A. Entropic Security and the Encryption of High Entropy Messages. vol. 3378. Berlin, Heidelberg. Springer. 2005;p. 556–577. https://doi.org/10.1007/978-3-540-30576-7_30.
12) Wu H, Feng X, Zhang J. Quantum Implementation of the SAND Algorithm and Its Quantum Resource Estimation for Brute-Force Attack. *Entropy*. 2024;26(3):216. https://doi.org/10.3390/e26030216.