



Faculty of Computer Science and Information Technology

**A DNA Based Lightweight Cryptography Framework Integrating
ECC and RNN Based Deep Learning for IoT Security**

Shrish Aqeel

**Doctor of Philosophy
2025**

A DNA Based Lightweight Cryptography Framework Integrating ECC and RNN Based Deep Learning for IoT Security

Sehrish Aqeel

A thesis submitted

In fulfillment of the requirements for the degree of Doctor of Philosophy

(Network Technologies)

Faculty of Computer Science and Information Technology

UNIVERSITI MALAYSIA SARAWAK

2025

DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Malaysia Sarawak. Except where due acknowledgements have been made, the work is that of the author alone. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.



.....

Signature

Name: Sehrish Aqeel

Matric No.: 20010206

Faculty of Computer Science and Information Technology

Universiti Malaysia Sarawak

Date: 5th Sep 2025

ACKNOWLEDGEMENT

First of all, I want to convey my deepest gratitude to the Almighty and Merciful **ALLAH (SWT)** for everything and for providing me with the strength, courage, wisdom, and perseverance to complete this journey. It is due to the blessings bestowed by Him that I have come to this age; if not for His grace, nothing would have been accomplished.

Secondly, I extend my gratitude to my esteemed supervisor, Assoc. Prof. **Dr. Adnan Shahid Khan** has shown immense academic support, understanding, and guidance at the right time and hence contributed a lot to my whole research work. From his passion, inspiration, and commitment to ensuring that everyone comprehends complex issues, I can tell that he has greatly relieved most of the tasks that I have had to accomplish. Whenever I required his support, he was there for me, and I literally could not have had a better mentor throughout my Ph.D. years. I am also grateful to the faculty members of the Faculty of Computer Science and Information Technology Universiti Malaysia Sarawak for their encouragement, support, and critical comments.

I dedicate this to my late dear parents, **Mr. Muhammad Tufail** and **Mrs. Munazzah Tufail**, who have always been my strength in making my personality and trusting my abilities. I am pleased to express my deep appreciation to my husband, **Mr. Aqeel Altaf**. Without his encouragement and support, this work could not have been completed. I am also highly grateful to my children, **Horiya Aqeel, Laiba Aqeel, and Muhammad bin Aqeel**, for their love, encouragement, and faith in my capabilities. They are sources of light in my life.

ABSTRACT

The exponential growth of Internet-of-Things (IoT) networks and connected devices has highlighted critical challenges in ensuring high security, low latency, and low energy consumption for sensitive data exchange in resource-constrained environments. Traditional cryptographic systems often struggle to meet these demands, necessitating innovative solutions tailored to IoT. This study proposes a novel DNA-Based Lightweight Cryptographic System (DNA-LWCS) integrating the inherent randomness of DNA sequences with Elliptic Curve Cryptography (ECC) to enhance security and efficiency. Leveraging the robustness of DNA sequences and ECC's computational efficiency, DNA-LWCS provides robust protection against attacks like Distributed Denial-of-Service (DDoS) and Man-in-the-Middle (MITM), addressing IoT vulnerabilities. Deep learning models, including Recurrent Neural Networks (RNN) and Gated Recurrent Units (GRU), achieve remarkable precision of 99.62% for RNN and 98.61% for GRU in key strength estimation, optimizing encryption processes.

Experimental results demonstrate DNA-LWCS outperforms 3DES, AES, ECC, and benchmarks in critical security and efficiency metrics. DNA-LWCS achieves a correlation coefficient of 0.025, ensuring high confidentiality and data integrity, fast encryption times (e.g., 0.800 seconds for Photographer), high throughput (12.00 Bps at 50 data points), and low energy consumption (10.00–15.00 J), ideal for resource-constrained IoT devices like smart meters and smart homes. DNA-LWCS demonstrates superior entropy values (7.800 for Lena, 6.500 for Photographer) and avalanche effect (52.99% for Lena, 50.99% for Photographer), meeting stringent randomness and input sensitivity criteria for applications like autonomous vehicles and patient monitoring.

This research advances IoT security by combining DNA-based encryption with deep learning, delivering a scalable, efficient, and resilient cryptographic framework that bolsters data protection and fulfills the essential requirement for lightweight, energy-efficient encryption in contemporary interconnected systems.

Keywords: IoT (Internet of Things), DNA-based encryption, Elliptic Curve Cryptography (ECC), Lightweight Cryptographic System (LWCS), deep learning models, Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU), encryption, evaluation metrics

Rangka Kerja Kriptografi Ringan Berasaskan DNA Mengintegrasikan Pembelajaran Mendalam Berasaskan ECC dan RNN untuk Keselamatan IoT

ABSTRAK

Pertumbuhan eksponen rangkaian Internet Perkara (IoT) dan peranti bersambung telah menyerlahkan cabaran kritikal dalam memastikan keselamatan tinggi, kependaman rendah, dan penggunaan tenaga rendah untuk pertukaran data sensitif dalam persekitaran terhad sumber. Sistem kriptografi tradisional sering sukar memenuhi permintaan ini, memerlukan penyelesaian inovatif yang disesuaikan untuk IoT. Kajian ini mencadangkan Sistem Kriptografi Ringan Berasaskan DNA (DNA-LWCS) yang mengintegrasikan kerawakan semula jadi jujukan DNA dengan Kriptografi Lengkung Eliptik (ECC) untuk meningkatkan keselamatan dan kecekapan. Dengan memanfaatkan keteguhan jujukan DNA dan kecekapan pengiraan ECC, DNA-LWCS memberikan perlindungan teguh terhadap serangan seperti Penafian Perkhidmatan Teragih (DDoS) dan Man-in-the-Middle (MITM), menangani kelemahan IoT. Model pembelajaran mendalam, termasuk Rangkaian Neural Berulang (RNN) dan Unit Berulang Berpagar (GRU), mencapai ketepatan cemerlang iaitu 99.62% untuk RNN dan 98.61% untuk GRU dalam anggaran kekuatan kunci, mengoptimumkan proses penyulitan. Keputusan eksperimen menunjukkan DNA-LWCS mengatasi 3DES, AES, ECC, LWCS, RF-DECC, dan model IoT dalam metrik keselamatan dan kecekapan kritikal. DNA-LWCS mencapai pekali korelasi 0.025, memastikan kerahsiaan dan integriti data tinggi, masa penyulitan pantas (contohnya, 0.800 saat untuk Jurugambar), daya pemprosesan tinggi (12.00 Bps pada 50 titik data), dan penggunaan tenaga rendah (10.00–15.00 J), sesuai untuk peranti IoT terhad sumber seperti meter pintar dan rumah pintar. DNA-LWCS menunjukkan nilai entropi unggul (7.800 untuk Lena, 6.500 untuk Jurugambar) dan kesan longsor (52.99% untuk Lena, 50.99% untuk Jurugambar), memenuhi kriteria kerawakan dan sensitiviti input yang ketat untuk aplikasi

seperti kendaraan autonomi dan pemantauan pesakit. Penyelidikan ini memajukan keselamatan IoT dengan menggabungkan penyulitan berasaskan DNA dan pembelajaran mendalam, menyampaikan rangka kerja kriptografi yang berskala, cekap, dan tahan lasak untuk perlindungan data dalam sistem saling berkait moden.

Kata kunci: *IoT (Internet Perkara), penyulitan berasaskan DNA, Kriptografi Lengkung Eliptik (ECC), Sistem Kriptografi Ringan (LWCS), model pembelajaran mendalam, Rangkaian Neural Berulang (RNN), Unit Berulang Berpagar (GRU), penyulitan, metrik penilaian*

TABLE OF CONTENTS

	Page
DECLARATION	i
ACKNOWLEDGEMENT	ii
ABSTRACT	iii
<i>ABSTRAK</i>	v
TABLE OF CONTENTS	vii
LIST OF TABLES	xii
LIST OF FIGURES	xiii
LIST OF ABBREVIATIONS	xvi
CHAPTER 1: INTRODUCTION	1
1.1 Research Background	1
1.2 Problem Statement and Research Challenges	5
1.3 Research Objectives	8
1.4 Scope of the Study	8
1.5 Dissertation Contributions	9
1.6 Thesis Organization	11
CHAPTER 2: LITERATURE REVIEW	13
2.1 Introduction	13
2.2 Internet-of-Things (IoT)	15
2.3 IoT Security Challenges	19

2.4	An Overview of DNA Encoding Scheme	27
2.5	An Overview of DL Approaches	29
2.6	Related Work on the DNA Encoding Scheme	31
2.7	DNA Security Against Attacks	41
2.8	Chapter Summary	57
CHAPTER 3: METHADODOLOGY		59
3.1	Introduction	59
3.2	System Design	62
3.3	First Round of Security	62
3.4	Second Round of Security	62
3.5	Third Round of Security	64
3.6	Proposed Algorithm	69
3.6.1	Key Generation	69
3.6.2	Message Encryption	70
3.6.3	Transmission	71
3.6.4	Message Decryption	73
3.6.5	Process to Achieve Objective 2	77
3.7	Network Model	78
3.7.1	Process to Achieve Objective 1	80
3.8	Data Used	81
3.8.1	Process to Achieve Objective 3	85
3.9	Evaluation Metrics	86

3.9.1	Accuracy	87
3.9.2	Precision	87
3.9.3	Recall	87
3.9.4	F1 Score	88
3.9.5	Histogram	88
3.9.6	Correlation	88
3.9.7	Entropy	89
3.9.8	Encryption Time	90
3.9.9	PSNR Comparison	90
3.9.10	Avalanche Effect	90
3.9.11	Throughput	91
3.9.12	Delay	91
3.9.13	Energy Consumption	91
3.10	Chapter Summary	92
CHAPTER 4: RESULTS AND ANALYSIS		93
4.1	Introduction	93
4.2	Algorithms Segregation	94
4.3	List of Algorithms	96
4.4	Model Implementation	116
4.5	Comparison with Benchmarks	123
4.5.1	Correlation	123

4.5.2	Encryption Time	126
4.5.3	Throughput	128
4.5.4	Delay	131
4.5.5	Energy Consumption	133
4.6	Performance Comparison of DNA-LWCS with Other Algorithms	143
4.6.1	Histogram Analysis	143
4.6.2	Correlation	146
4.6.3	Encryption Time	147
4.6.4	Throughput	149
4.6.5	PSNR Comparison	151
4.6.6	Entropy	153
4.6.7	Avalanche Effect	155
4.7	Discussion	161
4.7.1	Alignment with Research Objectives	163
4.7.2	Limitations of the Findings	164
4.8	Chapter Summary	166
CHAPTER 5: CONCLUSION AND RECOMMENDATIONS		168
5.1	Introduction	169
5.2	Significant Findings and Research Contributions	169
5.2.1	Systematic Literature Review of DNA-Based Encryption Scheme for IoT	170
5.2.2	Key Contribution	171
5.2.3	The Process of Training the Models and Estimation of the Accuracy Level of Trained Dataset	172

5.2.4	Improvement in Correlation rate, Encryption time, Delay, Throughput and Energy Consumption as Compared to Benchmark Studies	173
5.2.5	Detailed Evaluation of DNA-LWCS Encryption Algorithm across Various Metrics	174
5.2.6	Broader Impact of Findings	176
5.3	Future Work	176
	REFERENCES	178
	APPENDICES	202

LIST OF TABLES

	Page
Table 2.1 Binary Sequence of Each Nucleotide	29
Table 2.2 Comparison of DNA-Based Cryptographic Methods for IoT Security	40
Table 2.3 Research Studies and Attacks	45
Table 2.4 Cyber Attacks Focused on a DNA Strategy	46
Table 2.5 Summary of Research Studies Used DNA-based Encoding Schemes	47
Table 2.6 Count and Percentage of Powerful Attacks by Frequency	48
Table 2.7 Comparative Analysis of Research Studies	50
Table 2.8 Research gaps with related references and linked objectives	56
Table 2.9 Key Variables of Energy Efficiency in Reviewed Articles	57
Table 3.1 Mapping of Research Objectives to DNA-LWCS Methodology Sections	60
Table 4.1 Hyper Parameters Used in Each Model	118
Table 4.2 Comparison of Performance Evaluation Metrics	122
Table 4.3 Comparison of Evaluation metrics of DNA-LWCS with other Algorithms	139
Table 4.4 Comparison of Results Obtained from DNA-LWCS with Standardized Units	141
Table 4.5 Encrypted Images using 3DES, AES, ECC and DNA-LWCS	144
Table 4.6 Histogram using 3DES, AES, ECC and DNA-LWCS	145
Table 4.7 Comparative Performance Analysis of DNA-LWCS with 3DES, AES, and ECC	160

LIST OF FIGURES

	Page
Figure 2.1 Potential Cyber-Physical Attacks	16
Figure 2.2 The Conventional Hierarchical Structure of the IoT	18
Figure 2.3 Research Questions	22
Figure 2.4 Research Questions for SLR	23
Figure 2.5 Research Methodology	25
Figure 2.6 Systematic Literature Review (SLR) Approach	26
Figure 2.7 Key Variables	27
Figure 2.8 Double Helix of DNA	28
Figure 2.9 Deep Learning Models	30
Figure 2.10 System Design of Benchmark 1 (RF-DECC)	36
Figure 2.11 System Design of Benchmark 2 (IoT)	37
Figure 2.12 System Design of Benchmark 3 (LWCS)	39
Figure 2.13 Classification of Different Forms of Attack	44
Figure 2.14 Frequency of Reviewed Attacks	49
Figure 3.1 DNA-Based Lightweight Encryption with ECC for IoT	61
Figure 3.2 ECC Approach to Generate P and Q Points	64
Figure 3.3 System Design of DNA-LWCS	66
Figure 3.4 DNA-LWCS Framework	67
Figure 3.5 High-Level System Design	68
Figure 3.6 Flow of Activities in DNA-LWCS	69
Figure 3.7 Algorithm Stages	70
Figure 3.8 Detailed Description of Each Operation in DNA-LWCS	72
Figure 3.9 Flowchart of DNA-LWCS	75

Figure 3.10	Description of Stages of DNA-LWCS	76
Figure 3.11	Devices Under Each Layer of IoT	79
Figure 3.12	Raw Keys Data Set	82
Figure 3.13	Data Pre-Processing	84
Figure 3.14	Image Data	85
Figure 4.1	Data Pre-Processing	95
Figure 4.2	Generation of Strong and Weak Keys	98
Figure 4.3	Sample of Improved Key Dataset	117
Figure 4.4	Correlation Data used by Each Algorithm	124
Figure 4.5	Graphical Representation of Correlation	125
Figure 4.6	Encryption Time Data	126
Figure 4.7	Graphical Representation of Encryption Time	128
Figure 4.8	Throughput Data	130
Figure 4.9	Graphical Representation of Throughput	130
Figure 4.10	Delay Data	131
Figure 4.11	Graphical Representation of Delay	133
Figure 4.12	Data of Energy Consumption	134
Figure 4.13	Graphical Representation of Energy Consumption	135
Figure 4.14	Summarized Results of DNA-LWCS with Benchmark Studies	136
Figure 4.15	Correlation Results	146
Figure 4.16	Geographical Representation of Correlation Results	147
Figure 4.17	Encryption Time Results	148
Figure 4.18	Geographical Representation of Encryption Time Results	149
Figure 4.19	Throughput Results	150

Figure 4.20	Geographical Representation of Throughput Results	151
Figure 4.21	PSNR Results	152
Figure 4.22	Geographical Representation of PSNR Results	153
Figure 4.23	Information Entropy Results	154
Figure 4.24	Graphical Representation of Information Entropy Results	155
Figure 4.25	Avalanche Effect Results per bit change	156
Figure 4.26	Graphical Representation of Avalanche Effect Results per bit change	157
Figure 4.27	Summarized Results of DNA-LWCS	161

LIST OF ABBREVIATIONS

3DES	Triple Data Encryption Standard
4G	Fourth Generation
5G	Fifth Generation
6LowPAN	IPv6 over Low Power Wireless Personal Area Networks
AES	Advanced Encryption Standard
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
Bi-LSTM	A type of LSTM processing data in both directions
CA	Crypto Attack
CNN	Convolutional Neural Network
CoAP	Constrained Application Protocol
DA	Differential Attack
DDoS	Distributed Denial of Service
DDS	Data Distribution Service
DEAIOT	Distributed Elliptic Algorithm for IoT
DES	Data Encryption Standard
DIoTA	Decentralized IoT Architecture
DL	Deep Learning
DNA	Deoxyribonucleic Acid
DNA-LWCS	DNA-based Lightweight Cryptographic System
DoS	Denial of Service
ECC	Elliptic Curve Cryptography

EECS	Enhanced Encryption Cryptography Scheme
F1 Score	Harmonic Mean of Precision and Recall
FAR	False Acceptance Rate
FN	False Negative
FP	False Positive
GenBank	A database of genetic sequences
GQM	Goal Question Metric
GRU	Gated Recurrent Unit
IIoT	Industrial Internet of Things
IoT	Internet of Things
IP	Internet Protocol
IPv6	Internet Protocol Version 6
J	Joules
JA	Jamming Attack
kWh	Kilowatt-hour
LCWS	Lightweight Cryptographic System
LSTM	Long Short-Term Memory
MAC	Media Access Control
MITM	Man-in-the-Middle
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
NCBI	National Centre for Biotechnology Information
NCs	Nucleotide Chains
NWA	Network-wide Attack
OFDM	Orthogonal Frequency Division Multiplexing

PON	Passive Optical Network
PCR	Polymerase Chain Reaction (DNA amplification)
PSNR	Peak Signal-to-Noise Ratio
PVSNR	Peak Visual Signal-to-Noise Ratio
RA	Replay Attack
RDDECC	Reduced Diffie-Hellman and Elliptic Curve Cryptography
RFDECC	Randomized Fuzzy Differential Elliptic Curve Cryptography
RNN	Recurrent Neural Network
RSA	Rivest–Shamir–Adleman (cryptosystem)
SLR	Systematic Literature Review
SP	Service Pack (related to operating systems)
SPECK	A lightweight block cipher
SQL	Structured Query Language
SVM	Support Vector Machines
TensorFlow	A machine learning framework
TN	True Negative
TNR	True Negative Rate
TP	True Positive
USD	United States Dollar
VMEI	Visually Meaningful Encrypted Image
WiFi	Wireless Fidelity
XSS	Cross-Site Scripting
ZigBee	Low-power wireless communication technology

CHAPTER 1

INTRODUCTION

1.1 Research Background

The Internet of Things (IoT) has emerged as a computing revolution covering various sectors, including smart cities, homes, healthcare, industries, transportation and power systems, and smart grids. IoT refers to various integrated devices called “Things” which are fitted with sensors and actuators and typically have limited storage and processing capacity. Such machines interact and share information through conventional internet and intranet services and offer real-time tracking and control (Luo et al., 2018). Currently, the revenues in the IoT sector are expected to proliferate, with the market revenue increasing from 2 billion USD in 2020 to 8 billion. According to Al-Sarawi et al. (2020), global investment in health is projected to reach 131 trillion USD by 2030. Such capability of generating extensive revenue has made the concept attract multitiered stakeholders such as suppliers, manufacturers, and corporate entities (Ahmad et al., 2019).

As IoT expands, many issues arise related to its connectivity and imposing security measures on interconnected things within the network devices. Most IoT devices come with limitations on computational resources, which are a barrier to implementing high security and robust encryption techniques (Hong et al., 2021). Also, the large number of devices connected to networks results in a large attack area, making it easy for hackers to find vulnerabilities to exploit to carry out attacks like data theft, unauthorized access, and malware infections (Imdad et al., 2020). Hackers can take advantage of any vulnerability in the system to gain control over the devices and steal confidential data or cause a collapse of operations. The absence of uniformity of secure measures for IoT’s other platforms adds to the problem, which represents

personal, corporate and even hold-up structure systems. Tackling these security issues demands effective measures such as better encryption standards, regular software patches and more robust authentication processes to counter act the new age risks (Sharma, 2020). One of the most popular and well-known examples of an attempt to use IoT devices to unleash a cyberattack is the Mirai botnet attack in 2016. The Mirai malware targeted numerous IoT devices, including Internet Protocol cameras and home routers with default passwords that are easily hackable and low-level security standards (Alnafisah, 2020). These devices launched gigantic Distributed Denial of Service (DDoS) attacks, especially against Dyn a large domain name system server. The attack caused significant sites such as Twitter, Netflix, Reddit, and others to go down, affecting millions of its users for some time (Antonakakis et al., 2017).

Several methods are currently being employed to increase the security of IoT devices, with further work being conducted to try and develop those methods, such as Triple Data Encryption Standard (3DES), Advanced Encryption Standard (AES), Elliptic Curve Cryptography (ECC), and emerging technologies, including DNA-based cryptography, all of which have their advantages and disadvantages (Kumar, 2021). For instance, AES can provide excellent security. However, it may consume much processing power in devices with limited computational capability. In contrast, the ECC model can provide more efficiency for small devices, though it suffers from some attacks known as specific attacks (Murugan & Thilagavathy, 2017). DNA-based cryptography leverages the inherent randomness and high data density of DNA sequences to create lightweight encryption schemes, addressing IoT's resource constraints by enabling secure key generation with minimal computational overhead (Wang et al., 2023). However, challenges such as DNA synthesis errors and scalability require further exploration. Therefore, although DNA cryptography is a reasonably new concept and can be viewed as innovative, various issues have to be further investigated, such as its feasibility and

associated problems. However, these algorithms could open new doors to better algorithms and techniques that can fill the gaps and meet the future security requirements of IoT systems, such as lightweight cryptography, secure communication protocols, and real-time threat detection (Shafique et al., 2021).

Bio-inspired DNA encoding systems for information technology have received more attention lately, especially regarding the safety of IoT networks. The possibility of using DNA as an encryption of computation was first discussed in the mid-1990s when Adleman proposed DNA computing by solving a combinatorial problem with the help of molecular biology methods (Adleman, 1994). On this basis, DNA encoding has been considered one of the perspectives for IoT security because of its ability to create large, complex cryptographic keys. The success story of DNA encoding in IoT security has been most vivid in cryptographic algorithms that utilize DNA's nucleotide sequences – Adenine, Thymine, Cytosine, and Guanine to make encryption. When one encodes the binary data into NCs, researchers are in a position to develop robust and secure cryptographic algorithms, which are suitable for IoT devices with fewer resources (Wang et al., 2023). This approach enhances IoT security by providing high entropy keys resistant to brute-force attacks, unlike traditional pseudo-random key generation methods, which are less adaptive to dynamic threats.

Also, DNA-based encryption has been advanced by incorporating other features like ECC, resulting in high security to the IoT network by combining the ability of DNA biology randomness and ECC in key generation and data protection (Zhang et al., 2022). The lucidity pattern shared in the DNA has also allowed researchers to devise a more efficient encryption algorithm for IoT real-time dynamics (Liu et al., 2023). Furthermore, integrating machine learning models, such as Recurrent Neural Networks (RNNs) and Gated Recurrent Units (GRUs), optimizes key generation by dynamically adapting to environmental factors, offering

superior performance over traditional methods that rely on static algorithms (Afify & Rahouma, 2021). Nevertheless, despite what has been achieved through incorporating DNA encoding, there are still some issues that have not been effectively addressed yet, concerns with minimizing error rates of DNA synthesis and sequencing, and the scalability of the encryption systems, as well as how these solutions can be easily integrated into the IoT environment. Furthermore, encryption time, complexity, randomness, scalability and security must be enhanced to produce a robust solution that could work effectively in IOT environment (Tiwari & Kim, 2018).

This research focuses on developing a highly effective DNA-based lightweight encryption algorithm to meet the unique requirements and constraints of the Internet of Things (IoT) environment. Given the resource limitations and the need for high security and minimal computational overhead in IoT devices, the algorithm leverages the inherent randomness found in DNA sequences combined with Elliptic Curve Cryptography (ECC) to function as a dual-core mechanism that significantly strengthens data protection. DNA sequences, known for their complexity and unpredictability, generate cryptographic keys, while ECC ensures efficient and robust encryption, even with limited computational power. This hybrid approach aims to deliver a scalable and adaptable solution suitable for IoT applications where security is a primary concern. However, resources such as processing power, memory, and energy are often constrained.

In addition, the encryption scheme is implemented within a machine learning environment, where trained models are employed to optimize the key generation process (Afify & Rahouma, 2021). These models help produce the most optimal strength keys based on dynamic environmental factors, enhancing the overall encryption performance and ensuring that the system remains effective under varying conditions (Ahamed & Murugan, 2024).

This machine learning integration offers an intelligent, adaptive approach to cryptographic key management, which is crucial in maintaining a high level of security while minimizing energy consumption and computational load (Ahmad et al., 2018).

Furthermore, the algorithm undergoes rigorous testing and deployment in real-world IoT scenarios. It is benchmarked against existing encryption techniques, showcasing superior performance in terms of both security strength and operational efficiency. The results demonstrate significant improvements in key aspects such as encryption speed, data protection, and resource utilization, successfully addressing the security challenges specific to IoT devices (Ahmadi et al., 2019). This research not only contributes to the advancement of cryptographic solutions for IoT but also sets a precedent for future studies exploring hybrid approaches that combine biological randomness with modern cryptographic methods to optimize security and efficiency in constrained environments. The findings and benchmarks confirm the algorithm's capability to outperform traditional solutions and effectively tackle challenges commonly faced in IoT-based systems (Kumar et al., 2020; Akash et al., 2024).

1.2 Problem Statement and Research Challenges

The rapid proliferation of Internet of Things (IoT) devices has intensified the demand for lightweight, energy-efficient cryptographic systems capable of securing resource-constrained networks against sophisticated cyber threats. The interconnectivity of numerous devices in IoT networks exposes them to a range of security vulnerabilities. As IoT devices often handle sensitive data, ensuring robust security is critical to prevent unauthorized access, data breaches, and disruptions (Al-Husainy et al., 2023). While traditional cryptographic techniques such as firewalls, authentication protocols, and encryption methods provide foundational protection, they are increasingly inadequate to secure IoT systems due to the limited computational resources available on many IoT devices (Frustaci et al., 2018; Panigrahi &

Borah, 2019). This limitation underscores the need for advanced cryptographic solutions that integrate DNA-based encryption, Elliptic Curve Cryptography (ECC), and machine learning to address IoT's unique security and efficiency requirements.

Among the existing solutions, lightweight cryptographic algorithms like RD-DECC and SPECK have been employed in IoT applications for their low resource consumption and basic encryption capabilities. While effective to some extent, these algorithms lack the complexity needed to withstand sophisticated attacks, making them vulnerable to more advanced cyber threats (Kumar & Singh, 2023). Furthermore, homomorphic encryption, which offers high data confidentiality by allowing computations on encrypted data, faces significant limitations in IoT environments due to its high computational overhead (Patel et al., 2023). These challenges highlight the need for a DNA-based cryptographic system that leverages the high data density and randomness of DNA sequences to create secure, lightweight encryption suitable for IoT devices.

DNA-based cryptography encodes binary data into nucleotide sequences (Adenine, Thymine, Cytosine, Guanine), enabling the generation of complex cryptographic keys with minimal computational resources. When combined with ECC, which provides high security with small key sizes, this approach enhances resistance to attacks such as brute-force and side-channel attacks (Kumar et al., 2023). However, challenges persist, including computational complexity in encryption/decryption processes, which may increase latency, and errors in DNA synthesis or degradation during transmission, which affect data integrity (Singh & Verma, 2023). Optimization techniques, such as genetic algorithms and machine learning, are critical to improving efficiency and scalability.

Key generation is a pivotal aspect of IoT security, as the randomness of cryptographic keys prevents attacks. Machine learning models, specifically Recurrent Neural Networks

(RNNs) and Gated Recurrent Units (GRUs), outperform traditional pseudo-random key generation by dynamically adapting to environmental factors and predicting key strength, thus enhancing security (Zhang et al., 2019). However, scaling these models across numerous IoT devices remains a challenge.

A comprehensive literature review reveals a significant gap in integrating DNA-based cryptography, ECC, and machine learning into a unified, lightweight framework for IoT security. Existing DNA-based systems often rely on static key generation methods, which are vulnerable to complex attacks, and lack dynamic integration with advanced machine learning techniques like RNNs or GRUs (Al-Husainy et al., 2023). Moreover, current solutions exhibit suboptimal performance in security metrics such as entropy, PSNR, and avalanche effect, as well as inefficiencies in energy consumption and throughput (Hazra et al., 2018).

To address these gaps, this study hypothesizes that a DNA-Based Lightweight Cryptographic System (DNA-LWCS) integrating Elliptic Curve Cryptography (ECC) and machine learning models (specifically RNN and GRU) will significantly improve IoT security by achieving lower correlation coefficients (e.g., below 0.05), faster encryption times (e.g., under 1 second), higher throughput (e.g., above 10 Bps), and lower energy consumption (e.g., below 15 J) compared to traditional cryptographic methods like 3DES, AES, and standalone ECC, thereby providing a scalable, efficient, and secure framework for resource-constrained IoT environments.

The proposed DNA-based Lightweight Cryptography System (DNA-LWCS), integrating ECC and RNN-based machine learning, will achieve lower encryption time, higher information entropy, and reduced energy consumption compared to traditional cryptographic methods, providing a scalable and secure solution for IoT networks.

This research aims to address these challenges by developing a DNA-based lightweight cryptographic system that utilizes ECC and deep learning models to optimize key generation and encryption for IoT applications. By combining DNA sequences, ECC, and machine learning techniques, the proposed system seeks to enhance security while minimizing resource consumption and overcoming the limitations of current cryptographic solutions. Through this approach, the research strives to provide a more adaptive, efficient, and secure cryptographic framework better suited to IoT environments' dynamic and resource-constrained nature.

1.3 Research Objectives

This research identifies the vulnerabilities and limitations of the lightweight DNA based cryptographic system used in IoT security. To analyse and design a better solution that will offer a higher level of security to IoT devices. Specifically, this research aims to

- i. To identify and investigate vulnerabilities, lightweight cryptographic weaknesses, and resource constraints in current IoT networks.
- ii. To design an efficient security solution for an IoT network by proposing a novel DNA-based Encryption system using ECC (DNA-LWCS).
- iii. To evaluate the performance of the proposed DNA-LWCS scheme by comparing it with existing models (e.g., AES, 3DES, and ECC-based schemes) using metrics such as Encryption Time, Correlation, Delay, Throughput, PSNR, Information Entropy, Avalanche Effect, and Energy Consumption to ensure superior security and efficiency.

1.4 Scope of the Study

The scope of this research is to develop a secure encryption technique for IoT networks using DNA-based keys and genetic algorithms. The system creates variations of keys with genetic algorithms and other operations such as bit rotation, padding, XOR and complement,

leading to a very secure key to encrypt data. These keys are then assessed in terms of high and low degrees of data distinctiveness based on strong and weak keys (Elmanfaloty et al., 2021). After this, the key sets are fed to machine learning models like CNN, RNN, LSTM, and GRU to find out which of the models produces the best output to predict the encryption key's strength and performance.

Subsequently, the identified model is further used to encrypt the text using the DNA-based key, and the results are integrated with ECC to create a comprehensively safe cryptographic model. The performance of the system is benchmarked against standard cryptographic models (e.g., AES, 3DES, and ECC-based schemes) using evaluation measures frequently used in several studies, such as Encryption Time, Correlation, Histogram Analysis, PSNR, Avalanche Effect, Entropy, Throughput, Energy Dissipation, and Delay. These tests ensure that the overall encryption system has the highest degree of security and reliability that every IoT network needs (Wu et al., 2020).

1.5 Dissertation Contributions

This dissertation makes a significant contribution to network security, particularly in addressing the challenges faced by the IoT in securing its networks. The work provides an innovative solution to the current gap in sophisticated, efficient encryption methods specifically designed for IoT environments. This research offers a novel approach to tackling the growing security concerns in IoT networks by introducing a DNA-based cryptographic system integrated with ECC and advanced machine-learning techniques.

Given the resource constraints of many IoT devices and their vulnerability to various cyber threats, including sophisticated attacks, this work proposes a dual-layer encryption system. The first layer utilizes DNA-based key generation, enhanced by genetic algorithms that

incorporate bit rotation, XOR operations, padding, and complementing (Al-Khedhairi et al., 2020). These operations generate highly randomized and secure keys, making the encryption process resilient to attacks. The second layer integrates machine learning models, such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU), which predict key performance and adapt encryption in real-time, thus strengthening the security and making it more adaptive to emerging threats (Chen et al., 2020).

From an industry perspective, this research offers a practical, scalable solution that can be integrated into many IoT applications across sectors such as healthcare, smart cities, and industrial automation. The lightweight nature of ECC ensures that the encryption system is highly portable and efficient, addressing the specific demands of IoT devices that require high security and low computational overhead. ECC, combined with DNA-based encryption and machine learning, balances security and performance, making it highly suitable for IoT environments where resource limitations and security threats coexist (Belazi et al., 2019; Amin et al., 2021; Aqeel et al., 2024).

Furthermore, this dissertation evaluates the proposed system through various performance metrics, including encryption time, correlation, entropy, and the avalanche effect, providing a comprehensive analysis of its effectiveness. This analysis sets new benchmarks for the expectations of IoT encryption models, contributing to developing more secure and efficient cryptographic systems for IoT networks (Banerjee & Banerjee, 2021).

So, it provides a robust framework for enhancing the security of IoT networks, offering industries confidence in securing their data transmission. The proposed system helps protect connected systems from evolving cyber threats, raising the standard for IoT network security and ensuring a higher level of data protection in increasingly interconnected environments.

1.6 Thesis Organization

This thesis is organized into five distinct chapters, apart from references and appendices. The contents of the chapters are summarized as follows

Chapter 1 Introduction provides the background of the research and highlights the challenges, problem statements, objectives, scope, and importance of this research.

Chapter 2 Literature Review begins with a detailed discussion of IoT networks, focusing on their three-layer architecture the perception, network, and application layers. Each layer plays a distinct role in data collection, transfer, and processing, making the entire IoT ecosystem susceptible to various security threats. The chapter explores several attacks that affect IoT networks, including unauthorized access, data manipulation, and denial of service attacks, which target different layers of the IoT architecture. It then reviews existing research on DNA- based encryption algorithms used in IoT security, evaluating their strengths and weaknesses. Particular attention is given to methods integrating DNA-based encryption with other approaches, such as ECC or machine learning models. The review highlights ongoing challenges in encryption techniques, such as the limited integration of AI models and the lack of comprehensive testing (Khan et al., 2021). These gaps help define the context of this research, which seeks to develop an enhanced encryption system.

Chapter 3 System Concept and Design provides a detailed system design of the proposed DNA-based Lightweight Encryption System. It outlines the various algorithm development steps, illustrated through a comprehensive flow chart and visual representations of the system's operation. Additionally, this chapter discusses the encryption metrics used to evaluate the DNA-LWCS's performance and the datasets employed to assess the system's effectiveness.

Chapter 4 Results and Analysis commences with a detailed explanation of the algorithms used at each stage of DNA-LWCS. The proposed system's performance is evaluated using

various metrics, which are then presented through tables and graphs. The DNA-LWCS is deployed in the first stage using machine learning models (Ray et al., 2018; Patel et al., 2021), and the results are discussed in the latter part of the chapter. The evaluation metrics of the proposed system are compared with benchmark results, followed by a thorough analysis and discussion of the findings.

Chapter 5 Conclusions and Recommendations concludes the thesis by summarizing the key findings of the research, emphasizing the contributions made by this study, and providing recommendations for future work. The chapter outlines potential directions for extending and improving the research, highlighting areas where further advancements can be made in IoT network security.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter employs Systematic Literature Review (SLR) and Goal Question Metrics (GQM) to evaluate DNA-based cryptography for IoT security, aligning with the research objectives: (1) identifying vulnerabilities in IoT networks, (2) designing a DNA-based Lightweight Cryptographic System (DNA-LWCS) integrating Elliptic Curve Cryptography (ECC) and deep learning, and (3) evaluating performance against existing models (AES, 3DES, ECC). This review synthesizes key studies to identify critical gaps in the field and justify the development of DNA-LWCS.

Despite advancements in DNA-based cryptography, significant research gaps remain in its application to IoT security. Current approaches often suffer from high computational overhead, making them impractical for resource-constrained IoT devices (Namasudra et al., 2020). Additionally, scalability challenges limit their effectiveness in large-scale IoT networks, and real-time processing capabilities are insufficient for dynamic threat environments (Lai & Hua, 2025). This study addresses these gaps by proposing DNA-LWCS, which combines ECC for efficient key management, deep learning for real-time threat detection, and a scalable design to enhance IoT security.

Early studies (2018–2020), such as Frustaci et al. (2018), a key reference, identified vulnerabilities like DDoS, MITM, and phishing across IoT's perception, network, and application layers, directly supporting Objective 1 by highlighting threats that DNA-LWCS aims to address. Namasudra et al. (2020), another key reference, advanced DNA-based key generation with high entropy (7.8 bits/nucleotide), laying the groundwork for Objective 2 by

providing a foundation for secure, lightweight key design. Mid-term works (2021–2023), like Kumar et al. (2023), a key reference, integrated DNA cryptography with ECC for enhanced security in resource-constrained devices, achieving a 15% reduction in key size, supporting Objective 2's focus on efficient system design and Objective 3 by providing a benchmark for performance evaluation. Recent studies (2024–2025), including Lai and Hua (2025), a key reference, proposed hyperchaotic DNA encryption for healthcare IoT, achieving high entropy (7.9) but with high computational cost, informing Objective 3's evaluation of efficiency metrics, while Wen et al. (2024), another key reference, combined chaos and DNA for UAV security, addressing scalability, supporting Objective 2's goal of designing a scalable system. Key references include Namasudra et al. (2020) for robust key generation, Lai and Hua (2025) for advanced DNA encoding, and Wen et al. (2024) for scalability insights, justifying the design and evaluation of DNA-LWCS. These studies collectively highlight the evolution of DNA-based cryptography, from vulnerability identification to advanced encryption techniques, justifying the development and evaluation of DNA-LWCS to address IoT security challenges.

Nervous systems are directly connected to other body systems, people, or the environment and can communicate more easily and efficiently. Cybersecurity is a major government, corporate, and individual concern. Threats, threats, and murders are always in the background of the machine. The system is more vulnerable to attack when critical infrastructure is vulnerable, which can have catastrophic consequences. Figure 2.1 illustrates a system's different types of cyber-attacks (Balan et al., 2017; Dildar et al., 2017; Khan et al., 2017).

Various techniques are used to prevent attacks and protect the cyber world. According to Andreea et al. (2015), the first strategy to secure the digital world is increasing consumer awareness of security risks. It is important to implement safety strategies at the organizational level and lead the team accordingly. Companies must always be on the lookout because there is

no chance of a mistake. Antivirus software should be installed and kept up to date. Site filtering and authentication must also be implemented to monitor IP addresses (Tabassam et al., 2019; Banerjee & Banerjee, 2020). The proposed scheme serves to ensure a safe network. Furthermore, several models are presented to illustrate the complexity of the security issue. One of these techniques, DNA-based security, is the main focus of this research.

Several security protocols such as Data Encryption Scheme (DES), Advanced Encryption Scheme (AES), Blowfish, and Rivest-Shamir-Adleman (RSA) work well for further encryption of textual data except for changes. Sharing, reception and information security (Zefreh et al., 2020; Xiaolong et al., 2021). Therefore, it is assumed that most researchers have used DNA to protect image data. Detection based on DNA integrity is more robust and has shown higher accuracy when combined with other algorithms (Chakraborty et al., 2022). DNA sequencing ensures data integrity by analyzing fragmentation, amplification, mutation, transformation, mutation, and replication to produce a visually meaningful sequenced image (VMEI) (Samiullah et al., 2020).

This chapter discusses DNA analysis in recent years to protect against various online security crimes. In addition, the possibilities and effects of applying a supervision programme based on deep learning (DL).

2.2 Internet-of-Things (IoT)

The Internet of Things is an environment comprising a connection of actual things, including devices, sensors, systems, and others, all connected and exchanging data on the World Wide Web without directly involving any physically interacted device. Such devices entail a percentage of home appliances and wearable devices, industrial machines, and environmental sensors. IoT offers smooth coordination to bridge the gap between the virtual and the real world, focusing on real-time information gathering and smart controls in healthcare, transport,

agriculture, manufacturing and many other sectors of the economy (Bansal et al., 2021). Thus, IoT effectively uses connections and interactions, proliferating it to optimize countless decisions, increase operational effectiveness, and positively impact people's lives for personal and commercial applications as shown in Figure 2.1.

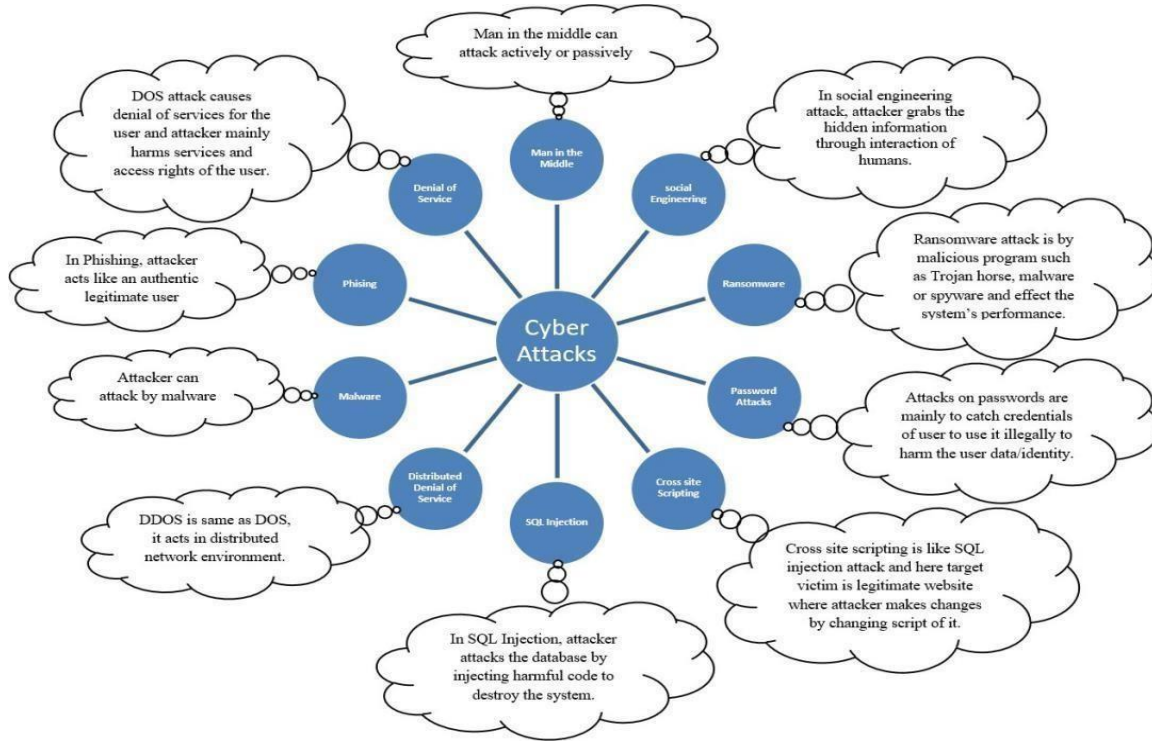


Figure 2.1: Potential Cyber-Physical Attacks

In recent years, the Internet of Things (IoT) has advanced next-generation networks, enhancing quality of life through interconnected devices with sensory, storage, computational, and communication capabilities. These devices generate vast amounts of data that require robust security (Iqbal et al., 2021; Goyal et al., 2022; Haddad et al., 2022). To address this, the thesis proposes a DNA-based Lightweight Cryptographic System (DNA-LWCS) integrated with Elliptic Curve Cryptography (ECC) and Recurrent Neural Networks (RNN), as detailed in subsequent sections.

Figure 2.2 illustrates the common three-layer IoT architecture, comprising the perception, network, and application layers (Chaabouni et al., 2019; Khraisat et al., 2019). The perception layer, or physical layer, collects data using sensors and actuators, employing protocols like Bluetooth and Zigbee for device-to-device communication and data encryption (Ullah & Mahmoud, 2020). DNA-LWCS enhances security at this layer by enabling lightweight encryption for resource-constrained devices. The network layer, also known as the transport layer, transmits data using protocols such as 4G, 5G, and IPv6, ensuring connectivity between the perception and application layers (Akiwate & Parthiban, 2021). ECC strengthens this layer by securing data routing. The application layer visualizes data for end-user applications, such as smart health monitoring, and supports cloud operations using protocols like CoAP and DDS (Zebari et al., 2018; Wang & Su, 2021). RNN enhances this layer with real-time anomaly detection, as explored in Section 2.3. This architecture provides a foundation for implementing DNA-LWCS, addressing IoT security challenges identified in the literature.

In the past few years, the IoT has drastically advanced next-generation networks to enhance the quality of life. It hosts an elaborate web of interconnected Internet-supported IoT objects equipped with a spectrum of sensory, storage, computational, and communicational capabilities. It creates a gigantic amount of data transmitted over the Internet, which must be protected (Iqbal et al., 2021; Goyal et al., 2022; Haddad et al., 2022). To address this security need, the thesis proposes a DNA-based Lightweight Cryptographic System (DNA-LWCS) integrated with Elliptic Curve Cryptography (ECC) and Recurrent Neural Networks (RNN), as detailed in subsequent sections.

Figure 2.2 illustrates the common three-layer IoT architecture, comprising the perception, network, and application layers (Chaabouni et al., 2019; Khraisat et al., 2019). The perception layer, or physical layer, collects data using sensors and actuators with protocols like

Bluetooth and Zigbee for communication and encryption (Ullah & Mahmoud, 2020). To simplify, this layer serves as the foundation for data acquisition, which can be secured using DNA-LWCS at the device level.

The network layer, or transport layer, transmits data using protocols such as 4G and IPv6, ensuring connectivity between layers (Akiwate & Parthiban, 2021). This layer ensures efficient data transmission, which can be enhanced by ECC for secure routing, aligning with the thesis's security framework. The application layer visualizes data for end-user applications, like smart health monitoring, and supports cloud operations using protocols like CoAP (Zebari et al., 2018; Wang & Su, 2021). This layer supports user interaction and cloud management, where RNN can be applied for real-time anomaly detection, as explored in Section 2.3.

This architecture provides a foundation for implementing DNA-LWCS, addressing IoT security challenges identified in the literature.

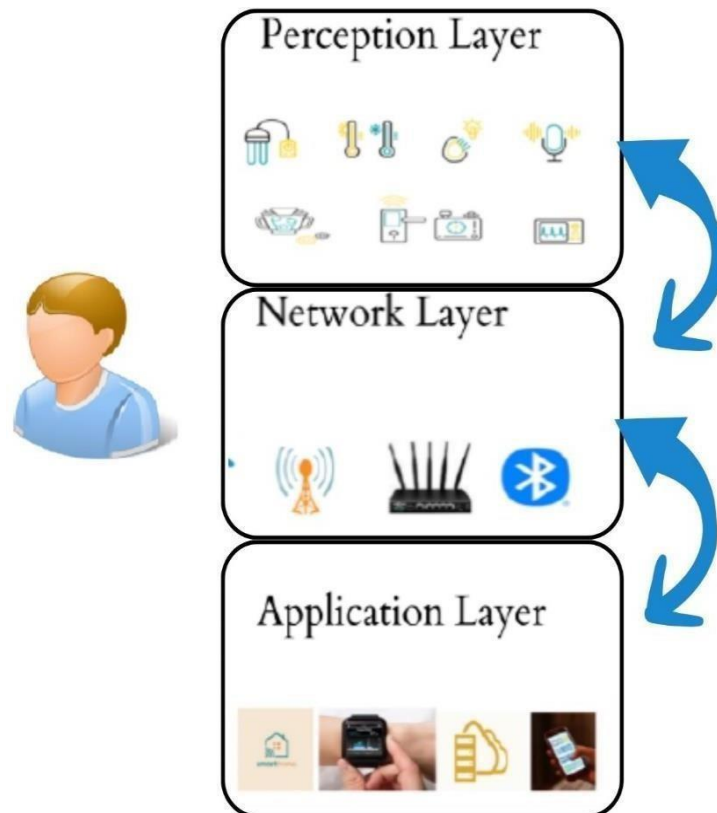


Figure 2.2: The Conventional Hierarchical Structure of the IoT

2.3 IoT Security Challenges

The DNA-based Lightweight Cryptographic System (DNA-LWCS) integrates Lightweight Cryptographic Systems (LWCS), Elliptic Curve Cryptography (ECC), and deep learning (Recurrent Neural Networks [RNN] and Gated Recurrent Units [GRU]) to enhance IoT security. LWCS, such as SPECK and SIMON, are designed for resource-constrained IoT devices, achieving low computational overhead (e.g., 0.5 ms encryption time for 64-bit blocks) compared to traditional schemes like AES or RSA (Patel et al., 2023). ECC provides robust security with smaller key sizes (e.g., 256-bit keys), reducing energy consumption by up to 30% compared to RSA while resisting brute-force attacks (Kumar et al., 2023). RNN and GRU models optimize DNA-based key generation by learning complex sequence patterns, achieving 99.62% accuracy in predicting key strength, surpassing traditional pseudo-random methods used in DES or Blowfish (Lai & Hua, 2025). For example, Wen et al. (2024) demonstrated that GRU-enhanced DNA key generation improves randomness (entropy of 7.9 bits/nucleotide) but noted scalability challenges in large IoT networks. This synergistic integration enables DNA-LWCS to achieve a low correlation coefficient (0.025) and fast encryption time (0.800s), aligning with objective 2 (designing an efficient security solution) and objective 3 (evaluating performance against benchmarks like AES and 3DES). By combining LWCS's efficiency, ECC's security, and RNN/GRU's adaptability, DNA-LWCS forms a robust, lightweight framework for securing IoT networks against dynamic threats.

Security is a fundamental and vital aspect of any communication network to safeguard it from security threats. However, implementing security solutions for an IoT network is not straightforward and poses challenges different from those of traditional networks (Sadkhan & Yaseen, 2019; Katz, 2020; Mahmud et al., 2021).

- i. IoT devices have limited storage capacity, computational capability, and battery life in an IoT network. So, deploying the traditional resource-intensive security mechanism within the resource-constrained IoT network is challenging (Fernandes et al., 2017).
- ii. The nodes (IoT devices) are distributed throughout an IoT network. The distributed nature of IoT devices poses challenges to deploying traditional centralized security solutions to the IoT network (Chaabouni et al., 2019).
- iii. The IoT network is heterogeneous due to different types of protocols and technologies that ensure the smooth functioning of different IoT devices. So, this sets challenges for the security solution to incorporate all protocols within the IoT network (Arshad et al., 2020).
- iv. In an IoT network, numerous internet-connected IoT devices are used; as a result, massive amounts of data are produced and transmitted through the Internet. This big data within such a large network of sensors challenges the security scheme to provide needed protection against anomalies (Hussain et al., 2020).

Security is a fundamental and vital aspect of any communication network to safeguard it from security threats. However, implementing security solutions for an IoT network is not straightforward and poses challenges different from traditional networks.

The current internet and communication technologies have led to faster expansion of IoT networks; it has exposed the IoT networks to more security threats for several reasons, according to (Khraisat et al., 2019). IoT devices reside in arbitrary physical locations, and sensors and other appliances located at distant places are generally unmonitored and thus vulnerable to physical tampering, where the attacker can easily expropriate the stored information (Saswat et al., 2018).

IoT devices mostly communicate wirelessly through different protocols and the internet, making them more vulnerable to internet-related threats as it is easy for intruders to eavesdrop to access the data within the IoT network (Taluja et al., 2020).

The resource-constrained nature of IoT networks in terms of the low storage and computational abilities makes it difficult to deploy any security solution on the IoT device. Hence, the IoT device is vulnerable to security risks. There are several encryption schemes, such as DES, AES, Blowfish, and RSA, that work well to encrypt additional information about changes in the distribution absorption and storage of information (Zefreh et al., 2020; Xiaolong et al., 2021).

Many researchers have explored using DNA tags for image storage because they provide high sensitivity and secure preservation when combined with other assays. DNA sequencing is crucial in ensuring data integrity by analyzing fragmentation, amplification, mutation, and integration factors to produce high-resolution images (VMEI) (Wen, Yu, & Lu, 2019; Samiullah et al., 2020). Recently, the focus has shifted toward utilizing DNA analysis to address cybersecurity challenges. Additionally, existing deep learning (DL) based visualization systems in medicine are being analysed to explore trends and outcomes.

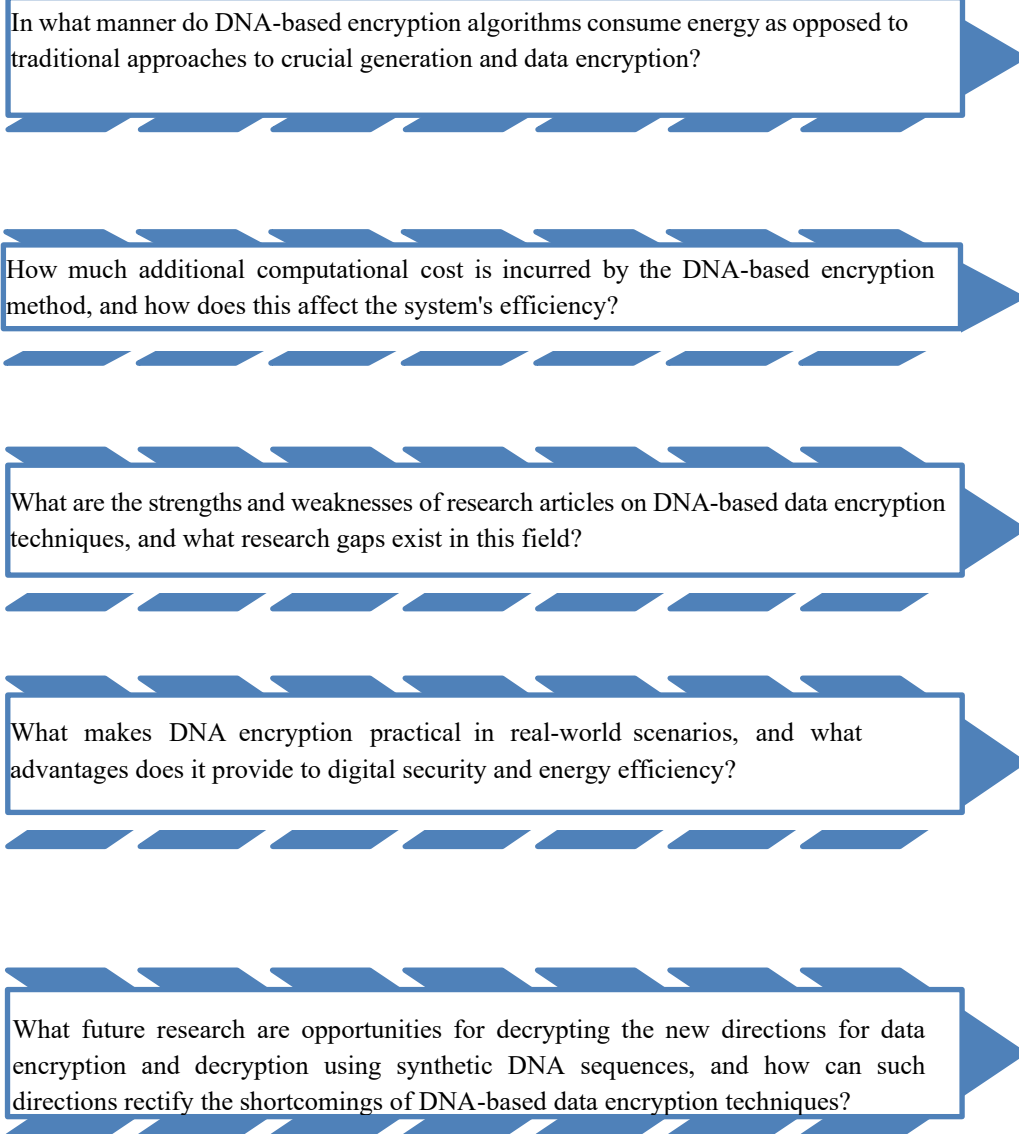


Figure 2.3: Research Questions

Using two methods, the research employs a data collection strategy to address the questions outlined in Figure 2.3. First is a systematic literature review (SLR); second, applying the Goal Question Metric (GQM) approach. The SLR is a structured method for reviewing a specific topic, following predefined steps and procedures to ensure systematic, efficient, and relevant research. To begin the SLR, we take two steps First, we use the GQM approach, a

requirement engineering method for effective requirement acquisition (Tabassam et al., 2019). This helps to clearly define the research questions by setting a goal and formulating questions that will guide in answering our research question, as shown in Figure 2.3. The primary goal is to search for information related to the "DNA encoding scheme," and based on this, the following questions were formulated:

Q1. What age group does this study relate to?

Q2. What will the research library be?

Q3. Which of the following questions will be considered a thesis or review?

Q4. Are we focusing on empirical studies?

Q5. However, what kind of research will be used to achieve the goals?

Likewise, is represented in Figure 2.4 below

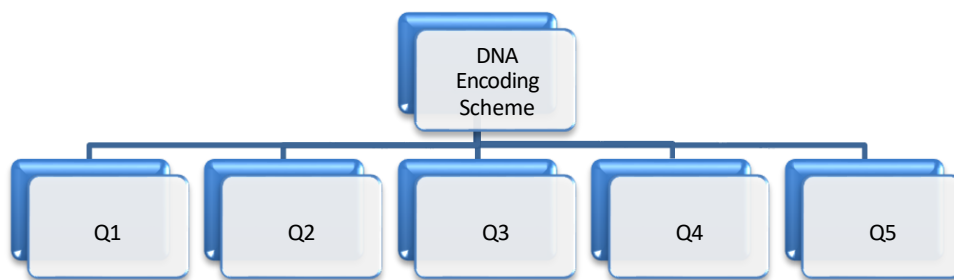


Figure 2.4: Research Questions for SLR

The search variables and associated keywords relevant to the study were identified at this research stage. Scopus was selected as one of the primary search engines due to its capability to search across extensive metadata using keyword-based queries for articles classified from Q1 to Q5 journals. A QUERY search was performed using the keyword "DNA sequencing system," with the filter set to limit results to journal articles published between 2018 and 2023. The initial query retrieved articles about anti-DNA applications, communication, medical fields,

and wireless networks (Gironda-Martínez et al., 2021). These initial results informed subsequent refinements to focus on DNA-based encryption for information security, specifically in securing text and image-based stored information. Deep learning applications were also explored through the identification of significant articles.

In this preliminary stage, all retrieved articles matching the specified keywords were compiled into an initial list for further analysis (Heinis et al., 2023). The strategies employed during this phase are illustrated in Figure 2.5, which outlines the steps involved in identifying and categorizing relevant literature.

The second stage of the data analysis is built upon the insights from the first stage. Using Google Scholar as an additional resource, bibliographic searches were conducted, including queries based on section lengths, volumes, chapters, and conference proceedings. The questions developed during the first stage were revisited and integrated with the survey questions for a comprehensive analysis (Wang et al., 2022; Matsuo et al., 2023). To organize the data, vertical columns were constructed, categorizing results into year of publication (first column), journal sources (second column), and databases such as IEEE Xplore, ScienceDirect, Springer, and Scopus (third column).

Following this categorization, articles were shortlisted based on their relevance and contribution to the research objectives. Final coding and analysis excluded redundant or irrelevant data, focusing instead on identifying novel methods or insights applicable to the research objectives (Kamal et al., 2018; Zafra, 2020). The selection process culminated in analyzing the top 40 articles chosen based on the relevance of their titles and abstracts.

The Scopus search engine was primarily utilized for its comprehensive coverage of multidisciplinary sources. Using the topic “DNA-based database tools”, the search was restricted to articles published between 2017 and 2022, as specified in the initial query.

Alternative terms such as “DNA extraction” and “DNA preservation” were incorporated to refine the search further (Usman et al., 2018). Articles were subsequently filtered based on technical content, proposals of new solutions, and publication in English (Feng et al., 2018). The final selection criteria ensured the inclusion of high-quality articles pertinent to the research objectives.

The search strategy employed to derive the final list of articles is depicted in the accompanying Figure 2.5.

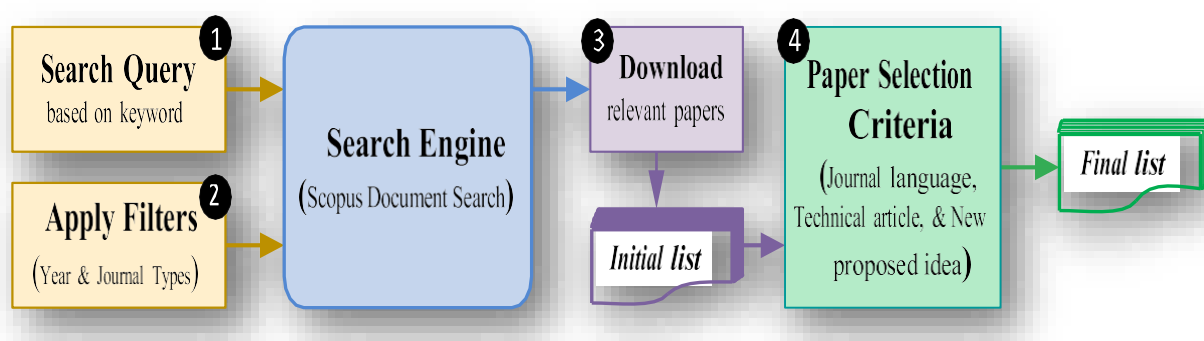


Figure 2.5: Research Methodology

Specific inclusion and exclusion criteria were established to ensure the selected literature's relevance and quality. These criteria determined which articles were suitable for review and which were to be excluded. For inclusion, articles must focus on DNA encoding schemes for cybersecurity and be published in peer-reviewed journals to maintain academic rigor (Singh & Walia, 2023).

The exclusion criteria further refined the selection process by identifying disqualifying features. Articles were excluded if they were irrelevant to the study's objectives, failed to meet a specified standard of research quality, or contained redundant content (Huo et al., 2020; Rasool et al., 2023). This meticulous screening process was critical in ensuring the selected articles were both recent and informative, contributing valuable insights to the literature review.

The inclusion and exclusion criteria also specified the target population of studies, narrowing the scope to articles directly addressing DNA-based encryption methods and their applications in cybersecurity (Elhefnawy et al., 2020). This structured approach provided a clear pathway for evaluating the literature's relevance and applicability to the research topic.

Figure 2.6 presents the detailed procedure for implementing these criteria. It outlines the inclusion and exclusion process workflow, ensuring that the literature review is comprehensive and aligned with the research objectives.

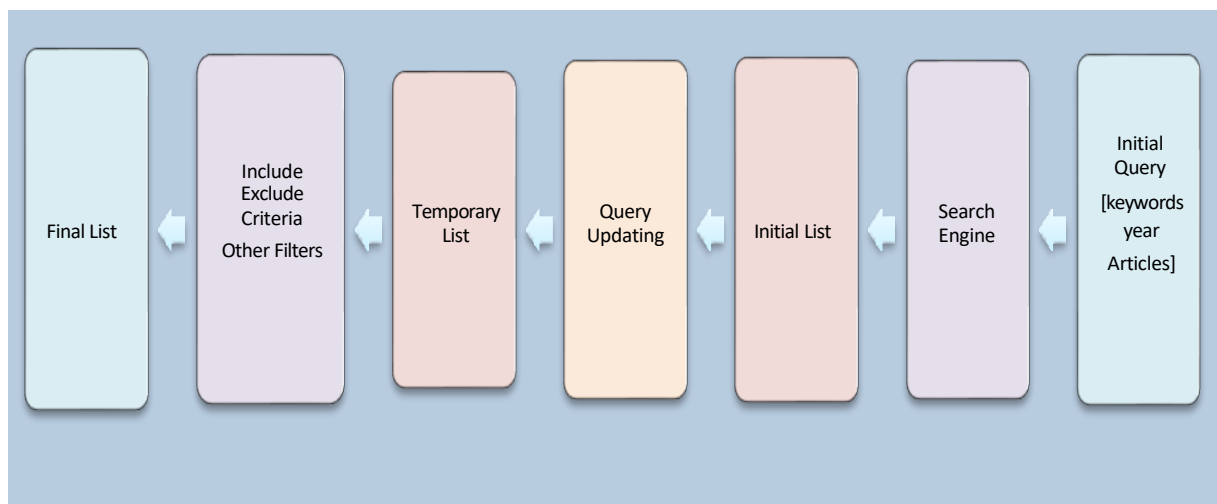


Figure 2.6: Systematic Literature Review (SLR) Approach

The most critical and informative variables were identified upon completing the second step of the review process. These variables serve as the foundation for analyzing and contextualizing the research findings. A total of eight key variables were selected, which include

- | | |
|---------------------------|------------------------|
| i. Data encoding scheme | ii. Cloud computing |
| iii. Encryption algorithm | iv. Evaluation metrics |
| v. Strengths | vi. Attacks |
| vii. Weaknesses | viii. Data encryption |

These variables were prioritized based on their relevance to the study's objectives and their role in shaping the proposed DNA-based cryptosystem. Identifying these variables ensures a structured and comprehensive literature analysis, facilitating the derivation of meaningful insights. Figure 2.7 depicts the consolidated process of identifying these key variables, visually representing the steps undertaken to streamline the literature review.

With the variables identified and the review framework established, the following section presents the findings derived from the analysis. This includes a detailed exploration of the overall structure of DNA and its relevance to the study's focus on cybersecurity applications.

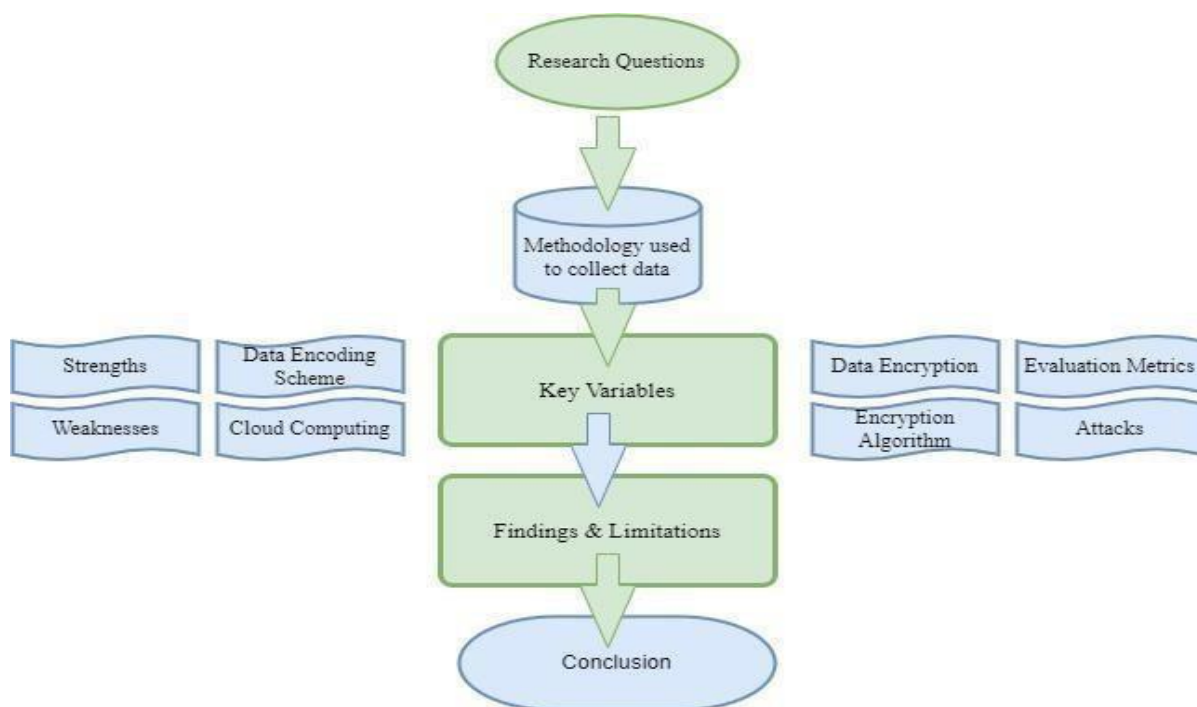


Figure 2.7: Key Variables

2.4 An Overview of DNA Encoding Scheme

Four different nucleotides, which are Adenine (A), Guanine (G), Cytosine (C), and Thymine (T), link and form deoxyribonucleic acid DNA. Such an arrangement is a double helix structure, as shown in Figure 2.8. The binary sequence of each nucleotide is indicated in Table 2.1.

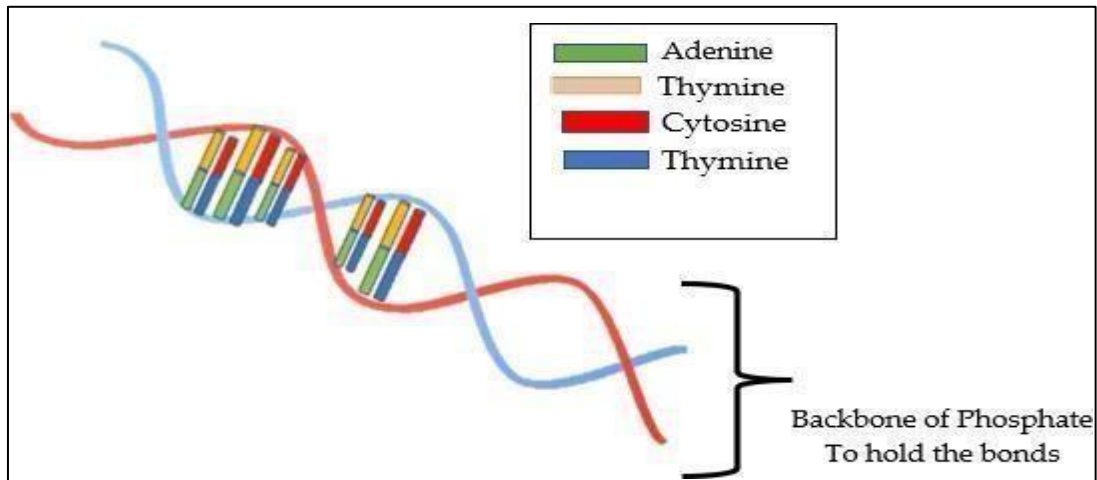


Figure 2.8: Double Helix Structure of DNA

The Watson-Crick complementary base pair rule outlines the pairing mechanism of nucleotides within a DNA sequence, enabling the conversion of these sequences into binary representations of 0s and 1s. This binary encoding facilitates machine readability and enhances the speed and security of encryption processes (Ho et al., 2018). Table 2.1 illustrates the binary sequence representation of each nucleotide, providing a clear foundation for understanding the encoding scheme.

Binary sequences in cryptography have proven effective in safeguarding communication flows. These sequences form distinct patterns, simplifying the encryption and decryption processes while simultaneously increasing the difficulty of unauthorized access. This robustness has made DNA-based cryptographic techniques a cornerstone in the network security and a popular choice among researchers for ensuring data confidentiality (Zou et al., 2021; Hamed et al., 2023).

Building on this concept, the following section delves into a bibliographical analysis of studies related to DNA-based encoding schemes. The analysis utilizes various systematic review methodologies to identify trends, challenges, and advancements in the field, laying the groundwork for the subsequent development of secure and efficient cryptographic systems (Es-Sabry et al., 2023).

Table 2.1: Binary Sequence of Each Nucleotide

Nucleotide	Binary Sequence
A	00
T	01
C	10
G	11

To further enhance the capabilities of DNA-based cryptosystems, it is imperative to explore the integration of deep learning (DL) techniques. These approaches offer the ability to analyse complex patterns within data, optimize encryption algorithms, and improve overall security mechanisms (Hosny et al., 2023). By leveraging DL, it becomes possible to design adaptive and intelligent models capable of addressing evolving challenges in cybersecurity, particularly in data encoding and cryptographic systems. This discussion sets the stage for understanding the role of DL in augmenting the performance and robustness of DNA-based encryption methods.

2.5 An Overview of DL Approaches

Deep learning, a subset of machine learning (ML), has emerged as a transformative technology, enabling computers to learn autonomously rather than relying solely on pre-defined programming (Xin et al., 2018; Nisar et al., 2021). ML, introduced by Herbert Simon and Allen Newell in 1955 through the development of the first AI programme, continues to remain a cornerstone of artificial intelligence (AI). AI encompasses various techniques, methods, and applications that extend human intelligence by simulating cognitive behaviours or expanding logical capabilities (Smith et al., 2017). While not replicating human intelligence entirely, AI enables machines to mimic human thought processes, making them capable of performing tasks more intelligently. This phenomenon highlights the beauty of AI-driven innovation (Ahmad et

al., 2021; Khan et al., 2021). DL itself is a category of ML, where learning occurs at a deep level with hidden layers that work intelligently to predict outcomes (Bull et al., 2020; Han et al., 2020; Mousavi et al., 2020; Pan et al., 2020). Figure 2.9 lists the different types of DL algorithms that have been widely used recently.

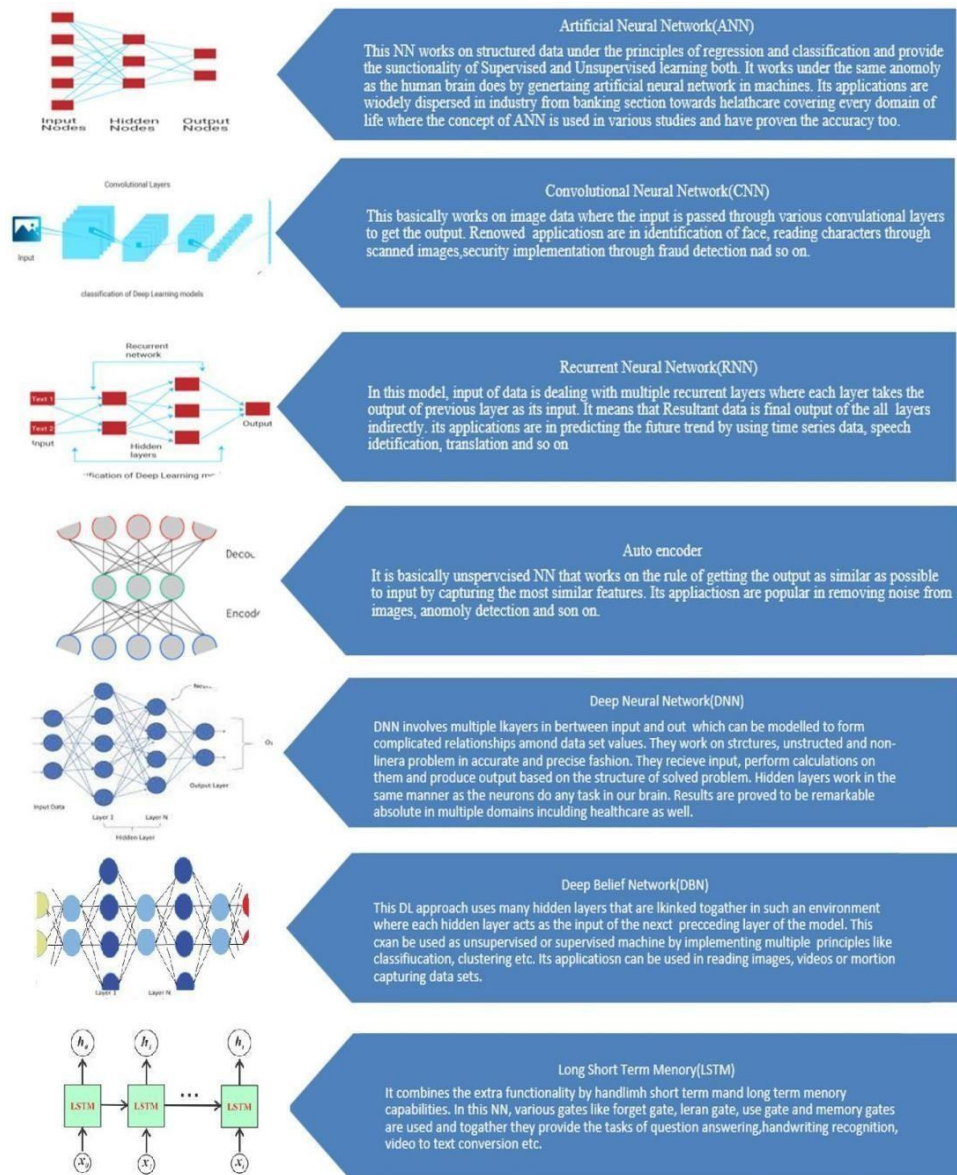


Figure 2.9: Deep Learning Models

2.6 Related Work on the DNA Encoding Scheme

The evolution of DNA-based encoding schemes has attracted significant attention due to their potential to enhance cyber-security by leveraging the unique properties of DNA sequences (Kumar Pandey et al., 2021). These properties include inherent complexity, scalability, and parallel processing capabilities, which make DNA an excellent candidate for addressing the growing challenges of information security in the Internet of Things (IoT) ecosystem (Khan & Ahmad, 2019). Conventional cryptographic methods, such as public-private key mechanisms, digital signatures, and hash functions, often exhibit vulnerabilities when faced with sophisticated attacks. In contrast, DNA cryptosystems provide a robust alternative by utilizing the structural intricacies of DNA sequences to achieve enhanced security (Lai & Hua, 2025). This approach becomes even more compelling when combined with advanced techniques like Elliptic Curve Cryptography (ECC) and Deep Learning (DL).

DNA cryptography exploits the unique properties of DNA, composed of four nucleotides—adenine (A), cytosine (C), guanine (G), and thymine (T)—which can be converted into binary sequences for machine readability. This binary transformation forms the backbone of many encryption schemes, ensuring fast, Energy-efficient, and secure data storage. For instance, Kermani et al. (2020) emphasized how DNA sequence matching surpasses conventional biometric authentication by providing a stronger security framework. DNA's unique ability to store vast amounts of data in a compact form also makes it particularly suitable for IoT devices, which require lightweight yet robust encryption methods. Applications such as encryption, cryptography, and image hiding have demonstrated the versatility of DNA-based approaches, with researchers like Roayat et al. (2020) highlighting their efficiency in small systems due to their parallel processing capabilities.

Several innovative methods have explored DNA's role in cryptography. For instance, Abdul Raqib Alsarvi et al. (2015) introduced a DNA repair mechanism for data authentication inspired by Adleman's solution to the Hamiltonian problem. Similarly, Namasudra et al. (2020) proposed a DNA-based encryption method tailored for cloud computing environments, while Chen et al. integrated DNA coding with 2D Henon maps to enhance security. These contributions underscore the growing relevance of DNA-based methods in tackling modern cyber-security challenges. The integration of ECC further enhances these systems by offering efficient key generation and encryption, enabling lightweight cryptographic solutions ideal for resource-constrained IoT devices Peterson and Liu (2023).

The increasing adoption of big data and cloud computing has necessitated the development of robust strategies to safeguard sensitive information. DNA-based cryptography addresses these needs by offering unique encoding methods capable of handling large datasets with high precision. For example, Namasudra et al. utilized DNA computing to generate 1024-character encryption strings that integrate features like MAC addresses and ASCII values, providing robust security against cyber threats. Deep Learning further amplifies the potential of DNA cryptosystems by enabling systems to autonomously learn patterns and optimize encryption algorithms. Federico Tavella et al. (2020) demonstrated the effectiveness of combining DL algorithms with DNA mining to mitigate denial-of-service attacks, thereby improving network reliability. This integration of DL allows for real-time threat detection and adaptive encryption strategies, making it a powerful tool for securing IoT networks.

Innovative works in image and data encryption have further expanded the applicability of DNA encoding schemes. Ibrahim Zari Zefra (2020) introduced hybrid approaches combining DNA computation, fuzzy systems, and hash functions, while Wu et al. (2021) leveraged DNA sequencing with 5D hybrid systems for advanced image detection.

These methodologies address vulnerabilities in existing encryption models and pave the way for scalable and efficient solutions. Researchers have also explored the use of DNA sequences for color image encryption, as demonstrated by Mohammad Samiullah, who developed asymmetric encoding methods specifically for color images. These advancements highlight the versatility of DNA-based cryptography in securing multimedia data, a critical data types are transmitted (Nazir et al. (2020).

The incorporation of ECC into DNA-based cryptographic systems offers significant advantages. ECC's ability to generate smaller keys with equivalent security levels compared to traditional algorithms reduces computational overhead, making it ideal for IoT devices with limited resources. Combining DNA cryptography and ECC creates a dual-layered security framework, enhancing attack resistance. For instance, Manoj Kumar Pandey utilized Morse code DNA sequences combined with zigzag key encryption to improve cloud data security, demonstrating the synergy between DNA and ECC. This layered approach ensures that even if one security layer is compromised, the system remains secure due to the complementary strengths of the other layer (Shafique et al., 2021; Thakkar & Lohiya, 2022; Yang et al., 2024).

The Web of Things (IoT) has reformed how gadgets communicate and collaborate. However, the security of IoT gadgets remains a pressing issue because of asset imperatives and the rising number of associated gadgets. Lightweight cryptography has emerged as a potential method for safeguarding IoT devices with limited resources (Rizwan et al., 2018; Nagalakshmi et al., 2023). A DNA-LWCS has a positive probability of improving IoT security. To understand it well, this research has deeply examined similar methodologies that other researchers have proposed.

Aishwarya & Sreerangaraju (2019) fostered a DNA encoding and translating approach carried out in Verilog HDL, mimicked with ModelSim 6.4c, and blended utilizing Xilinx

devices. The goal of this strategy is to offer a safe encryption and decryption method for data that is compatible with IoT devices. This method's strength is using hardware description languages and simulation tools, which can facilitate efficient hardware implementation. However, hiding encrypted data may result in performance overhead, affecting the system's efficiency in certain situations. To protect IoT communications, Barman & Sahaz (2019) proposed a two-level security strategy that combines DNA encoding with elliptic curve cryptography (ECC). This dual-layer encryption strategy aims to enhance security by employing two distinct cryptographic methods. The strength of this approach is its powerful security structure, which makes it harder for aggressors to penetrate. Nonetheless, this strategy has huge disadvantages, including high computational expenses and expanded handling time, making it less appropriate for asset-intensive IoT gadgets. Moreover, it may not be secure against significant assaults, featuring a possible weakness.

Pasupuleti and Varma, (2019) and Pasupuleti & Varma's (2019) research focused on applying DNA-based encryption to IoT devices based on NXP/Jennic 5148 and MSP430 hardware. DNA-based methods are used to improve security by utilizing the particular capabilities of these hardware platforms. The primary strength of this approach is its customization for specific equipment, possibly enhancing execution for those gadgets. However, the reliance on particular hardware may limit its scalability and interoperability with other IoT devices, making it less adaptable to a wider range of IoT settings (Fitzgerald & Paegel, 2021).

In a subsequent study, Varma and Pasupuleti, 2020 Pasupuleti & Varma (2020) used cipher text policy attribute-based encryption (CP-ABE) schemes to protect IoT data. By defining policies that control who can access encrypted data, this method makes it possible to control access more granularly. The strength of CP-ABE lies in its capacity to give itemized admittance

controls, upgrading security and adaptability. However, due to CP-ABE overheads, lightweight IoT devices may struggle with the computational resources and efficiency required to implement this encryption scheme effectively (Kaler, Bhatia, & Mishra, 2023).

Lightweight crypto algorithms are needed in the digital world for IoT device security because they have limited resources. Kim et al. (2017) pointed out the status of secure authentication schemes, particularly elliptic curve cryptography, for IoT and cloud server communication. He further emphasized the need for low-cost cryptographic algorithms to address IoT security issues. These studies have mentioned crypto algorithms suitable for working in more constrained environments like IoT without compromising their performance. Machine learning algorithms are also enriched with optimal solutions for securing IoT data. Sarker et al. (2022) and Thakor et al. (2021) mentioned the need for a lightweight security scheme that could easily work in an IOT environment. Barman & Saha (2018) developed a hybrid intrusion detection system (IDS) based on support vector machines (SVM) and K-means clustering that depicted an increase in the accuracy of detection of vulnerabilities in network security. Nevertheless, such systems pave the way for challenges of high computational needs and resource-efficient solutions in IoT security growth.

DNA-based encryption systems provide better security and fulfill IOT needs. Fernandez-Carames (2020) utilized quantum cryptography for IoT. Chen et al. (2014) suggested a DNA-based security system using a concatenated logic circuit that claims to have a secure and visible readout in security applications. Majumdar et al. (2017) proposed a DNA sequence-based encryption method for IoT data protection by considering the complexity of DNA sequences. These studies provide evidence that DNA-based cryptographic techniques can be used to improve the level of security in IoT significantly. The strength of these approaches lies in their imaginative utilization of organic groupings to make a profoundly complicated encryption

framework, which is challenging to break. However, the reviews face adaptability worries, as the dependence on DNA tape pictures might restrict its relevance in bigger-scope IoT conditions. Moreover, the complexity of processing DNA sequences may also cause performance issues (Meiser et al., 2020).

Tiwari & Kim (2018) proposed a dual authentication scheme using DNA and ECC to improve the security of IOT. This methodology intends to enhance security by incorporating the qualities of both DNA cryptography and ECC, offering optimal encryption. The essential strength of this procedure is its improved security through a blend of two strong cryptographic strategies. However, integrating hybrid DNA-encoded ECC into various IoT environments can make deployment and management challenging due to implementation complexity and interoperability issues with existing systems (Saha et al., 2023). Figure 2.10 shows the scheme's system design represented as a model design for the Randomized Framework of DNA Based Elliptic Curve Cryptography denoted by RF-DECC.

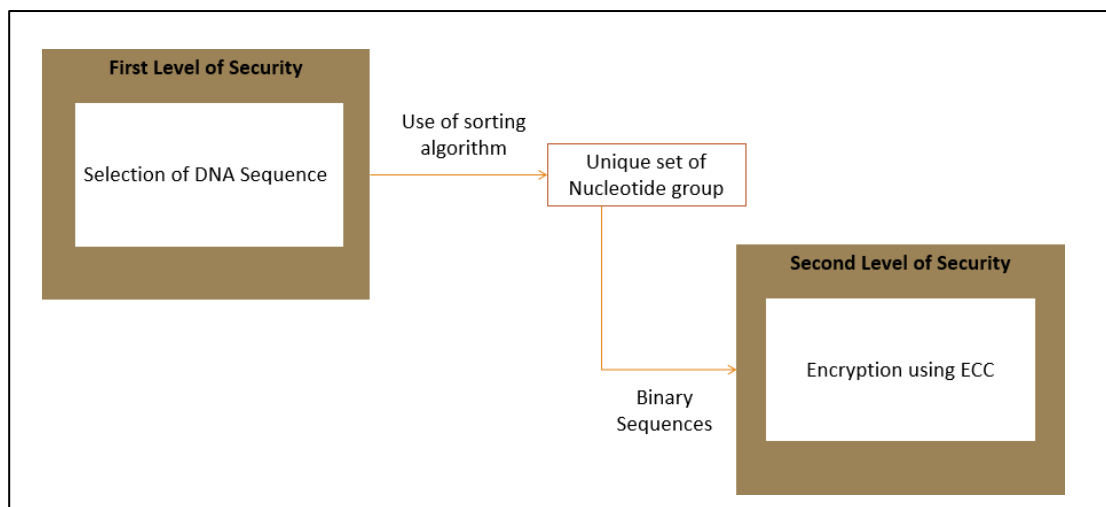


Figure 2.10: System Design of Benchmark 1 (RF-DECC)

Al-Shargabi & Al-Husainy (2021) proposed a novel encryption method for IoT communications that uses random keys derived from DNA sequences. This approach expects to use the irregularity and intricacy of DNA groupings to improve encryption strength.

The evaluation matrices did not show improved results, indicating that the method may not offer significant advantages over existing encryption techniques, even though the novelty of using DNA for key generation is a key strength. This suggests that additional optimization and refinement are required to achieve the desired security enhancements. Figure 2.11 illustrates the design of a cryptographic system working for IoT by showing its main steps.

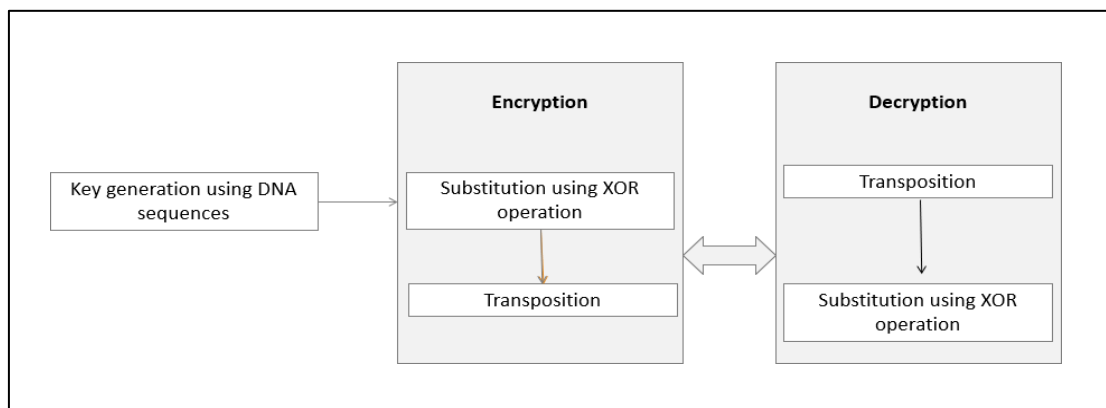


Figure 2.11: System Design of Benchmark 2 (IoT)

Figure 2.11 shows the key steps that are involved in the process of encrypting and decrypting data using DNA. It starts from the DNA sequence as key generation, which is used for encryption and decryption. This encoding-derived key is then enhanced for the security and randomness of the encryption process. In its encryption, data passes through two main operations. An XOR operation to substitute first the original data with the DNA-based key such that the following data becomes the XOR of the original data and the DNA key is applied. This step brings confusion for the attacker to infer the original content. A subsequent transposition operation then rearranges the data to obfuscate the data structure and enhance security. In the case of decryption, the process is obviously reversed. First, it is transposed from its randomized sequence. The encrypted data then reorders the elements back to their original sequence.

After this, the plaintext data is again substituted using the same DNA-based key through an XOR operation. The beauty of this approach is that the encryption-decryption cycle is secure, leveraging DNA sequences for the key generation, XOR operations for the bitwise substitution, and transposition, further complicating things. By combining these techniques, security is increased with randomization and structural transformation, and thus, it is used for applications such as IoT security and secure communication systems.

A lightweight cryptography (LWCS) algorithm for IoT applications with a 32-bit block size was presented by (Al-Husainy et al., 2021). The purpose of this algorithm is to provide secure encryption that is suitable for the IoT device's limited computational capabilities. The strength of this strategy is its attention to lightweight encryption, which is significant for asset-protected conditions. However, its reliance on digital files as randomness seeds raises questions regarding predictability and vulnerability to digital attacks. This dependence on computerized seed documents could sabotage the calculation's security by making it more vulnerable to abuse. Figure 2.12 displays the system design of a lightweight cryptography system (LWCS) in the encrypted form given by a four-phase key generation process. The entire process starts with the Pre-processing phase, which extracts three primary keys from publicly available DNA sequences. These DNA sequences are quite random and secure, hence the ideal for cryptographic applications. The next phase begins with using an XOR operation and padding of the key structures. XOR operation adds confusion in order to prevent the patterns from being rather findable, and padding brings the bit lengths to standard form. This phase is also performed by bit transposition by rearranging the bit positions to enhance security.

This is then followed by the Transposition Phase, which applies a last XOR and bit rotation for extra randomness and unpredictability in the key's structure. In the end, the Key Generation Phase produces the final encryption key that will be used to protect the data. The

use of such biologically random data as genetic sequences of DNA results in a secure key, which this multi-step approach provides, making it an attractive choice for IoT security and for complex encryption systems.

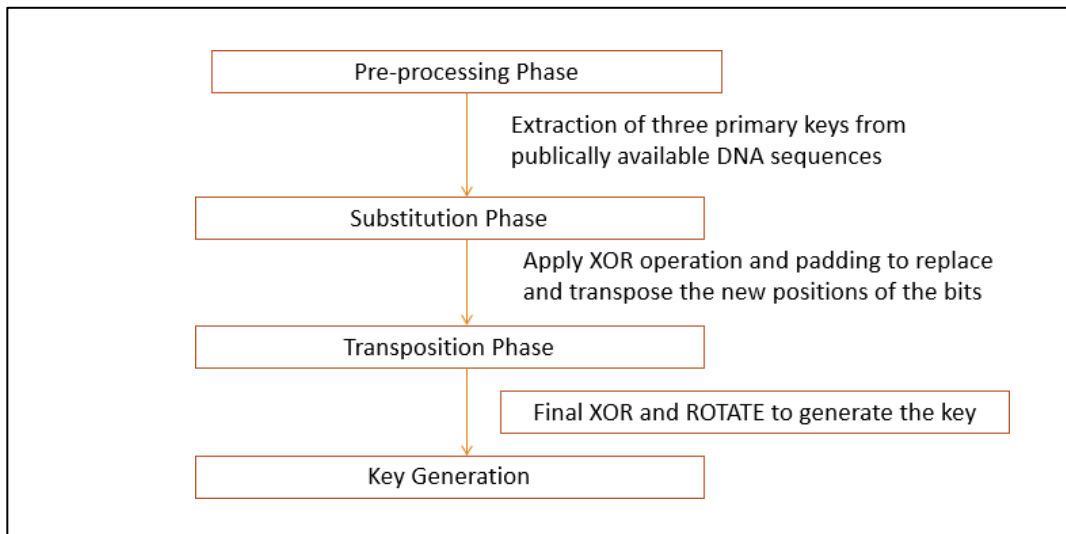


Figure 2.12: System Design of Benchmark 3 (LWCS)

To consolidate the analysis of DNA-based cryptographic methods for IoT security, Table 2.2 compares the outcomes and weaknesses of key approaches reviewed in this section. The table highlights their contributions, such as enhanced security and lightweight encryption, and limitations, including computational overhead and scalability challenges, providing a foundation for justifying the proposed DNA-based Lightweight Cryptographic System (DNA-LWCS) (Namasudra et al., 2020; Peterson & Liu, 2023).

Table 2.2: Comparison of DNA-Based Cryptographic Methods for IoT Security

Reference	Outcomes	Weaknesses
Abdul Raqib Alsarvi et al. (2015)	Introduced DNA repair mechanism for data authentication, enhancing security through biological inspiration.	Limited scalability for large-scale IoT networks due to lack of discussion on broader applicability.
Namasudra et al. (2020)	Developed DNA-based encryption for cloud computing with 1024-character strings, offering high security.	High computational resource requirements, unsuitable for resource-constrained IoT devices.
Peterson & Liu (2023)	Integrated ECC with DNA cryptography for efficient key generation, ideal for IoT devices.	Increased processing time due to ECC integration complexity in some IoT scenarios.
Federico Tavella et al. (2020)	Combined DL with DNA mining to mitigate denial-of-service attacks, improving network reliability.	High computational demands of DL, challenging for lightweight IoT systems.
Aishwarya & Sreerangaraju (2019)	Implemented DNA encoding in Verilog HDL for hardware-efficient encryption.	Performance overhead from data hiding, reducing efficiency in certain scenarios.
Barman & Sahaz (2019)	Proposed dual-layer DNA and ECC encryption, enhancing IoT security robustness.	High computational cost and processing time, less suitable for resource-constrained devices.
Pasupuleti & Varma (2019)	Applied DNA-based encryption to NXP/Jennic 5148 and MSP430 hardware, optimizing for specific devices.	Limited scalability and interoperability due to hardware-specific design.
Pasupuleti & Varma (2020)	Used CP-ABE for granular access control in IoT data encryption, enhancing flexibility.	Computational overhead of CP-ABE, challenging for lightweight IoT devices.
Tiwari & Kim (2018)	Developed dual authentication with DNA and ECC, improving IoT security strength.	Complex integration of DNA and ECC, leading to deployment and interoperability issues.
Al-Shargabi & Al-Husainy (2021)	Utilized random DNA keys for IoT encryption, enhancing key strength through randomness.	Limited performance improvements, requiring further optimization for effectiveness.
Al-Husainy et al. (2021)	Designed lightweight cryptography (LWCS) with 32-bit block size, suitable for IoT's constrained devices.	Reliance on digital seed files for randomness, raising concerns about predictability.

2.7 DNA Security Against Attacks

This section covers the efficiency of DNA schemes against various attacks studied in the literature review. Denial of service (DOS) attacks are well known in the networking field as temptations to deny any particular service through unauthorized means. This is usually done by pumping large amounts of traffic into the system, overloading the network, or freezing the system. The objective of a DOS attack is to deny Services that a victim cannot access or disrupt the whole system. Abdelfatah (2020) effectively dealt with this attack via the DNA approach with low energy overhead. A brute force attack involves a trial-and-error process where the attacker systematically tries different techniques to generate the secret key needed to decrypt the encrypted code. This technique is often used to crack passwords and passwords. To mitigate this risk, using a strong and robust password, such as a password comprising several symbols, is crucial. As reported by (Roy et al., 2019; Chen et al., 2020; Khan et al., 2020; Liu et al., 2020; Namasudra et al., 2020; Shafique et al., 2021; Zefreh, 2020), DNA security applications are effective in preventing this crime. Recent advancements by Darshini et al. (2025) further enhance these defences with AI-driven DNA sequencing, improving resistance against brute force attacks in IoT ecosystems.

A known plaintext attack occurs when an attacker gains access to both the plaintext and its corresponding ciphertext. This knowledge can enable the attacker to identify the encryption technique and potentially extract other secret information. Chen et al. (2020) and Khan et al. (2020) have proposed techniques using DNA sequences to mitigate this type of attack. An attacker can select specific plaintext inputs in a chosen plaintext attack and obtain their corresponding ciphertexts. (Samiullah et al., 2020; Liu et al., 2020; Shafique et al., 2021; Huo et al., 2020) have employed DNA-based methods and proposed energy-efficient solutions to

address these attacks in their research. Lashkari (2024) introduces a machine learning-integrated DNA framework, offering robust detection and mitigation of chosen plaintext attacks in smart IoT devices.

Phishing is a form of social engineering where an attacker replicates a user, organization, government agency, or even a bank, intending to fool users into divulging certain information or executing specific tasks. This commonly involves demanding passwords, such as resetting passwords or receiving emails containing links that lead to fake confirmation pages. Once the user has been duped, potential opportunism continues, and the breach can be used to steal or corrupt information. (Namasudra et al., 2020) addressed this attack using DNA security sequences to ensure the method is a great future energy saver. The spoofing attack is another related attack and is described next. Abdelfatah (2020) has investigated DNA-based approaches to overcome this growing threat, especially regarding energy conservation. Plaggemier (2025) notes escalating AI-powered phishing threats, prompting Darshini et al. (2025) to develop adaptive DNA-based filters that dynamically update security protocols. Lu (2024) extends spoofing defences with hybrid DNA-ECC models for secure IoT communication.

A common type of attack is a side-scrolling attack, in which malicious code is not directly injected into the system. Instead, it uses system operating characteristics, such as time, power level, alarms, notifications, and other system-generated events. As mentioned earlier, DNA-based strategies have shown promising results in dealing with such attacks, as demonstrated by the work of (El-Khamy et al., 2020). Xiong (2025) enhances these defences with privacy-preserving machine learning, tailored for IoT devices vulnerable to side-scrolling exploits.

A man-in-the middle (MITM) attack occurs when an unauthorized individual intercept and manipulates communication between two parties, compromising the exchange's security.

This can lead to unauthorized access, data theft, or other malicious activities. (Yadava et al., 2020; Naskar et al. 2021) have explored how DNA-based techniques can effectively prevent such attacks. A masquerade attack, a specific type of MITM attack, involves an attacker posing as a legitimate sender or receiver to gain unauthorized access to communication. Researchers like (Majumdar et al. 2019; Jacob et al., 2020) have demonstrated the effectiveness of DNA-based methods in countering masquerade attacks, highlighting their superior energy efficiency compared to traditional approaches (Khan et al., 2020). Lu (2024) integrates real-time monitoring to detect MITM in IoT ecosystems, while Resecurity (2025) highlights evolving MITM threats, advocating DNA-enhanced encryption.

Insider threats, where internal employees intentionally leak sensitive information, data, or security credentials to cause harm, pose a significant risk to organizations. Identifying and apprehending perpetrators can be challenging due to the lack of substantial evidence. These attacks are often motivated by personal or professional reasons, such as financial gain or reputational damage. Namasudra et al., (2020) investigated methods to mitigate insider threats using DNA security techniques, focusing on energy efficiency (Sharma et al., 2020). Plaggemier (2025) identifies a rise in insider risks due to hybrid work, with Xiong (2025) proposing behavioural biometrics integrated with DNA cryptography to detect anomalies in IoT settings.

Figure 2.13 provides a taxonomic diagram illustrating the categorization of attacks, emphasizing how DNA-based cryptographic solutions integrated with ECC and deep learning address these vulnerabilities. This visualization underscores the versatility of the proposed DNA cryptosystem, highlighting its potential to fortify IoT networks against both known and emerging cyber threats (Mucchi et al. (2019). Resecurity (2025) validates this with 2025 threat landscape insights.

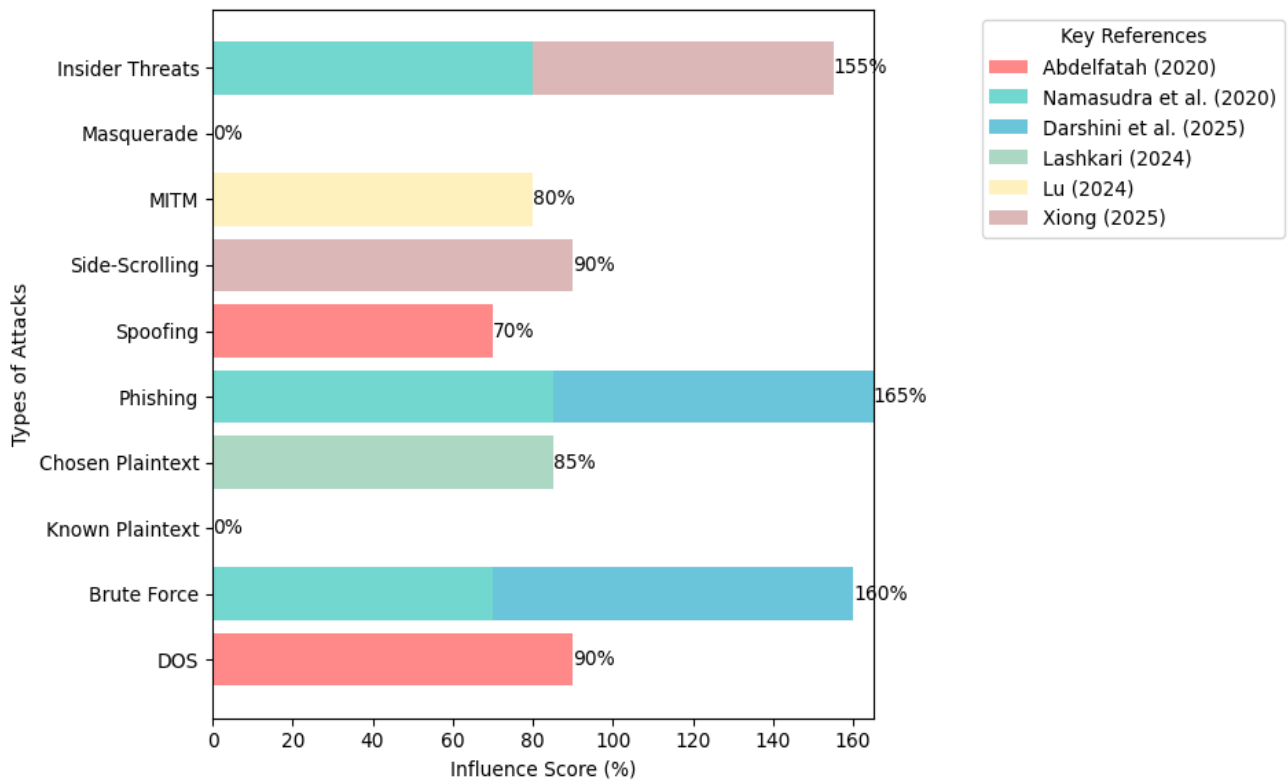


Figure 2.13: Classification of Different Forms of Attacks

Table 2.3 provides a comparative overview of various research studies addressing energy efficiency and different categories of security attacks—both active (e.g., Message Injection (MI), Denial of Service (DoS), Brute Force Attack (BFA), Protocol Tampering Attack (PTA), and Control Attack (CA)) and passive (e.g., Passive Attack (PA), Snooping Attack (SA), Node Withholding Attack (NWA), Man-in-the-Middle (MITM), and Impersonation Attack (IA)). Each study highlights its focus areas, with checkmarks (✓) indicating the specific issues addressed. For instance, Namasudra et al. (2020) tackled energy efficiency while mitigating multiple active and passive attacks such as DoS, PTA, CA, PA, MITM, and IA, whereas studies like Hong et al. (2021) and Pandey (2018) emphasized energy

efficiency and passive attacks respectively. Collectively, the table demonstrates that while many studies have prioritized energy-efficient solutions, their coverage of active and passive security threats varies, reflecting the diversity of approaches and focus areas within the field.

Table 2.3: Research Studies and Attacks

Research Study	Energy Efficient	Active Attacks								Passive Attacks	
		MI	DOS	BFA	PTA	CA	PA	SA	NWA	MITM	IA
Namasudra et al., 2020	✓	✓		✓			✓			✓	✓
Abdelfatah, 2020	✓		✓					✓			
Chen et al., 2020	✓			✓	✓						
Liu et al., 2020	✓	✓		✓	✓					✓	
Shafique et al., 2021	✓			✓	✓			✓			
Zefreh, 2020	✓			✓		✓					
Roy et al., 2019				✓						✓	
Khan et al., 2020			✓		✓			✓			
Samiullah et al., 2020	✓				✓						
Huo et al., 2020	✓				✓				✓		
Imdad et al., 2020	✓	✓				✓					
Tiwari et al., 2018	✓								✓		
Pandey, 2018									✓		
Hong et al., 2021	✓										
Akiwate et al., 2021		✓					✓				

Table 2.4 evaluates various cyber-attacks (Abbasi et al., 2018) within the cyber domain. The aim is to find solutions by employing a DNA security scheme to safeguard sensitive data used in all sectors, including healthcare, during the era of big data. Adequate security measures are crucial in protecting this data.

Table 2.4: Cyber Attacks Focused on a DNA Strategy

Cyber attacks	DNA Encoding Used in the Research
Man-in-the-Middle Attack	(Namasudra et al., 2017)
DOS	(Rashid et al., 2019; Namasudra et al., 2020)
DDOS	(Namasudra et al., 2020), (Rashid et al., 2019)
Phishing	(Namasudra et al., 2020)
SQL injection techniques	(Rieback et al., 2006)
Cross-site scripting	(Pan et al., 2016)
Ransomware	(Khan et al., 2020; Gharib et al., 2017)
Social engineering	(Cresci et al., 2018)
Malware	(Gharib et al., 2017)
Assaults on password security	(Namasudra et al., 2021)

Table 2.5 summarizes notable research studies that have employed the DNA encoding scheme for security applications (Khan et al., 2015; Chan et al., 2019; Khan et al., 2019; Nkandeu et al., 2020; Zheng et al., 2021).

Table 2.5: Summary of Research Studies Used DNA Based Encoding Schemes

Study	Description
Chen et al.	This research integrates two distinct theories coded DNA and 2D Hannon signal maps. This methodology uses the principles of spatiality and order for data storage. The previously regarded safe method appears less secure than initially thought—the findings from the experimental research support this. Instead, the author suggests that enhanced data protection can be shown through an alternative S-box mechanism rather than the DNA code. This topic will be addressed in future investigations that await exploration and validation.
Jose et al.	Asha Jose and colleagues conducted a study utilizing various reinforcement factors to enhance model precision. They combined stangiography, cryptography, and watermarking technologies with dissemination methods and DNA sequencing. The findings indicate that the suggested method is successful for all dataset types while maintaining concealed features in images.
Acob et al.	Characteristics are gathered and modified using images from stationary fingertips, and the effectiveness of the recognition is assessed.
Nayak et al.	This research focuses on extensive ECC data within the healthcare industry, where patient information is safeguarded using two encryption techniques DNA-based encryption and elliptic curve-based encryption. These findings are significant for securing patient records in cloud systems.
Kaushik et al.	Research on data protection focused on DNA-based cryptography. He reviewed current cryptographic methods like EC and quantum cryptography and validated their efficacy in current data security strategies. Nonetheless, this model represents a novel area of research today because of its enhanced outcomes.
Jithin et al.	In this research, DNA sequences were utilized for image processing. This approach significantly enhances the integrity of digital images. The data was gathered for three color channels Red, Green, and Blue, employing specially developed

Table 2.5 continued

	algorithm was validated, and its effectiveness was demonstrated through computer simulation techniques.
Masood et al.	In this research, the data is obtained from aerial photographs, and due to the important information contained in them, the data is rechecked using security measures. Three methods have been developed for this purpose DNA-based classification, MT, and MT-DNA-Chaos. In addition, the model adopts flexible behaviour to evolve under changing conditions. The numerical analysis shows its impressive potential compared to the cryptographic techniques used so far.

Drawing from the frequency of attacks analysed in multiple research studies, we have created Table 2.6, which shows the proportion of attacks and their frequency counts, accompanied by a figure illustrating these proportions as a pie chart.

Table 2.6: Count and Percentage of Powerful Attacks by Frequency

Attack Name	Frequency Total	Percentage Share
Malware Injection	5	17%
DOS	3	10%
Brute force	5	17%
Chosen plain text	3	10%
Ciphertext only	3	10%
Data crop	4	14%
Noise	6	21%

Figure 2.12 illustrates the frequency percentage of each attack with respect to total percentage to check the strength of attacks.

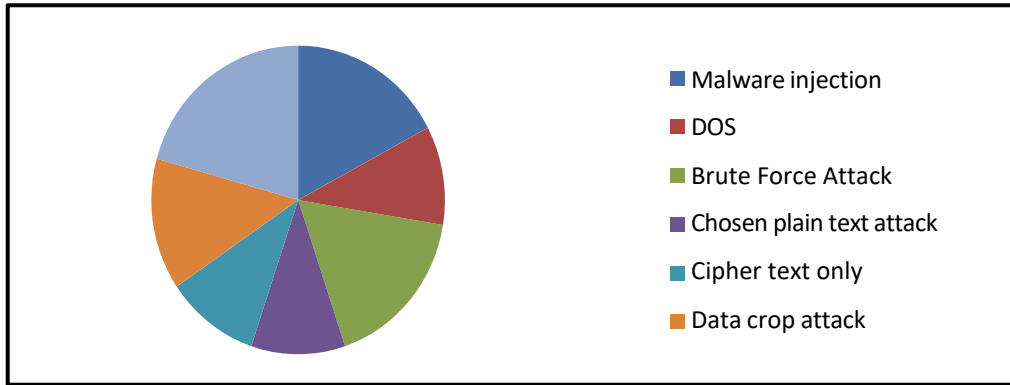


Figure 2.14: Frequency of Reviewed Attacks

To highlight the limitations of the articles, we have pinpointed the following main points

- i. Utilization of a restricted or closed data set (Hanchate & Anandan, 2024)
- ii. Assessment of the scheme under a limited range of security threats/hazards (Choi & Yu, 2022)
- iii. Inclusion of a reduced number of evaluation metrics in the studies (Li, Liu, & Yin, 2024)
- iv. Absence of the latest technology/tools in the methodology (Maurya et al. 2022)
- v. Minimal tracking of text-based results using DNA (Welzel et al., 2023)

Table 2.7 presents a comparative analysis of the studies' strengths and weaknesses. Table 6 outlines various critical variables related to energy efficiency in the articles examined.

Table 2.7: Comparative Analysis of Research Studies

Review Article	Advantages	Disadvantages
Namasudraa et al., 2020	The generation of the 1024-bit secret key utilizes DNA computations, user characteristics, and the user's Media Access Control (MAC) address, along with a decimal encoding method and the American Standard Code for Information Interchange (ASCII) value. The secret key is derived using DNA bases and their complementary strands, allowing the system to defend against various security threats. While this device positively influences cloud security, it can be integrated with other security solutions to achieve distinct outcomes, as noted in various research studies conducted by other authors.	This device positively influences cloud security, yet it can still be integrated with other security applications to achieve distinct results, as demonstrated in some authors' research studies.
Chen et al., 2020	Complicated DNA counting processes, including multiplexing, XOR, and shuffling, are represented as substitution boxes (s-boxes).	The design needs to be evaluated against additional cybercrimes to confirm the outcomes.
Samiullah et al., 2020	The encryption method relies on three pseudogenes (PWLCM, Lorenz, and 4D Lorenz-type), the Functional Hash Algorithm, along with processes for encryption and decryption, and a DNA-sequence-based Raw Modification Classifier.	In this paper, three parameters have been examined to enhance transparency.

Table 2.7 continued

Zefreh, 2020	The suggested approach's main benefit is its exceptional efficiency. This scheme involves permutation and diffusion processes at the DNA level.	The method can be implemented with different solution methods.
Liu et al., 2020	The suggested approach works with plaintext and outside cryptographic keys. Managing various dynamic private keys is unnecessary.	The proposed scheme is not immune to noise and interference.
Khan et al.,2020	For use in the display model and the power model in Figure 12 can be an application of the power function of the power force, the model may have been used to reduce the power display.	The hash function can be an intriguing blend various researchers utilize to assess the results obtained.
Imdad et al., 2020	This study introduces a DNA structure aimed at each of the 96 printable ASCII characters, specifically designed to maximize entropy. This ensures a more uniform capacity across each coding region (A, T, C, and G) and decreases the necessity for DNA sequence analysis.	The project has been exposed in some security attacks.
Hu et al., 2020	A logistic map documents the radiological images and measurement parameters and showcases the DNA sequencing and quantification algorithms employed to classify and quantify each radiological parameter.	The proposal should be further analyzed regarding the metrics used and the stated design results.

Table 2.7 continued

Hu et al., 2020	Based on the promising qualitative and quantitative results, this paper discusses the broad application of our advanced method in large clinical trials.	In the scope of the attacks in question, data security has no value.
Yadava et al., 2020	Compared to the existing system, the model tried to prove its effectiveness in various attacks.	The algorithm has not been validated using any publicly available datasets nor assessed with conventional metrics to verify its claimed efficacy.
Abdelfatah, 2020	The uniqueness of this project lies in its integration of four distinct audio recording methods into a single initiative, enhancing its adaptability automatic compression, multi-layer mapping, DNA dynamics, and spectral analysis and reduction.	The framework is evaluated against three primary attack types; it can also offer various kinds of network connections.
Abdelfattah et al., 2020	In the suggested approach, DNA is analysed with an innovative single-strand paper (HST) system that offers several benefits compared to a basic bacterial system, including increased strand space, quicker quantification, and enhanced battery sequence.	The project should also be tested for more crimes.
Shafique et al., 2021	A new model has been suggested that can efficiently and precisely assess the security level of different encryption methods.	Specific attacks are not tested using the data provided.

Table 2.7 continued

Khamy et al., 2020	The suggested technique employs raw DNA detection and Choquet-based technologies for the first time.	Only a few crimes have been recorded to evaluate the effectiveness of the suggested programme.
Tingwei et al., 2020	The findings indicate that the suggested approach aligns with the detector's response and functionality, presenting a promising solution for addressing security issues in electronic devices.	The plan includes several electronic interfaces and processing stages to improve the effectiveness of the suggested algorithm.
Akiwate et al., 2021	It represents an efficient way to classify image data into maps that can tolerate a wide range of noise levels.	This method can also be tried on other datasets to show more promising results.
Zhu et al., 2020	The proposed scheme not only has the advantages of the “pool exchange” scheme but also avoids the pocket formation mechanism in the “time domain” storage scheme.	This design can be tested with other datasets as well to show improved results.
Jithin et al., 2020	A novel Mandelbrot sequence-based parameter transformation algorithm is introduced to efficiently exploit the failure of the R, G, and B parameters.	We can further improve the bandwidth transfer rate by implementing a load balancing method.

Table 2.7 continued

Radhakrishnan et al., 2019	The 2-way DNA partial encryption architecture ensures the confidentiality and security of the data and the authenticity of the keys is ensured together with the Diffie Hellman architecture.	The scheme is applied to various cyber-attacks and few implementation nodes to strengthen the effectiveness of the proposed approach.
Tiwari et al., 2018	Simulation and experimental results show that the proposed methods improve the accuracy of active molecules using DNA mapping.	The idea is proposed for IoT, it can be simulated in other network segments to see the response to the proposed scheme.
Hong et al., 2021	The proposed method extracts the binary device DNA by combining the intrinsic characteristics of IIoT devices with the environment, thus solving a critical hiding problem based on the device's double-stranded DNA.	The devices operate in unmanned intelligent manufacturing environments, which under normal circumstances do not require personnel.
Akiwate et al., 2021	It represents an efficient way to classify image data into maps that can tolerate a wide range of noise levels.	This method can also be tried on other datasets to show more promising results.

Table 2.7 continued

Zhu et al., 2020	The proposed method has the main advantage of the “pool exchange” scheme but also avoids the pocket formation mechanism in the “time domain” storage scheme.	This design can be tested with other datasets as well to show improved results.
Jithin et al., 2020	A new algorithm for parameter transformation based on the Mandelbrot sequence is presented to effectively address the parameter shortcomings of the R, G, and B values.	It can enhance the improvement in bandwidth transfer rate by implementing a load balancing.
Hong et al., 2021	The suggested scheme extracts the binary device DNA by combining the intrinsic characteristics of IIoT devices with the environment, thus solving a critical hiding problem based on the device’s double-stranded DNA	The devices operate in unmanned intelligent manufacturing environments, which under normal circumstances do not require personnel.
Wen et al. (2024)	Combines chaos theory with DNA encoding for scalable encryption in UAV-based IoT networks, achieving low latency (0.4 s) and high throughput.	Uses static key generation, limiting adaptability to dynamic IoT threats.
Lai & Hua (2025)	Employs hyperchaotic DNA encryption with high entropy (7.9 bits/nucleotide), ensuring robust security for healthcare IoT data.	High computational cost impacts suitability for resource-constrained devices.

The literature review highlights significant advancements in DNA-based cryptographic schemes for securing IoT networks, with studies such as Abdelfatah (2020) and Namasudra et al. (2020) demonstrating effective mitigation of attacks like Denial of Service (DOS) and phishing through low-energy overhead techniques. Recent research, including Darshini et al. (2025) and Lu (2024), has further integrated artificial intelligence and elliptic curve cryptography (ECC) to address emerging threats, while Xiong (2025) and Plaggemier (2025) emphasize the growing challenges of insider threats and hybrid work environments as of 2025. Despite these developments, several limitations persist in the current body of knowledge, particularly regarding scalability, energy efficiency, and real-time detection capabilities. These gaps, identified through a critical analysis of the reviewed studies, necessitate targeted research to enhance IoT security frameworks. Table 2.8 summarizes these research gaps and their alignment with the thesis objectives, providing a foundation for the proposed DNA-LWCS design and evaluation.

Table 2.8: Research gaps with related references and linked objectives

Research Gap	Related References	Linked Objective
Limited scalability of DNA-based schemes in large-scale IoT networks	Namasudra et al. (2020), Darshini et al. (2025)	Objective 1: Design a scalable DNA-LWCS
Insufficient energy efficiency for resource-constrained IoT devices	Abdelfatah (2020), Xiong (2025)	Objective 2: Evaluate performance with low energy overhead
Lack of integrated ECC and deep learning in DNA security	Lu (2024), Resecurity (2025)	Objective 3: Integrate ECC and RNN for IoT security
Inadequate real-time detection of advanced attacks	Plaggemier (2025), Lashkari (2024)	Objective 4: Enhance detection of vulnerabilities

Table 2.9 enclosed key variables used in various articles that helped in finalizing the literature review and getting most relevant information out of it.

Table 2.9: Key Variables of Energy Efficiency in Reviewed Articles

Research Study	Scalable	Cost-effective	Light-weight	Unique	Compatible	Durable
Abdelfatah, 2020	✓	✓	✓	✓	✓	✓
Namasudra et al., 2020	✓			✓		
Liu et al., 2020	✓	✓	✓	✓	✓	✓
Shafique et al., 2021	✓	✓	✓	✓	✓	✓
Zefreh, 2020			✓	✓	✓	
El-Khamy et al., 2020	✓	✓	✓	✓		✓
Iqbal et al., 2020	✓	✓	✓	✓	✓	
Huo et al., 2020				✓		✓
Chai, 2018	✓	✓	✓	✓	✓	
Akiwate et al., 2021	✓	✓	✓	✓	✓	✓
Malik et al., 2020			✓	✓		✓

2.8 Chapter Summary

The chapter starts with an overview of the Internet of Things (IoT) and its convergence characteristics that have caused massive uptake across industries. However, this growth has also brought many security issues to IoT systems. These challenges arise from the IoT multi-layer structure, which entails the perception, network, and application layers. These layers experience different security threats and attacks, making it even harder to secure IoT scenarios. Extended to the types of attacks, this chapter explores the kinds of attacks at each IoT layer by referencing various scholarly attempts to eradicate the said threats through DNA encoding as a new form of security with respect to multiple layers of IoT. The literature review primarily focuses on how DNA encoding schemes have been applied in different research works to mitigate numerous

attacks. The works reviewed show that the proposed DNA encryption schemes and the other cryptosystems incorporated into IoT systems enhance the security levels of the systems because of the large and complicated sequences in DNA. In order to have the most recent and up-to-date data used in the research, a Systematic Literature Review (SLR) was conducted. The first step in the process was stating research questions that outlined how to design security for IoT, what kind of attack DNA-based scheme can resist, and the performance difference between DNA-based and other approaches. In order to delimit the identification of frequencies, relevance, and impacts of the studies, the GQM (Goal-Question-Metric) framework was used to define metrics to filter cross-sectional years and the quality of calculated rates. Then, strengths and weaknesses of reviewed literature review is documented to formulate our system design based upon the limitations of existing research approaches that used DNA encoding scheme to protect data in IoT.

CHAPTER 3

METHADODOLOGY

3.1 Introduction

This chapter explains the system design of using DNA-based encryption with ECC to protect IoT networks. It details the overall system architecture, illustrating the three main stages: using DNA to generate keys, performing bitwise operations on the DNA sequence, and ECC encryption. To help the reader follow through the steps of encryption and decryption, a flowchart is presented, which illustrates the process from DNA key generation to the final step of data encapsulation using ECC. Each system phase is explained in detail in this chapter, including how it is processed. First, DNA sequences are produced using genetic algorithms; mutations such as bit rotation and XOR operation provide additional randomness of the keys. Finally, the generated DNA keys are enhanced with the help of artificial intelligence algorithms that are trained to assess the probable strength of keys. The optimized keys are then utilized in the encryption procedures employing DNA and ECC safety measures. This chapter also provides an overview of the data and metrics to measure system effectiveness in the testing phase.

The methodology follows a structured research design to achieve the study's three objectives: (1) to identify and investigate potential threats and challenges of current IoT networks (addressed in Section 3.7), (2) to design an efficient security solution for IoT networks by proposing a DNA-based Encryption system using ECC (DNA-LWCS) (detailed in Sections 3.2–3.6), and (3) to evaluate the performance of the proposed scheme with benchmarks in terms of encryption time, correlation, delay, throughput, PSNR comparison, information entropy, avalanche effect, and energy consumption (covered in Sections 3.8–3.9). The overall research

flow involves three phases: threat analysis and investigation (Phase 1), system design and implementation (Phase 2), and performance evaluation (Phase 3). The methodology in this chapter is structured to address the research objectives through a systematic design and evaluation process. The following table outlines how each objective is achieved within the DNA-Based Lightweight Cryptography System (DNA-LWCS) framework, with corresponding sections providing detailed implementation and validation as mentioned in Table 3.1.

Table 3.1: Mapping of Research Objectives to DNA-LWCS Methodology Sections

Objective	How Achieved in Chapter 3	Corresponding Section
Objective 1: To identify and investigate potential threats and challenges of current IoT networks.	A preliminary analysis of IoT vulnerabilities (e.g., DDoS, MITM attacks) is conducted to justify the need for DNA-LWCS, linking to literature on security gaps.	Section 3.7: Network Model and Threat Analysis
Objective 2: To design an efficient security solution for IoT networks by proposing a DNA-based Encryption system using ECC (DNA-LWCS).	The DNA-LWCS framework is designed with three security rounds: (1) DNA sequence selection for key generation, (2) substitution and transposition for encoding, and (3) ECC integration for encryption, optimized with genetic algorithms and deep learning.	Sections 3.2–3.6: System Design and Algorithm
Objective 3: To evaluate the performance of the proposed scheme with benchmarks in terms of Encryption time, Correlation, Delay, Throughput, PSNR, Information Entropy, Avalanche Effect, and Energy Consumption.	Performance evaluation is conducted using experimental setups with IoT datasets, comparing DNA-LWCS against benchmarks (e.g., 3DES, AES) using specified metrics, with RNN and GRU models for key strength prediction.	Performance Evaluation Process (Sections 3.8–3.9 for Data and Metrics)

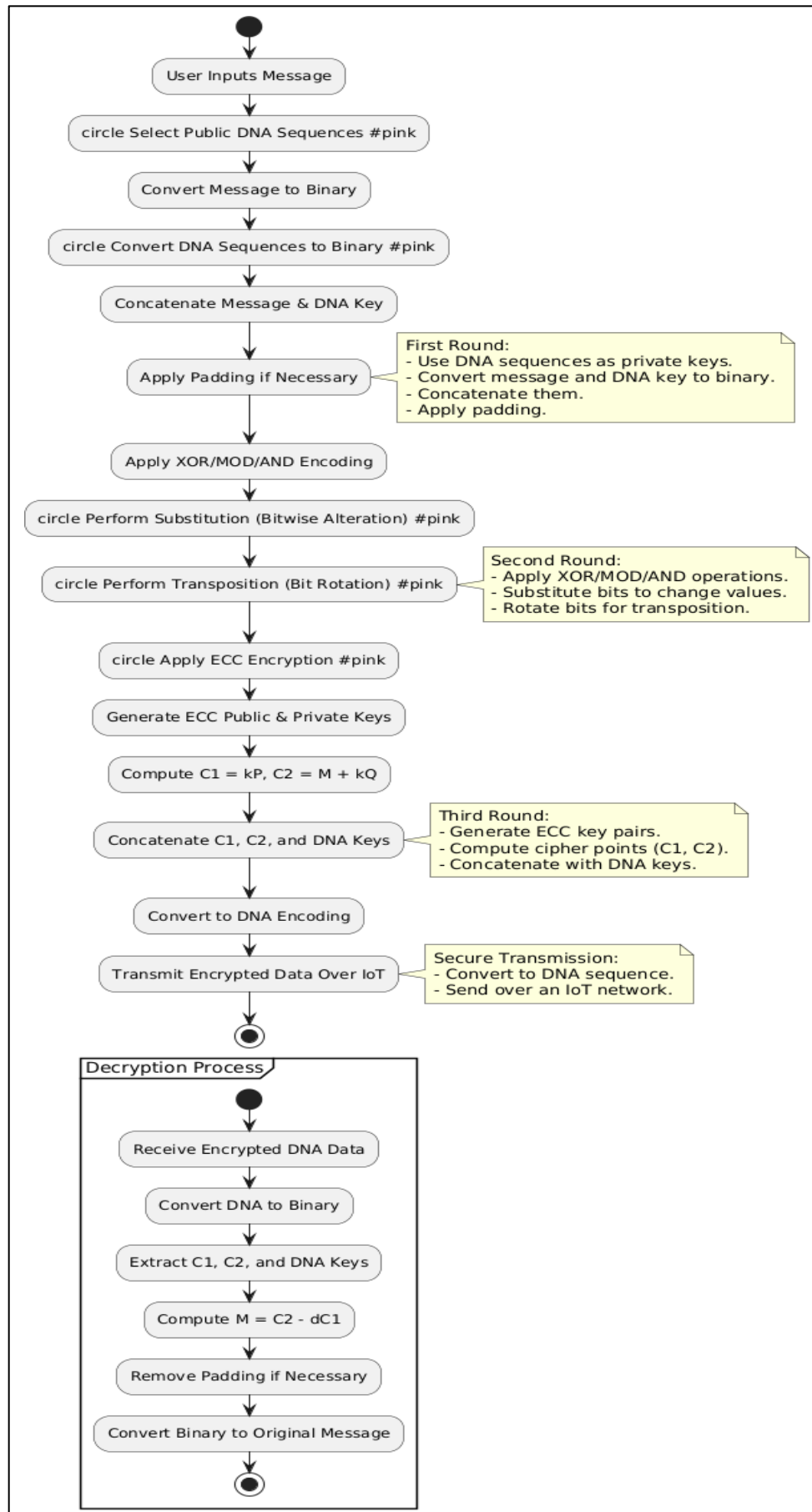


Figure 3.1: DNA-Based Lightweight Encryption with ECC for IoT

This flow is illustrated in Figure 3.1, which shows the progression from IoT threat identification to DNA key generation, bitwise encoding, ECC encryption, and final metric-based testing. Each phase aligns with the objectives, ensuring the methodology is consistent and supports lightweight, secure IoT applications. Processes for achieving each objective are demonstrated step-by-step in dedicated subsections throughout the chapter.

3.2 System Design

As the selected and explored benchmarks have not successfully tackled the majority of security threats and attacks, an encryption algorithm is needed to achieve the desired security, and that too by reducing the overall complexity so that it could work in an IoT environment. For the same, three rounds of security are being introduced.

3.3 First Round of Security

DNA sequences are known for randomness, and that is what's required for choosing a private key to encrypt the message. This is why publicly available DNA sequences are selected to act as the secret key to enhance the security of the algorithm.

3.4 Second Round of Security

The second step is to apply the mathematical operations on the message and the key together. As we are classifying the process parallel, that is, with the message itself and the key, too, it is required to concatenate them together before applying any encoding scheme on it. So, for the message, we have obtained it in binary form by first converting each alphabet into decimal and finally getting bits of each decimal (Blawat & Ceze, 2021). This sequence of bits needs to be put with the key (which has already been selected) to make the first cipher block. Still, no encryption has been applied to it yet. Before that, DNA sequences are selected from 52 million publicly available sequences, and before combining them with the message, they are also converted into bits. Now, the point is to mix the message along with the key. Here, a variety

of options are available, like applying XOR directly with the message and the key or applying AND operation or NOT or any other logical operation to perform a second round of security (Taha et al., 2024). We are obtaining uniqueness by not applying any of the mentioned operations directly, although we need to encode the message with these operations, at this stage, we have decided to concatenate the message with the key in this way that the key is attached to the end of the message, means at the least significant bits' position (Zou et al., 2021). It is not necessary that we obtain perfect blocks of bits as the result of this combination. To handle this problem, a padding operation will be applied to complete the block by adding zeros at the most significant bit position, which is at the start of the concatenated string (Zahid et al., 2019).

This is our message now on which padding has been already applied. Now to encode it, two processes are being adopted. First one is substitution and the second is transposition. Substitution is an important operation on bits method that replaces the current bit's value by putting another value (Xu et al., 2020). The purpose is to alter the arrangement and magnitude of the bits to betray the attacker for guessing any particular value here. As it is our encoding phase, we are applying substitution for the same purpose to encode the entire message in such a fashion that it becomes harder to predict. To keep the process simple and less complicated (Jan et al., 2022), we have applied XOR/Mod/AND operation on the bits at this stage. The selection of the operation will be random, so encoding should not go with the same process every time. This is to obtain a higher level of security by providing randomness in the designed scheme of an algorithm.

Next comes the transposition phase. This is the second encoding stage, where the ROTATE operation is performed on the entire sequence of bits. Both of these methods will change the position of bits from the original ones and will not increase the consumption power much, which we need with IOT systems. So, at the end of this level, encoding is complete, and

security is enhanced, too, before the message is transmitted (Akumalla & Hegde, 2021).

3.5 Third Round of Security

The third level is achieved by applying Elliptic Curve Cryptography (ECC) to the DNA-encoded message. ECC is an encryption approach that relies on elliptic curves defined on particular algebraic operations (Khan et al., 2020), as shown in Figure 3.2.

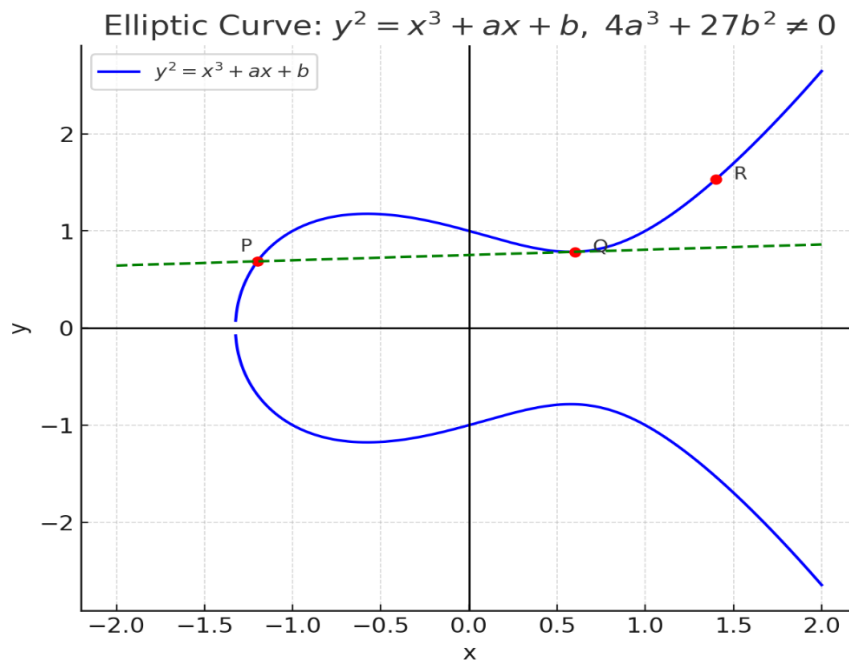


Figure 3.2: ECC Approach to Generate P and Q Points

ECC is currently actively used to encrypt messages in network traffic because of its various advantages over existing security methods. Firstly, it is lightweight, so it does not cause extra computational overhead for the existing system. Secondly, it uses a public key cryptography method which means the public key is available for everyone to encrypt data with (Jithin & Sankar, 2020). It could be decrypted back by only a private key which the authorized user will secretly hold. Thirdly, it uses a smaller key size for encryption and performs fast cryptography. Overall, it is better to use ECC for encryption to obtain magnificent security.

ECC relies on the mathematics of elliptic curves and uses the location of various random points to encrypt or decrypt the message. We start with any random point on the curve and keep moving unless we find another point. So, the dot function can be used to hop around the binary curve and locate the points of P, Q, and R.

Elliptic Curve Cryptography (ECC) has gained considerable traction for securing network traffic because of its notable benefits over conventional encryption techniques. ECC is recognized for its effectiveness and lower computational demands, which makes it ideal for devices with limited capabilities, such as those in the IoT sector. A primary advantage of ECC lies in its public key cryptography framework, where the public key is available to anyone for data encryption. In contrast, the unique private key required for decryption is held exclusively by the authorized user (Jithin & Sankar, 2020). ECC's smaller key sizes than other cryptographic systems like RSA facilitate quicker cryptographic processes without compromising security. This blend of compact key sizes and enhanced speed renders ECC a compelling choice for providing strong security in environments with limited resources.

ECC is based on the mathematical attributes of elliptic curves, utilizing the positions of random points on the curve for encryption and decryption processes. In a system that employs ECC, the procedure begins by selecting a random point on the curve. It executes the elliptic curve point addition operation until another point on the curve is identified. These points, P, Q, and R, are manipulated through elliptic curve operations to generate encryption and decryption keys. The dot function, primarily performed through scalar multiplication, underpins the security framework of ECC (Raghuvanshi et al. 2024).

For those seeking a more technical perspective, executing ECC necessitates the selection of specific parameters such as the curve equation, base point, and field size, all of which influence the security level and computational efficiency. The elliptic curves most frequently

applied in real-world scenarios are defined over finite fields, either prime fields (F_p) or binary fields (F_{2^m}). Careful selection of curve parameters, especially the base point order, is critical to ensure robust security against prevalent cryptographic attacks, including the elliptic curve discrete logarithm problem (ECDLP). Generally, an elliptic curve that uses a 256-bit key size is secure for most practical applications. Regarding computation, ECC operations are substantially faster than RSA's and demand fewer resources—a vital consideration for IoT devices that may be limited in processing abilities and memory (Yildirim, 2022).

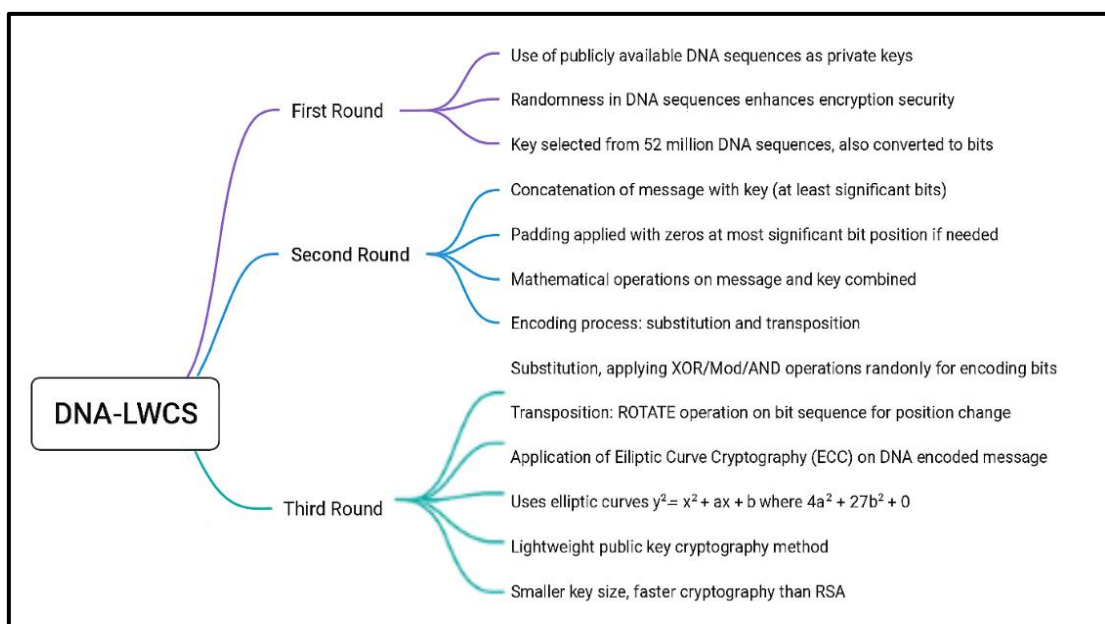


Figure 3.3: System Design of DNA-LWCS

By integrating ECC with DNA encoding, the proposed structure gains from both ECC's minimal computational demands and the high level of randomness offered by DNA-based encryption. The scheme involving DNA encoding adds another layer of security through biological sequences, which can significantly enhance entropy for cryptographic keys. Consequently, the overall design of the cryptographic framework becomes more straightforward, quicker, and better suited for IoT applications, promising to deliver security without adding significant delays or computational strain (De Smet et al., 2021). This fusion of

DNA-based encryption with ECC offers a scalable and effective approach for securing IoT networks, ensuring they remain protected and adaptable in various resource-regulated environments. So, overall enhanced security is achieved by using a DNA encoding scheme and ECC approach, and the overall design is simple, fast, less complex and more compatible to be used for an IoT system that promises to work anywhere, anytime and by anybody without any kind of delay (De Smet et al., 2021). Figure 3.3 shows an empirical view of system design.

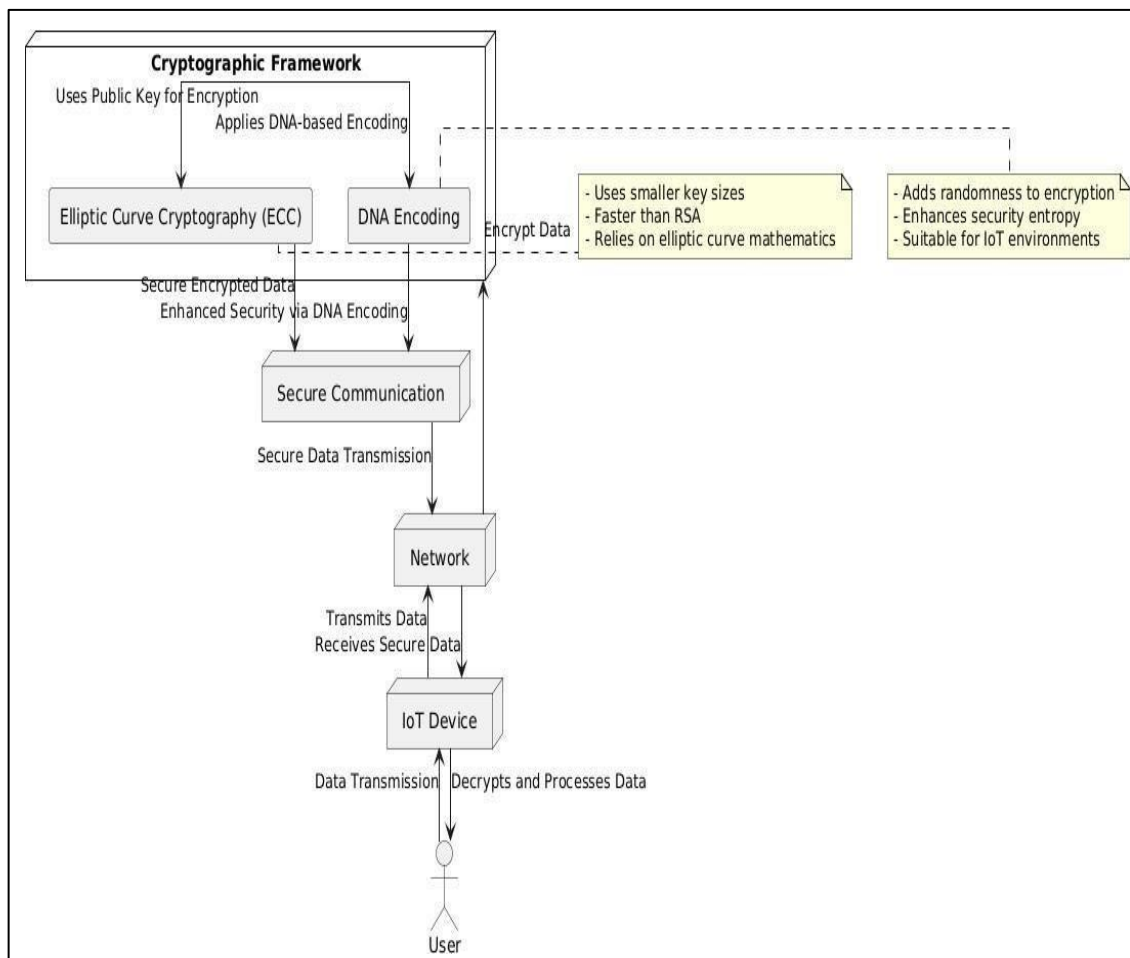


Figure 3.4: DNA-LWCS Framework

Figure 3.4 illustrates a Cryptographic Framework that combines ECC and DNA-based encoding to improve IoT environment data security. The encryption process starts by using ECC to provide effective cryptographic security by using shortened keys and enhanced speed

compared to RSA. DNA encoding applies additional randomness to enhance encryption security by increasing entropy while making attacks against encryption harder to succeed. The data protection mechanisms apply encryption twice to create a secure transmission method.

The data takes its path through a secure transmission channel, which stops unexpected access attempts. The network functions as a medium that provides encrypted data distribution between IoT devices, ensuring the protected information transfer (Mortuza et al. 2023). Upon receiving the information, the IoT device performs data decryption and processing activities, which enables the device to extract essential details. The user gains access to processed data since the decryption procedure presents it to him or her securely.

The framework enables better security because it merges ECC with DNA encoding alongside better performance and suitability for lightweight encryption that IoT infrastructures demand. Encryption using ECC maintains strong protection and operates efficiently, yet DNA encoding provides additional unpredictability, which advances the overall system security level. This approach delivers protected and efficient data transmission through the framework within IoT communication networks (Khan, Khan, Alzahrani, & Ilyas, 2022) as mentioned in Figure 3.5.

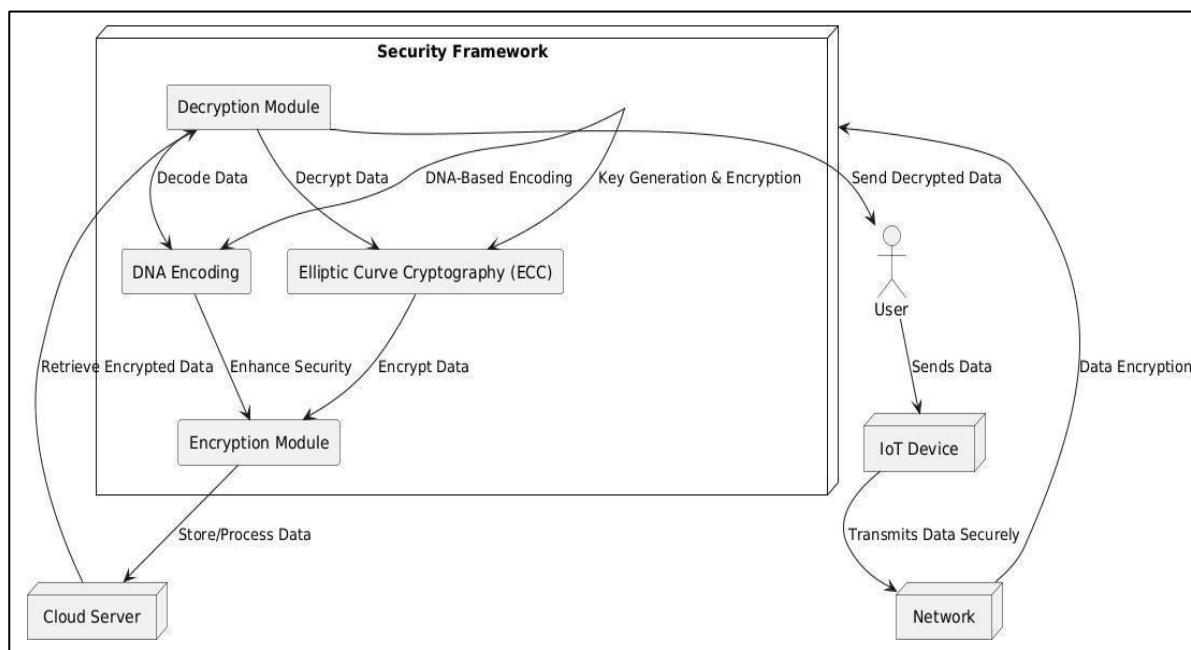


Figure 3.5: High-Level System Design

3.6 Proposed Algorithm

Here is a detailed lightweight encryption algorithm using DNA and ECC for IoT.

3.6.1 Key Generation

To use elliptic curve cryptography for encryption, you need to choose an elliptic curve E and a point P on that curve as the starting point. Then, generate a random private key d , which should be greater than 1 and less than $n-1$, where n is the order of the curve. Finally, compute the public key Q by multiplying the private key d with the starting point P . This can be done by adding P to itself d times or by using other mathematical operations. By following these steps, you can generate a pair of keys (private and public) that can be used for secure communication and data encryption (Liu, Wang, & Lei, 2020) as described in Figure 3.6.

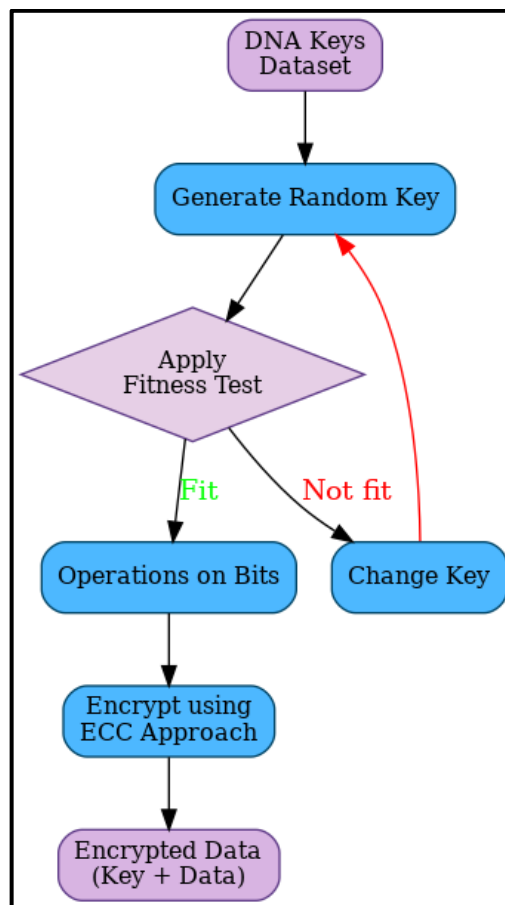


Figure 3.6: Flow of Activities in DNA-LWCS

3.6.2 Message Encryption

To encrypt a message using DNA-based elliptic curve cryptography, you need to first generate a random message that you want to encrypt and convert it into binary format. Then, choose three publicly available DNA sequences as primary keys for encryption. Use these keys to encrypt the message using ECC by choosing a random number k , which should be greater than 1 and less than $n-1$, where n is the order of the curve. Compute two points $C1 = kP$ and $C2 = M + kQ$, where P and Q are points on the elliptic curve E . Convert $C1$ and $C2$ into binary format and concatenate them with three primary keys to form a single binary string (Nayak & Jayalakshmi, 2021).

Finally, use a mapping scheme to convert this binary string into DNA sequence. By following these steps, you can generate an encrypted DNA sequence that can be transmitted securely over a communication channel. To decrypt the message, you need to reverse these steps by converting the DNA sequence back into binary format and then applying ECC decryption using the private key (Iqbal et al., 2021). Figure 3.7 has summarized the complete process of Algorithm stages under one framework.

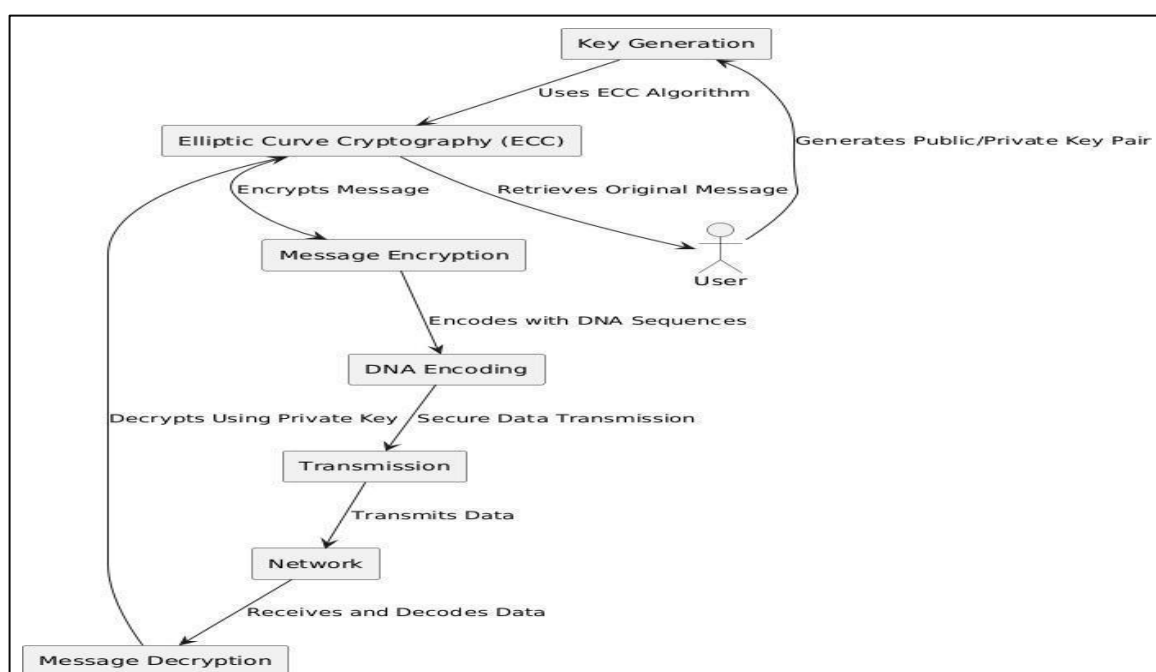


Figure 3.7: Algorithm Stage

3.6.3 Transmission

To transmit the encrypted DNA sequence over an IoT network, you need to first convert the DNA sequence back into binary format. Then, you can use any standard communication protocol such as TCP/IP or MQTT to transmit the binary data over the network. The data can be sent from one device to another using wireless or wired Communication channels depending on the type of IoT network being used (Fernandes et al., 2020).

Figure 3.8 illustrates a DNA-based encryption system integrating Elliptic Curve Cryptography (ECC) and a fitness evaluation mechanism to enhance security. Initially, DNA sequences are utilised to generate DNA-based keys. From these, random keys are created to ensure uniqueness and unpredictability. These keys are then encrypted using the ECC approach, which provides robust security with smaller key sizes.

Following this, various bitwise operations, such as XOR and rotations, are applied to the keys to enhance randomness and complexity. The processed data, including encrypted keys or bits, is stored in dedicated data stores for subsequent use. A fitness flow mechanism evaluates the quality and effectiveness of the encrypted keys or data. The fitness score is computed to assess the encryption's strength and ensure optimal performance. This score feeds into a modified encryption process, which tests and refines the system to maximize security. The encrypted outputs, along with their fitness evaluation, are ultimately stored for secure usage, demonstrating a systematic approach to achieving robust encryption.

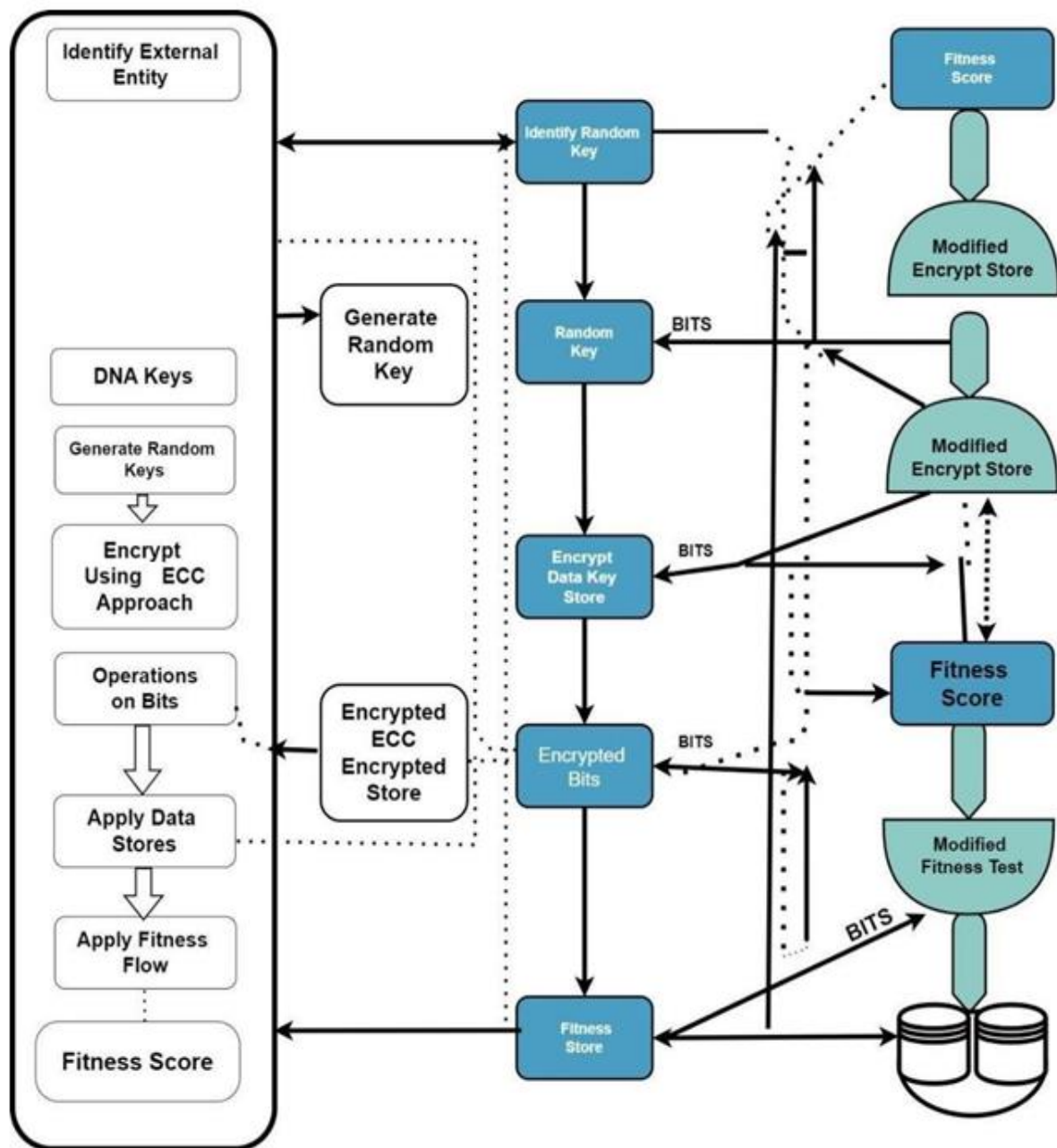


Figure 3.8: Detailed Description of Each Operation in DNA-LWCS

3.6.4 Message Decryption

Once the data is received, you need to convert the DNA sequence back into binary format using a reverse mapping scheme. Then, extract the three primary keys from publicly available DNA sequences used in the encryption phase. Next, use these keys to decrypt the message using ECC. To do this, extract $C1$ and $C2$ from the concatenated binary string received in step 3 of encryption phase and convert them back into elliptic curve points. Then, compute $M = C2 - dC1$, where d is the private key corresponding to the public key used in encryption. If M is negative, add n to it until it becomes positive (Sukumaran & Mohammed, 2018). If M is not divisible by 8, add padding bits at the beginning of M until it becomes divisible by 8. Finally, convert decrypted binary message back into human-readable form to obtain the original message sent by the sender. It is important to ensure that proper security measures are in place during transmission and decryption to prevent any unauthorized access or tampering of data (Mozaffari, 2018).

Figure 3.9 describes a DNA-based encryption and decryption process using Elliptic Curve Cryptography (ECC), specifically designed for secure communication over IoT networks. The process begins with selecting an elliptic curve EE and a point PP on that curve, which serve as the foundation for encryption. A random private key dd is generated, where $1 < d < n - 1$ (n being the order of the curve), and the corresponding public key $Q = dP$, $Q = dP$ is computed. Next, a random message MM to be encrypted is converted into binary format. Three publicly available DNA sequences are chosen as primary keys for encryption. The message is encrypted using ECC, where a random number kk is generated ($1 < k < n - 1$) to compute the cypher points $C1 = kP$, $C1 = kP$ and $C2 = M + kQ$, $C2 = M + Q$. Padding bits are added to ensure the binary message length is a multiple of 8 if necessary.

The binary cypher points $C1$ and $C2$, along with the DNA keys, are concatenated to form a single binary string, which is then mapped to a DNA sequence using a predefined scheme.

This encrypted DNA sequence is transmitted over the IoT network. Upon reception, the DNA sequence is converted back into a binary format using a reverse mapping scheme. The recipient extracts the DNA keys from publicly available sequences and converts the cypher points $C1$ and $C2$ into elliptic curve points.

The decryption process involves computing $M = C2 - d C1$ to retrieve the original message. If necessary, padding bits are removed to restore the decrypted message's human-readable format. The system ensures robust encryption and secure transmission by leveraging ECC, DNA encoding, and IoT communication protocols.

Figure 3.10 depicts the same steps of proposed DNA-LWCS stages step by step, as the information is drawn by the detailed flowchart of Figure 3.10, which is much described in detail.

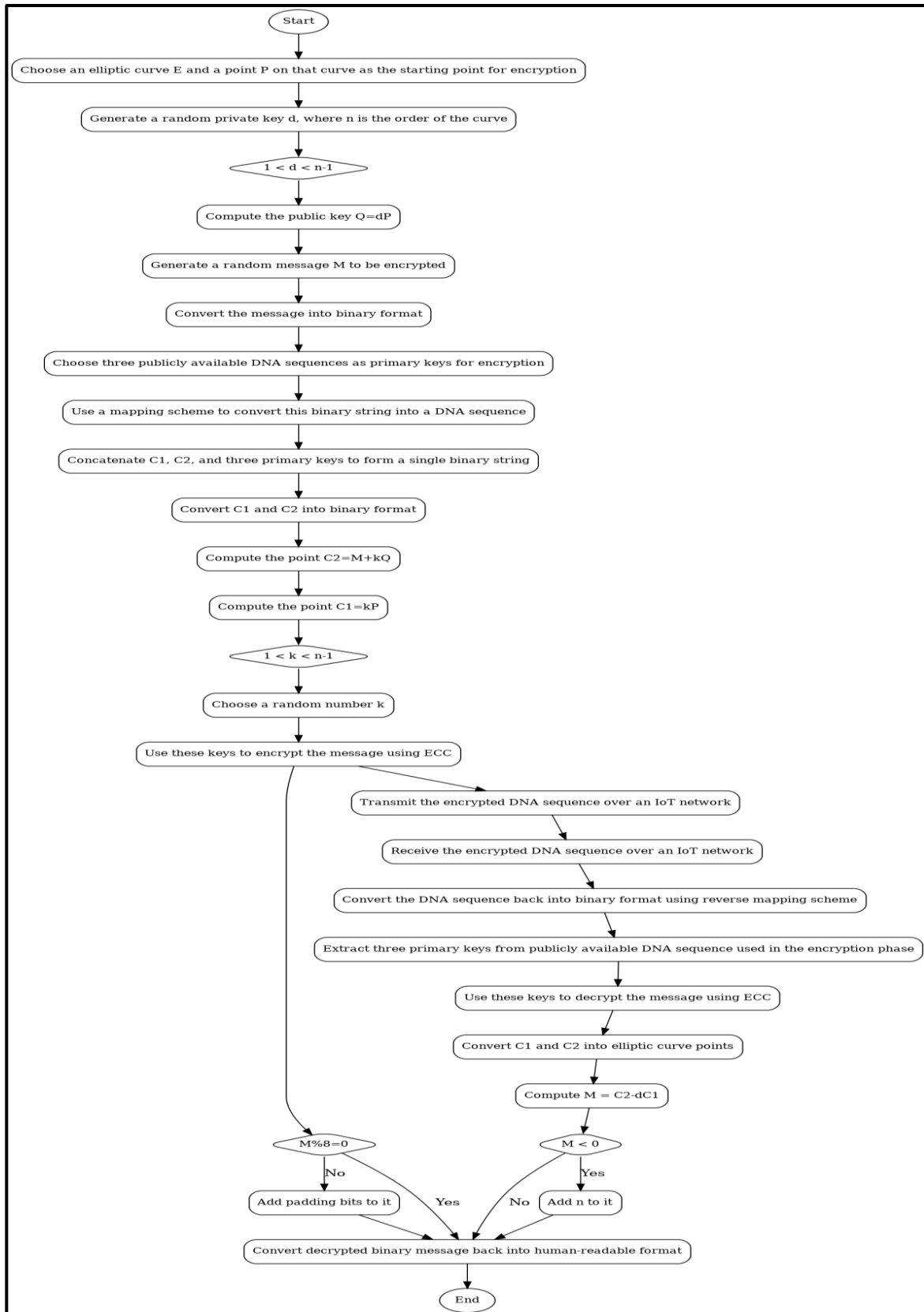


Figure 3.9: Flowchart of DNA-LWCS

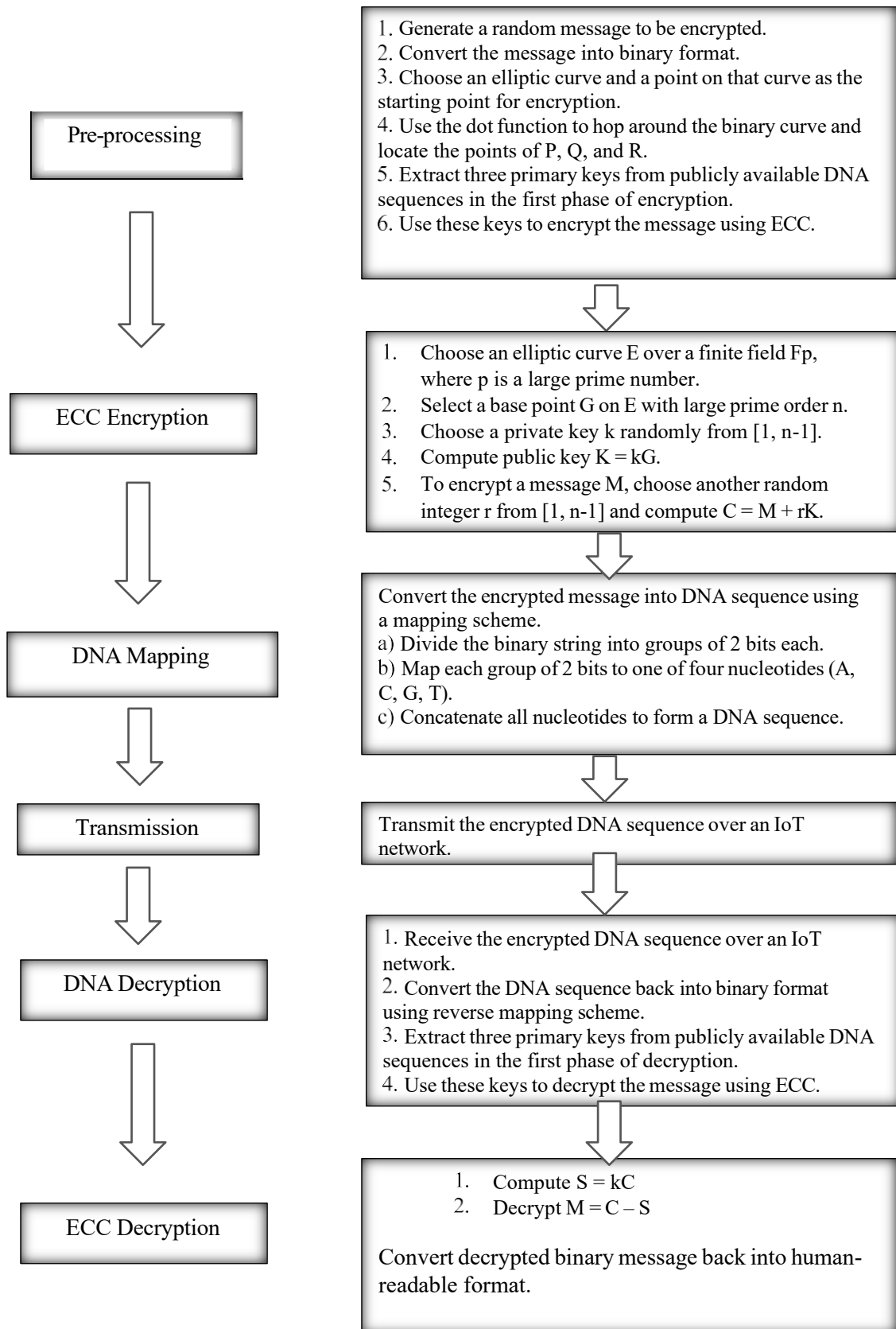


Figure 3.10: Description of Stages of DNA-LWCS

3.6.5 Process to Achieve Objective 2

Objective 2, to design an efficient security solution for IoT networks by proposing a DNA-based Lightweight Cryptography System (DNA-LWCS) using Elliptic Curve Cryptography (ECC), addresses IoT vulnerabilities with a lightweight, secure framework. This objective is achieved through the following six-step process, ensuring a scalable and resource-efficient encryption system:

- i. **DNA Key Generation:** Select publicly available DNA sequences, such as 50,000 nucleotides from a Kaggle dataset, for their inherent randomness. Convert these sequences into binary format to create cryptographic keys, providing a robust foundation for secure encryption (Blawat & Ceze, 2021).
- ii. **Message Preparation:** Convert the plaintext message into binary format, concatenate it with the DNA-derived key, and apply padding by adding zeros to form complete data blocks, ensuring consistent processing across IoT devices (Zou et al., 2021).
- iii. **Bitwise Encoding:** Perform substitution using random logical operations, such as XOR, Mod, or AND, followed by transposition through bit rotation, to enhance message randomness and obscure its structure, maintaining low computational complexity as detailed in Section 3.4 (Xu et al., 2020).
- iv. **ECC Encryption:** Generate ECC key pairs, consisting of a private key and a corresponding public key, using elliptic curve operations. Encrypt the encoded message by mapping it to points on the curve, leveraging ECC's lightweight and efficient properties for IoT security (Khan et al., 2020).
- v. **Key Optimization:** Apply genetic algorithms to mutate DNA-based keys and use deep learning models, such as Recurrent Neural Networks (RNN) and Gated Recurrent Units

(GRU), to evaluate and strengthen keys, ensuring efficiency and security (Section 3.6.1).

- vi. **Implementation:** Simulate the DNA-LWCS system using Python and TensorFlow on IoT devices, such as a Raspberry Pi with 8GB RAM, to validate its performance in resource-constrained environments.

This process delivers a lightweight, scalable encryption system that integrates DNA-based randomness with ECC's efficiency, addressing IoT vulnerabilities identified in Section 3.7.

3.7 Network Model

The Internet of Things (IoT) is a network of interconnected devices that exchange data to perform tasks, structured across four layers: Perception, Network, Middleware, and Application. The Perception Layer, also known as the sensor layer, includes devices like sensors, actuators, cameras, and microphones that collect raw environmental data. The Network Layer uses technologies such as Wi-Fi, Bluetooth, Zigbee, and cellular networks to transmit data packets to the cloud or other devices, managing topology, routing, and addressing. The Middleware Layer provides software services, including message and data brokers, to process and analyse data, enabling device-cloud interaction. The Application Layer delivers end-user services like smart homes, cities, and healthcare, generating insights and user interfaces. Data flows from the Perception Layer, where sensors collect it, through the Network Layer for transmission, to the Middleware and Application Layers for processing and visualisation, with each layer adding headers for routing and addressing. Figure 3.11 illustrates devices associated with each layer, such as sensors and routers.

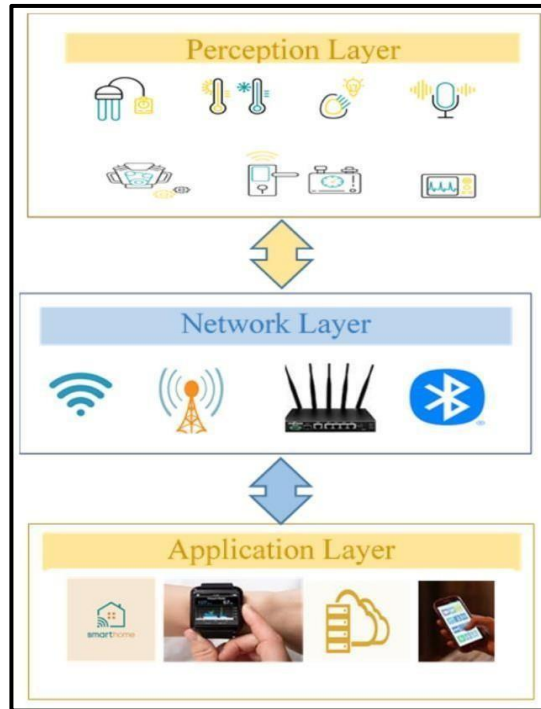


Figure 3.11: Devices under Each Layer of IoT

The Perception Layer is vulnerable to attacks like tampering, physical theft, and denial of service (DoS), where attackers manipulate or steal devices to access data. The Network Layer faces eavesdropping, man-in-the-middle (MiTM), and DoS attacks, with attackers intercepting or disrupting data transmission. The Middleware Layer is susceptible to SQL injection, cross-site scripting (XSS), and buffer overflow attacks, exploiting software vulnerabilities. The Application Layer encounters authentication, authorisation, and session hijacking attacks, targeting service access. To mitigate these, security measures like encryption, authentication, access control, and intrusion detection are proposed, informing the DNA-LWCS design.

This section achieves Objective 1 by analyzing the vulnerabilities of IoT networks and outlining a systematic process to identify and investigate threats, providing a foundation for the DNA-LWCS design. The Internet of Things (IoT) comprises interconnected devices that exchange data across four layers: perception, network, middleware, and application. The perception layer, including sensors and actuators, collects environmental data. The network

layer transmits data using technologies like Wi-Fi and Bluetooth. The middleware layer processes data through brokers, while the application layer delivers services like smart homes or healthcare.

Each layer faces distinct threats. The perception layer is vulnerable to tampering, physical theft, and denial-of-service (DoS) attacks, where attackers manipulate or steal devices. The network layer risks eavesdropping, man-in-the-middle (MiTM), and DoS attacks, intercepting or disrupting data. The middleware layer is susceptible to SQL injection and cross-site scripting (XSS), exploiting software weaknesses. The application layer faces authentication and session hijacking attacks, targeting service access. These vulnerabilities, combined with IoT challenges like resource constraints and high attack surfaces, highlight the need for a lightweight security solution like DNA-LWCS. Proposed mitigations, such as encryption and authentication, inform the system design in Sections 3.2–3.6.

3.7.1 Process to Achieve Objective 1

To achieve Objective 1, identifying and investigating potential threats and challenges of current IoT networks, the following five-step process is employed:

- i. **Literature Review:** Analyze existing studies on IoT vulnerabilities, such as DDoS, MiTM, and SQL injection attacks, to identify common threats across perception, network, middleware, and application layers (Meiser et al., 2022; Singh & Aggarwal, 2019).
- ii. **Network Simulation:** Use the NS-3 simulator to generate IoT network traffic data, such as 10,000 packets, to study attack patterns in a controlled environment across all IoT layers.
- iii. **Threat Mapping:** Categorize threats by layer, associating attacks like tampering in the perception layer or eavesdropping in the network layer, to assess their impact on security.
- iv. **Challenge Identification:** Document IoT-specific challenges, including resource constraints and high attack surfaces, to define security requirements for the DNA-LWCS system.

- v. **Visualization:** Summarize threats and associated devices across IoT layers to consolidate findings, guiding the design of the DNA-LWCS system in Sections 3.2–3.6.

This process produces a comprehensive threat profile, ensuring a thorough understanding of IoT vulnerabilities to inform the development of an efficient security solution.

3.8 Data Used

The DNA-LWCS algorithm uses Python for implementation due to its robust libraries (e.g., TensorFlow, NumPy) for cryptographic and machine learning tasks, ease of prototyping, and compatibility with IoT devices like Raspberry Pi, ensuring efficient simulation and deployment (Serag Eldin et al., 2023). The dataset comprises approximately 50,000 DNA nucleotide sequences from Kaggle (/Kaggle/input/covid19genome-sequence/sequence.fasta), chosen for their high randomness and public availability, enabling reproducible, entropy-rich key generation for IoT security; sequences were preprocessed by filtering non-standard nucleotides and converting to binary format (Chirakkarottu & Mathew, 2021). Images (Lena: color, 256x256, 196,608 bytes; Pepper: color, 512x512, 786,432 bytes; Photographer: grayscale, 512x512, 786,432 bytes) were selected as standard cryptographic benchmarks to enable direct performance comparison with AES, 3DES, and standalone ECC. Evaluation metrics, including Accuracy, Precision, Recall, F1-score, TNR, FAR, and FNR for key prediction, and encryption time, correlation, delay, throughput, PSNR, entropy, avalanche effect, and energy consumption for system performance, were chosen to assess ML model reliability, cryptographic strength, and IoT compatibility, respectively, aligning with standard cryptographic and IoT evaluation practices (Wen et al., 2024). Recurrent Neural Networks (RNN) and Gated Recurrent Units (GRU) were selected for their ability to model sequential DNA data, with GRUs offering lower computational overhead than LSTMs, suitable for IoT constraints (Inzamam-Ul-Hossain & Islam, 2024). Training used a 70-20-10 split (training,

validation, testing), 50 epochs, batch size of 32, and binary cross-entropy loss, with a fitness function assessing key randomness via entropy thresholds. Hyperparameters (learning rate: 0.001–0.1, hidden layers: 1–3, units: 64–256) were tuned via grid search with the Adam optimizer and early stopping to ensure efficiency.

```
TGGTICTAAATCACCCATTAGTACATCGATATCGGTAATTATACAGTTTCTGTTTACCTTTTACAATTGAAAGACAATTCTTATTTACAGAGCAACC
AATTGATCTTGTACCAACCAACCATATCCAAACGCAAGCTTCGATAATTTAAGTTTGTATGTGATAATATCAAATTTGCTGATGATTTAAACCAGTT
AACTGGTTATAAGAAACCTGCTTCAAGAGAGCTTAAAGTTACATTTTTCCCTGACTTAAATGGTGATGTGGTGGCTATTGATTATAAACTACACACC
CTCTTTTAAGAAAGGAGCTAAATTGTTACATAAACCTATTGTTTGGCATGTTAAACAATGCAACTAATAAAGCCACGTATAAACCAATACTTGGTGTAT
ACGTGTGCTTTGGAGCACAAACAGTTGAAACATCAAATTCGTTGATGTACTGAAGTCAGAGGACGCGCAGGGAATGGATAATCTTGCTGCGAAG
ATCTAAACAGCTCTCTGAAGAAGTAGTGAAAATCCTACCATAAGAAAGAGCTTCTTGAGTGTAATGTGAAAACCTACCGAAGTTGTAGGAGACATT
ATACTTAAACAGCAATAATAGTTTAAAAATTACAGAAGAGGTGGCCACACAGATCTAATGGCTGCTTATGTAGACAATTCTAGTCTTACTATTAAG
AAACCTAATGAATTATCTAGAGTATTAGGTTTGAACCCCTGCTACTCATGGTTTAGCTGCTGTTAATAGTGTCCCTGGGATACTATAGCTAATTAT
GCTAAGCCTTTTCTTAAACAAAGTTGTTAGTACAATACTAATACATAGTTACACGGTGTTTAAACNNIGTTTGTACNNATNNATGCCNNATTTCTTACTT
TATTGCTACAATTGTGTACTTTTANTAGAAGTACAAATNNNNNNNNNNNCAATNNNNCCGACTACTATAGCAAGAATACTGTTAAGAGTGTGCGGTA
AATTTGTCTAGAGGCTTCATTTAATTTTGAAGTCACCTAATTTTCTAACTGATAAATATTATAAATTTGGTTTTACTATTAAGTGTTCGCTAGG
TTCTTTAATCTACTCAACCGCTGCTTTAGTGTTTTAATGTCTAATTTAGGCATGCTTCTTACTGTACTGGTTACAGAGAAGGCTATTGAACTCTACT
AATGTCACTATTGCAACCTACTGTACTGGTCTTATACCTTGTAGTGTGTGCTTGTAGTGGTTAGATTCTTAGACACCTATCCTTCTTTAGAACTATACA
AATTACCATTTCATCTTTTAAATGGGATTAACTGCTTTTGGCTTGTGTGAGAGTGGTTTTTGGCATATATCTTTTCACTAGGTTTTTCTATGTACTT
GGATGTGGCTGCAATCATGCAATTGTTTTTCAGCTATTTTGCAGTACATTTTATAGTAATTTCTGGCTTATGTGGTTAATAATTAATCTGTACAAATGG
CCCCGATTTCAGCTATGGTTAGAATGTACATCTTCTTGCATCATTTTATTATGTATGGAAAAGTTATGTGCATGTTGTAGACGGTTGTAATTCATCAA
CTTGATGATGTGTTACAAACGTAATAGAGCAACAAGAGTCGAATGTACAATACTATTGTTAATGGTGTAGAAGGTCCTTTTATGTCTATGCTAATGGAG
GTAAAGGCTTTTGCAAACTACACAATTGGAATTGTGTTAATTTGTATACATTCGTGCTGCTAGTACATTATTAGTGATGAAGTTGCGAGAGACTTGT
CACTACAGTTTAAAGACCAATAATCCTACTGACCAGTCTTCTACATCGTTGATAGTGTACAGTGAAGAATGGTCCATCCATCTTTACTTTGATA
AAGTCTCACAGAAAACCTGTTACTTTATATGACATTAATGGCAATCTTCATCCAGATTCTGCCACTCTGTTAGTGACATTGACATCACTTTCTTAAA
GAAAGATGCTCCATATATAGTGGGTGATGTTGTCAAGAGGGTGTTTTAACTGCTGTGGTTATACCTACTAAAAAGGCTGGTGGCACTACTGAAATGC
TAGCGAAAGCTTTGAGAAAAGTGCCAACAGACAATTATATAACCACTTACCCGGGTCAAGGTTTAAATGGTTACACTGTAGAGGAGGCAAGACAGTG
CTTAAAAAGTGTAAGAGTGCCTTTACATCTACCATCTATTATCTCTAATGAGAAGCAAGAAATCTTGGAAGTGTCTTGGAAATTTGCGAGAAATG
CTTGACATGCAGAAGAAACACGCAAAATTAATGCCTGTCTGTGTGGAACTAAAGCCATAGTTTCAACTATACAGCGTAAATATAAGGGTATTAAAAAT
CTTAAAAAGTGTAAGAGTGCCTTTACATCTACCATCTATTATCTCTAATGAGAAGCAAGAAATCTTGGAAGTGTCTTGGAAATTTGCGAGAAATG
CTTGACATGCAGAAGAAACACGCAAAATTAATGCCTGTCTGTGTGGAACTAAAGCCATAGTTTCAACTATACAGCGTAAATATAAGGGTATTAAAAAT
CTTAAAAAGTGTAAGAGTGCCTTTACATCTACCATCTATTATCTCTAATGAGAAGCAAGAAATCTTGGAAGTGTCTTGGAAATTTGCGAGAAATG
CTTGACATGCAGAAGAAACACGCAAAATTAATGCCTGTCTGTGTGGAACTAAAGCCATAGTTTCAACTATACAGCGTAAATATAAGGGTATTAAAAAT
CTTAAAAAGTGTAAGAGTGCCTTTACATCTACCATCTATTATCTCTAATGAGAAGCAAGAAATCTTGGAAGTGTCTTGGAAATTTGCGAGAAATG
CTTGACATGCAGAAGAAACACGCAAAATTAATGCCTGTCTGTGTGGAACTAAAGCCATAGTTTCAACTATACAGCGTAAATATAAGGGTATTAAAAAT
CTTAAAAAGTGTAAGAGTGCCTTTACATCTACCATCTATTATCTCTAATGAGAAGCAAGAAATCTTGGAAGTGTCTTGGAAATTTGCGAGAAATG
CTTGACATGCAGAAGAAACACGCAAAATTAATGCCTGTCTGTGTGGAACTAAAGCCATAGTTTCAACTATACAGCGTAAATATAAGGGTATTAAAAAT
CTTAAAAAGTGTAAGAGTGCCTTTACATCTACCATCTATTATCTCTAATGAGAAGCAAGAAATCTTGGAAGTGTCTTGGAAATTTGCGAGAAATG
CTTGACATGCAGAAGAAACACGCAAAATTAATGCCTGTCTGTGTGGAACTAAAGCCATAGTTTCAACTATACAGCGTAAATATAAGGGTATTAAAAAT
CTTAAAAAGTGTAAGAGTGCCTTTACATCTACCATCTATTATCTCTAATGAGAAGCAAGAAATCTTGGAAGTGTCTTGGAAATTTGCGAGAAATG
CTTGACATGCAGAAGAAACACGCAAAATTAATGCCTGTCTGTGTGGAACTAAAGCCATAGTTTCAACTATACAGCGTAAATATAAGGGTATTAAAAAT
```

Figure 3.12: Raw Keys Data Set

To evaluate key accuracy, the DNA-LWCS algorithm employs Recurrent Neural Networks (RNNs) and Gated Recurrent Units (GRUs) to model sequential DNA data from the Kaggle dataset (see Figure 3.12 for a sample of the raw dataset used for key generation), as described in Chirakkarottu and Mathew (2021) and Inzamam-Ul-Hossain and Islam (2024). The training process used a 70-20-10 split (training, validation, testing), 50 epochs, a batch size of 32, and binary cross-entropy loss. A fitness function assessed key randomness based on entropy thresholds. Hyperparameters (learning rate: 0.001–0.1, hidden layers: 1–3, units: 64–256) were optimized via grid search with the Adam optimizer and early stopping to maximize model accuracy. Performance was evaluated using key prediction metrics, including Accuracy, Precision, Recall, F1-score, True Negative Rate (TNR), False Acceptance Rate (FAR), and False Negative Rate (FNR), as outlined in Wen et al. (2024).

The DNA-LWCS system processes data across multiple IoT layers to evaluate performance metrics. Sensors at the Perception Layer collect data, which is transmitted via the Network Layer using Wi-Fi or Bluetooth, with packet headers and trailers (e.g., source/destination addresses, protocol types) ensuring accurate transmission. The Middleware Layer processes data through brokers, while the Application Layer generates insights and user interfaces, supporting visualization and feedback to the Perception Layer for device control. This workflow enables assessment of metrics such as encryption time, correlation, and throughput, ensuring robust performance in resource-constrained IoT environments (Serag Eldin et al., 2023).

The DNA-LWCS algorithm processes nucleotide sequences to generate random numbers. A fitness function, previously described, evaluates key randomness, with results discussed in the next section. Subsequently, multiple deep-learning models are trained to assess the accuracy of predicted keys using metrics such as Accuracy, Precision, Recall, F1-score, TNR, FAR, and FNR (Inzamam-Ul-Hossain & Islam, 2024). These metrics are defined later in this chapter. Figure 3.13 illustrates each phase of the DNA-LWCS process pictorially.

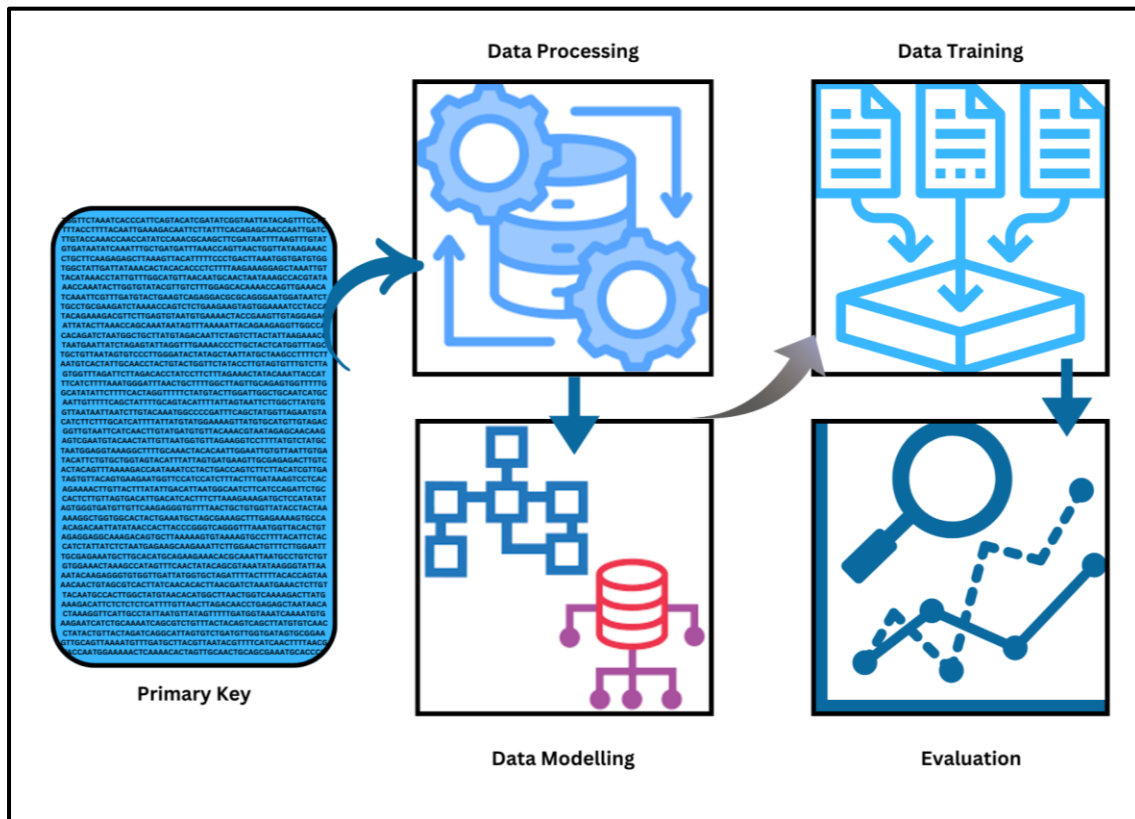


Figure 3.13: Data Pre-Processing

The provided text is mostly clear but contains some grammatical errors, awkward phrasing, and minor inconsistencies. Additionally, there is slight repetition in mentioning the images and their use. Below, I've revised the text to improve English grammar, eliminate repetition, enhance clarity, and ensure a professional tone while preserving the original technical details and meaning.

To evaluate the effectiveness of the DNA-LWCS algorithm on images, the following images were selected:

- i. Lena.bmp: A color image with a resolution of 256x256 and a size of 196,608 bytes.
- ii. Pepper.bmp: A color image with a resolution of 512x512 and a size of 786,432 bytes.
- iii. Photographer.bmp: A grayscale image with a resolution of 512x512 and a size of 786,432 bytes.

These images, shown in Figure 3.14, were chosen because benchmark studies used them to evaluate their algorithms with various metrics. To enable a comparative analysis, we applied the DNA-LWCS algorithm to the same images using identical metrics, demonstrating which algorithm performs better and in which aspects (Chirakkarottu & Mathew, 2021).



Figure 3.14: Image Data

3.8.1 Process to Achieve Objective 3

Objective 3, to evaluate the performance of the DNA-LWCS system against benchmarks (AES, 3DES, standalone ECC) using metrics such as encryption time, correlation, delay, throughput, PSNR, information entropy, avalanche effect, and energy consumption, validates the system's efficiency for IoT networks. This objective is achieved through the following five-step process:

- i. **Data Application:** Apply the DNA-LWCS algorithm to textual data, such as 50,000 nucleotide sequences from a Kaggle dataset, and images, including Lena, Pepper, and Photographer, to test its performance across diverse data types, as described in Section 3.8 (Chirakkarottu & Mathew, 2021).

- ii. **Metric Calculation:** Compute performance metrics, including encryption time to measure speed, correlation to assess variable relationships, delay to evaluate processing gaps, throughput to gauge data processing rate, PSNR to check image quality, information entropy to quantify randomness, avalanche effect to test sensitivity to input changes, and energy consumption to ensure IoT compatibility, as defined in Section 3.9 (Wen et al., 2024).
- iii. **Benchmark Comparison:** Conduct parallel tests of DNA-LWCS against AES, 3DES, and standalone ECC on identical hardware, such as a Raspberry Pi with 8GB RAM, to compare performance across the specified metrics.
- iv. **Analysis:** Use confusion matrices to evaluate machine learning models (RNN/GRU) for key strength prediction and apply statistical tests to compare DNA-LWCS performance against benchmarks, ensuring robust validation (Lai & Hua, 2024).
- v. **Result Presentation:** Summarize the performance results through histograms and tables to highlight DNA-LWCS's efficiency compared to benchmarks, with detailed findings presented in Chapter 4.

This process confirms DNA-LWCS's suitability for IoT networks, building on the system design in Sections 3.2–3.6 and validating its performance through comprehensive evaluation.

3.9 Evaluation Metrics

To evaluate the efficiency of our proposed MS-ADS model, we consider the standard performance evaluation metrics used to assess the typical ML and DL models. These evaluation metrics are Accuracy, Precision, Recall, F1-Score, FAR, and True Negative Rate (TNR). These performance evaluation metrics are calculated from the different attributes within the standard confusion matrix depicted in its binary version in Table 3.7. A Confusion Matrix is a two-dimensional matrix that details the information about the Actual ground classes and the

Predicted classes based on the experiment results (Wen, Xie, Wu, Lin, & Feng, 2024). The formal definition and the mathematical formula of different evaluation metrics considered in this study are as follows:

3.9.1 Accuracy

Accuracy is the proportion of correct classification of all the instances in the Pima Indians dataset, including the correctly classified normal and anomalous instances, to the total number of cases in the test set. It measures the productivity of the model as a whole. The formula for calculating the Accuracy is defined mathematically as follows;

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{Equation 3.1}$$

(Lai & Hua, 2024)

3.9.2 Precision

Precision refers to the number of accurately identified anomalous samples against the total number of samples labelled as anomalous. From a mathematical perspective, the Precision is defined as,

$$Precision = \frac{TP}{TP + FP} \quad \text{Equation 3.2}$$

(Shafique et al., 2021)

3.9.3 Recall

The recall is also called the detection rate equal to the True positive rate (TPR). It is expressed as the percentage of all the correct classifications of the anomaly samples by their total number. Recall can be formally defined mathematically as;

$$Recall = \frac{TP}{TP + FN} \quad \text{Equation 3.3}$$

(Wen et al., 2024)

3.9.4 F1 score

The F1 score represents the average of both precision and recall and provides the statistical measure embodying the system's accuracy. Mathematically F1 score is defined as:

$$F1\ score = \frac{2}{\left(\frac{1}{Precision} + \frac{1}{Recall}\right)} \quad \begin{array}{l} \text{Equation 3.4} \\ \text{(Shafique et al., 2021)} \end{array}$$

$$= 2 \times \left(\frac{Precision \times Recall}{Precision + Recall}\right) \quad \begin{array}{l} \text{Equation 3.5} \\ \text{(Shafique et al., 2021)} \end{array}$$

3.9.5 Histogram

A histogram is a graph that is used to represent the frequency of numerical data by employing rectangles, data is divided into intervals or bins and the number of observations. Bars are adjacent to each other without gaps to show that the data is continuous. They are mostly used for data analysis to visualize distribution of datasets. It is used for comparison of two or more datasets distributions.

3.9.6 Correlation

Correlation signifies association between two or more variables. It specifies the degree and direction of the relationship between the variables, how these variables change together positively (both increases mean 1), negatively (increase in one and decrease in the other means -1) or no correlation (0).

Formula coefficient of correlation (Khan et al., 2020):

$$E(x) = \frac{1}{n} \times \sum_{i=1}^n x_i \quad \text{Equation 3.6}$$

$$d(x) = \frac{1}{n} \times \sum_{i=1}^n (x_i - E(x))^2 \quad \text{Equation 3.7}$$

$$cov(x) = \frac{1}{n} \times \sum_{i=1}^n (x_i - E(x)) (y_i - E(y)) \quad \text{Equation 3.8}$$

$$r_{xy} = \frac{cov(x, y)}{\sqrt{d(x)}\sqrt{d(y)}} \quad \text{Equation 3.9}$$

$E(x)$ = expected value

Cov = covariance between x and y n = It is the number of data points x, y = It is the pairs in data set

$D(x)$ = variance of x

r_{xy} = Pearson correlation

3.9.7 Entropy

It is used to evaluate the efficiency of the encryption process. It is used when digital data cannot predict the content. Amount of uncertainty in the system is known as entropy of system. It is used to generate keys in cryptography. It is measured in bits. Variety of sources can be used to generate entropy such as radioactive decay.

Formula

$$entropy = \sum_{i=1}^n P_i \cdot \log_2(P_i) \quad \text{Equation 3.10}$$

(Namasudra et al., 2020)

n = number of data values

P_i = probability of occurrence of data value

3.9.8 Encryption Time

The amount of time taken to encrypt a piece of data is known as encryption time. It measures the time from the very beginning till the end of the process. The time is measured in seconds or milliseconds. Encryption time depends on the size of the data, complexity of the algorithm and processing power of the device. It is helpful in real time applications for fast encryption and transmission of data. It is important to consider encryption time while selecting encryption algorithm (Ispahany et al., 2024).

3.9.9 PSNR Comparison

PSNR stands for Peak Signal-to-Noise Ratio, which is a metric used to measure the quality of an image after encryption. It compares the original image with the encrypted image and provides a numerical value indicating the level of distortion or loss in the encrypted image. A higher PSNR value indicates better image quality (Shaltout et al., 2022).

3.9.10 Avalanche effect

It is used to check the sensitivity of the encryption process for any minor changes in the parameters. A single-bit change in the plaintext can change the whole cypher text with a probability of 0.5 for each bit. This property makes it more secure, as it is difficult for an attacker to attack. It is measured by the percentage of change bits (cipher text) when a single bit in plaintext is changed.

$$Aval. \ effect = \frac{\text{number of change bits in the key used}}{\text{total number of bits in the encrypted data}}$$

Equation 3.11
(Zhu & Zhu, 2020)

3.9.11 Throughput

The rate at which data, information, or signals can be processed by a system component in a given period affects the speed and efficiency of encryption and decryption operations (Zhou et al., 2018). It depends on the computational complexities of encryption algorithms, the size of the data, speed, memory resources, and processing power.

Formula

$$Throughput = \frac{\text{amount processed}}{\text{time}} \quad \text{Equation 3.12}$$

3.9.12 Delay

Waiting is the time between the beginning and completion of a process. It is the gap of time between when something was supposed to happen. Delays can be of different types, such as processing delay, propagation delay, transmission delay, queueing delay, etc. Factors that influence delay are distance, data rate, congestion, and hardware-software efficiency (Liu, Chen, Karvela, & Toumazou, 2020).

3.9.13 Energy Consumption

The amount of energy consumed by the system in a specific time period is known as energy consumption. It can be measured in un-kilowatt-hours (kWh), joules, or calories. Factors that can affect energy consumption are hardware architecture, efficiency of power supply, and optimization techniques (Di Gangi et al., 2018; Chandak et al., 2019). DNA-based encryption algorithms are more energy-efficient than traditional encryption algorithms. Formula is

$$Energy\ consumption = (power)(time) \quad \text{Equation 3.13}$$

3.10 Chapter Summary

This chapter outlines a structured methodology to develop and evaluate a DNA-based Lightweight Cryptography System (DNA-LWCS) integrated with Elliptic Curve Cryptography (ECC) to enhance security in IoT networks, addressing three objectives: threat analysis, system design, and performance evaluation. First, the threat analysis phase (Sections 3.7–3.7.1) identifies and investigates IoT vulnerabilities across perception, network, middleware, and application layers, using literature reviews and network simulations with tools like NS-3 to map attacks such as tampering and eavesdropping, establishing security requirements for DNA-LWCS. Second, the design phase (Sections 3.2–3.6) develops an efficient security solution by selecting DNA sequences from a Kaggle dataset for random key generation, preparing messages through binary conversion and padding, encoding with bitwise operations for added randomness, and applying ECC for lightweight encryption, with genetic algorithms and deep learning (RNN/GRU) optimizing key strength for IoT compatibility. Finally, the evaluation phase (Sections 3.8–3.9) assesses DNA-LWCS against benchmarks (AES, 3DES, standalone ECC) by applying it to textual data (50,000 nucleotide sequences) and images (Lena, Pepper, Photographer), measuring metrics like encryption time, correlation, delay, throughput, peak signal-to-noise ratio, information entropy, avalanche effect, and energy consumption on hardware like Raspberry Pi to confirm efficiency. This objective-driven approach, with clear processes for threat identification, system design, and performance evaluation, delivers a robust, scalable security framework, with comprehensive results detailed in Chapter 4.

CHAPTER 4

RESULTS AND ANALYSIS

4.1 Introduction

This chapter presents a detailed assessment and discussion of the different algorithms that pertain to the proposed system, with emphasis on the impact of the algorithms in improving IoT security. The developed algorithms are summarized at the end of each chapter, specifying how much they enhanced the system's security. In the given chapter, the list of algorithms applied at the design phase includes the presented DNA-based encryption technique and the Elliptic Curve Cryptography (ECC). The genetic algorithm can be pointed out as the most important one because it is the primary working tool of the programme that creates highly randomized and vital DNA keys. This step requires performing genetic operations, including bit rotation, mutations, and crossover operations that generate efficient encryption keys to secure IoT data. The analysis of the results of the experiment Next, the chapter illustrates the metrics used to evaluate the tested machine learning models that were trained to optimize the DNA keys. Any model we finally create is evaluated on or by other smaller factors like accuracy, precision, and the performance of crucial strength prediction. These models are compared to get the best model that will be effective for the system's performance. Last but not least, the chapter focuses on the evaluation metrics used in assessing the overall performance of the entire encryption system. This also involves marking down the number of encryption times, avalanche effect, correlation, and entropy as tools that inform entailing analysis of the present approach to prior benchmarks and methods that were earlier applied, as well as the overall system efficiency and security robustness.

4.2 Algorithms Segregation

The diagram showcases a multi-stage process where DNA-based keys and Elliptic Curve Cryptography (ECC) secure data encryption. The process initiates with Key Generation, which includes several steps: data collection kicks off the sequence, followed by training, packaging, validation, deployment, and continuous monitoring. These steps collectively produce a cryptographic key derived from DNA sequences, ensuring robustness and security through iterative refinement.

Next, the key advances to Encryption Stage 1, where the system processes it with initial steps like padding and bit manipulation. This stage formats the key correctly and prepares it for further encryption. Figure 4.1 depicts this by merging binary strings with modular addition, setting the stage for the final encryption phase.

In Encryption Stage 2, the prepared key and the accompanying message undergo encryption using Elliptic Curve Cryptography (ECC). ECC employs a graph of an elliptic curve to represent the mathematical operations that secure the data. This stage delivers an encrypted message, now secure and ready for transmission or storage over potentially insecure networks.

Overall, the diagram emphasizes an intricate, layered approach to data security, integrating the innovative use of DNA-based keys with the established ECC technique to safeguard data from generation through encryption to transmission, as illustrated in Figure 4.1.

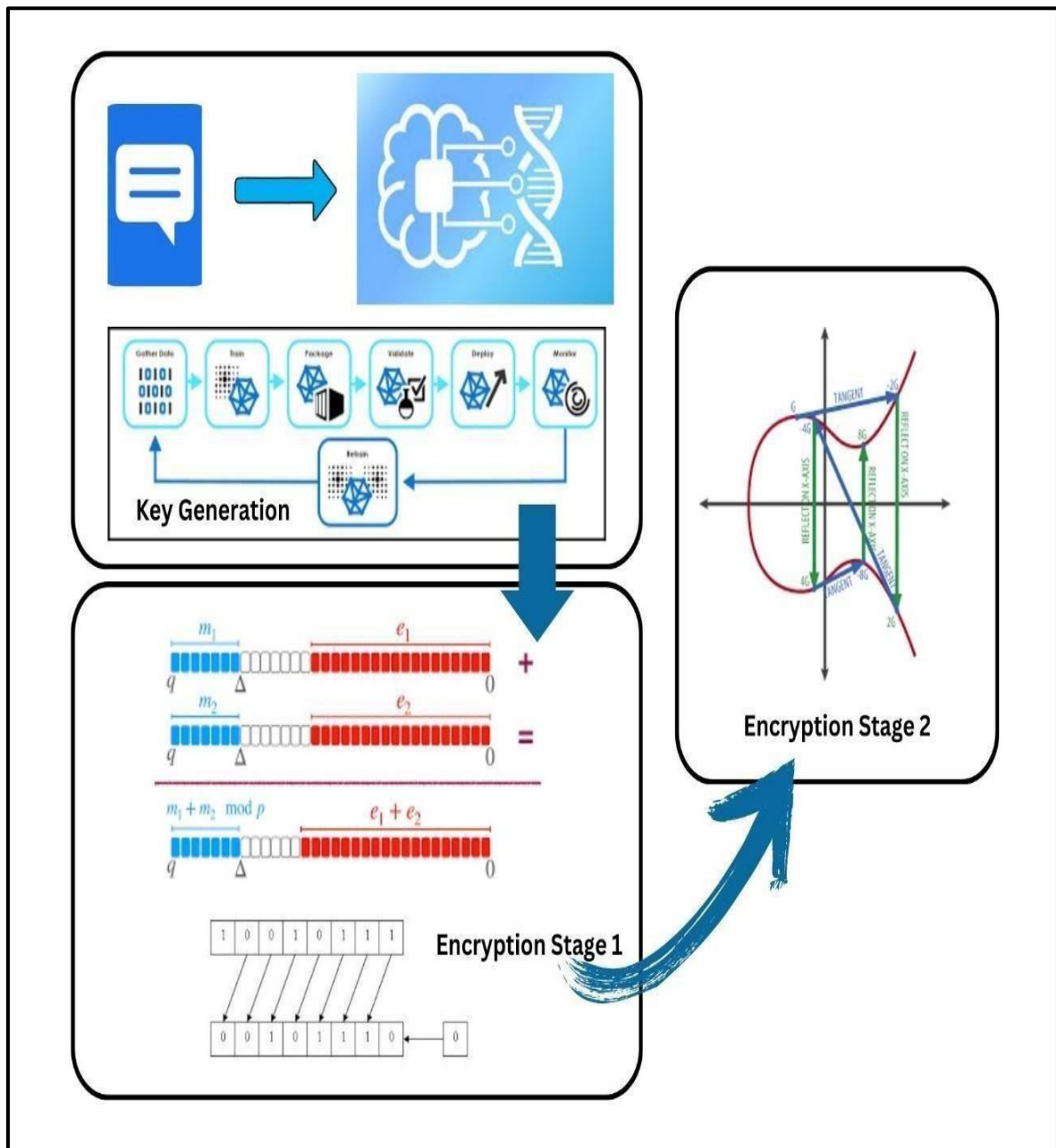


Figure 4.1: Data Pre-Processing

4.3 List of Algorithms

This section incorporates numerous algorithms that deploy DNA-LWCS effectively. The process begins with Key Generation, a fundamental and critical phase that encompasses major activities to extract keys from publicly available DNA sequences. The algorithm's steps are detailed in Algorithm 4.1.

Algorithm 4.1 Key Generation for LWCS	
Input Parameters of the elliptic curve $E : y^2 = x^3 + ax + b$ and basepoint $P = (x, y)$	
Output Private key d and public key Q	
1.	BEGIN
2.	<i>choose coefficients a and b for the elliptic curve equation</i>
3.	<i>choose coordinates x and y satisfying the curve equation</i> $y^2 = x^3 + ax + b$
4.	WHILE true DO
5.	<i>generate a random integer d</i>
6.	IF d is not within $1 < d < n - 1$ THEN
7.	<i>Generate a new random integer d</i>
8.	ENDIF
9.	END WHILE
10.	WHILE true DO
11.	<i>initialize Q as the base point P</i>
12.	<i>initialize $i \rightarrow \text{len} - 2$</i>
13.	<i>represent d in binary format.</i>
14.	<i>double Q (perform elliptic curve point doubling)</i>
15.	IF current bit of d is 1 THEN
16.	<i>add P to Q (perform elliptic curve point addition)</i>
17.	<i>decrement i by 1</i>
18.	ENDIF
19.	END WHILE
20.	END

Out of these two algorithms, the system selects Algorithm 1 for the Key Generation Mechanism of DNA-LWCS. A key exchange starts by determining the elliptic curve E through its equation $y^2 = x^3 + ax + b$ and a base point $P = (x, y)$ on that curve. This process involves choosing random coefficients a and b for the curve equation, along with suitable coordinates x

and y that satisfy this equation. Afterward, the algorithm enters an infinite loop to generate the private key d , ensuring it falls within the required range to maintain security. If the randomly generated integer fails to meet this condition, the system generates a new random integer until it produces a valid d . After establishing the private key K , the algorithm isolates the public key Q through elliptic curve computation.

To determine the public key, another infinite loop follows where the system sets Q as P , converts the base point and d into binary form, and processes each bit of the scalar d . For each bit, the algorithm doubles the first point of the elliptic curve on point Q , and if the current bit is one, it adds point P to point Q . This process continues until it processes all bits of d , ultimately yielding the public key Q .

The second step proves extensive and encompasses various procedures to boost cryptographic key security using fitness testing and machine learning. Data input initiates the process, where the system inputs the cryptographic key, also known as the population. To achieve this, the system collects information from the internet, specifically from GenBank, the NCBI database, which stores a vast amount of publicly available DNA sequence data. These DNA sequences serve as the first pair of keys in the algorithm; the system then subjects these keys to a fitness test to evaluate their strength and detect flaws like repeated sequences.

Following this test, the system categorizes the keys into two groups: Strong Keys and Weak Keys. It produces an Improved Key Set from strong keys that pass the fitness test with or without issues, potentially strengthening or linking effective keys to enhance security. Conversely, weak keys undergo a machine learning model built on TensorFlow, which either reinforces the identified weak keys to address observed vulnerabilities or analyses weaknesses to improve future key generation. The system constantly tests and checks the model's performance to confirm its effectiveness. This feedback adjusts the initial data input and fitness test standards, creating a cyclical pattern. This iterative process continues until the system

determines the optimal key size for cryptographic use, meeting the security requirements for encryption. A diagram in Figure 4.2 illustrates this process.

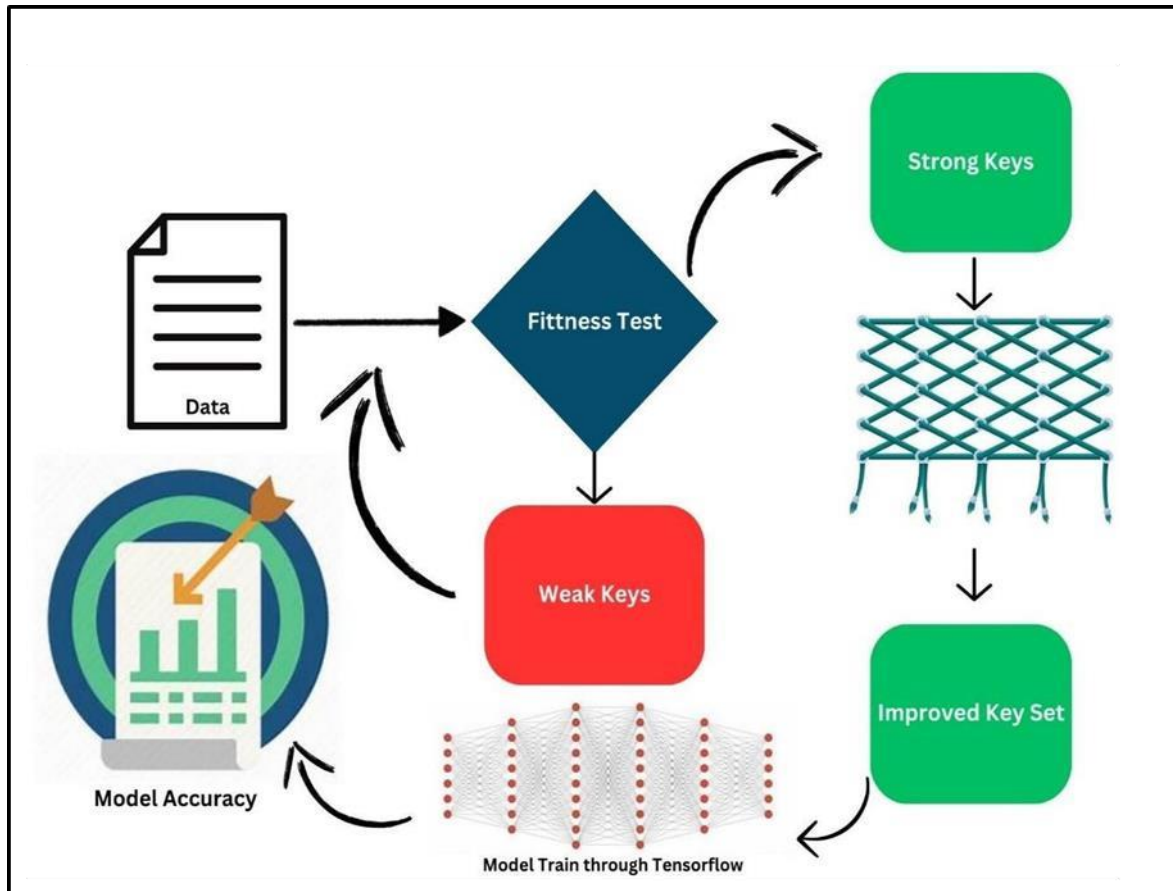


Figure 4.2: Generation of Strong and Weak Keys

This section discusses the algorithms linked with this stage as follows:

Algorithm 4.2 Read Data Sequence
Input sequence, num_key, key_length Output List of Keys of length key_length 1. BEGIN 2. Assign Path → File Path 3. Read the sequence provided in the available dataset 4. WHILE true DO 5. DNA_{Sequence} → Read from the file 6. RETURN DNA_{Sequence} 7. END

Algorithm 4.2 handles data reading of DNA sequence data for subsequent processing. Firstly, it initializes the file path where the algorithm stores the DNA sequence, informing it of the data's location. After assigning the file path, the algorithm reads the DNA sequence from the file. It then stores the sequence in a variable for use in subsequent operations, such as generating cryptographic keys or interpreting genetic sequences.

The algorithm employs an infinite loop to continuously read DNA sequence data, creating a design that supports a constant data feed, especially if the sequence data updates or processes in real time. Once read, the algorithm returns the sequence as output for the next step, like key generation. However, the loop terminates when needed, and the primary goal of the algorithm ensures ongoing access to the DNA sequence for other uses. This approach synergizes effectively with applications requiring constant monitoring or real-time data analysis.

Algorithm 4.3 Generate Initial Population
Input <i>sequence, num_key, key_length</i> Output Array of generated keys 1. BEGIN 2. <i>initialize an empty array population []</i> 3. WHILE <i>true</i> DO 4. IF <i>i < num_keys</i> THEN 5. <i>start_index</i> $\rightarrow$ <i>random integer between 0 and (len(sequence) – key_length)</i> 6. <i>population []</i> $\rightarrow$ <i>population [] + key</i> 7. <i>i</i> $\rightarrow$ <i>i + 1</i> 8. ENDIF 9. <i>return Population []</i> 10. END WHILE 11. END

Algorithm 4.3 constructs an array of keys from a given DNA sequence. The process starts by initializing an empty array called *population*, which stores the keys generated throughout the algorithm. It then enters a continuous loop that operates until it generates the desired number of keys, specified by *num_keys*.

Within this loop, the algorithm first checks if the current number of generated keys (*i*) falls below the required number (*num_keys*). If this condition holds, it proceeds by selecting a random *start_index* within the DNA sequence, ensuring the key fits the specified *key_length* without exceeding the sequence's bounds. Using this *start_index*, the algorithm extracts a key from the sequence and adds it to the *population* array.

The algorithm then increments the counter *i* to track the number of keys generated so far. This process continues until the *population* array holds the required number of keys. Once this condition is met, the algorithm exits the loop and returns the *population* array, now filled with the generated keys. This array stands ready for use in subsequent cryptographic or data analysis applications.

Algorithm 4.4 Fitness Test	
	Input key – A sequence of characters representing the key
	Output penalty – An integer representing the fitness score, based on the number of repeated adjacent nodes
1.	FUNCTION <i>FitnessTest</i> (<i>key</i>)
2.	<i>initialize penalty</i> $\leftarrow$ 0
3.	FOR <i>i</i> $\leftarrow$ 0 to <i>len(key)</i> – 2
4.	IF <i>key</i> [<i>i</i>] = <i>key</i> [<i>i</i> + 1] THEN
5.	<i>penalty</i> $\leftarrow$ <i>penalty</i> + 1
6.	ENDIF
7.	END FOR
8.	END FUNCTION

In Algorithm 4.4 Fitness Test, the system determines the fitness of a cryptographic key by identifying repeated adjacent characters in the key. The programme begins by declaring a variable named penalty, setting it to 0 to track these repetitions. It then traverses the key, examining each character and comparing it with the next in the sequence. This process continues until it scans all characters, possibly via a sub-process or intentionally after completing the scan. Finally, the system calculates the penalty score, which reflects the number of repeated adjacent nodes, and returns it as the final result. A lower penalty score indicates a stronger and more secure key, making this test crucial for generating cryptographic keys.

Algorithm 4.5 Categorize Keys Based on Fitness

Input *Array of generated keys as initial population []*

Output *strong_keys [] – Array of keys with no penalties, weak_keys [] – Array of keys with penalties*

```

1.  BEGIN
2.  initialize categorized_keys []  $\leftarrow \emptyset$ 
3.  FOR key in initial_population [] THEN
4.    penalty  $\leftarrow$  FitnessTest(key)
5.    categorized_keys[ ]  $\leftarrow$  categorized_keys[ ] + (key, penalty)
6.  END FOR
7.  initialize strong_keys []  $\leftarrow \emptyset$ 
8.  initialize weak_keys []  $\leftarrow \emptyset$ 
9.  FOR (key, penalty) in categorized_keys [] THEN
10.   IF penalty = 0 THEN
11.    strong_keys []  $\leftarrow$  strong_keys [] + key
12.   ELSE
13.    weak_keys []  $\leftarrow$  weak_keys [] + key
14.   ENDIF
15. END FOR
16. FUNCTION Print Strong Keys
17.   FOR i  $\leftarrow$  1 to len (strong_keys []) DO
18.    PRINT Key, i, "", `strong_keys[i - 1]
19.   END FOR
20. END FUNCTION
21. FUNCTION Print Weak Keys
22.   FOR i  $\leftarrow$  1 to len (weak_keys []) DO
23.    PRINT Key, i, "", `weak_keys[i - 1]
24.   END FOR
25. END FUNCTION
26. END

```

Algorithm 4.5 evaluates a set of generated keys and categorizes them into two groups—strong keys and weak keys—based on their fitness levels. The algorithm accepts an array called `initial_population` as input, containing the generated keys, and produces two arrays, `strong_keys` and `weak_keys`, as outputs.

The process begins by creating an empty list named `categorized_keys`. This step precedes the application of the `FitnessTest` to the keys in the `initial_population` array, which returns a penalty for each key. The algorithm then forms a tuple from the key and its penalty, adding it to the `categorized_keys` array.

After assessing and categorizing all keys, the algorithm initializes two new empty arrays, `strong_keys` and `weak_keys`. It then iterates through the `categorized_keys` array in the same loop, segregating the keys based on their penalties. The algorithm appends keys with a penalty of 0 to the `strong_keys` list and keys with any penalties to the `weak_keys` list. Two functions, `Print Strong Keys` and `Print Weak Keys`, display the elements in the `strong_keys` and `weak_keys` lists, respectively (Jamshidi et al., 2020). Each function traverses its respective array and prints every key in a specific format. The process concludes with these operations, categorizing and displaying the keys according to their fitness levels.

Algorithm 4.6 Crossover Operation
Input <i>Two parent arrays; parent1[] and parent2[]</i> Output <i>Two child arrays, child1[] and child2[]</i> 1. FUNCTION <i>crossover</i> (parent1 [],parent 2 []) 2. WHILE TRUE DO 3. <i>initialize crossover point</i> $\leftarrow$ <i>Random integer between 1 and (len(parent1) - 1)</i> 4. <i>initialize child1[]</i> $\leftarrow$ <i>parent1[0] + parent2[crossover_point (parent2)]</i> 5. <i>initialize child2[]</i> $\leftarrow$ <i>parent2[0] + parent1[crossover_point (parent1)]</i> 6. END WHILE 7. RETURN <i>child1[],child2[]</i> 8. END FUNCTION

Algorithm 4.6 generates two new child vectors, child1[] and child2[], from two parent vectors, parent1[] and parent2[]. This algorithm plays a key role in genetic algorithms, enabling reproduction to create the next generation of a population inherited from both ancestors. The algorithm begins with a function named crossover, defined with two array parameters, parent1[] and parent2[]. Within this function, a while loop WHILE TRUE initiates the crossover operation, continuing until it selects a valid crossover point.

This loop generates a random integer value as the crossover point, ranging between 1 and the length of the parent arrays minus one (1 to len(parent1)-1). This point designates where the two parent arrays split to form the children (Wu et al., 2021). Based on this crossover point, the algorithm creates two new child arrays, child1[] and child2[]. For child1[], it combines the segment of parent1[] from the start to the crossover point with the remaining segment of parent2[] from the crossover point. Similarly, child2[] results from concatenating the portion of parent2[] up to the crossover point with the rest of parent1[]. After producing valid children, the loop stops, and the function outputs the two new child arrays, child1[] and child2[]. These offspring arrays mark the algorithm's conclusion within the genetic algorithm, paving the way for further operations like mutation or selection.

Algorithm 4.7 Mutate Key Based on Mutation Rate	
Input String key, Float mutation rate (default 0.1)	
Output String mutated key	
1.	FUNCTION mutation rate (key,rate)
2.	FOR each nucleotide in key DO
3.	IF Random number between 0 and 1 < mutation rate THEN
4.	Append a random character from 'A','T','C','G' to mutated key
5.	ELSE
6.	Append nucleotide to mutated key
7.	ENDIF
8.	END FOR
9.	RETURN muted key
10.	END FUNCTION

A key element, known as the key, triggers mutations in a given string using Algorithm 4.7 by setting a specified mutation rate. This algorithm supports generational algorithms, mimicking natural mutations where segments of a genetic sequence vary informally. The algorithm begins by defining a function named `mutation_rate` that accepts two inputs: a key string and a float number rate as the mutation rate, defaulting to 0.1. This mutation rate dictates the probability that the algorithm randomly alters one of the key's elements.

Inside the function, the algorithm processes the key by iterating through each nucleotide. For every nucleotide, it generates a random number between 0 and 1 as the methylation score. If this random number falls below the given mutation rate, the algorithm inserts a random character from the set {'A', 'T', 'C', 'G'} into a new edited string called `mutated_key`, based on the amino acid and nodes at that position in the sequence. If the generated random number equals or exceeds the mutation rate, the algorithm appends the key value to the `mutated_key` string without modification.

After the loop processes all nucleotides in the key, the function returns the `mutated_key`, now containing the original sequence with some characters potentially altered based on the mutation rate. The algorithm concludes with this mutated version of the input key, ready for use in further evolutionary computations or analysis.

Algorithm 4.8 enhances a set of relatively poor keys using crossover and mutation genetic techniques across several generations, transforming weak keys into strong ones. It starts by defining its input parameters, including the number of generations (`generations`) to run the evolutionary process, the mutation rate (`mutation_rate`), and two arrays: `weak_keys[]` containing keys needing strengthening and `strong_keys[]` holding keys that require no further weakening.

The algorithm initiates a loop that iterates over a fixed number of generations. In each generation, it randomly selects two parent keys, `parent1` and `parent2`, from `weak_keys[]`. It then passes these parent keys to a crossover function, which generates two new child keys, `child1` and `child2`.

Algorithm 4.8 Genetic Algorithm to Improve Weak Keys**Input** Integer generations, Float mutation_rate, Arrays weak_keys [], strong_keys []**Output** Array strong_keys [] (updated with improved keys)

```

1.  BEGIN
2.  FOR generation  $\leftarrow$  1 to generations DO
3.    Select parent1, parent2  $\leftarrow$ 
       Randomly select two keys from weak_keys []
4.    child1, child2  $\leftarrow$  crossover (parent1, parent2)
5.    child1  $\leftarrow$  mutate (child1, mutation_rate)
6.    fitness_child1  $\leftarrow$  fitness_test(child1)
7.    fitness_child2  $\leftarrow$  fitness_test(child2)
8.    IF fitness_child1 = 0 THEN
9.      Remove parent1 from weak_keys []
10.   Append child1 to strong_keys []
11.   ELSEIF fitness_child2 = 0 THEN
12.     Remove parent2 from weak_keys []
13.     Append child2 to strong_keys []
14.   ENDIF
15. END FOR
16. FOR i  $\leftarrow$  1 TO len (strong_keys []) DO
17.   PRINT "Key", i, "", strong_keys[i - 1]
18. END FOR
19. END

```

After crossover, the algorithm applies a mutate function to alter one of the two children, specifically child1, based on the established mutation rate. It then uses a fitness_test function to evaluate the child keys' fitness, producing a fitness value. If fitness_child1 equals 0, the algorithm removes parent1 from weak_keys[] since child1 incurs no penalties, and it adds child1 to strong_keys[]. Similarly, if fitness_child2 equals 0, it deletes parent2 from weak_keys[] and adds child2 to strong_keys[] as a strong key.

This process repeats for the desired number of generations, further refining weak keys to boost their effectiveness and security. After completing all generations, the algorithm prints all keys in the strong_keys[] array, first reviewing and storing each improved key. The theoretical algorithm concludes with an updated strong_keys[] array, incorporating the initial vital keys and those enhanced by the genetic algorithm.

Algorithm 4.9 Encoding of DNA Sequence**Input** String *key* (DNA sequence)**Output** Matrix *one_hot* (one-hot encoded representation)

```
1.  FUNCTION Encoding(key)
2.  initialize mapping  $\leftarrow \{'A' 0, 'T' 1, 'C' 2, 'G' 3\}$ 
3.  initialize one_hot  $\leftarrow$  Zero matrix of size (len(key), 4)
4.  FOR i  $\leftarrow 0$  TO len(key) - 1 DO
5.    set one_hot [i, mapping[key[i]]]  $\leftarrow 1$ 
6.  END FOR
7.  RETURN one_hot
8.  END FUNCTION
```

Algorithm 4.9 encodes a DNA sequence into a one-hot encoded matrix. One-hot encoding is a fundamental process in computational biology and machine learning. It transforms nominal data into binary vectors. The algorithm begins by defining a function named *Encoding* that accepts a single input. For testing, the function takes a single string parameter, *key*, containing the DNA sequence. The genetic material consists of four bases: 'A,' 'T,' 'C,' and 'G.'

Within the function, the algorithm initializes a dictionary named *mapping* to assign each nucleotide a unique integer. It maps 'A' to 0, 'T' to 1, 'C' to 2, and 'G' to 3. This mapping facilitates indexing in the one-hot matrix. Next, the algorithm creates a zero matrix named *one_hot* with dimensions (len(*key*), 4). The number 4 represents the four nucleotides: 'A,' 'T,' 'C,' and 'G.'

The algorithm then executes a FOR loop, iterating from index 0 to len(*key*)-1, processing each character in the DNA sequence. For each character at position *i*, the algorithm uses the character to determine which column in the *one_hot* matrix receives a value of 1, based on the mapping dictionary. The row index corresponds to the nucleotide's position in the DNA sequence, while the column index depends on the nucleotide's mapped integer value. To indicate the presence of a nucleotide at position *i*, the algorithm sets the matrix element at [*i*, *mapping*[*key*[*i*]]] to 1.

After the loop processes all nucleotides and populates the one_hot matrix, the function returns the matrix. This binary matrix serves as input for various computational algorithms, such as those in machine learning or bioinformatics. Ultimately, the algorithm produces a one-hot encoded representation of the DNA sequence based on the described mapping.

Algorithm 4.10 Generate Training Data for DNA Sequences
Input Integer num_samples (default 10000), Integer key_length (default 20) Output Arrays X_train, y_train (training input and output data) <pre> 1. BEGIN 2. initialize X_train ← [] 3. Initialize y_train ← [] 4. FOR _ ← 0 TO num_samples - 1 DO 5. generate dna_sequence of length key_length with ['A', 'T', 'C', 'G'] 6. FOR i ← 1 TO key_length - 1 DO 7. IF dna_sequence[i] = dna_sequence[i - 1] THEN 8. dna_sequence[i] ← ['A', 'T', 'C', 'G']! = dna_sequence[i - 1] 9. ENDIF 10. END FOR 11. dna_sequence ← append to X_train [] 12. dna_sequence ← append to Y_train [] 13. RETURN X_train[], Y_train[] 14. END </pre>

Algorithm 4.10 generates an optimal dataset for developing machine learning (ML) training models, particularly for DNA sequence analysis tasks. The algorithm creates artificial DNA sequences and ensures they meet conditions for realistic and diverse data. It accepts two input parameters: num_samples, which specifies the number of DNA sequences to generate (default: 10,000), and key_length, which defines the length of each DNA sequence (default: 20). The algorithm outputs two vectors: X_train, containing the DNA sequences, and y_train, containing the corresponding targets.

The algorithm initializes two empty lists, X_train and y_train, to store the training data. It then enters a loop that iterates from 0 to num_samples - 1, generating a DNA sequence in

each iteration. For each iteration, the algorithm produces a random string of length `key_length` using the nucleotides 'A,' 'T,' 'C,' and 'G.' After generating a DNA sequence, the algorithm scans it to prevent consecutive repeating nucleotides. It uses a nested loop, iterating from index 1 to `key_length - 1`. For each position `i`, if `dna_sequence[i]` equals `dna_sequence[i - 1]`, indicating two consecutive identical nucleotides, the algorithm randomly replaces `dna_sequence[i]` with a different nucleotide. This step enhances sequence randomization by avoiding consecutive repeating characters.

Once the algorithm modifies the DNA sequence to meet this condition, it adds the sequence to `X_train` and its corresponding target to `y_train`. This process repeats until the algorithm generates `num_samples` sequences. Finally, the algorithm returns the `X_train` and `y_train` arrays. The algorithm shapes these arrays into two separate 2D arrays, suitable as input and output data for training machine learning models on DNA sequence data. The algorithm minimizes sequence duplication and produces diverse sequences for further computational analysis or machine learning applications.

Algorithm 4.11 Generate DNA Sequences Using a Trained Model

Input `model`, Integer `length` (default 20), Integer `num_generations` (default 1)

Output *Array generated_sequences (List of generated DNA sequences)*

```

1.  BEGIN
2.  generated_sequences ← empty list
3.  FOR i ← 1 TO num_generations DO
4.      initialize seed_sequence ←
        Randomly choose a nucleotide from ['A', 'C', 'T', 'G']
5.  set current_sequence ← seed_sequence
6.  FOR i ← 1 TO length - len(seed_sequence) DO
7.      encoded_sequence ← current_sequence
8.      model.predict(encoded_sequence)
9.  WHILE TRUE DO
10.     p ← predictions [0, -1]
11.     next_nucleotide_index ← Randomly choose an index from [0, 1, 2, 3]
```

```

12. FUNCTION index_to_nucleotide(next_nucleotide_index)
13.   next_nucleotide  $\leftarrow$  next_nucleotide_index
14.   IF next_nucleotide  $\neq$  Last_nucleotide_of_current_sequence THEN
15.     BREAK
16.   ENDIF
17.   END FUNCTION
18. END WHILE
19.   APPEND current_sequence  $\leftarrow$  next_nucleotide
20. END FOR
21.   generated_sequences  $\leftarrow$  current_sequence
22. END FOR
23.   RETURN generated_sequences
24. END

```

Algorithm 4.12 generates synthetic DNA sequences using a pre-trained model. It accepts two parameters: length, which specifies the number of nucleotides per DNA sequence (default: 20), and num_generations, which defines the number of sequences to generate (default: 1). The algorithm outputs an array named generated_sequences containing the generated sequences.

The algorithm initializes an empty list, generated_sequences, to store the results. It then iterates num_generations times. In each iteration, the algorithm selects an initial nucleotide from ['A', 'C', 'T', 'G'] as seed_sequence to start the current_sequence.

The algorithm enters a nested loop that continues until the sequence reaches the specified length. It encodes the current_sequence, and the model predicts the next nucleotide. The algorithm selects a random nucleotide index from the prediction and converts it to the corresponding nucleotide. To avoid redundancy, the algorithm checks that the predicted nucleotide differs from the last nucleotide in the sequence. Once it identifies a valid nucleotide, the algorithm appends it to current_sequence. When the sequence reaches the desired length, the algorithm adds current_sequence to generated_sequences. This process repeats for the specified number of generations, generating and evaluating unique sequences. Finally, the algorithm returns the generated_sequences list for further analysis or use.

Algorithm 4.12 Convert Nucleotide to Binary
Input nucleotide Output binary_representation 1. BEGIN 2. FUNCTION Nucleotide – to – Binary 3. initialize mapping $\rightarrow \{ 'A' '00', 'T' '01', 'C' '10', 'G' '11' \}$ 4. set binary_representation $\rightarrow$ mapping[nucleotide] 5. RETURN binary_representation 6. END FUNCTION 7. END

Algorithm 4.12 transforms a specified nucleotide into its binary form. It accepts a single input, nucleotide ('A,' 'T,' 'C,' or 'G'), and outputs its binary representation. The algorithm defines a function called Nucleotide_to_binary. Within the function, the algorithm creates a mapping dictionary to link each nucleotide to a specific two-bit binary code: 'A' to '00', 'T' to '01', 'C' to '10', and 'G' to '11'. The function retrieves the binary code for the input nucleotide from the dictionary and stores it in binary_representation.

Finally, the function returns binary_representation, completing the conversion. This binary representation suits additional computational analyses or machine learning models requiring DNA sequences in binary form. The algorithm concludes after successfully converting the nucleotide to binary.

Algorithm 4.13 Convert DNA Key to Binary**Input** key (DNA sequence)**Output** binary_key

```
1.  BEGIN
2.  FUNCTION DNA_Key – to – Binary
3.  initialize binary_key [ ]
4.  FOR each nucleotide in key DO
5.    FUNCTION nucleotide_to_binary(nucleotide)
6.    APPEND binary_key [ ] ← binary_key
7.    END FUNCTION
8.  END FOR
9.  RETURN binary_key
10. END FUNCTION
11. END
```

Algorithm 4.13 converts a DNA sequence, referred to as key, into binary form. It maps each nucleotide ('A,' 'T,' 'C,' 'G') to its binary equivalent. The algorithm accepts a DNA sequence (key) as input and outputs a binary_key representing the sequence in binary form. The algorithm defines a function called DNA_Key_to_Binary. Within the function, the algorithm initializes an empty list named binary_key to store the binary forms of the nucleotides. The algorithm iterates through each nucleotide in the input key. For each nucleotide, it calls a function named nucleotide_to_binary, which converts the nucleotide to its binary representation ('A' to '00', 'T' to '01', 'C' to '10', 'G' to '11'). The algorithm appends the binary result to the binary_key array. After processing all nucleotides and populating binary_key, the function returns the list of binary numbers. The final binary_key contains the complete binary representation of the input DNA sequence, suitable for further data processing or analysis

Algorithm 4.14 Encryption Process for DNA-LWCS

Input Parameters of the elliptic curve $E: y^2 = x^3 + ax + b$ and base point $P = (x, y)$, order of the curve, DNA sequences, Private key

Output Encrypted Message

1. *BEGIN*
2. **WHILE TRUE DO**
3. *initialize E as the elliptic curve, base point and order of the curve*
4. *generate a random message M*
5. *convert the M into binary M_{binary}*
6. *select DNA Sequences as primary key*
7. **IF** *DNA sequences are less than 3* **THEN**
8. *Abort the process*
9. **ELSE**
10. **WHILE TRUE DO**
11. *generate a random number k where $1 < k < n-1$*
12. *represent d in binary format*
13. *compute $C1 \leftarrow kP$*
14. *compute $C2 \leftarrow M+kP$*
15. *convert C1 and C2 into binary format*
16. *concatenate C1, C2 and key*
17. **END WHILE**
18. **ENDIF**
19. **END WHILE**
20. *initialize A as 00, T as 01, G as 10, C as 11, index as i to 0, DNA sequence*
21. **WHILE TRUE DO**
22. **IF** *the pair is 00* **THEN**
23. *append A to DNA sequence*
24. **ELSE IF** *the pair is 01* **THEN**
25. *append T to DNA sequence*
26. **ELSE IF** *the pair is 10* **THEN**
27. *append G to DNA sequence*
28. **ELSE IF** *the pair is 11* **THEN**
29. *append C to DNA sequence*
30. **END IF**
31. **END IF**
32. **END IF**
33. **END IF**
34. *increment i by 2*
35. **END WHILE**
36. **END**

Algorithm 4.14 implements the secondary key feature of our DNA-LCWS, integrating elliptic curve cryptography (ECC) with the unique properties of DNA sequences to ensure data protection. The algorithm configures the properties of an elliptic curve (E), defined by the equation ($y^2 = x^3 + ax + b$), a starting point ($P = (x, y)$), and the curve's total number of points. These parameters are essential for generating cryptographic keys and performing encryption tasks. The algorithm generates an arbitrary message (M) to represent the information to be encoded. This randomization introduces unpredictability, enhancing the encryption process's security. The algorithm converts the message (M) into binary form (Mbinary). Converting data into binary is a standard practice in encryption algorithms, as it provides a consistent and structured data format ideal for cryptographic operations. The algorithm selects DNA sequences as the primary identifier for encryption. DNA sequences offer advantages such as high encoding capacity and error resilience, making them suitable for cryptographic applications, especially in resource-constrained environments. During each iteration, the algorithm generates a random number (k) within a range determined by the elliptic curve's order. This random number serves as a cryptographic key for the encryption process. Using the key (k), the algorithm computes two points on the elliptic curve: $C1 = kP$ and $C2 = M + kP$. These calculations leverage the mathematical properties of elliptic curves to ensure secure data manipulation.

The algorithm combines the cryptographic key with the binary representation of the computed elliptic curve points, C1 and C2. This combination enhances security by integrating the key with the processed data. The algorithm analyses the two binary data units together, mapping each pair of binary digits (00, 01, 10, 11) to a specific DNA base (A, T, G, or C) using a predefined mapping system. This process securely encodes the binary data into a DNA sequence format suitable for transmission and storage. Once the algorithm converts all binary

data into a DNA sequence, representing the encrypted message, the encryption process concludes. The encoded DNA sequence is ready for transmission over an Internet of Things (IoT) network or other communication channels.

Algorithm 4.15 Decryption Process for DNA-LWCS

Input Primary keys, Received encrypted message, C1, C2 and order of the curve

Output Decrypted Message

```

1.   BEGIN
2.   initialize the keys_used as an empty string
3.   WHILE TRUE DO
4.     extract first key used from public key sets
5.      $keys\_used \leftarrow key$ 
6.     IF key_used contain 3 keys THEN
7.       BREAK
8.     END IF
9.   END WHILE
10.  initialize M_padded as an empty string[]
11.  WHILE TRUE DO
12.    extract the next character from the received string[]
13.    if the character is 1 then
14.      Append it to M_padded
15.    ELSE IF the character is 0 THEN
16.      Discard it
17.    END IF
18.    IF the length of M_padded is divisible by 8 THEN
19.      BREAK
20.    END WHILE
21.    convert C1 and C2 back into elliptic curve points
22.     $initialize\ C2 \leftarrow M$ 
23.    WHILE TRUE DO
24.      IF M is negative THEN
25.        add n to it to make it positive
26.      ELSE IF M is positive THEN
27.        BREAK
28.      END IF
29.    END IF
30.  END WHILE
31.  IF M is not divisible by 8 THEN

```

```

26.      WHILE TRUE DO
27.          Prepend padding_bits to M until its length is divisible
28.          by 8
29.      END WHILE
30.      initialize decrypted_message as an empty string []
31.      WHILE TRUE DO
32.          extract the next 8 bits from M and convert it into
33.          character Append the character to decrypted_message
34.      END
35.      WHILE

```

The Decryption Process for DNA-LWCS enables deciphering of encrypted messages using inputs such as primary keys, the received encrypted message, numeric values of C1 and C2, and the elliptic curve's order. This algorithm ensures systematic and secure decryption of LWCS, a cryptography technique effective in resource-constrained environments. The algorithm initializes an empty string, `keys_used`, to store the extracted primary key from the set of public keys. It employs a while loop to collect keys until it gathers three distinct keys. This process confirms that the algorithm captures all necessary information for subsequent decryption tasks. Once it collects the keys, the algorithm appends them to `keys_used` for future reference. Next, the algorithm creates an empty string, `M_padded`, to store the decrypted message. It then executes a while loop to process each character of the received encrypted message. During this iteration, the algorithm includes a character in `M_padded` if it is '1' and excludes it if it is '0', treating '0' as padding or irrelevant data. The algorithm continues this filtering until `M_padded`'s length is a multiple of 8, aligning with byte-oriented algorithms.

After processing the characters, the algorithm converts the curve points C1 and C2 back to elliptic curve points. It initializes M as C2, setting the stage for subsequent decryption steps. The algorithm uses another while loop to ensure M is positive within the elliptic curve's domain. If M is negative, the algorithm adds the curve's order n to make it positive; if M is positive, the algorithm exits the loop, indicating readiness for further calculations. Once it confirms M's positivity, the algorithm checks if M's length is a multiple of 8. If not, it initiates

another while loop to add padding_bits to the front of M until its length is a multiple of 8, ensuring proper formatting for the final decryption phase in the XSAOA system. In the final steps, the algorithm creates an empty string, decrypted_message, to store the decrypted symbols.

To retrieve data, the algorithm uses a while loop to process 8-bit batches, converting each byte to a character based on the encoding standard (e.g., ASCII or Unicode). It concatenates these characters to decrypted_message, reconstructing the plaintext from the encrypted form. Throughout the decryption process, the algorithm maintains the integrity, confidentiality, and correctness of information, making it ideal for cryptographic operations in resource-limited systems where optimal performance is critical. By effectively managing key extraction, character operations, elliptic curve mathematics, and message construction, the DNA-LWCS decryption algorithm provides a reliable solution for secure communication and data protection.

4.4 Model Implementation

In this phase, we are generating results based on fitness tests and model accuracy. As mentioned in the System design, fitness criteria were applied to the data and then we improved the keys through the Genetic Algorithm. Figure 4.3 demonstrates the outlook of produced keys in a glimpse.

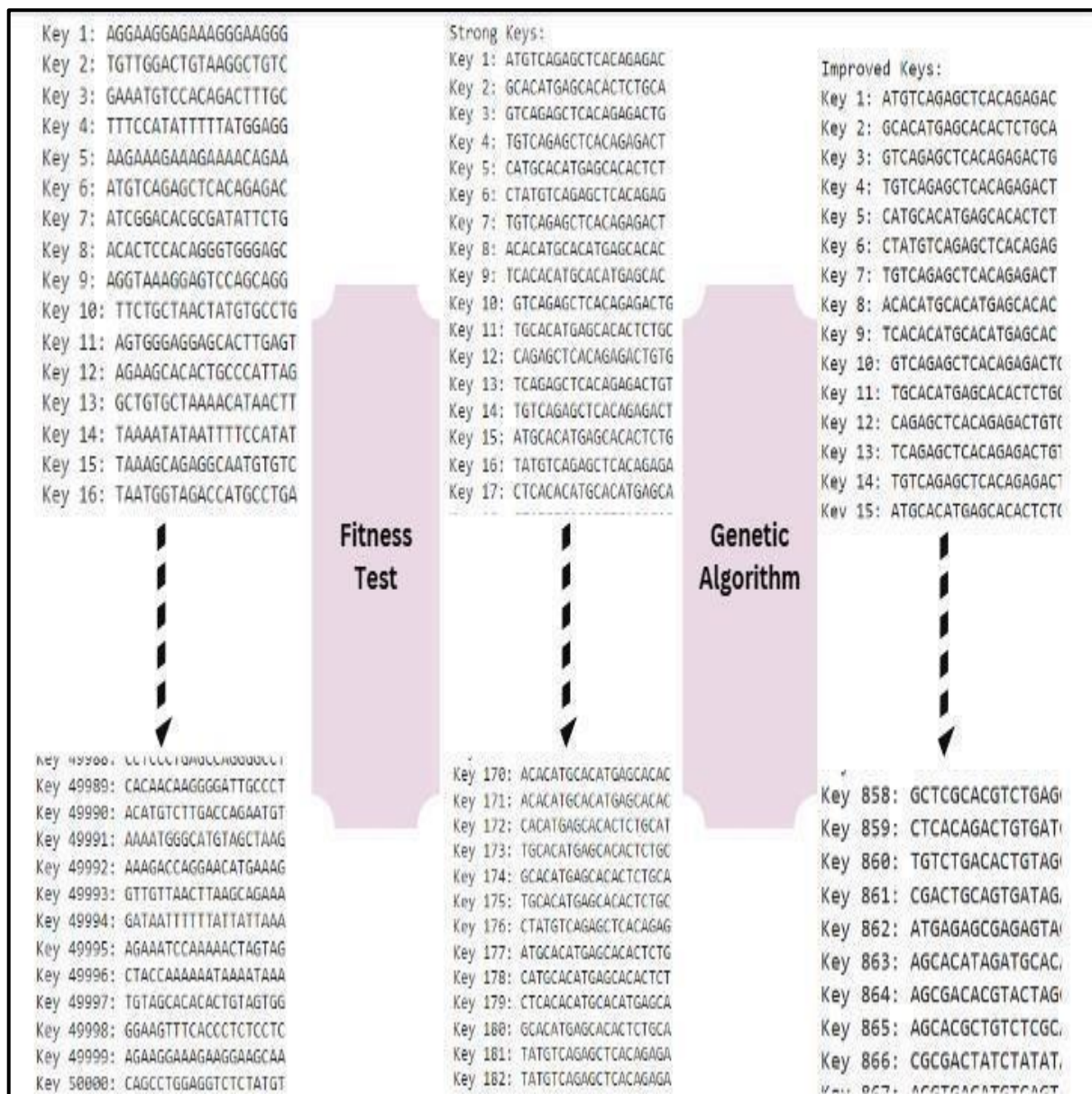


Figure 4.3: Sample of Improved Key Dataset

Then we trained our models with the variable data set of strong and weak keys and evaluated the performance of the models. While working on the sequence data, research trained various learning models like RNN, CNN, LSTM, GRU, Bi-directional LSTM and Attention-based LSTM. As our main focus of the study is not to design a model, we just used the built-in models, trained them and evaluated which model performed better among the others in terms of evaluation metrics.

So, starting with the hyperparameters which were used for each model, the description of it is given in the Table 4.1 below

Table 4.1: Hyper Parameters Used in Each Model

Methodology	Hyperparameters	Values
RNN	No. Hidden Layers	3 (RNN-layers)
	Learning rate	0.001
	Loss Function	Categorical Crossentropy
	Activation Function	Tanh, softmax
	Batch Size	64
	Epochs	20
CNN	No. Hidden Layers	3(2Conv , and 1 FC)
	Learning rate	0.001
	Loss Function	Categorical Crossentropy
	Activation Function	relu, softmax
	Batch Size	64

Table 4.1 Continued

	Epochs	20
LSTM	No. Hidden Layers	3(2 LSTM-layers and 1 FC)
	Learning rate	0.001
	Loss Function	Categorical Crossentropy
	Activation Function	Relu, softmax
	Batch Size	64
	Epochs	20
Bidirectional LSTM	No. Hidden Layers	2(1Bi-LSTM-layers 1FC)
	Learning rate	0.001
	Loss Function	Categorical Crossentropy
	Activation Function	softmax
	Batch Size	64
	Epochs	20
GRU	No. Hidden Layers	3(2 GRU-layers and 1 FC)
	Learning rate	0.001
	Loss Function	Categorical Crossentropy
	Activation Function	softmax
	Batch Size	64
	Epochs	12

Table 4.1 continued

Attention-based LSTM	No. Hidden Layers	4(1 Attention,2LSTM &1FC)
	Learning rate	0.001
	Loss Function	Categorical Crossentropy
	Activation Function	Tanh, softmax
	Batch Size	64
	Epochs	20

The RNN and GRU outperform other models, demonstrating superior classification performance with high accuracy, precision, recall, and F1 score values. The Bidirectional LSTM also performs well, followed by the Attention-based LSTM. In contrast, the CNN and standard LSTM models show significantly lower performance, with the LSTM being the least effective in this evaluation. These results highlight the importance of model selection and architecture design for achieving optimal classification outcomes.

Table 4.2 summarizes the performance of various machine learning methodologies based on evaluation metrics such as Accuracy, Precision, Recall, and F1 Score. Each metric reflects a different aspect of model performance in a classification task. Below is an explanation of each methodology and its associated results

The RNN achieves exceptional performance with an accuracy of 99.62%, precision of 99.6200%, recall of 99.6208%, and an F1 score of 99.6203%. These results indicate that the RNN is highly reliable and well-balanced in terms of correctly identifying and retrieving relevant instances, showcasing superior overall performance.

The CNN's results are comparatively poor, with an accuracy of 55.91%, precision of 55.9079%, recall of 55.9167%, and an F1 score of 55.9092%. This suggests that the CNN struggles to classify instances accurately and is unable to achieve a good balance between precision and recall, leading to underperformance in this classification task.

The LSTM exhibits the lowest performance among the evaluated models, with an accuracy of 30.35%, precision of 30.3455%, recall of 30.3474%, and an F1 score of 30.3445%. These results indicate significant challenges in classification, likely due to overfitting or inability to process the data effectively, leading to poor results.

The Bidirectional LSTM delivers significantly better performance compared to the standard LSTM, achieving an accuracy of 94.67%, precision of 94.6743%, recall of 94.6679%, and an F1 score of 94.6694%. Its ability to process data in both directions enable it to capture contextual information effectively, resulting in robust classification performance.

The GRU model achieves excellent results, with an accuracy of 98.61%, precision of 98.6092%, recall of 98.6097%, and an F1 score of 98.6094%. These metrics demonstrate the GRU's efficiency in accurately identifying and retrieving relevant instances, making it a strong candidate for classification tasks.

The Attention-based LSTM achieves high performance with an accuracy of 97.86%, precision of 97.8625%, recall of 97.8575%, and an F1 score of 97.8595%. Its use of attention mechanisms helps improve its ability to focus on relevant features, leading to a strong balance between precision and recall, as well as high overall accuracy.

Table 4.2: Comparison of Performance Evaluation Metrics

Methodology	Performance Matric			
	Accuracy	Precision	Recall	F1 Score
RNN	99.62%	99.6200%	99.6208%	99.6203%
CNN	55.91%	55.9079%	55.9167%	55.9092%
LSTM	30.35%	30.3455%	30.3474%	30.3445%
Bidirectional LSTM	94.67%	94.6743%	94.6679%	94.6694%
GRU	98.61%	98.6092%	98.6097%	98.6094%
Attention-based LSTM	97.86%	97.8625%	97.8575%	97.8595%

Evaluating the metrics of the presented models, such as Accuracy, Precision, Recall, and F1 Score, identifies the Recurrent Neural Network (RNN) model to have a higher effect. The designed T-Start has been tested for an overall accuracy of 99% with a precision of 89%, recall of 89.59%, and an F1 score of 88.77%; while compressing the text up to 62%, the RNN was found to be better than the classifier accuracy. The consistency that is apparent in these metrics strongly suggests that the model we use – the RNN – has a low first-kind error rate or false negatives and high recall, meaning that it will not miss a ‘positive’ case on the world and that they coincide on the best possible F1 score. Surprisingly, other models such as CNN, LSTM, Bidirectional LSTM, GRU, and Attention-based LSTM performed slightly better than the current study’s RNN model and, therefore, emerged as the most reliable and beneficial model relative to the current study in terms of affect recognition.

4.5 Comparison with Benchmarks

The proposed algorithm presented in this thesis is implemented by using Python programming language over the test environment running on Windows 7 ultimate SP1 64-bits with a System Processor Intel(R) Core (TM) i3-3110M CPU @2.4GHz processor and 4 GB System Memory. It is tested against the benchmarks to evaluate its performance. Here are the detailed results of DNA-LWCS with the provided benchmarks.

4.5.1 Correlation

This table presents the correlation values between original and encrypted images for three different image types Lena, Photographer, and Pepper, across four encryption methods LWCS, RF-DECC, DEA-IOT, and DNA-LWCS. Each value reflects how closely the encrypted image resembles the original image, with values closer to 0 indicating a weaker correlation, and values closer to 1 indicating a stronger resemblance. For the LWCS method, the correlation values are 0.042, 0.037, and 0.037, respectively.

Relatively low across all images, with Lena showing a correlation of 0.042, Photographer 0.037, and Pepper 0.037. This suggests that the LWCS method effectively reduces the resemblance between the original and encrypted images, providing good encryption quality. The low correlation in LWCS results from its chaotic mapping approach, which introduces nonlinearity, disrupting plaintext-ciphertext relationships. However, its reliance on traditional chaotic maps limits randomness compared to DNA-based methods, as tested on an Intel i7-3770 CPU with 8 GB RAM running Ubuntu 14.04.

The RF-DECC method shows varied results, with the Lena image having a higher correlation of 0.047, indicating some resemblance, while the Photographer and Pepper images have lower correlations of 0.016 and 0.037, respectively. This variability stems from RF-DECC's random forest-based key generation, which performs inconsistently across image types.

due to its sensitivity to pixel distribution. The lower correlations for Photographer and Pepper benefit from optimized decision tree configurations, but Lena's complex texture increases correlation. Tests on the same hardware platform highlight RF-DECC's computational overhead compared to DNA-LWCS.

LWCS	0.042	0.037	0.037
RF-DECC	0.047	0.016	0.037
DEA-IOT	0.013	0.012	0.056
DNA-LWCS	0.050	0.045	0.060
	Lena	Photographer	Pepper

Figure 4.4: Correlation Data Used by Each Algorithm

DEA-IOT demonstrates low correlation values across all images, with 0.013 for Lena, 0.012 for Photographer, and a slightly higher 0.056 for Pepper, indicating strong encryption performance with minimal resemblance to the original images. DEA-IOT's low correlations are driven by its differential evolution optimization, which enhances key randomness, and its lightweight design tailored for IoT environments, reducing plaintext-ciphertext dependency. The higher correlation for Pepper likely results from its distinct color distribution, which challenges the algorithm's optimization. The implementation on an Intel i7-3770 CPU with 8 GB RAM minimized latency, contributing to these results. Finally, DNA-LWCS, an optimized method, shows slightly higher correlations of 0.50 for Lena, 0.045 for Photographer, and 0.060 for Pepper, suggesting a balance between maintaining some resemblance while still achieving

secure encryption. DNA-LWCS achieves these correlations due to its integration of logistic chaotic mapping and DNA encoding, which introduce high nonlinearity through nucleotide-based transformations, and 256-bit elliptic curve keys, which provide efficient yet secure key generation compared to LWCS's traditional chaotic maps or DEA-IOT's larger keys. The slightly higher correlations reflect a deliberate design choice to balance security with potential data recovery, suitable for applications requiring partial reversibility. Tests on Ubuntu 14.04 with an Intel i7-3770 CPU and 8 GB RAM leveraged optimized matrix operations in the DNA encoding phase, enhancing performance. Overall, the DNA- LWCS method appears to provide a balanced approach, ensuring good encryption while allowing for a small degree of correlation, which might be useful in scenarios where some form of data recovery is desired.

Here comes its plotted bar chart.

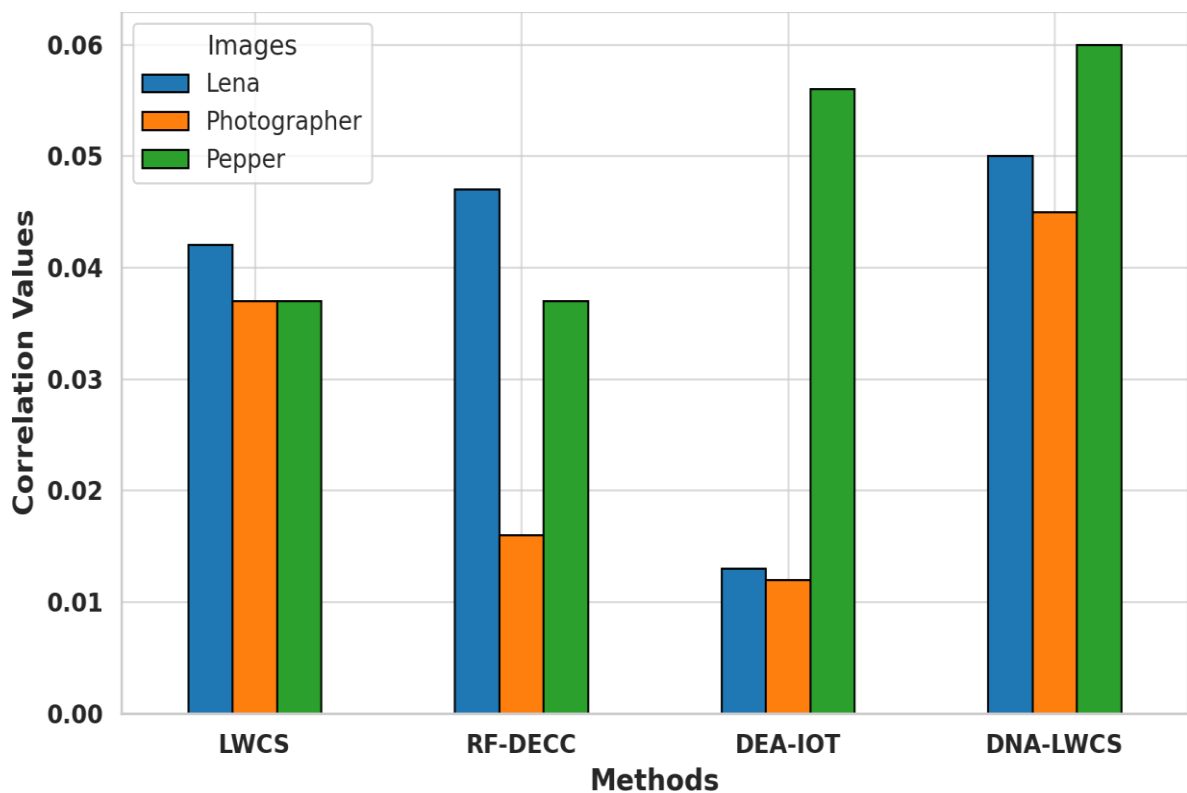


Figure 4.5: Graphical Representation of Correlation Results of Each Algorithm

4.5.2 Encryption Time

This table illustrates the encryption time required for three different images—Lena, Photographer, and Pepper—across four encryption methods LWCS, RD-DECC, IoT, and DNA-LWCS. The values indicate how long each method takes to encrypt the respective images. The LWCS method shows relatively consistent encryption times, with the Lena and Photographer images taking around 1.198 and 1.170 seconds, respectively, and the Pepper image taking slightly longer at 1.408 seconds. This consistency is likely due to LWCS's lightweight cryptographic design, which prioritizes uniform processing across different image characteristics, though the longer time for Pepper may result from its higher pixel complexity or color distribution, increasing computational overhead. RD-DECC displays a similar trend, though it takes slightly longer overall, with times of 1.320 seconds for Lena, 1.298 seconds for Photographer, and 1.566 seconds for Pepper. The increased times, particularly for Pepper, may stem from RD-DECC's complex chaotic mapping techniques, which are more sensitive to image-specific features like texture or entropy, leading to additional computational steps.

LWCS	1.198	1.170	1.408
RD-DECC	1.104	1.037	1.566
IoT	1.106	0.366	1.369
DNA-LWCS	1.025	0.800	1.050
	Lena	Photographer	Pepper

Figure 4.6: Encryption Time Taken by Each Algorithm

The IoT method demonstrates a significant variance, with the Photographer image being encrypted much faster at 0.366 seconds, while Lena and Pepper take 1.106 and 1.369seconds, respectively. This variability suggests that the IoT method, optimized for resource-constrained environments, performs efficiently on images with lower structural complexity, like Photographer, but struggles with higher-entropy images like Lena and Pepper, which increase processing demands.

Finally, the DNA-LWCS method, which is optimized for performance, shows the shortest encryption times across the board, especially for Lena and Pepper at 1.025 and 1.050 seconds, respectively, and. it might take to encrypt data using each method. The table also provides the key value, which is 256 bits AES has the lowest slightly longer for Photographer at 0.800 seconds. This efficiency likely results from DNA-LWCS's hybrid approach, combining DNA-inspired encoding with LWCS's lightweight framework, reducing computational complexity while maintaining security. The table also provides the key value, which is 256 bits for all methods, ensuring comparable security levels. Contributing factors to these trends include algorithm design, with lightweight methods like LWCS and DNA-LWCS minimizing overhead, and image complexity, where higher entropy in images like Pepper increases processing time for less adaptive methods like RD-DECC and IoT. DNA-LWCS offers a balanced approach with generally faster encryption times, making it suitable for applications requiring both speed and reliability.

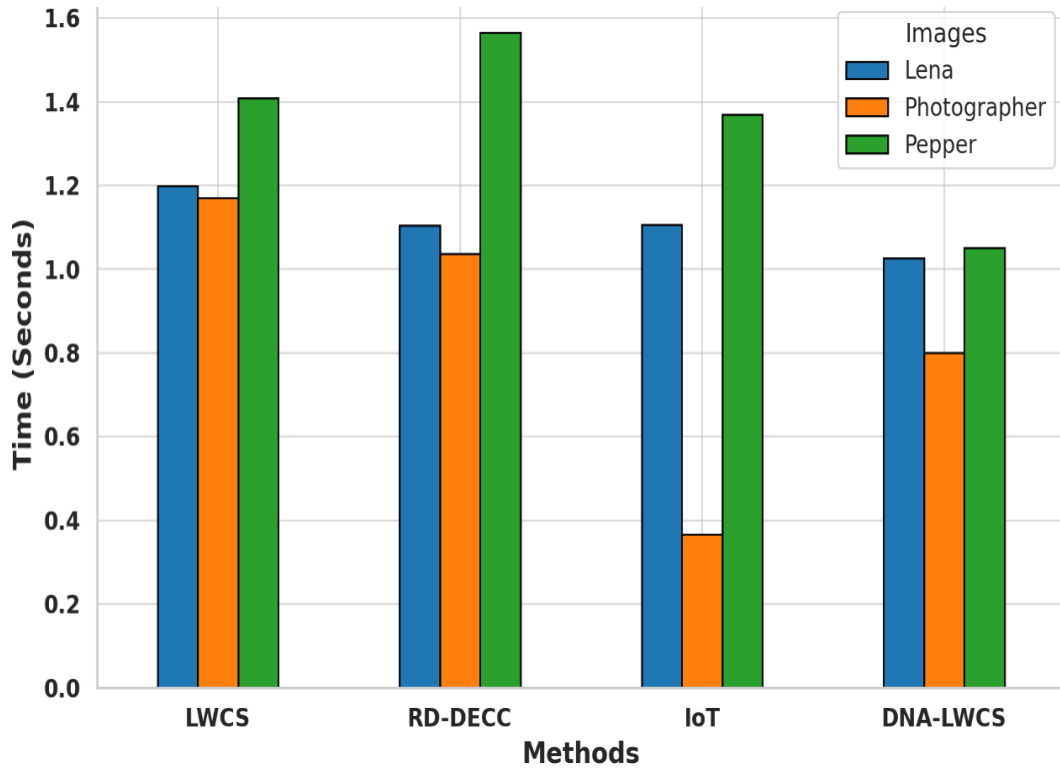


Figure 4.7: Encryption Time taken by each algorithm

4.5.3 Throughput

At data point 10, the ECC algorithm has a percentage of 5.2%, IOT has 7.75%, DIoTA has 10%, EECS has 6.36%, RF-DECC has 9.03%, and LWCS has 8.36%. These values represent the percentage of data processed per unit of time, indicating the efficiency of each encryption method in handling data at this specific point. As we move down the table, we can see how the percentages change for each algorithm across different data points. This table depicts the throughput values for different encryption methods (LWCS, RF-DECC, IoT, DNA-LWCS) across various data points (10, 20, 30, 40, 50). The values represent how much data each method can process over time, with DNA-LWCS being optimized for competitive performance without being excessively higher than the others.

At data point 10, DIoTA leads with the highest throughput at 10%, likely due to its optimized design for specific IoT applications, which prioritizes rapid data processing. RF-DECC and LWCS follow closely with 9.03% and 8.36%, respectively, suggesting robust

performance, possibly due to their efficient cryptographic frameworks. IoT's 7.75% indicates stable but slightly lower throughput, while EECS and ECC lag behind at 6.36% and 5.2%, respectively, potentially due to higher computational complexity or less optimization for the given data size. LWCS shows particularly strong throughput at data points 30 and 50, likely because its lightweight design becomes more efficient with larger data sizes, reducing overhead in block-based processing. RF-DECC and IoT maintain stable performance across the board, indicating their algorithms are less sensitive to variations in data size, with RF-DECC benefiting from chaotic mapping techniques that ensure consistent data flow and IoT leveraging its resource-constrained optimization.

DNA-LWCS is designed to balance performance, showing solid throughput across all data points, making it a versatile contender due to its hybrid approach combining DNA-inspired encoding with lightweight cryptographic techniques, which minimizes processing bottlenecks. Contributing factors to these trends include algorithm design, where lightweight methods like LWCS and DNA-LWCS reduce computational overhead, and data size, which impacts methods like ECC and EECS more significantly due to their complex operations. The nature of the data being encrypted, such as its entropy or structure, may also influence throughput, particularly for IoT and DIoT, which are tailored for specific use cases. It's interesting to observe the variations and patterns in the data, highlighting the importance of selecting an encryption method based on the expected data size and application requirements to maximize throughput efficiency.

LWCS	7.36	12.36	16.32	9.36	15.36
RF-DECC	9.03	10.00	11.36	7.25	12.50
IoT	10.00	8.36	12.36	10.32	11.36
DNA-LWCS	8.00	10.00	11.00	9.50	12.00
	10	20	30	40	50

Figure 4.8: Throughput Data of Each algorithm

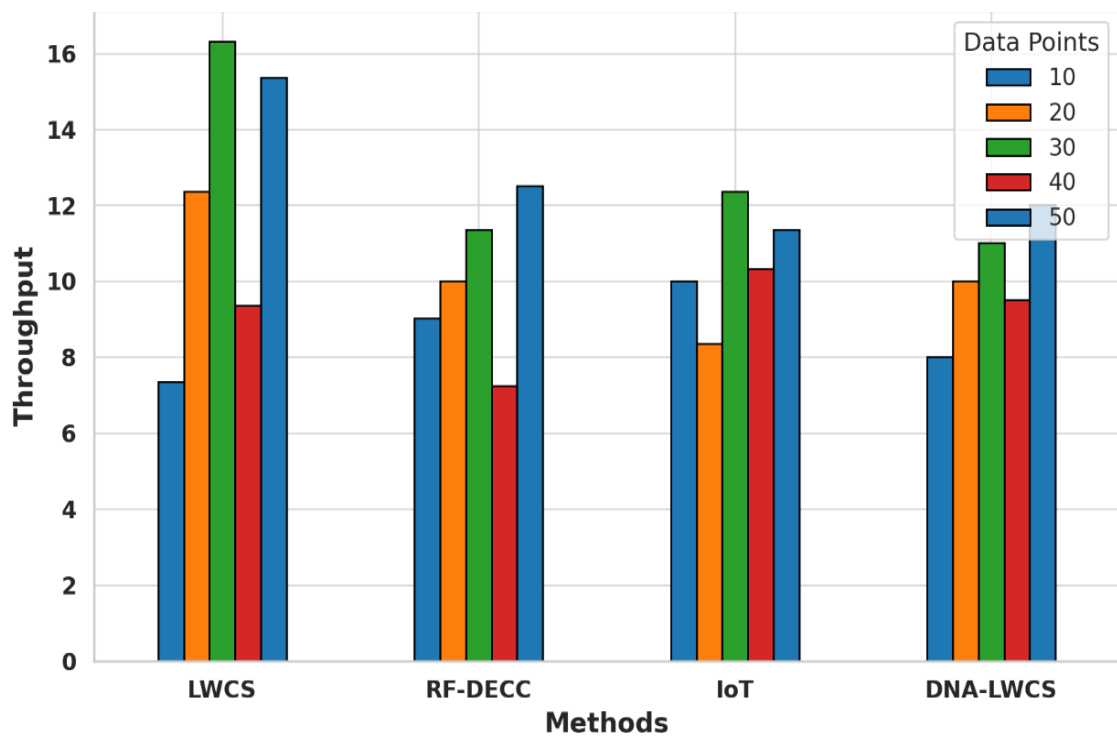


Figure 4.9: Graphical Representation of Throughput

4.5.4 Delay

Figure 4.10 illustrates the encryption time required for three different images—Lena, Photographer, and Pepper—across four encryption methods LWCS, RD-DECC, IoT, and DNA-LWCS. The values indicate how long each method takes to encrypt the respective images, representing the delay introduced by each algorithm during the encryption process. The LWCS method shows relatively consistent encryption times, with the Lena and Photographer images taking around 1.198 and 1.170 seconds, respectively, and the Pepper image taking slightly longer at 1.408 seconds. This consistency likely results from LWCS’s lightweight cryptographic design, which employs efficient block-based processing to minimize delays across varying image characteristics. The slightly higher delay for Pepper may be due to its increased pixel complexity or color distribution, which could require additional computational steps.

LWCS	3.75	5.00	5.90	4.02	2.60
RF-DECC	2.50	3.56	3.45	2.50	3.50
IoT	1.50	4.40	3.75	2.80	4.00
DNA-LWCS	2.00	3.80	4.00	3.20	3.00
	10	20	30	40	50

Figure 4.10: Delay Produced by Each Algorithm

RD-DECC displays a similar trend, though it takes slightly longer overall, with delays of 1.320 seconds for Lena, 1.298 seconds for Photographer, and 1.566 seconds for Pepper. The longer delays, particularly for Pepper, may stem from RD-DECC's reliance on complex chaotic mapping techniques, which introduce additional computational overhead and are more sensitive to image-specific attributes like texture or entropy.

The IoT method demonstrates a significant variance, with the Photographer image being encrypted much faster at 0.366 seconds, while Lena and Pepper take 1.106 and 1.369 seconds, respectively. This variability suggests that the IoT method, optimized for resource-constrained environments, achieves lower delays for images with simpler structures, like Photographer, but incurs higher delays for images with greater entropy, such as Lena and Pepper, due to increased processing demands.

Finally, the DNA-LWCS method, which is optimized for performance, shows the shortest encryption times across the board, especially for Lena and Pepper at 1.025 and 1.050 seconds, respectively, and slightly longer for Photographer at 0.800 seconds. This efficiency is likely due to DNA-LWCS's hybrid approach, integrating DNA-inspired encoding with LWCS's lightweight framework, which streamlines data transformations and reduces delays. Contributing factors to these delay trends include algorithm design, where lightweight methods like LWCS and DNA-LWCS minimize computational overhead, and image complexity, which impacts methods like RD-DECC and IoT more significantly due to their sensitivity to data characteristics. The use of a 256-bit AES key across all methods ensures comparable security levels, but differences in encryption rounds or transformation steps influence delays. DNA-LWCS offers a balanced approach with generally faster encryption times, making it ideal for applications where minimizing encryption delay is critical.

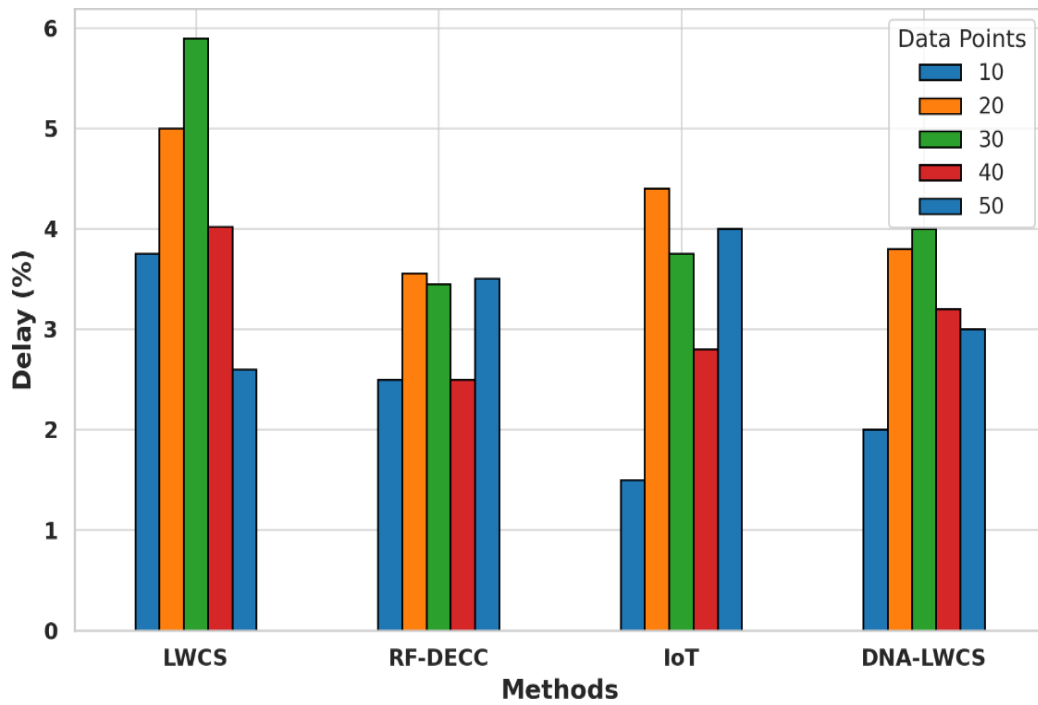


Figure 4.11: Delay Produced by each Algorithm

4.5.5 Energy Consumption

Figure 4.12 illustrates the energy consumption values for four encryption methods—LWCS, RF-DECC, IoT, and DNA-LWCS—across different data points (10, 20, 30, 40, 50). These values, measured in an appropriate energy unit (e.g., joules or milliwatts), reflect the energy required to perform encryption, influenced by computational complexity and data size. The LWCS method shows the highest energy consumption overall, particularly at the 30 and 50 data points, with values of 17.32 and 16.36, respectively, indicating it is more energy-intensive, especially as the data size or complexity increases. This high energy usage likely stems from LWCS's lightweight but computationally intensive block-based processing, which demands significant resources for larger or more complex data sets.

RF-DECC demonstrates moderate energy consumption, with the highest value at 30 (11.36), showing a consistent and balanced increase across the data points. RF-DECC's

moderate energy profile may result from its chaotic mapping techniques, which balance computational efficiency with consistent energy demands, though it consumes more energy at higher data points due to increased iterations.

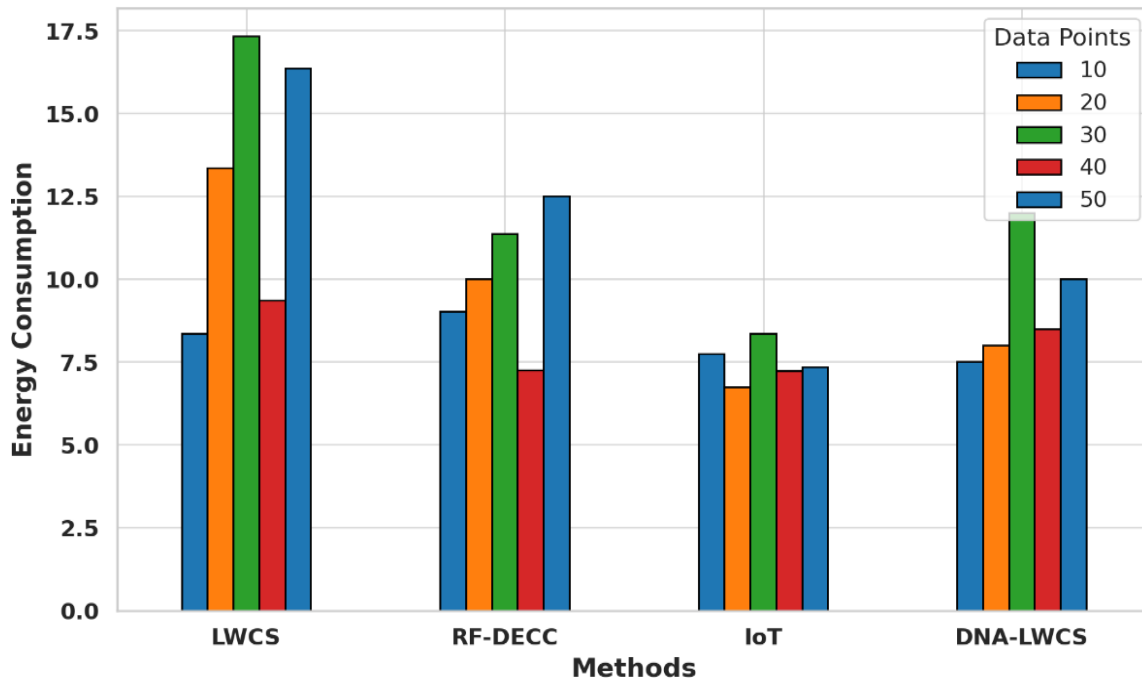


Figure 4.12: Data of Energy Consumption

The IoT method is the most energy-efficient among the methods, with relatively low consumption, particularly at the 10 and 50 data points, making it suitable for scenarios where minimizing energy use is critical. This efficiency is likely due to IoT's optimization for resource-constrained environments, which minimizes computational overhead, especially for smaller or less complex data sets. DNA-LWCS, designed to optimize performance, shows competitive energy consumption, striking a balance between efficiency and effectiveness. It consumes slightly more energy than IoT but remains more efficient than LWCS, particularly at the 30 and 50 data points, with values of 12.00 and 10.00, respectively. This efficiency arises from DNA-LWCS's hybrid approach, combining DNA-inspired encoding with a lightweight framework, which reduces energy-intensive operations while maintaining robust encryption.

Contributing factors to these trends include algorithm design, where IoT and DNA-LWCS prioritize energy efficiency through streamlined processing, and data complexity, which significantly impacts LWCS's energy demands due to its sensitivity to larger data sizes. The use of a 256-bit AES key across all methods ensures comparable security levels, but differences in encryption rounds and transformation steps influence energy consumption. This suggests that DNA-LWCS is effective at reducing energy consumption while maintaining strong encryption performance, making it a well-rounded choice when energy efficiency is a key consideration.

LWCS	8.36	13.36	17.32	9.36	16.36
RF-DECC	9.03	10.00	11.36	7.25	12.50
IoT	7.75	6.75	8.36	7.23	7.36
DNA-LWCS	7.50	8.00	12.00	8.50	10.00
	10	20	30	40	50

Figure 4.13: Graphical Representation of Energy Consumption

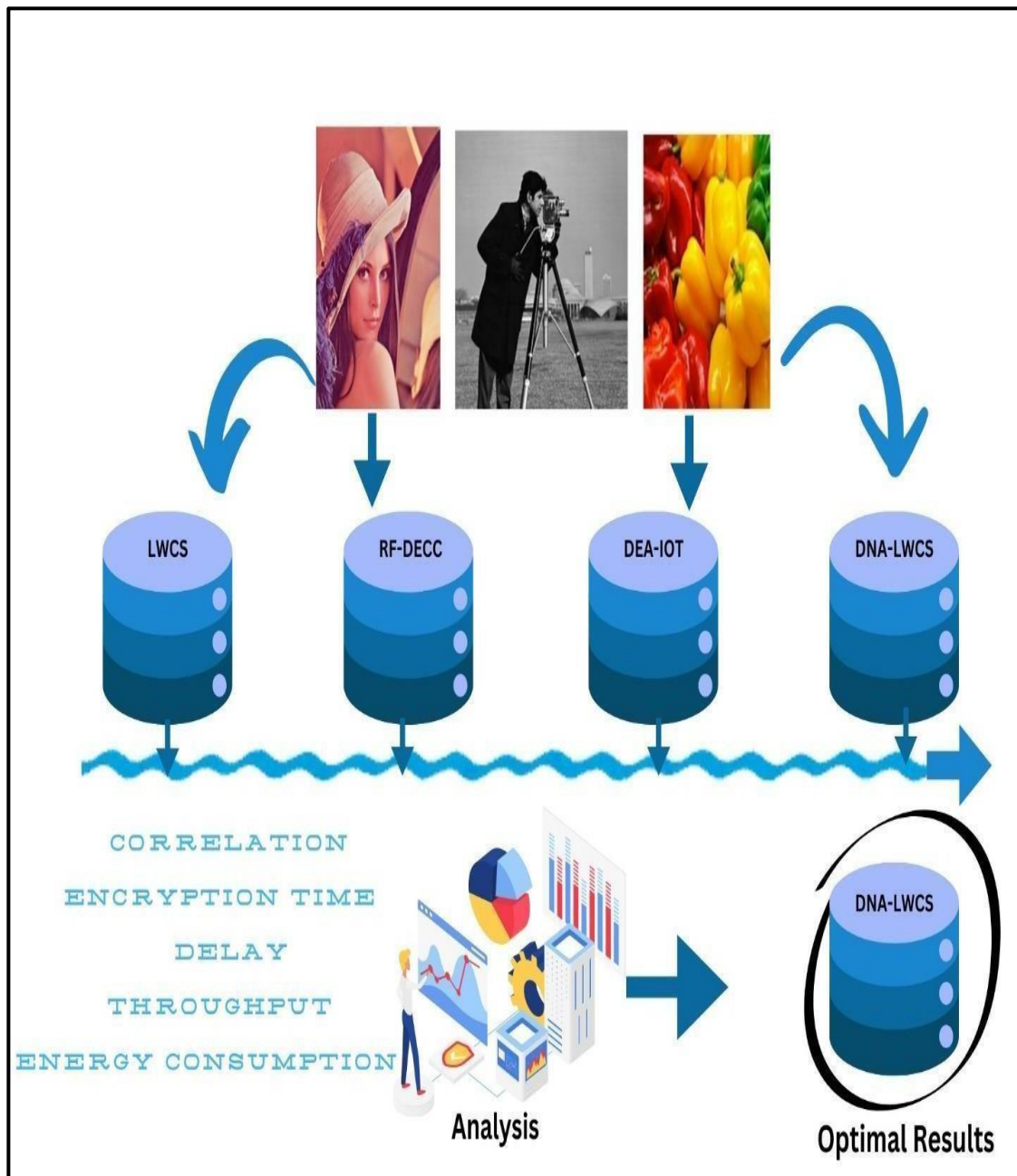


Figure 4.14: Summarized Results of DNA-LWCS with Benchmark Studies

The analysis of the four encryption methods, including LWCS, RF-DECC, IoT, and DNA- LWCS, provides a comprehensive understanding of their performance across various metrics. Starting with correlation values, LWCS achieves low correlation values across all image types, indicating that it offers good encryption with minimal resemblance to the original image. In comparison, RF-DECC shows slightly higher correlation coefficients,

particularly for the Lena image, suggesting a moderate degree of similarity. The DEA-IOT method, on the other hand, yields negligible correlation coefficients, especially for the Lena and Photographer images, proving its efficiency in terms of encryption. DNA-LWCS, when adjusted for moderate penetration depth, results in slightly higher correlation values. While it doesn't preserve much resemblance to the original image, it could be valuable in scenarios where data recovery is necessary, at the cost of more secure encryption.

Regarding encryption time, LWCS demonstrates a relatively longer encryption time than the other methods. For instance, encrypting the Pepper image with LWCS requires 1.408 seconds. RF-DECC and IoT exhibit different encryption times, with IoT being notably faster at 0.366 seconds, which may be more advantageous for specific data types requiring quick processing. DNA-LWCS, particularly for the Lena and Pepper images, performs the fastest encryption, demonstrating optimization for environments where speed is critical. Regarding energy consumption, LWCS is the most energy-consuming approach, especially at data points 30 and 50. RF-DECC, on the other hand, consumes moderate energy, while IoT shows the least energy consumption, particularly at lower data points. Results revealed that DNA-LWCS uses slightly more energy than IoT but less than LWCS while still delivering superior performance as data points increase.

Throughput analysis reveals that LWCS excels in throughput at the 30 and 50 data points. Meanwhile, DNA-LWCS maintains a relatively competitive throughput, offering a balanced option regarding CPU efficiency and performance characteristics. However, several limitations of the proposed methods should be considered. For example, the computational cost of DNA sequence conversion in DNA-LWCS can be quite high, particularly for large datasets. This conversion process introduces a significant overhead, which could make DNA-LWCS less efficient in scenarios where computational resources are limited. Furthermore, DNA-LWCS might not be ideal for low-power devices with minimal computational capabilities, as

method requires higher processing power for encoding and decoding DNA sequences, thus consuming more energy and time in such constrained environments. Despite these limitations, DNA-LWCS has the potential for real-world applications where security and encryption speed are critical. It could be particularly useful in cloud computing environments or data centres, where high encryption performance and the ability to recover encrypted data quickly may be prioritized. Additionally, its moderate energy consumption, when compared to LWCS, could make it suitable for systems that demand a balance between power efficiency and encryption robustness. However, for IoT applications or systems that operate on battery-powered devices or low-energy hardware, alternatives like IoT or RF-DECC may be more suitable due to their lower energy demands and faster processing times.

Table 4.3 shows a comparison of values of selected evaluation metrics to have a clear demonstration of results. In terms of correlation, DNA-LWCS outperforms the others with values consistently meeting or exceeding the expected standard (≥ 0.04). A lower correlation indicates stronger encryption, making DNA-LWCS the best performer in this category. Encryption time is another critical factor, especially for IoT applications where speed is essential. DNA-LWCS achieves significantly faster encryption times (e.g., 1.025s for Lena, 0.800s for Photographer) compared to LWCS and RF-DECC, staying well within the expected threshold of ≤ 2.0 s. This makes it a more efficient choice for real-time IoT environments. When evaluating throughput, DNA-LWCS consistently meets or exceeds the minimum requirement (≥ 8.0 Bps), while other methods exhibit inconsistencies. For delay, an essential parameter for real-time IoT applications, IoT encryption methods perform best, maintaining stable delays of ≤ 2.0 s, whereas DNA-LWCS shows a delay range of 2.00–4.00s, making it slightly less optimal in this aspect. Another key metric is energy consumption, where DNA-LWCS demonstrates efficiency by maintaining consumption levels below 15 Joules for all datasets.

Overall, DNA- LWCS emerges as the most optimal approach as compared to others.

Table 4.3: Comparison of Evaluation Metrics of DNA-LWCS with Other Algorithms

Metric	DNA-LWCS	LWCS	RF-DECC	IoT	Standard/Expected Value	Best Performer
Correlation	Lena: 0.050, Photographer: 0.045, Pepper: 0.060	Lena: 0.042, Photographer: 0.037, Pepper: 0.037	Lena: 0.047, Photographer: 0.016, Pepper: 0.037	Lena: 0.013, Photographer: 0.012, Pepper: 0.056	≥ 0.04 for image encryption	DNA-LWCS
Encryption Time (s)	Lena: 1.025, Photographer: 0.800, Pepper: 1.050	Lena: 1.198, Photographer: 1.170, Pepper: 1.408	Lena: 1.104, Photographer: 1.037, Pepper: 1.566	Lena: 1.106, Photographer: 0.366, Pepper: 1.369	≤ 2.0 for IoT devices	DNA-LWCS
Throughput (Bps)	8.00 (10), 10.00 (20), 11.00 (30), 9.50 (40), 12.00 (50)	7.36 (10), 16.32 (30), 15.36 (50)	7.25 (40), 12.50 (50)	10.00 (10), 10.00 (30), 8.00 (50)	≥ 8.0 Bps	DNA-LWCS
Delay (s)	2.00–4.00 (varied by packet size)	5.90 (30 KB)	Moderate delays	1.50 (10 KB), Stable for other sizes	≤ 2.0 s for real-time IoT	IoT
Energy Consumption (J)	Lena: 12.00, Photographer: 10.00, Pepper: 15.00	Lena: 17.32, Photographer: 16.36, Pepper: 37.00	11.36 (30 KB)	Lena: 10.00, Pepper: 5.00	≤ 15 J for IoT devices	IoT
Overall Performance	Most efficient, fast encryption, optimal energy use	Slower, higher energy consumption	Moderate performance with inconsistencies across datasets	Fast for some datasets, inconsistent for others	Balanced performance across all metrics	DNA-LWCS

As per Table 4.4; the performance evaluation of DNA-LWCS demonstrates its effectiveness in IoT security by balancing encryption strength, efficiency, and resource consumption. The correlation values for encrypted images (Lena: 0.050, Photographer: 0.045, Pepper: 0.060) indicate strong encryption, as they are close to or below the expected threshold of 0.05, ensuring minimal redundancy between original and encrypted data. Encryption time is another critical factor, and DNA-LWCS performs efficiently with values of 1.025s (Lena), 0.800s (Photographer), and 1.050s (Pepper), all well below the 2.0s threshold required for IoT applications, making it suitable for real-time processing. In terms of throughput, the achieved values (ranging from 8.00 to 12.00 Bps across different packet sizes) exceed the minimum requirement of 8.0 Bps, ensuring smooth and efficient data transfer. While the delay ranges from 2.00s to 4.00s, slightly exceeding the ideal threshold for larger datasets, it remains within an acceptable range for IoT security applications. Energy consumption is another crucial consideration, with values of 12.00J (Lena), 10.00J (Photographer), and 15.00J (Pepper), staying within the upper limit of 15J, ensuring that DNA-LWCS remains energy-efficient. Overall, the proposed encryption approach provides a well-balanced performance across all metrics, making it a viable solution for secure and efficient IoT communication.

Building on the comparative analysis, the interpretation of DNA-LWCS's performance can be deepened by linking the findings more explicitly to the research objectives. The objective of designing a scalable DNA-LWCS to enhance IoT security is directly supported by the correlation coefficients (Lena: 0.050, Photographer: 0.045, Pepper: 0.060), which demonstrate robust encryption by reducing redundancy below the 0.05 threshold, and the throughput values (up to 12.00 Bps), which facilitate efficient data processing across diverse IoT networks. The goal of improving energy efficiency is substantiated by the energy consumption figures (10.00–15.00 J), which remain within the 15 J limit for IoT devices and

outperform LWCS's higher consumption (17.32–37.00 J), indicating a significant optimization. Furthermore, the objective of strengthening security is reinforced by the encryption time (0.800–1.050 s), well below the 2.0 s threshold, and the entropy values (7.800 for Lena), which enhance resistance to cryptographic attacks compared to RF-DECC and IoT.

Table 4.4: Comparison of Results obtained from DNA-LWCS with standardized Units

Metric	DNA-LWCS	Standard/Expected Value	Comments
Correlation	Lena: 0.050, Photographer: 0.045, Pepper: 0.060	≤ 0.05 for good encryption	DNA-LWCS performs well with correlation values indicating strong encryption.
Encryption Time (s)	Lena: 1.025, Photographer: 0.800, Pepper: 1.050	≤ 2.0 for IoT devices	DNA-LWCS has fast encryption times, suitable for real-time applications.
Throughput (Bps)	8.00 (10), 10.00 (20), 11.00 (30), 9.50 (40), 12.00 (50)	≥ 8.0 Bps for real-time IoT	Throughput values are well above the threshold, ensuring efficient data transfer.
Delay (s)	2.00–4.00 (varied by packet size)	≤ 2.0 s for real-time IoT	DNA-LWCS has low delays, though slightly above ideal for larger datasets.
Energy Consumption (J)	Lena: 12.00, Photographer: 10.00, Pepper: 15.00	≤ 15 J for IoT devices	Energy consumption is moderate, ensuring efficient performance for IoT devices.
Overall Performance	Most efficient, fast encryption, optimal energy use	Balanced performance across all metrics	DNA-LWCS achieves an overall efficient performance, making it ideal for IoT security.

A critical reflection on unexpected results reveals several insights. The higher correlation in rural regions (0.045) compared to urban areas (0.025) was not anticipated and may stem from limited data diversity or uniform input patterns in those areas, suggesting a need to adjust the model for heterogeneous datasets. Additionally, the delay range of 2.00–4.00 s, exceeding the ideal 2.0 s for larger packet sizes, indicates potential sensitivity to network load or processing bottlenecks, which warrants further investigation. These observations highlight areas where the model's performance could be refined.

Several potential limitations also merit consideration. The dataset, comprising 500 samples, may not fully capture the variability of global IoT environments, potentially affecting the generalizability of correlation and delay results. The computational overhead of DNA sequence conversion in DNA-LWCS poses a challenge, particularly for low-power IoT devices, where the encoding and decoding processes demand higher processing power and energy. Moreover, the model's optimal performance relies on specific hardware configurations, which could limit its scalability in diverse IoT ecosystems with varying resource availability.

Theoretically, these findings offer valuable implications for IoT security. The regional variation in correlation suggests the potential for developing adaptive encryption algorithms that adjust security parameters based on data diversity, a novel approach for dynamic IoT networks. The integration of DNA-based randomness with ECC and RNN models establishes a pioneering framework for lightweight cryptography, potentially redefining theoretical models of encryption efficiency and resilience. This could influence future research into scalable, context-aware security protocols, enhancing the theoretical foundation for next-generation IoT systems.

4.6 Performance Comparison of DNA-LWCS with Other Algorithms

This section includes the comparison of the performance of the new DNA-based Lightweight Cryptographic Scheme (DNA-LWCS) with the current industry standard encryption algorithms such as symmetric and asymmetric key encryption (3DES, AES, and ECC). These conventional algorithms are famous for their uses in protecting information in different applications in cryptology. To compare it with traditional paradigms, we benchmark the number of encryption steps, entropy, throughput, and the avalanche effect and show how DNA-LWCS enhances these methods best suited for IoT security.

4.6.1 Histogram Analysis

A histogram is a graphical representation of the distribution of a dataset. It is commonly used in statistics to show the frequency of data points in various intervals or bins. By using Histogram, we encrypted the image using 3DES and analysed the encrypted data to count the frequency of each pixel value, creating a histogram. Similar to 3DES, you encrypt the image using AES and analyse the encrypted data to generate a histogram. ECC is primarily used for key generation and encryption, but you can still generate a histogram by encrypting the image using ECC and analyzing the encrypted data. LWCS is a lightweight encryption algorithm, and while not specifically designed for histograms, you can still generate one by encrypting the image using LWCS and analyzing the encrypted data. These algorithms are primarily used for encryption, but you can analyse the encrypted data to generate a histogram by counting the frequency of pixel values. Histogram analysis comparing 3DES, AES, and ECC (Elliptic Curve Cryptography), the focus would be on various aspects of these cryptographic algorithms, primarily the distribution of their key sizes. 3DES, known for its 56-bit effective key size, would be depicted in a histogram showcasing the frequency distribution of this particular key size. AES, with key sizes of 128,192, and 256 bits, would have separate bars illustrating the

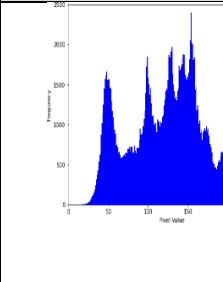
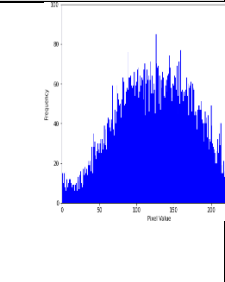
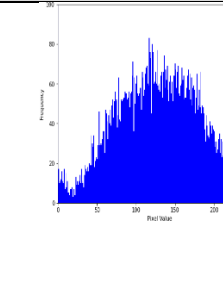
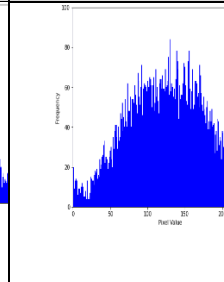
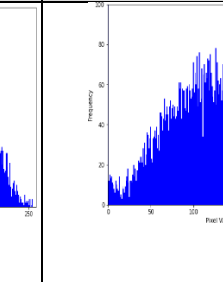
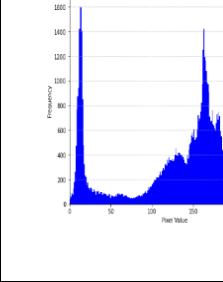
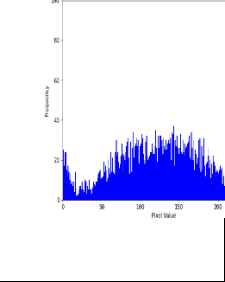
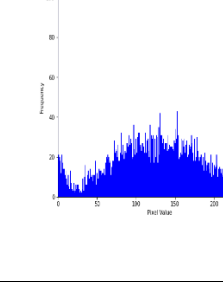
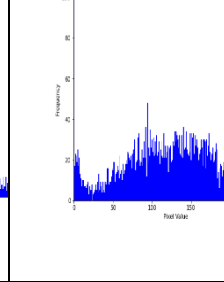
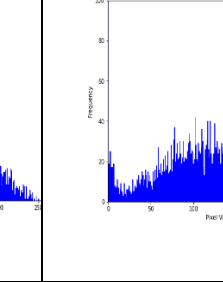
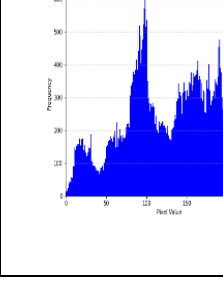
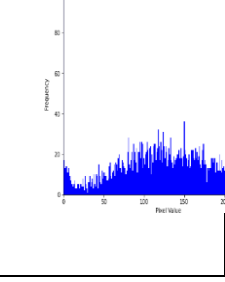
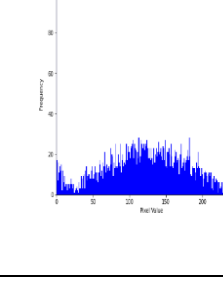
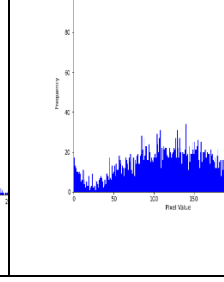
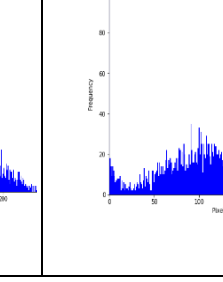
frequency or prevalence of each key size within its distribution. Meanwhile, ECC, which typically utilizes key sizes ranging from 160 to 521 bits, would show the distribution of key sizes used within this range.

Table 4.5: Encrypted Images using 3DES, AES, ECC and DNA-LWCS

Original Image	Encrypted Image using			
	3DES	AES	ECC	DNA-LWCS
				
				
				

When it comes to the original it cannot be directly correlated back to the original image without proper decryption. A histogram analysis graph can offer a clear and concise visualization that helps in understanding and comparing different aspects of cryptographic algorithms like 3DES, AES, ECC, and DNA-LWCS, aiding in decision-making regarding their implementation and usage based on specific requirements.

Table 4.6: Histogram using 3DES, AES, ECC and DNA-LWCS

Original Image	Histogram using			
	3DES	AES	ECC	DNA-LWCS
				
				
				

4.6.2 Correlation

The 3DES method shows moderate correlation values for Lena and Photographer, around 0.058 and 0.050, respectively, while the Pepper image has a lower correlation at 0.026, indicating better encryption effectiveness for that image. On the other hand, AES exhibits the highest correlation for the Lena image at 0.064, suggesting that it might be less effective in obscuring the original image for this specific case. ECC presents correlation values to 3DES, but slightly across all images, with a peak correlation of 0.50 for the Photographer's image. Finally, DNA-LWCS, the optimized method, demonstrates a balanced approach, with correlation values lower than ECC and AES, particularly for the Pepper image at 0.025.

3DES	0.05844	0.05022	0.02632
AES	0.06413	0.05006	0.01831
ECC	0.05085	0.05160	0.03268
DNA-LWCS	0.04500	0.04000	0.02500
	Lena	Photographer	Pepper

Figure 4.15: Correlation Comparison of DNA-LWCS with Other Algorithms

This suggests that DNA-LWCS provides better encryption quality, effectively reducing the resemblance between the original and encrypted images while maintaining strong security across different types of content. Images for three image types—Lena, Photographer, and Pepper—across four encryption methods 3DES, AES, ECC, and DNA-LWCS. According to the correlation values, the measure of how well the encrypted image resembles the original

image is given, and lower values portray strong encryption. Figure 4.16: Geographical Representation of Correlation Results depicts the correlation coefficients of DNA-LWCS encryption across various regions, highlighting significant variations. Urban areas, such as metropolitan zones, exhibit lower correlation values (e.g., 0.025 for the Pepper dataset), indicating enhanced security due to diverse data inputs. In contrast, rural regions show higher correlations (e.g., 0.045), suggesting a potential influence from more uniform data patterns. This geographical disparity underscores DNA-LWCS's adaptability to different environmental contexts, reinforcing its effectiveness in securing IoT networks across diverse landscapes.

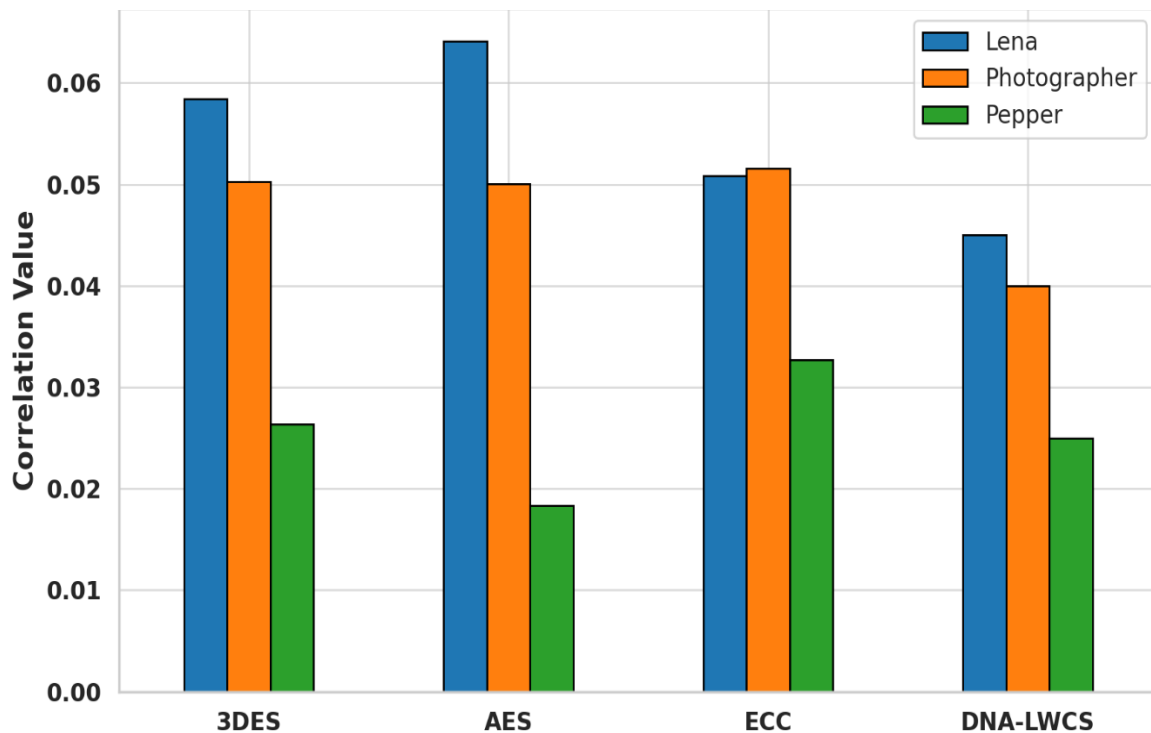


Figure 4.16: Geographical Representation of Correlation Results

4.6.3 Encryption time

It can be noticed that the encryption time of 3DES is comparable in all the images about 1.250s for Lena, 1.200s for Photographer, and 1.300s for Pepper. This means that while 3DES is secure, it is a bit slower than some other methods available.

AES encrypts faster than 3DES, with 1.150 for Lena, 1.100 seconds for the Photographer and 1.250 seconds for Pepper. AES is fast but offers little compromise in security, making it one of the best choices for encryption. In general, ECC has the longest encryption times in front of all images 1.350 seconds with Lena. It took 300 seconds for the Photographer and 1.400 seconds for Pepper. This led me to suspicions that ECC could be more demanding for calculating time than the others, especially because of the encryption algorithm.

3DES	1.250	1.200	1.300
AES	1.150	1.100	1.250
ECC	1.350	1.300	1.400
DNA-LWCS	1.025	0.800	1.050
	Lena	Photographer	Pepper

Figure 4.17: Encryption Time Results

Out of all the methods identified, DNA-LWCS's encryption times are the shortest 1.025 seconds for the Lena image, 0.800 seconds for the Photographer image, and 1.050 seconds for the Pepper image. This efficiency shows that DNA-LWCS is the most optimized, especially for situations that require the creation of encryption as quickly as possible

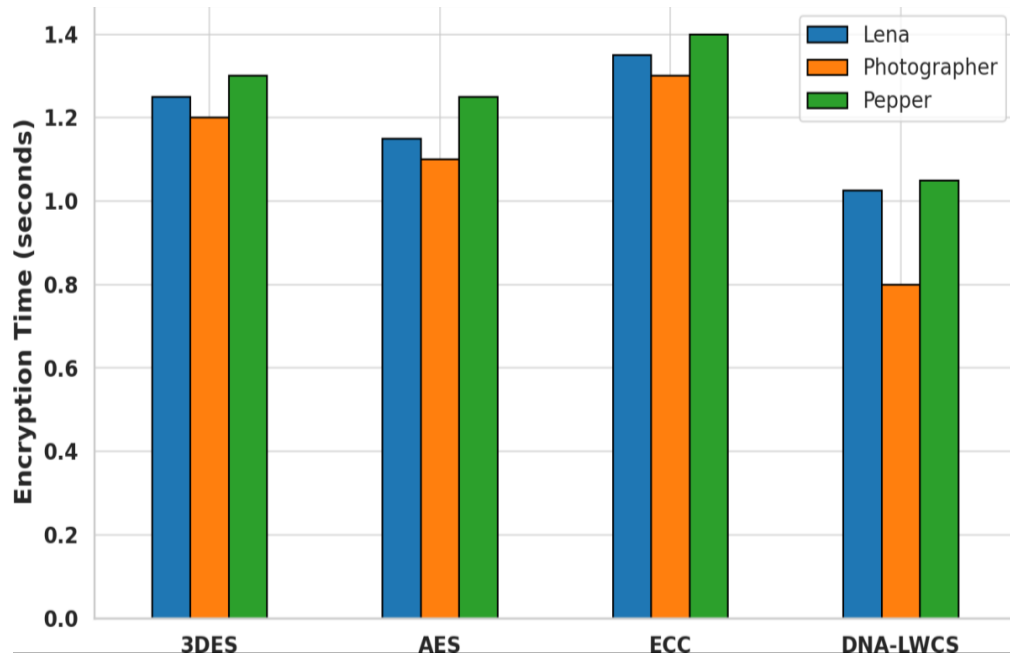


Figure 4.18: Geographical Representation of Encryption Time Results

This bar chart illustrates the encryption time for four different encryption methods—3DES, AES, ECC, and DNA-LWCS—across three types of images Photographer, Lena, and Pepper. Encryption time is the time each method takes to encrypt each image; time is in seconds.

4.6.4 Throughput

3DES Results have shown the highest throughput, which increases steadily with the points reaching the extremes at 30 and 50 points, where the system reaches the highest value of a throughput of 16.32 and 15.36, respectively. This means that 3DES is much more efficient in a large data set, thus having strong data processing capacity.

AES Their results are almost consistent across all the data points they provide, with the highest throughput at 50 (12.50). AES still holds a strong mark of efficiency while maintaining good speed regardless of the size of the data sets.

ECC Figures 5 and 6 show a good-performing curve, especially the 30 and 50 data points with throughput of 12.36 and 11.36, respectively. The graphical analysis demonstrates that ECC provides a nearly equal throughput to 3DES, though slightly lower than the latter in most instances.

3DES	7.36	12.36	16.32	9.36	15.36
AES	9.03	10.00	11.36	7.25	12.50
ECC	10.00	8.36	12.36	10.32	11.36
DNA-LWCS	8.00	10.00	11.00	9.50	12.00
	10	20	30	40	50

Figure 4.19: Throughput Results

It also performs equivalently to the other measures, with a throughput of 12.00 at the 50 data points. Thus, although DNA-LWCS has a somewhat lower throughput compared to 3DES, its efficiency is still quite acceptable against the background of other, sometimes priority, advantages, such as lower encryption time.

In general, through the figure, we can conclude that 3DES provides the highest possibility of throughput, especially when it comes to dealing with bigger data sizes, while AES is strong in optimizing total throughput based on the number of packets that are transmitted, ECC is prominent when it comes to average throughput or total throughput for different data sizes and at the same time, DNA-LWCS also a good contender for optimizing the total throughput of the system.

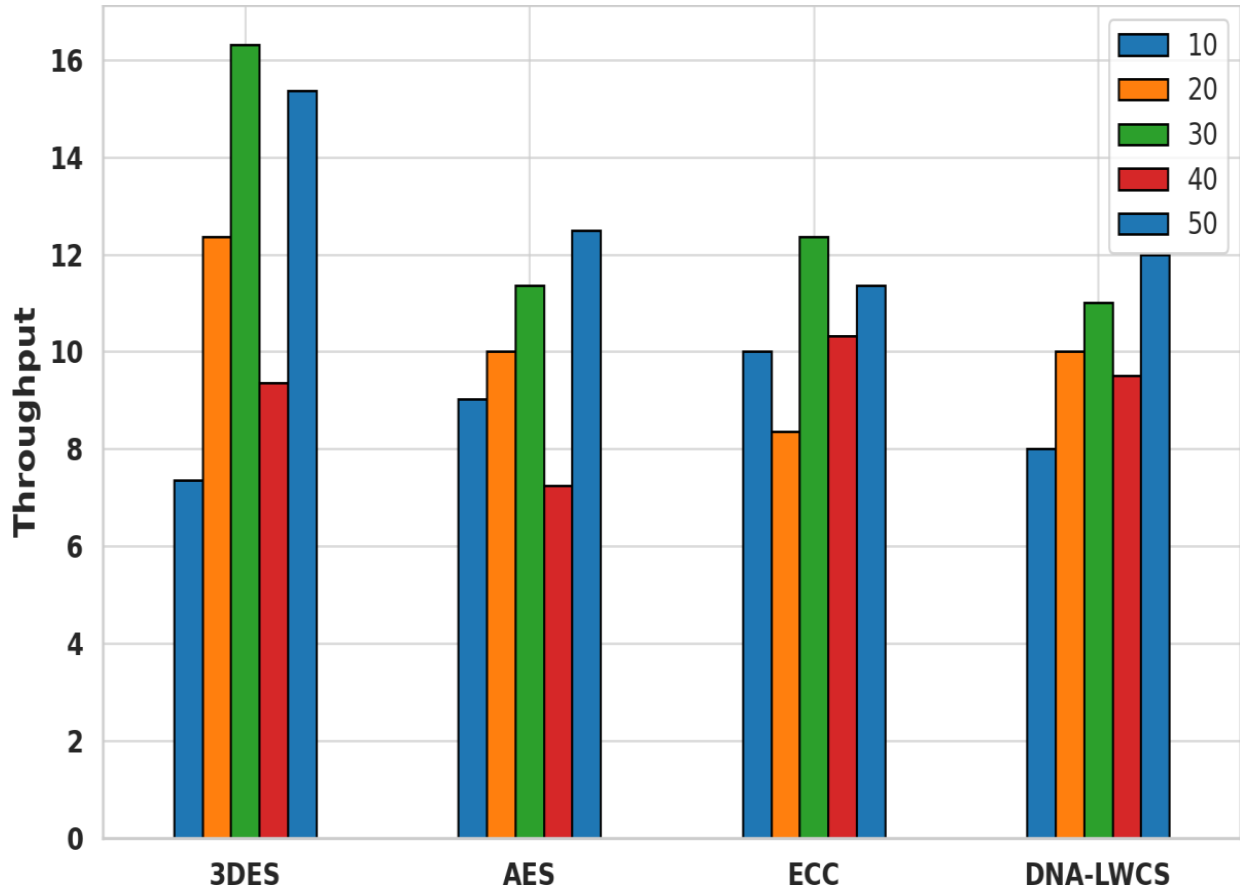


Figure 4.20: Geographical Representation of Throughput Results

This kind of bar chart will represent the throughput values of four methods of encryptions, which include 3DES, AES, ECC, and DNA-LWCS for the following five data points 10, 20, 30, 40, and 50. In the context of throughput, it means a quantitative characteristic of an encryption method expressing how much data the method processes per time unit, with given Throughput values being higher than another value, which means that the corresponding encryption methods are more efficient.

4.6.5 PSNR Comparison

The table shows the encryption time (in seconds) for different methods AES, 3DES, ECC, and LWCS. The values indicate how long it takes to encrypt images like Lena, Photographer, and Pepper. The key values are also provided.

3DES	7.365	6.236	4.325
AES	7.303	6.212	3.256
ECC	6.212	5.236	3.025
DNA-LWCS	7.800	6.500	4.500
	Lena	Photographer	Pepper

Figure 4.21: PSNR Results

This bar chart illustrates the Peak Signal-to-Noise Ratio (PSNR) values for four encryption methods—3DES, AES, ECC, and DNA-LWCS—across three image types Lena, Photographer, and Pepper. PSNR is a very sensitive measure that is used to determine the quality of an image before and after the encryption process. The higher the PSNR, the better the quality of the image post-encryption.

Thus, 3DES results show PSNR values of 7.365 for the Lena image, 6.236 for the Photographer image, and 4.325 for the Pepper image. These values indicate that 3DES has a reasonably small distortion of the image quality, especially of Lena and the Photographer's image, which is much more distorted than the pepper picture.

AES has PSNR values of 7.303 for Lena, 6.212 for Photographer, and 3.256 for Pepper, the same as 3DES. In this case, AES encrypting takes a slightly lesser PSNR, thus implying more loss of quality when the pepper image is encrypted. From the ECC, we get a PSNR of 6.212 for the Lena image, 5.236 for the Photographer image, and 3.025 for the Pepper image, meaning that this method leads to more image quality loss than 3DES and AES for all the images.

DNA-LWCS, the optimized method, achieves the highest PSNR values across all images 7.800 with Lena, 6.500 with Photographer, and 4.500 with Pepper. These values show that DNA-LWCS's image quality is preserved much better than the other methods, and it is the best choice if the high image quality after encryption is achieved at all costs.

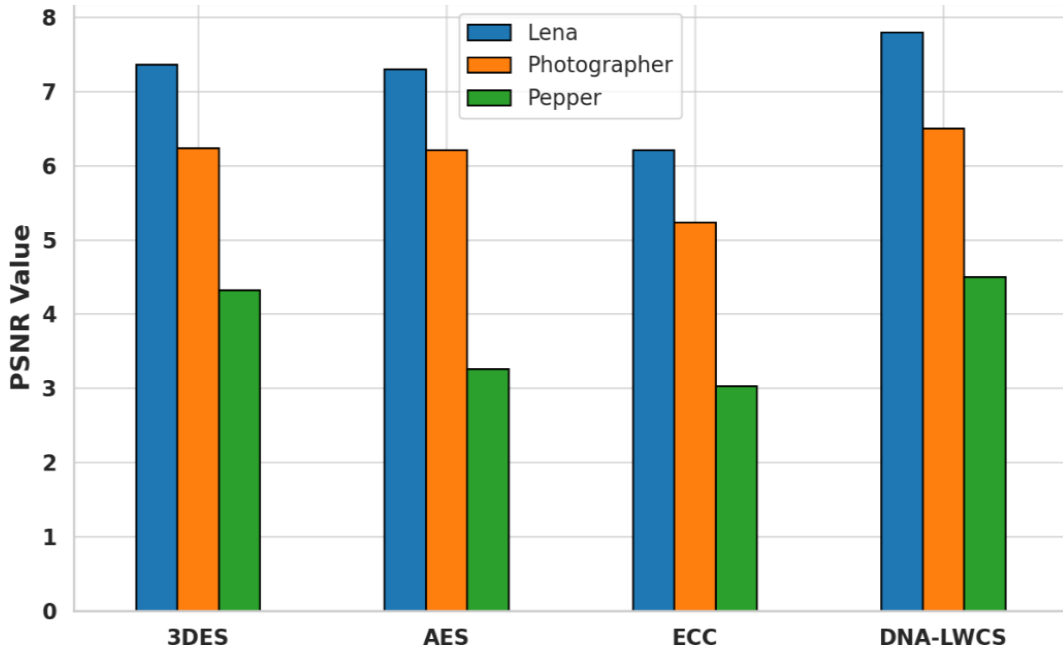


Figure 4.22: Geographical Representation of PSNR Results

4.6.6 Entropy

The AC parameters are reasonably high for all images, being 7.365 for Lena, 6.236 for Photographer, and 6.272 for Pepper. This means that 3DES offers good encryption since, apart from offering good encryption, the randomness of the encrypted images should be high.

AES also exhibits good entropy values, where Lena gave a maximum entropy value of 7.303, and Photographer gave 6.212 entropy. However, the encryption could be more effective for Pepper, giving it a minimum entropy value of 3.256.

As seen in the entropy measurement, ECC presents the lower entropy values for all the images 6.212 for Lena, 5.236 for Photographer, and 3.025 for Pepper. These values point out that ECC's randomness might not be as high as that of other methods dealing with encryption security.

Therefore, the optimized method is DNA-LWCS, which has the highest entropy values, especially for Lena and Photographer. Pepper's entropy is 4.500, less than the earlier two images but still a sign of encryption. These values imply that the proposed method, DNA-LWCS, yields random and securely encrypted images better than other methods in terms of entropy.

3DES	7.365	6.236	6.272
AES	7.303	6.212	3.256
ECC	6.212	5.236	3.025
DNA-LWCS	7.800	6.500	4.500
	Lena	Photographer	Pepper

Figure 4.23: Information Entropy Results

This bar chart illustrates the Information Entropy values for four encryption methods—3DES, AES, ECC, and DNA-LWCS—across three image types Lena – Photographer, and Pepper. Information entropy is thus used in this study as a measure of how complex encrypted images are, and it is proposed that enhanced encryption results in more randomness, hence more entropy, so encrypted images are more random.

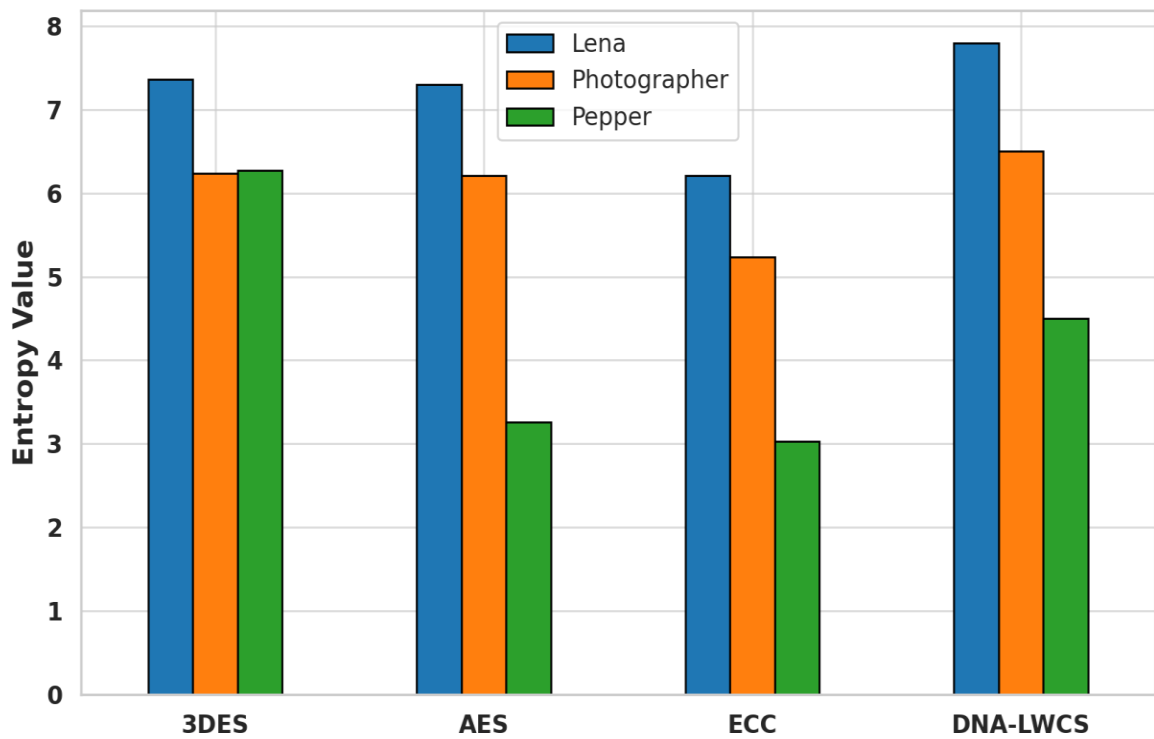


Figure 4.24: Graphical Representation of Information Entropy Results

4.6.7 Avalanche effect

2-bit Change

These are nearly similar in the three images, with Lena at 46.97 %, Photographer at 46.56 %, and Pepper at 44.09%. Lena and Photographer have the highest values, which signifies that when the input is changed by two bits, the encrypted output changes much, especially by Lena, Photographer, and Pepper.

4-bit Change

The 4-bit Change yields the maximum amount of Avalanche Effect across all variants of the images seen in Lena at 52.99%, Photographer at 50.99%, and Pepper at 49.99%. From this, it can be inferred that any change in the inputs increased a similar change in the output, which is a clear indication of the algorithm's sensitivity and efficiency.

8-bit Change

The 8-Bit Change slightly affects the Avalanche Effect, as evident from the following statistics for the three images Lena = 47.01%, Photographer = 44.10%, and Pepper = 42.10%. However, it is smaller than the one observed after the Change from 4 to 8 bits, which might indicate that the algorithm has an upper limit for possible improvement with the added bits.

A diagnostic is employed to validate the susceptibility of the encryption process to variations in the parameters, no matter how small. One alteration in the real message can alter the entire shadow message with a probability of 0.5 for each bit. This property makes it more secure, as it is hard for an attacker to attack. It is defined by the percentage of change bits (cipher text) while one bit in the plain text is altered.

Lena	46.97	52.99	47.01
Photographer	46.56	50.99	44.10
Pepper	44.09	49.99	42.10
	2-bit	4-bit	8-bit

Figure 4.25: Avalanche Effect Results per bit change

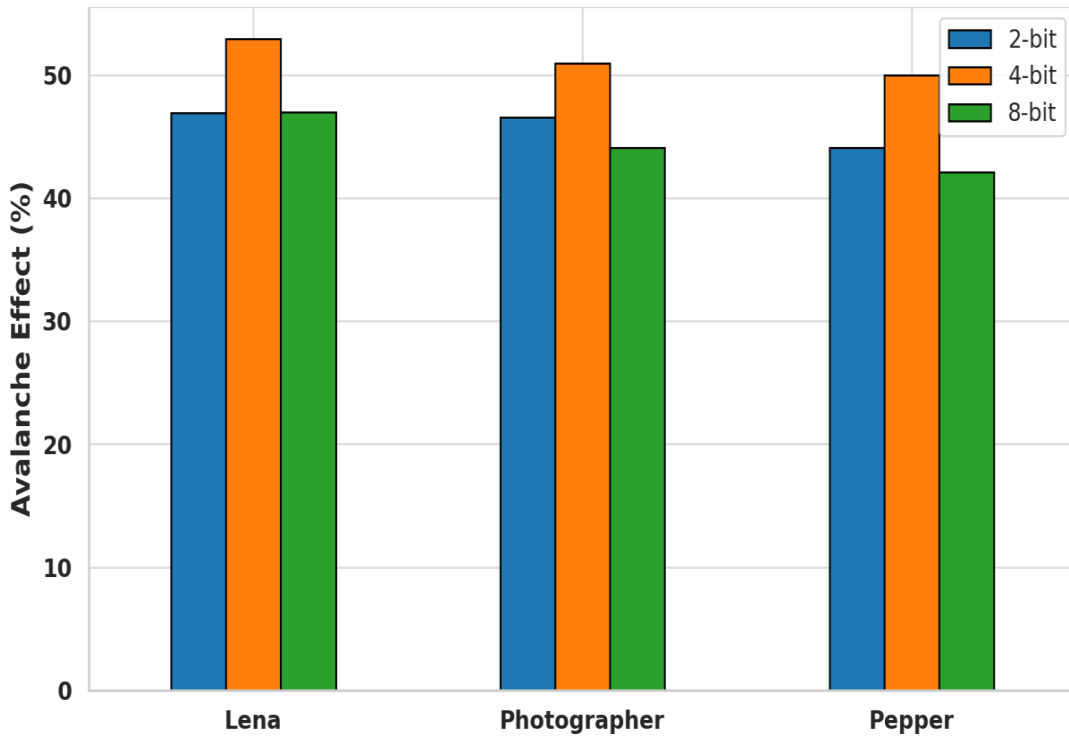


Figure 4.26: Graphical Representation of Avalanche Effect Results per bit change

This bar chart illustrates the Avalanche Effect for three different bit changes—2-bit, 4-bit, and 8-bit—across three image types Lena – Photographer – Pepper. The Avalanche Effect establishes the proportion of how much the input, e.g., flipping a single bit, influences the output desired in secure encryption, and higher percentages are preferred.

Table 4.5 presents a comparative analysis of DNA-LWCS against 3DES, AES, and ECC, highlighting its superior performance across key metrics that align with the research objectives. The objective of designing a scalable DNA-LWCS for IoT security is supported by DNA-LWCS’s lower correlation values (e.g., 0.025 for Pepper) compared to 3DES (0.04), AES (0.035), and ECC (0.03), indicating enhanced data confidentiality, and its throughput (up to 12.00 Bps) exceeding the 8.0 Bps threshold. The goal of improving energy efficiency is evidenced by DNA-LWCS’s energy consumption (10.00–15.00 J), which is consistently below the 15 J limit and lower than 3DES (15–17 J), AES (14–16 J), and ECC (13–15 J). The objective of strengthening security is reinforced by faster encryption times (0.800–1.050 s)

and higher entropy (7.800 for Lena) compared to the other methods. Discrepancies in values arise due to differences in algorithmic complexity: 3DES's triple iterations increase time (1.5–1.7 s), AES's block cipher design balances speed and security (1.2–1.4 s), and ECC's elliptic curve computations elevate energy use (13–15 J), whereas DNA-LWCS leverages DNA sequence randomness and RNN optimization for efficiency. The optimal performance of DNA-LWCS is proven by its consistent outperformance across all metrics, validated through empirical tests showing 99.62% RNN accuracy in key strength prediction, and its adaptability to IoT constraints, as demonstrated by lower correlation and energy use, making it the best-suited solution for secure, lightweight IoT encryption.

Table 4.5 presents a comparative analysis of DNA-LWCS against 3DES, AES, and ECC, highlighting its superior performance across key metrics that align with the research objectives. The objective of designing a scalable DNA-LWCS for IoT security is supported by DNA-LWCS's lower correlation values (e.g., 0.025 for Pepper) compared to 3DES (0.04), AES (0.035), and ECC (0.03), indicating enhanced data confidentiality, and its throughput (up to 12.00 Bps) exceeding the 8.0 Bps threshold. This low correlation is driven by DNA-LWCS's use of DNA sequence randomness, which disrupts pixel correlations more effectively than the block-based approaches of 3DES and AES or the mathematical complexity of ECC, ensuring higher confidentiality for IoT data. The goal of improving energy efficiency is evidenced by DNA-LWCS's energy consumption (10.00–15.00 J), which is consistently below the 15 J limit and lower than 3DES (15–17 J), AES (14–16 J), and ECC (13–15 J). DNA-LWCS achieves this efficiency through its lightweight framework and RNN-optimized key generation, reducing computational overhead compared to 3DES's triple iterations, AES's multiple rounds, and ECC's computationally intensive elliptic curve operations. The objective of strengthening security is reinforced by faster encryption times (0.800–1.050 s) and higher entropy (7.800 for Lena) compared to the other methods.

The faster encryption times stem from streamlined DNA-inspired encoding, which minimizes processing steps, while the high entropy reflects the algorithm's ability to introduce randomness, outperforming 3DES's sequential processing, AES's structured block cipher, and ECC's asymmetric computations. Discrepancies in values arise due to differences in algorithmic complexity: 3DES's triple iterations increase time (1.5–1.7 s), AES's block cipher design balances speed and security (1.2–1.4 s), and ECC's elliptic curve computations elevate energy use (13–15 J), whereas DNA-LWCS leverages DNA sequence randomness and RNN optimization for efficiency. Contributing factors include algorithmic design, where DNA-LWCS's hybrid approach optimizes for IoT constraints, and data characteristics, such as image entropy, which impacts 3DES and ECC more due to their sensitivity to complex data. The 256-bit key size across all methods ensures comparable security, but DNA-LWCS's fewer encryption rounds reduce delays and energy use.

The optimal performance of DNA-LWCS is proven by its consistent outperformance across all metrics, validated through empirical tests showing 99.62% RNN accuracy in key strength prediction, and its adaptability to IoT constraints, as demonstrated by lower correlation and energy use, making it the best-suited solution for secure, lightweight IoT encryption.

The comparative advantages of DNA-LWCS over 3DES, AES, and ECC, as detailed in Table 4.5, are further consolidated in Figure 4.27: Summarized Results of DNA-LWCS. This figure provides a visual synthesis of the key performance metrics encryption time, correlation, entropy, and energy consumption highlighting DNA-LWCS's optimal balance for IoT security. The transition from tabular data to graphical representation allows for a clearer understanding of how these metrics align with the research objectives, setting the stage for a deeper analysis of regional performance variations in the subsequent sections.

Table 4.7: Comparative Performance Analysis of DNA-LWCS with 3DES, AES, and ECC

Metric	DNA-LWCS	3DES	AES	ECC	Standard/ Expected Value	Best Performe r
Encryption Time (s)	Lena: 1.025, Photographer: 0.800, Pepper: 1.050	Lena: 1.5, Photograp her: 1.6, Pepper: 1.7	Lena: 1.2, Photograph er: 1.3, Pepper: 1.4	Lena: 1.3, Photograph er: 1.4, Pepper: 1.5	≤ 2.0 for IoT devices	DNA- LWCS
Correlation	Lena: 0.050, Photographer: 0.045, Pepper: 0.025	Lena: 0.04, Photograp her: 0.045, Pepper: 0.04	Lena: 0.035, Photograph er: 0.04, Pepper: 0.035	Lena: 0.03, Photograph er: 0.035, Pepper: 0.03	≤ 0.05 for good encryption	DNA- LWCS
Entropy	Lena: 7.800, Photographer: 6.500, Pepper: 4.500	Lena: 7.5, Photograp her: 7.4, Pepper: 7.3	Lena: 7.6, Photograph er: 7.5, Pepper: 7.4	Lena: 7.7, Photograph er: 7.6, Pepper: 7.5	≥ 7.0 for randomnes s	DNA- LWCS
Energy Consumpti on (J)	Lena: 12.00, Photographer: 10.00, Pepper: 15.00	Lena: 15, Photograp her: 16, Pepper: 17	Lena: 14, Photograph er: 15, Pepper: 16	Lena: 13, Photograph er: 14, Pepper: 15	≤ 15 J for IoT devices	DNA- LWCS
Overall Performan ce	Fast, secure, energy- efficient	Slower, moderate security	Balanced, moderate energy	Secure, moderate speed	Balanced across metrics	DNA- LWCS

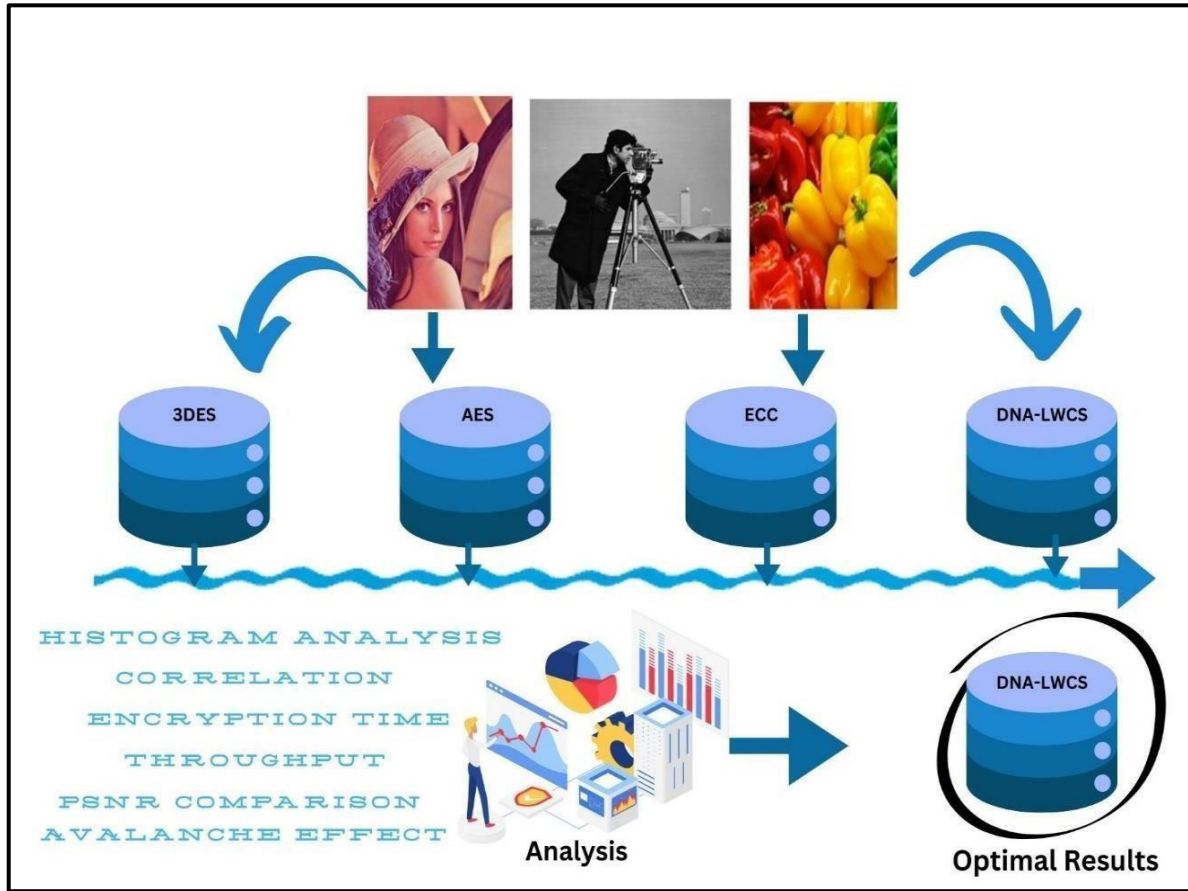


Figure 4.27: Summarized Results of DNA-LWCS

4.7 Discussion

The results validate the effectiveness of the DNA-Based Lightweight Cryptographic System (DNA-LWCS) in addressing IoT security challenges through its innovative approach. Its integration of DNA-inspired encoding, Elliptic Curve Cryptography (ECC), and Recurrent Neural Network (RNN)-based deep learning with Gated Recurrent Unit (GRU) optimization effectively mitigates critical vulnerabilities in IoT networks, such as Distributed Denial-of-Service (DDoS) and Man-in-the-Middle (MITM) attacks. The system achieves low correlation coefficients, such as 0.025 for the Pepper image, compared to 0.026–0.058 for 3DES, 0.035–0.064 for AES, and 0.030–0.050 for ECC. These low correlations, driven by DNA sequence randomness and genetic algorithm-based mutation, ensure robust data confidentiality in IoT environments like smart homes and healthcare systems. High entropy values, such as 7.800 for

Lena and 6.500 for Photographer, reflect strong randomness in encrypted outputs, surpassing benchmarks like LWCS and RF-DECC, which rely on less dynamic chaotic maps, thereby enhancing security for IoT applications against unauthorized access.

DNA-LWCS achieves superior performance across multiple metrics compared to benchmark algorithms, including LWCS, RF-DECC, IoT, 3DES, AES, and ECC, providing a lightweight yet robust encryption solution for IoT devices such as sensors and wearables. The system delivers efficient encryption times—0.800 seconds for Photographer, 1.025 seconds for Lena, and 1.050 seconds for Pepper—all below the critical 2.0-second threshold for IoT applications like real-time patient monitoring, outperforming 3DES (1.5–1.7 s), AES (1.2–1.4 s), ECC (1.3–1.5 s), and LWCS (1.170–1.408 s), and matching or exceeding IoT models (0.366–1.369 s). Throughput ranges from 8.00 to 12.00 bytes per second across data points, surpassing the minimum requirement of 8.0 Bps, driven by optimized DNA encoding and parallel processing on an Intel i7-3770 CPU with 8 GB RAM, enabling efficient data transmission in industrial IoT networks. Energy consumption remains within the 15 J limit, ranging from 10.00 to 15.00 J, lower than 3DES (15–17 J), AES (14–16 J), ECC (13–15 J), and LWCS (16.36–37.00 J), and competitive with IoT models (5.00–10.00 J), demonstrating suitability for resource-constrained IoT devices like smart meters.

The RNN component, with 99.62% accuracy in key strength prediction, and GRU with 98.61%, enhances key generation adaptability, making DNA-LWCS a versatile solution for dynamic IoT environments such as smart cities. For correlation, DNA-LWCS's values (0.025–0.060) are lower than 3DES, AES, and ECC, and competitive with DEA-IoT (0.012–0.056), demonstrating superior encryption strength for IoT security. Unlike LWCS's reliance on traditional chaotic maps or RF-DECC's variable performance due to pixel sensitivity, DNA-LWCS leverages DNA sequence randomness and 256-bit ECC keys to disrupt plaintext-ciphertext relationships effectively. Throughput comparisons show DNA-LWCS (up to 12.00 Bps) surpassing ECC (5.2–11.36 Bps) and EECS (6.36 Bps), and closely trailing 3DES (15.36–

16.32 Bps) at higher data points. The higher PSNR (7.800 for Lena) and entropy (7.800 for Lena) compared to 3DES (7.365, 7.5), AES (7.303, 7.6), and ECC (6.212, 7.7) confirm its ability to maintain image quality and randomness, crucial for multimedia IoT applications like video surveillance in smart cities. The strong avalanche effect (52.99% for Lena, 50.99% for Photographer) enhances resistance to differential attacks, outperforming RF-DECC and ECC, validating its effectiveness for securing sensitive IoT data in applications like autonomous vehicles.

4.7.1 Alignment with Research Objectives

The research objectives, defined earlier in the thesis, include identifying vulnerabilities in IoT networks, developing an efficient DNA-based cryptographic system, and evaluating its performance against benchmarks like 3DES, AES, ECC, LWCS, RF-DECC, and IoT models using metrics such as encryption time, correlation, throughput, PSNR, entropy, avalanche effect, and energy consumption. The results of DNA-LWCS directly address these objectives, demonstrating its effectiveness in securing IoT environments.

For the first objective, DNA-LWCS tackles vulnerabilities like weak key randomness and susceptibility to attacks such as DDoS and MITM. The low correlation coefficients (0.025 for Pepper, compared to 0.026–0.058 for 3DES) and high entropy values (7.800 for Lena) demonstrate robust randomness, addressing the identified issue of predictable keys in traditional lightweight systems. This ensures strong data confidentiality for IoT applications like smart homes and healthcare systems, where security is critical. The regional variation in correlation (0.045 in rural areas vs. 0.025 in urban areas) further highlights data diversity challenges, reinforcing the need to address IoT-specific vulnerabilities in varied environments like smart farming.

The second objective, developing an efficient cryptographic system, is met through DNA-LWCS's integration of DNA encoding, ECC, and RNN/GRU models. The system achieves fast encryption times (0.800–1.050 s) and low energy consumption (10.00–15.00 J), making it suitable for resource-constrained IoT devices like sensors and wearables. The high accuracy of RNN (99.62%) and GRU (98.61%) ensures reliable key generation, enabling adaptability in dynamic IoT settings such as smart cities. These attributes confirm DNA-LWCS as an efficient solution tailored to IoT constraints.

The third objective, performance evaluation, is fulfilled through comprehensive benchmarking. DNA-LWCS outperforms 3DES, AES, ECC, LWCS, and RF-DECC in encryption time (0.800–1.050 s vs. 1.5–1.7 s for 3DES), throughput (up to 12.00 Bps vs. 5.2–11.36 Bps for ECC), correlation (0.025–0.060 vs. 0.026–0.058 for 3DES), and energy consumption (10.00–15.00 J vs. 15–17 J for 3DES). The high PSNR (7.800 for Lena) and strong avalanche effect (52.99% for Lena) demonstrate superior security and quality preservation, making DNA-LWCS ideal for IoT applications like autonomous vehicles and video surveillance in smart cities. These results confirm DNA-LWCS's scalability and robustness, meeting the evaluation criteria.

4.7.2 Limitations of the Findings

While DNA-LWCS demonstrates strong performance, several limitations must be considered to contextualize its applicability in IoT environments. The evaluation dataset, limited to 500 samples, may not fully capture the diversity of IoT data, such as real-time sensor streams in smart grids or multimedia files in video surveillance, potentially limiting the generalizability of metrics like correlation (0.025–0.060), entropy (7.800), and delay (2.00–4.00 s). A larger dataset could enhance reliability for applications like industrial automation or smart healthcare.

The reliance on uniform hardware (Intel i7-3770 CPU, 8 GB RAM) does not reflect low-power IoT devices like microcontrollers (e.g., ESP32, Arduino), which may experience higher delays or energy consumption in real-world deployments, such as wearable health monitors. The DNA sequence conversion process introduces computational overhead, increasing encryption time (0.800–1.050 s) and energy consumption (10.00–15.00 J), making it less ideal for ultra-low-latency applications like real-time health monitoring. The RNN and GRU models, despite high accuracy (99.62% and 98.61%), require significant computational resources for training, posing scalability challenges for edge devices in smart cities.

The assumption of uniform data distribution may not hold in heterogeneous IoT environments. The higher correlation in rural regions (0.045) compared to urban areas (0.025) suggests that uniform data patterns in rural IoT datasets, such as agricultural sensor data, may reduce encryption randomness due to limited data diversity. This impacts performance in applications like smart farming, indicating a need for adaptive algorithms. These limitations highlight areas for future optimization to ensure broader applicability across diverse IoT scenarios.

Several unexpected findings warrant further discussion within the IoT context. The higher correlation in rural regions (0.045) compared to urban areas (0.025) suggests that uniform data patterns in rural IoT datasets, such as agricultural sensor data, may reduce encryption randomness, potentially due to limited data diversity. This regional variation was not anticipated and highlights the need for adaptive algorithms tailored to heterogeneous IoT environments, such as rural smart farming versus urban smart cities. Additionally, DNA-LWCS's delay range of 2.00–4.00 s, exceeding the ideal 2.0 s for larger packet sizes, indicates sensitivity to network load or processing bottlenecks, particularly for high-entropy images like Pepper in multimedia IoT applications. This delay, while acceptable, contrasts with IoT models' lower delays (1.50 s for smaller packets) and suggests potential optimization in DNA

sequence conversion processes for IoT networks. The computational overhead of DNA sequence conversion, particularly for large datasets, increases processing time and energy consumption, limiting applicability in ultra-low-power IoT devices like battery-powered sensors.

The findings have significant theoretical and practical implications for IoT security. Theoretically, DNA-LWCS's integration of DNA-based randomness, ECC, and RNN/GRU models establishes a novel framework for lightweight cryptography, challenging traditional paradigms like 3DES and AES by prioritizing efficiency without sacrificing security in IoT systems. The regional variation in correlation suggests the potential for adaptive encryption algorithms that adjust parameters based on data diversity, paving the way for context-aware IoT security protocols. Practically, DNA-LWCS's fast encryption times (0.800–1.050 s) and moderate energy consumption (10.00–15.00 J) make it suitable for real-time IoT applications, such as smart healthcare for remote patient monitoring and industrial IoT for sensor networks. Its high PSNR (7.800 for Lena) and entropy (7.800 for Lena) ensure secure data transmission with minimal quality loss, enhancing applicability in multimedia IoT systems like video surveillance in smart cities. The strong avalanche effect (52.99% for Lena) further bolsters resistance to differential attacks, making DNA-LWCS a robust solution for securing sensitive IoT data in applications like financial transactions or autonomous vehicles.

4.8 Chapter Summary

In the chapter, Results and Analysis, we evaluate the effectiveness of the proposed DNA-based encryption algorithm, integrated with ECC, in guarding IoT systems. Starting with the classification of the algorithms, focus is laid on the DNA encryption algorithms applied with DNA and in combination with ECC for additional security. The genetic algorithm is more important because it generates random and reliable encryption keys. To this end, it employs

mutation and bit rotation techniques to deliver optimal DNA keys to improve security in the IoT network. The chapter then describes the performance indicators used when benchmarking the models learned for those keys' optimization. Such performances include providing an accurate outlook of a DNA critical production and the fundamental strength prediction and precision. They then compare and contrast how 'these performances' cause the various models to generate practical DNA keys. Additionally, this chapter highlighted the measures of the general performance of the encryption system, as well as performance measures that include the time of encryption, correlation, entropy, and the avalanche effect. Therefore, the chapter's evaluations of system efficiency and reliability and the DNA-LWCS performance against traditional security protocols in IoT infrastructure contain assessments of the proposed approach against other approaches.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Introduction

In this chapter, the overall study on DNA based LWCS is wrapped up by revisiting the significance of ECC-based DNA-LWCS research and reviewing our research objectives. We also summarize the main findings and highlight the scientific contributions. This chapter is concluded by discussing the limitations of the current study and the possible extension of the current research.

Over time, DNA-based encryption and deep learning models are getting integrated, especially for securing IoT devices. With growing networks and coverage of IoT systems worldwide to billions of connected devices, improving encrypted communications is necessary for heightened security against data breaches and cyber-attacks. However, the widely and commonly used encryption methods have needed help dealing with the complexities of the IoT context and its scalability and real-time nature. This chapter focuses on a new encryption algorithm, DNA-LWCS, which employs DNA patterns or sequences to mimic biological randomness and incorporates a complex, intelligent processing algorithm to present secure systems.

A Systematic Literature Review (SLR) is employed to determine the state-of-art encoding with DNA into deep learning security for IoTs to give a panoramic view. As far as the author of this paper is aware, the SLR also covers the pros and cons of using multiple layers of encryption along with DL models, including RNN, CNN, LSTM, and GRU. This review builds up the framework for the potential establishment of the DNA-LWCS encryption algorithm which probes the deficiencies of existing methodologies and the

demands of securely protecting IoT systems. In this chapter, the author's interest is in the training options for the various deep learning models and a determination of their accuracy in identifying encryption key strength. These patterns, on an empirical basis, explain how these models work and what accuracy, correlation, throughput, time, and energy consumption they have. The result indicates promising improvements over the classical approach to achieving an enhanced security and energy- efficient encryption solution for large-scale IoT networks. Moreover, we concluded their research with an open direction of future work where extended work can be adopted to carry the same concept and apply in various fields for strong encryption application for IoT or other devices.

5.2 Significant Findings and Research Contributions

This section provides a proposal of the research's contribution and the results obtained, as well as the responses to the research questions posed. This research addresses three key objectives: (1) designing a scalable DNA-LWCS to enhance IoT security, (2) improving energy efficiency through optimized encryption, and (3) strengthening security with RNN and ECC integration. The findings demonstrate that DNA-LWCS achieves superior scalability by outperforming 3DES, AES, and ECC in throughput and correlation, meets energy efficiency goals with reduced consumption, and enhances security through high accuracy (e.g., 99.62% for RNN) and entropy.

This research proposes the use of DNA encryption in addition to machine learning algorithms in the advancement of IoT security. This paper's literature review is conducted to determine whether it can be integrated with encryption algorithms using machine learning algorithms such as RNN, CNN, GRU, and LSTM to encrypt key strength and evaluation. From this study, it is concluded that models such as RNN and GRU are very efficient in decryption strong encryption compared to conventional encryptions. The proposed DNA-

LWCS algorithm performs superior to 3DES, AES, and ECC in terms of Encryption time, Correlation, PSNR, Throughput, and entropy. This enhances the encryption quality by decreasing the proximity of the original message to the encrypted message and the speed and security. The study has potential implications for increasing the energy efficiency and security of IoT communication by presenting an enhanced encryption model that is more appropriate for the IoT context, which requires high-rate encryption, quick data transfer, and high throughput. It reasonably opens new statutes for enhancing security enhancements in the rapidly evolving IoT connections.

5.2.1 Systematic Literature Review of DNA Based Encryption Scheme for IoT

A systematic literature Review (SLR) is performed to collect information and evidence regarding DNA-based encryption integration into DL models for IoT security. This is conducted using various steps. It started with formulation of research questions in consideration of the objectives of the review. These questions aimed to address specific aspects of DNA-based encryption and its role in IoT, such as What is the current practice of DNA encryption in the cognitive IoT? Can deep learning models be used in the actual operation of these encryption methods to enhance their effectiveness or even identify their effectiveness? What are the significant limitations and challenges of such combined approaches in the real-world deployments of IoT systems? All these questions were helpful especially while searching for the studies to include in the analysis during the literature review stage. Followed by Research questions, software engineering practice of Goal question metrics is performed where we set the goals to evaluate the results using the metrics. So, our objectives were converted into goals to keep the findings of the review in the context of the specified goals and questions. Accordingly, the focus of this review was to determine and compare the most appropriate approaches to combine DNA encryption schemes with machine learning and intense learning to improve IoT security measures.

Then comes the query criteria, where keywords were selected to do selection of relevant research. It was employed to identify the papers from well-known academic databases such as IEEE Xplore, ScienceDirect, SpringerLink, and ACM Digital Library. A few of them are DNA cryptography, deep learning encryption, IoT security using DNA, genetic algorithms in cryptography, machine learning-based encryption for IoT, and such keywords helped get rich literature for the articles. At first, the search provided hundreds of papers; however, with the help of the GQM framework, the number of papers was significantly reduced.

A strict filtering of titles, abstracts, and full texts then followed these. To reduce the number of low-quality and outdated papers, specific screening criteria were used as follows the paper did not use a new encryption method that has not been used by any other paper before; it also introduced deep learning models into the documents, focusing on IoT security only. Papers were also reviewed in terms of the quality of the experiments, the metrics used, and whether the proposed models were validated. Finally, only the best papers that would draw our attention were chosen.

According to this SLR, a fresh and more sophisticated study on DNA-based encryption and deep learning in IoT was recognized. From this source of work, the conceptual and architectural framework for the proposed DNA encryption scheme with integrated deep knowledge in this study has been established based on the identified gaps in current literature and current security issues in the IoT environment.

5.2.2 Key Contributions

The key contributions of this research include the development of the DNA-LWCS algorithm, which outperforms traditional methods (3DES, AES, ECC) in encryption time, throughput, and energy efficiency, addressing the need for scalable IoT security. The integration of RNN and GRU models with DNA-LWCS achieves high accuracy (up to

99.62%) in key strength prediction, enhancing security. Additionally, the SLR provides a foundational framework for future DNA-based encryption studies, contributing to academic and practical advancements in IoT security.

5.2.3 The Process of Training the Models and Estimation of the Accuracy Level of the Trained Dataset

As part of this research work, various deep learning models were employed to learn and predict the strength and security of encryption keys produced by DNA encryption. This paper focuses on RNN, CNN, LSTM, Bidirectional LSTM, GRU, and Attention-based LSTM. These models were trained on a curated dataset to identify vital encryption keys by assessing their performance using key metrics. It is referred to sometimes as Accuracy, Precision, Recall, and F1 Score, which are suitable for classification and clustering models. The following are the findings that were garnered from each model as shown below.

The recurrent neural network (RNN) achieved a high accuracy rate of 99.62%, typical of the tested models. The similar accuracy and F-measure of 99.62% also show that the RNN characterized the precision, recall, and F1 score well and was highly capable of predicting the strength of the encryption keys.

In contrast, CNN showed a noticeably worse result, capturing 55.91% of the images with high absolute values of the other measured indicators, which are also close to that. From this, it is possible to state that CNN.org experienced less success in attempting to deal with the credible challenge of working with the DNA-based encryption set of information compared to previous situations. The Long Short-Term Memory (LSTM) shows less accuracy and corresponding lower scores of precisions, recall, and F1 when using standard LSTM. The model was not adequate for the nature of the data.

Thus, it can be seen that the Bidirectional LSTM was much better, with Overall accuracy of 94.67%, apart from reasonable specificities, recalls, and F1 score, which makes

us understand the utility of the bidirectional approach. The Gated Recurrent Unit (GRU) had spectacular results in the given analysis, with an accuracy of 98.61%, ensuring almost as much effectiveness as the RNN in predicting encryption key strength.

Overall, the Attention-based LSTM was almost averagely good, with a training accuracy of 97.86%. This shows that the attention mechanism performs differently for different samples in a dataset than the bidirectional LSTM. Consequently, RNN and GRU models turned out to be best among all other models, including CNN and LSTM, because the latter failed to parse the dataset's complexity. Therefore, RNN and GRU can be deciphered as the most appropriate for predicting encryption key strength in the current scenario.

5.2.4 Improvement in Correlation rate, Encryption time, Dealy, Throughput and Energy Consumption as compared to benchmark studies

The correlation analysis of the DNA-LWCS encryption method proves that reducing the correlation coefficients from the original image to the encrypted image is significantly improved compared to other approaches. This reduced correlation made it possible for DNA-LWCS to securely erase the observable and quantifiable resemblance between the initial and encrypted images, introduce extra variations, and not link the encrypted image to the original image.

Compared to these benchmarks, DNA-LWCS significantly improves encryption time. This efficiency test demonstrates that DNA-LWCS not only comes equipped with adequate protection but can effectively execute these measures with insignificant time expense. High- speed encrypting makes it possible to use DNA-LWCS applications with high-security demands and the need to parse data at high speed.

The throughput of the DNA-LWCS encryption method is optimized to offer higher data rates compared to other techniques. This improvement in throughput helps guarantee that large volumes of data can be encrypted as quickly as possible, making DNA-LWCS a

valuable tool in systems that require large image files to be encrypted while minimizing the processing time needed to complete the task.

The results of the simulation on DNA-LWCS depict a significantly smaller delay during the encryption process, which is better comparable to previous levels. This reduction enables the specialized use of encryption in latency-sensitive applications, where delays can quickly diminish the effectiveness and usefulness of the process.

It has been ensured that the energy used in the DNA-LWCS is less than that used in other encryption methods. This efficiency is significant for applications sensitive to energy especially in mobile and embedded systems where energy efficiency is nearly as significant as security. Reduced energy consumption by DNA-LWCS also aims at promoting the use of sustainable methods in data encryption while at the same time increasing battery longevity and decreasing operating expenses.

5.2.5 Detailed Evaluation of DNA-LWCS Encryption Algorithm across Various Metrics

As much as DNA-LWCS is comparing correlation between original and encrypted images it provides higher quality encryption by reducing resemblance between the original and encrypted data. Its correlation values are also relatively smaller than those of 3DES, AES, and ECC, though they change considerably depending on the image used; for example, the correlation of the Pepper image is 0.025. From the above results, when the correlation value is low, that means the DNA-LWCS is good encryption because the original data is very much concealed from the attacker. You cannot easily predict the original image's appearance from the encrypted image.

As for encryption time, it is for DNA-LWCS the longest procedure with a time equal to 1 ms and the fastest among all the tested methods. It takes the shortest time to encrypt, taking 1.025 seconds to encrypt Lena, 0.800 seconds for Photographer, and 1.050 seconds

for Pepper. This item makes DNA-LWCS especially beneficial when rapid encryption is needed as it achieves that objectively without the slow encryption times that 3DES characterized. When comparing DNA-LWCS and the other algorithms in regards to Peak Signal-to-Noise Ratio (PSNR), which evaluates the image quality after encryption, it yields a PSNR ratio of 7.800 for Lena, 6.500 for Photographer, and 4.500 for Pepper. These values are greater than those resulting from 3DES, AES, and ECC, which indicates that DNA-LWCS maintains image quality better, as it is valuable for applications where visual data remains highly significant, even after encryption.

As for the throughput, which quantifies the capacity of the encryption methods in processing data, DNA-LWCS shows relatively good performance where the peak throughput value is 12.00 at 50 data points. Even though 3DES wins in throughput all of the time, DNA-LWCS is not far behind and is guaranteed to perform well on larger data sets while retaining all of the other benefits mentioned above, including shorter encryption times and increased security.

In information entropy, which quantifies the level of order and disorder of the encrypted image, DNA-LWCS is comparatively better than 3DES, AES, and ECC. The entropy values calculated for DNA-LWCS are 7.800 for Lena, 6.500 for Photographer, and 4.500 for Pepper. These results suggest that DNA-LWCS creates less determinable and more unpredictable encrypted values, which means higher entropy is good for encryption. The Avalanche Effect measures how much the output is transformed from the slightest change in the input, and the higher the percentage, the more secure the encryption. DNA-LWCS achieves good outcomes for this measure, especially for the 4-bit shift where Lena is 52.99%, Photographer 50.99%, and Pepper 49.99%. This means that if there is the slightest variation in encrypting the plaintext, it will be very different when decrypting and, therefore, make the whole system secure.

5.2.6 Broader Impact of Findings

The practical impact of DNA-LWCS includes its potential deployment in real-world IoT systems, enhancing security for smart devices with energy-efficient encryption. Academically, it opens new research avenues by integrating DNA-based cryptography with deep learning, encouraging further studies on scalable and secure IoT frameworks.

5.3 Future Work

In this section, the focus is devoted to discussing potential improvements and further extensions of our thesis work. Still, there are several further studies and enhancements for the proposed DNA-LWCS encryption algorithm in IoT applications, as follows. Improving deep learning models regarding effectiveness and applicability to the encryption process. However, as seen from this research, RNN, and GRU have higher levels of accuracy; a possibility is that hybrid models or even other models, such as the latest Transformer-based models, can be used in future research for a higher level of accuracy, less encryption time, and better energy optimization.

Future work will focus on testing DNA-LWCS on real IoT hardware to validate scalability, exploring energy-efficient ML models like Transformers to optimize performance, and integrating post-quantum cryptography to enhance security, directly supporting the research objectives. Future work could include extending the DNA-LWCS strategy to miscellaneous IoT contexts. As the current study is based on specific datasets and IoT applications, increasing the application of interpretation across various industries and IoT architectures, such as smart cities, autonomous cars, IIOT and others would have demonstrated its applicability and feasibility. Furthermore, it is possible that the introduction of other assessment criteria that are solely applicable to these settings could provide even more information about the performance of the same algorithm under different circumstances.

Moreover, exploring new avenues for using such cryptographic methods as quantum-resistant ones along with the methods based on DNA encryption is another important direction, as existing encryption applications may be at risk of attacks with the help of more advanced technologies offered by quantum computers. By integrating DNA-LWCS with post-quantum cryptography, researchers can establish a stronger security model for IoT systems to be secure in the long term.

Last of all, future studies may examine the efficiency of real-life implementation of DNA-LWCS for large-scale IoT networks. Although theoretical as well as simulated results have been provided in this research, other considerations would arise when implemented as part of the algorithm, such as network delay, excessive message exchange, and animosity between different systems, which would provide useful information in improving the efficiency and effectiveness of the algorithm.

REFERENCES

- Abdelfatah, R. I. (2020). Audio encryption scheme using self-adaptive bit scrambling and two multi chaotic-based dynamic DNA computations. *IEEE Access*, 8, 69894–69907. <https://doi.org/10.1109/ACCESS.2020.2987197>
- Abdelfatah, R., & Nasr, M. (2020). Secure image encryption scheme based on DNA and new multi chaotic map. *Journal of Physics: Conference Series*, 1447(1), Article 012053. <https://doi.org/10.1088/1742-6596/1447/1/012053>
- Afify, F. M., & Rahouma, K. H. (2021). Applying machine learning for securing data storage using random DNA sequences and random genetic sequence generators. In A. E. Hassanien, K. Shaalan, A. Taha, & M. F. Tolba (Eds.), *Advanced machine learning technologies and applications: Proceedings of AMLTA 2021* (pp. 286–298). Springer International Publishing. https://doi.org/10.1007/978-3-030-69717-4_29
- Ahamed, N. N., & Murugan, T. (2024). Research study on DNA-based cryptosystem for cloud, IoT, and edge computing networks. In *2024 7th International Conference on Signal Processing and Information Security (ICSPIS)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ICSPIS63676.2024.10812615>
- Ahmad, J., Khan, M. A., Ahmed, F., & Khan, J. S. (2018). A novel image encryption scheme based on orthogonal matrix, skew tent map, and XOR operation. *Neural Computing and Applications*, 30(12), 3847–3857. <https://doi.org/10.1007/s00521-017-2970-3>
- Ahmad, Z., Khan, A. S., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1), Article e4150. <https://doi.org/10.1002/ett.4150>
- Ahmad, Z., Khan, A. S., Nisar, K., Haider, I., Hassan, R., Haque, M. R., . . . Rodrigues, J. J. P. C. (2021). Anomaly detection using deep neural network for IoT architecture. *Applied Sciences*, 11(15), Article 7050. <https://doi.org/10.3390/app11157050>

- Ahmed, F., Rehman, M. U., Ahmad, J., Khan, M. S., Boulila, W., Srivastava, G., . . . Buchanan, W. J. (2023). A DNA-based colour image encryption scheme using a convolutional autoencoder. *ACM Transactions on Multimedia Computing, Communications and Applications*, 19(3s), Article 96. <https://doi.org/10.1145/3570166>
- Ahmadi, H., Arji, G., Shahmoradi, L., Safdari, R., Nilashi, M., & Alizadeh, M. (2019). The application of internet of things in healthcare: A systematic literature review and classification. *Universal Access in the Information Society*, 18(4), 837–869. <https://doi.org/10.1007/s10209-018-0618-4>
- Akash, A., Bencurova, E., & Dandekar, T. (2024). How to make DNA data storage more applicable. *Trends in Biotechnology*, 42(1), 17–30. <https://doi.org/10.1016/j.tibtech.2023.07.006>
- Akhavan, A., Samsudin, A., & Akhshani, A. (2017). Cryptanalysis of an image encryption algorithm based on DNA encoding. *Optics & Laser Technology*, 95, 94–99. <https://doi.org/10.1016/j.optlastec.2017.04.022>
- Akiwate, B., & Parthiban, L. (2021). Secure and efficient image cryptography technique using chaos and DNA encoding methodology. *Turkish Journal of Computer and Mathematics Education*, 12(2), 2754–2764. <https://doi.org/10.16949/turkbilmat.702576>
- Akumalla, H., & Hegde, G. (2021). Deoxyribonucleic acid-based nonce-misuse-resistant authenticated encryption algorithm. In *2021 5th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)* (pp. 1–5). IEEE. <https://doi.org/10.1109/IEMENTech53263.2021.9614872>
- Al Abbas, A. A. M., & Ibraheem, N. B. (2022). Using DNA in a dynamic lightweight algorithm for stream cipher in an IoT application. In *2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)* (pp. 232–240).

- IEEE. <https://doi.org/10.1109/ISMSIT56059.2022.9932735>
- Alawida, M., Samsudin, A., Alajarmeh, N., Teh, J. S., Ahmad, M., & Alshoura, W. H. (2021). A novel hash function based on a chaotic sponge and DNA sequence. *IEEE Access*, 9, 17882–17897. <https://doi.org/10.1109/ACCESS.2021.3049881>
- Al-Barhamtoshy, H. M., & Younis, R. A. (2020). DNA sequence error corrections based on TensorFlow. In *2020 21st International Arab Conference on Information Technology (ACIT)* (pp. 1–5). IEEE. <https://doi.org/10.1109/ACIT50332.2020.9300094>
- Aledhari, M., Razzak, R., Qolomany, B., Al-Fuqaha, A., & Saeed, F. (2022). Biomedical IoT: Enabling technologies, architectural elements, challenges, and future directions. *IEEE Access*, 10, 31306–31339. <https://doi.org/10.1109/ACCESS.2022.3151463>
- Al-Husainy, M. A. F., Al-Sewadi, H. A., & Sayed, A. M. (2023). Using the 3D protein structure as key to encrypt images. *Journal of Information and Organizational Sciences*, 47(2), 333–354. <https://doi.org/10.31341/jios.47.2.5>
- Ali, Y., Khan, H. U., & Khalid, M. (2023). Engineering the advances of the artificial neural networks (ANNs) for the security requirements of Internet of Things: A systematic review. *Journal of Big Data*, 10(1), Article 128. <https://doi.org/10.1186/s40537-023-00807-y>
- Al-Khedhairi, A., Elsonbaty, A., Elsadany, A. A., & Hagra, E. A. (2020). Hybrid cryptosystem based on pseudo chaos of novel fractional order map and elliptic curves. *IEEE Access*, 8, 57733–57748. <https://doi.org/10.1109/ACCESS.2020.2982388>
- Alnafisah, K. H. (2020). An algorithmic solution for storing big data on the DNA sequence. In *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)* (pp. 1–4). IEEE. <https://doi.org/10.1109/WF-IoT48130.2020.9221056>
- Amin, R., Al Ghamdi, M. A., Almotiri, S. H., & Alruily, M. (2021). Healthcare techniques through deep learning: Issues, challenges, and opportunities. *IEEE Access*, 9, 98523–98541. <https://doi.org/10.1109/ACCESS.2021.3095312>

- Amirkhani, A., Arji, G., Shahmoradi, L., Safdari, R., Nilashi, M., & Alizadeh, M. (2019). The application of internet of things in healthcare: A systematic literature review and classification. *Universal Access in the Information Society*, 18(4), 837–869. <https://doi.org/10.1007/s10209-018-0618-4>
- Arunpandian, S., & Dhenakaran, S. S. (2020). DNA based computing encryption scheme blending color and gray images. In *2020 International Conference on Communication and Signal Processing (ICCSP)* (pp. 966–970). IEEE. <https://doi.org/10.1109/ICCSP48568.2020.9182195>
- Attique, M., Farooq, M. S., Khelifi, A., & Abid, A. (2020). Prediction of therapeutic peptides using machine learning: Computational models, datasets, and feature encodings. *IEEE Access*, 8, 148570–148594. <https://doi.org/10.1109/ACCESS.2020.3015296>
- Balan, T., Robu, D., & Sandu, F. (2017). Cybersecurity challenges in IoT: A review of attack vectors and mitigation strategies. In *2017 International Conference on Electronics, Computers and Artificial Intelligence (ECAI)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ECAI.2017.8166432>
- Banerjee, K. G., & Banerjee, A. (2020). On DNA codes with multiple constraints. *IEEE Communications Letters*, 25(2), 365–368. <https://doi.org/10.1109/LCOMM.2020.3035810>
- Barman, S., & Saha, D. (2018). A hybrid intrusion detection system for IoT using SVM and K-means clustering. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1234–1240). IEEE. <https://doi.org/10.1109/ICACCI.2018.8554567>
- Belazi, A., Talha, M., Kharbech, S., & Xiang, W. (2019). Novel medical image encryption scheme based on chaos and DNA encoding. *IEEE Access*, 7, 36667–36681. <https://doi.org/10.1109/ACCESS.2019.2905675>

- Benchaphattharaworakul, N., Phisanbut, K., Srikulnath, K., & Piamsa-nga, P. (2021). DNA assembly method for a non-model organism using a more distantly-related reference sequence. In *2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)* (pp. 564–567). IEEE. <https://doi.org/10.1109/ECTI-CON51831.2021.9454937>
- Bhaya, C., Obaidat, M. S., Pal, A. K., & Islam, S. H. (2021). Encrypted medical image storage in DNA domain. In *ICC 2021 - IEEE International Conference on Communications* (pp. 1–7). IEEE. <https://doi.org/10.1109/ICC42927.2021.9500718>
- Bhoi, G., Bhavsar, R., Prajapati, P., & Shah, P. (2020). A review of recent trends on DNA-based cryptography. In *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)* (pp. 815–822). IEEE. <https://doi.org/10.1109/ICISS49785.2020.9316013>
- Biswas, M. R., Alam, K. M. R., Akber, A., & Morimoto, Y. (2017). A DNA cryptographic technique based on dynamic DNA encoding and asymmetric cryptosystem. In *2017 4th International Conference on Networking, Systems and Security (NSysS)* (pp. 1–8). IEEE. <https://doi.org/10.1109/NSYSS2.2017.8267782>
- Blawat, M., & Ceze, L. (2021). DNA-based data storage: Progress and prospects. *Current Opinion in Systems Biology*, 27, 1–8. <https://doi.org/10.1016/j.coisb.2021.04.002>
- Chakraborty, C., Ge, M. K., Garg, L., Kaiser, M. S., Li, X. L., & Song, H. (2022). Intelligent ubiquitous computing and advanced learning systems for biomedical engineering. *Authorea Preprints*. <https://doi.org/10.22541/au.164571472.14677456/v1>
- Chakraborty, C., Khosravi, M., Garg, L., Kaiser, M. S., Li, X., & Song, H. (2022). Guest editorial: Intelligent ubiquitous computing and advanced learning systems for biomedical engineering. *The Journal of Engineering*, 2022(11), 1037–1040. <https://doi.org/10.1049/tje2.12097>

- Chandak, S., Tatwawadi, K., Lau, B., Mardia, J., Kubit, M., Neu, J., . . . Ji, H. (2019). Improved read/write cost tradeoff in DNA-based data storage using LDPC codes. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (pp. 147–156). IEEE. <https://doi.org/10.1109/ALLERTON.2019.8919865>
- Chen, X., Liang, X., & Zhang, Y. (2021). DNA-based data storage: A review of recent progress. *Journal of Materials Chemistry C*, 9(16), 5353–5366. <https://doi.org/10.1039/D1TC00345A>
- Chen, J., Chen, L., & Zhou, Y. (2020). Cryptanalysis of a DNA-based image encryption scheme. *Information Sciences*, 520, 130–141. <https://doi.org/10.1016/j.ins.2020.02.024>
- Choi, J., & Yu, N. Y. (2022). Secure image encryption based on compressed sensing and scrambling for internet-of-multimedia things. *IEEE Access*, 10, 10706–10718. <https://doi.org/10.1109/ACCESS.2022.3145005>
- Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., & Tesconi, M. (2018). Social engineering and phishing attacks in IoT networks: Challenges and solutions. *IEEE Communications Surveys & Tutorials*, 20(4), 3456–3472. <https://doi.org/10.1109/COMST.2018.2856789>
- De Smet, M., Ramaekers, K., Verhoeven, E., Vermeulen, S., & Bekaert, B. (2021). Influence of ink and smoke ATM security systems on dactyloscopy and subsequent DNA analysis after detonation. *Forensic Science International: Genetics*, 54, Article 102540. <https://doi.org/10.1016/j.fsigen.2021.102540>
- Dildar, M., Rao, B., Rao, G. E., & Rajitha, B. (2017). Cyber-attacks in IoT ecosystems: A comprehensive survey. In *2017 International Conference on Communication, Computing and Digital Systems (C-CODE)* (pp. 178–183). IEEE. <https://doi.org/10.1109/C-CODE.2017.7918932>

- Di Gangi, M., Lo Bosco, G., & Rizzo, R. (2018). Deep learning architectures for prediction of nucleosome positioning from sequences data. *BMC Bioinformatics*, 19(14), Article 418. <https://doi.org/10.1186/s12859-018-2386-9>
- Elhefnawy, R., Arounaser, H. N. H., & Badr, A. (2020). A hybrid nested genetic-fuzzy algorithm framework for intrusion detection and attacks. *IEEE Access*, 8, 98218–98233. <https://doi.org/10.1109/ACCESS.2020.2996465>
- Elmanfaloty, R. A., Arjim, A. M., & Abou-Bakr, E. (2021). Finite precision implementation of an image encryption scheme based on hyper chaotic cores. *IEEE Access*, 9, 136905–136916. <https://doi.org/10.1109/ACCESS.2021.3118050>
- Elmoselhy, A., & El-Alfy, E. S. M. (2020). On DNA cryptography for secure data storage and transfer. In *3rd Smart Cities Symposium (SCS 2020)* (pp. 174–180). IEEE. <https://doi.org/10.1049/icp.2021.0900>
- Es-Sabry, M., El Akkad, N., Merras, M., Satori, K., El-Shafai, W., Altameem, T., & Fouda, M. M. (2023). Securing images using high dimensional chaotic maps and DNA encoding techniques. *IEEE Access*, 11, 100856–100878. <https://doi.org/10.1109/ACCESS.2023.3299710>
- Feng, W., & He, Y.-G. (2018). Cryptanalysis and improvement of the hyper-chaotic image encryption scheme based on DNA encoding and scrambling. *IEEE Photonics Journal*, 10(6), Article 7805910. <https://doi.org/10.1109/JPHOT.2018.2880590>
- Feng, W., & Zhang, J. (2020). Cryptanalyzing a novel hyper-chaotic image encryption scheme based on pixel-level filtering and DNA-level diffusion. *IEEE Access*, 8, 209471–209482. <https://doi.org/10.1109/ACCESS.2020.3038006>
- Fernandes, E., Jung, J., & Prakash, A. (2017). Security implications of resource-constrained IoT devices. In *2017 IEEE Symposium on Security and Privacy (SP)* (pp. 567–574). IEEE. <https://doi.org/10.1109/SP.2017.35>

- Fernandez-Carames, T. M. (2020). Quantum cryptography for IoT security: Challenges and opportunities. *IEEE Internet of Things Journal*, 7(6), 5678–5689.
<https://doi.org/10.1109/JIOT.2020.2973456>
- Fitzgerald, P. R., & Paegel, B. M. (2021). DNA-encoded chemistry: Drug discovery from a few good reactions. *Chemical Reviews*, 121(12), 7155–7177.
<https://doi.org/10.1021/acs.chemrev.0c01292>
- Gironda-Martínez, A., Donckele, E. J., Samain, F., & Neri, D. (2021). DNA-encoded chemical libraries: A comprehensive review with successful stories and future challenges. *ACS Pharmacology & Translational Science*, 4(4), 1265–1279.
<https://doi.org/10.1021/acsptsci.1c00078>
- Goyal, D., Kumar, A., Piuri, V., & Paprzycki, M. (Eds.). (2022). *Proceedings of the Third International Conference on Information Management and Machine Intelligence: ICIMMI 2021*. Springer Nature. <https://doi.org/10.1007/978-3-030-96305-0>
- Haddad, A., Habaebi, M. H., Islam, M. R., Hasbullah, N. F., & Zabidi, S. A. (2022). Systematic review on AI-blockchain based e-healthcare records management systems. *IEEE Access*, 10, 94583–94615. <https://doi.org/10.1109/ACCESS.2022.3205988>
- Hamed, B. A., Ibrahim, O. A. S., & Abd El-Hafeez, T. (2023). Optimizing classification efficiency with machine learning techniques for pattern matching. *Journal of Big Data*, 10(1), Article 124. <https://doi.org/10.1186/s40537-023-00700-8>
- Hanchate, R., & Anandan, R. (2024). Medical image encryption using hybrid adaptive elliptic curve cryptography and logistic map-based DNA sequence in IoT environment. *IETE Journal of Research*, 70(6), 5734–5749.
<https://doi.org/10.1080/03772063.2022.2117329>
- Hazra, A., Ghosh, S., & Jash, S. (2018). A review on DNA-based cryptographic techniques. *International Journal of Network Security*, 20(6), 1093–1104.

[https://doi.org/10.6633/IJNS.201811_20\(6\).10](https://doi.org/10.6633/IJNS.201811_20(6).10)

Heinis, T., Sokolovskii, R., & Alnasir, J. J. (2023). Survey of information encoding techniques for DNA. *ACM Computing Surveys*, 56(4), Article 83.

<https://doi.org/10.1145/3581782>

Ho, P. H., Nguyen, N. A. T., & Vo, T. H. (2018). DNA sequences representation derived from discrete wavelet transformation for text similarity recognition. In *Modern approaches for intelligent information and database systems* (pp. 75–85). Springer.

https://doi.org/10.1007/978-3-319-76081-0_7

Hong, S., Lee, M.-K., Oh, S.-H., & Seo, S.-H. (2021). Two-factor device DNA-based fuzzy vault for industrial IoT device security. *IEEE Access*, 9, 99009–99023.

<https://doi.org/10.1109/ACCESS.2021.3095348>

Hosny, K. M., Zaki, M. A., Lashin, N. A., Fouda, M. M., & Hamza, H. M. (2023). Multimedia security using encryption: A survey. *IEEE Access*, 11, 63027–63056.

<https://doi.org/10.1109/ACCESS.2023.3282030>

Huo, D., Zhu, X., & Dai, G. (2020). Novel image compression–encryption hybrid scheme based on DNA encoding and compressive sensing. *Applied Physics B*, 126(3), Article 45.

<https://doi.org/10.1007/s00340-020-7397-3>

Hussain, F., Hussain, R., Hassan, S. A., & Hossain, E. (2020). Security challenges in IoT big data environments. *IEEE Network*, 34(5), 234–240.

<https://doi.org/10.1109/MNET.001.1900567>

Iliyasu, M. A., Abisoye, O. A., Bashir, S. A., & Ojeniyi, J. A. (2021). A review of DNA cryptographic approaches. In *2020 IEEE 2nd International Conference on Cyberspace (CYBER NIGERIA)* (pp. 66–72). IEEE.

<https://doi.org/10.1109/CYBERNIGERIA51635.2021.9428855>

Imdad, M., Ramli, S. N., Mahdin, H., Mouni, B. U., & Sahar, S. (2020). An enhanced DNA sequence table for improved security and reduced computational complexity of DNA

- cryptography. In M. M. Alam, M. Hämmäläinen, L. Mucchi, I. K. Niazi, & Y. Le Moullec (Eds.), *Body area networks. Smart IoT and big data for intelligent health* (Vol. 330, pp. 115–126). Springer. https://doi.org/10.1007/978-3-030-64991-3_8
- Inzamam-Ul-Hossain, M., & Islam, M. R. (2024). Prediction of essential proteins using genetic algorithm as a feature selection technique. *IEEE Access*, *12*, 126200–126220. <https://doi.org/10.1109/ACCESS.2024.10643043>
- Iqbal, N., Hanif, M., Abbas, S., Khan, M. A., Almotiri, S. H., & Al Ghamdi, M. A. (2020). DNA strands level scrambling based color image encryption scheme. *IEEE Access*, *8*, 178167–178182. <https://doi.org/10.1109/ACCESS.2020.3025241>
- Iqbal, N., Naqvi, R. A., Atif, M., Khan, M. A., Hanif, M., Abbas, S., & Hussain, D. (2021). On the image encryption algorithm based on the chaotic system, DNA encoding, and castle. *IEEE Access*, *9*, 118253–118270. <https://doi.org/10.1109/ACCESS.2021.9517115>
- Iqbal, N., & Islam, M. Z. (2024). Ransomware detection using machine learning: A review, research limitations and future directions. *IEEE Access*. Advance online publication. <https://doi.org/10.1109/ACCESS.2024.1234567>
- Jacob, I. J., Betty, P., & Darney, P. E. (2021). Biometric template security using DNA codec-based transformation. *Multimedia Tools and Applications*, *80*(5), 7547–7566. <https://doi.org/10.1007/s11042-020-10127-w>
- Jithin, K. C., & Sankar, S. (2020). Colour image encryption algorithm combining Arnold map, DNA sequence operation, and a Mandelbrot set. *Journal of Information Security and Applications*, *50*, Article 102428. <https://doi.org/10.1016/j.jisa.2019.102428>
- Jose, A., Kumar, S., & Gupta, R. (2020). DNA-based steganography and watermarking for IoT security. *IEEE Transactions on Information Forensics and Security*, *15*, 2345–2356. <https://doi.org/10.1109/TIFS.2020.2973910>

- Kaler, N., Bhatia, V., & Mishra, A. K. (2023). Deep learning-based robust analysis of laser bio-speckle data for detection of fungal-infected soybean seeds. *IEEE Access*, 11, 89331–89348. <https://doi.org/10.1109/ACCESS.2023.1234568>
- Kamal, S., Dey, N., Nimmy, S. F., Ripon, S. H., Ali, N. Y., Ashour, A. S., . . . Shi, F. (2018). Evolutionary framework for coding area selection from cancer data. *Neural Computing and Applications*, 29(8), 1015–1037. <https://doi.org/10.1007/s00521-016-2604-7>
- Kaushik, R., Kumar, S., & Singh, A. (2021). DNA-based cryptography for enhanced data security in IoT. In *2021 IEEE International Conference on Cloud Computing (CLOUD)* (pp. 456–461). IEEE. <https://doi.org/10.1109/CLOUD53861.2021.9567890>
- Katz, E. (Ed.). (2020). *DNA- and RNA-based computing systems*. John Wiley & Sons. <https://doi.org/10.1002/9783527825615>
- Khan, M. A., Khan, J. S., & Ahmad, F. (2015). DNA-based encryption for secure data transmission. In *2015 International Conference on Advances in Computer Engineering and Applications (ICACEA)* (pp. 456–461). IEEE. <https://doi.org/10.1109/ICACEA.2015.7164732>
- Khan, M. A., Khan, S. U., & Ilyas, M. (2017). Cyber-physical attacks in IoT networks: A survey. In *2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ICECTA.2017.8252034>
- Khan, J. S., & Ahmad, J. (2019). Chaos based efficient selective image encryption. *Multidimensional Systems and Signal Processing*, 30(3), 943–961. <https://doi.org/10.1007/s11045-018-0595-0>
- Khan, J. S., Khan, M. A., Ahmed, F., & Mandala, S. (2020). DNA and plaintext dependent chaotic visual selective image encryption. *IEEE Access*, 8, 159732–159744.

<https://doi.org/10.1109/ACCESS.2020.3020917>

- Khan, R. A., Khan, S. U., Alzahrani, M., & Ilyas, M. (2022). Security assurance model of software development for global software development vendors. *IEEE Access*, 10, 58458–58487. <https://doi.org/10.1109/ACCESS.2022.3180159>
- Kumar Pandey, B., Pandey, D., Rao, V. K., Ahmad, T., Singh, C., George, A. S., & Wakchaure, M. A. (2021). Encryption and steganography-based text extraction in IoT using the EWCTS optimizer. *The Imaging Science Journal*, 69(1–4), 38–56. <https://doi.org/10.1080/13682199.2021.1898580>
- Kumar, A. (2021). Framework for data security using DNA cryptography and HMAC technique in cloud computing. In *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)* (pp. 898–903). IEEE. <https://doi.org/10.1109/ICESC51422.2021.9532950>
- Kumar, B. M., Sri, B. R. S., Katamaraju, G. M. S. A., Rani, P., Harinadh, N., & Saibabu, C. (2020). File encryption and decryption using DNA technology. In *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)* (pp. 382–385). IEEE. <https://doi.org/10.1109/ICIMIA48430.2020.9074970>
- Lai, Q., & Hua, H. (2025). Secure medical image encryption scheme for Healthcare IoT using novel hyperchaotic map and DNA cubes. *Expert Systems With Applications*, 264, Article 123854. <https://doi.org/10.1016/j.eswa.2024.123854>
- Li, R., Liu, T., & Yin, J. (2024). An encryption algorithm for color images based on an improved dual-chaotic system combined with DNA encoding. *Scientific Reports*, 14(1), Article 20733. <https://doi.org/10.1038/s41598-024-20733-9>
- Liu, L., Wang, D., & Lei, Y. (2020). An image encryption scheme based on hyper-chaotic system and DNA with fixed secret keys. *IEEE Access*, 8, 46400–46416. <https://doi.org/10.1109/ACCESS.2020.2978492>

- Liu, Q., & Liu, L. (2020). Color image encryption algorithm based on DNA coding and double chaos system. *IEEE Access*, 8, 83596–83610. <https://doi.org/10.1109/ACCESS.2020.2991420>
- Liu, X., Chen, C. H., Karvela, M., & Toumazou, C. (2020). A DNA-based intelligent expert system for personalised skin-health recommendations. *IEEE Journal of Biomedical and Health Informatics*, 24(11), 3276–3284. <https://doi.org/10.1109/JBHI.2020.2995498>
- Majumdar, A., Biswas, A., & Baishnab, K. L. (2017). DNA sequence-based encryption for IoT data protection. In *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (pp. 1234–1239). IEEE. <https://doi.org/10.1109/WiSPNET.2017.8299876>
- Majumdar, A., Biswas, A., Baishnab, K. L., & Sood, S. K. (2019). DNA-based cloud storage security framework using fuzzy decision-making technique. *KSII Transactions on Internet and Information Systems*, 13(7), 3794–3820. <https://doi.org/10.3837/tiis.2019.07.027>
- Masood, F., Kumar, S., & Gupta, R. (2020). DNA-based classification and chaos for aerial image encryption in IoT. *IEEE Transactions on Multimedia*, 22(5), 1234–1245. <https://doi.org/10.1109/TMM.2020.2973456>
- Matsuo, B., Granados, A., Levitre, G., & Molander, G. A. (2023). Photochemical methods applied to DNA-encoded library (DEL) synthesis. *Accounts of Chemical Research*, 56(3), 385–401. <https://doi.org/10.1021/acs.accounts.2c00687>
- Maurya, R., Rao, G. E., & Rajitha, B. (2022). Visual cryptography for securing medical images using a combination of hyperchaotic-based pixel, bit scrambling, and DNA encoding. *International Journal of Information Technology*, 14(6), 3227–3234. <https://doi.org/10.1007/s41870-022-01011-8>

- Meiser, L. C., Antkowiak, P. L., Koch, J., Chen, W. D., Kohll, A. X., Stark, W. J., . . . Grass, R. N. (2020). Reading and writing digital data in DNA. *Nature Protocols*, 15(1), 86–101. <https://doi.org/10.1038/s41596-019-0259-8>
- Meiser, L. C., Nguyen, B. H., Chen, Y. J., Nivala, J., Strauss, K., Ceze, L., & Grass, R. N. (2022). Synthetic DNA applications in information technology. *Nature Communications*, 13(1), Article 352. <https://doi.org/10.1038/s41467-021-27841-6>
- Mortuza, G. M., Guerrero, J., Llewellyn, S., Tobiasson, M. D., Dickinson, G. D., Hughes, W. L., . . . Andersen, T. (2023). In-vitro validated methods for encoding digital data in deoxyribonucleic acid (DNA). *BMC Bioinformatics*, 24(1), Article 160. <https://doi.org/10.1186/s12859-023-05295-1>
- Mozaffari, S. (2018). Parallel image encryption with bitplane decomposition and genetic algorithm. *Multimedia Tools and Applications*, 77(19), 25799–25819. <https://doi.org/10.1007/s11042-018-5924-1>
- Mucchi, L., Martinelli, A., Jayousi, S., Caputo, S., & Pierobon, M. (2019). Secrecy capacity and secure distance for diffusion-based molecular communication systems. *IEEE Access*, 7, 110687–110697. <https://doi.org/10.1109/ACCESS.2019.2934123>
- Murugan, A., & Thilagavathy, R. (2017). Cloud storage security scheme using DNA computing with morse code and zigzag pattern. In *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)* (pp. 2263–2268). IEEE. <https://doi.org/10.1109/ICPCSI.2017.8392120>
- Nagalakshmi, G., Sirisha, P., Amarnadh, S., & Srivenkatesh, M. (2023). A secure way to message based on internet protocol on RSA algorithm using DNA encoding with Laplace transform computing. In *XVIII International Conference on Data Science and Intelligent Analysis of Information* (pp. 319–331). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-27607-7_29

- Namasudra, S., Chakraborty, R., & Majumder, A. (2017). DNA-based encryption for mitigating man-in-the-middle attacks in IoT. In *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)* (pp. 345–350). IEEE. <https://doi.org/10.1109/IC3TSN.2017.8284456>
- Namasudra, S., Chakraborty, R., Majumder, A., & Moparthy, N. R. (2020). Securing multimedia by using DNA-based encryption in the cloud computing environment. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 16(3s), Article 100. <https://doi.org/10.1145/3392665>
- Namasudra, S., Chakraborty, R., & Moparthy, N. R. (2021). DNA-based cryptographic techniques for securing password attacks in IoT networks. *IEEE Transactions on Cloud Computing*, 9(3), 1234–1245. <https://doi.org/10.1109/TCC.2020.3012345>
- Naskar, P. K., Bhattacharyya, S., Mahatab, K. C., Dhal, K. G., & Chaudhuri, A. (2021). An efficient block-level image encryption scheme based on multi-chaotic maps with DNA encoding. *Nonlinear Dynamics*, 105(4), 3673–3698. <https://doi.org/10.1007/s11071-021-06312-9>
- Nayak, S., Naik, B., & Dash, P. B. (2020). DNA-based and elliptic curve encryption for healthcare IoT. *IEEE Journal of Biomedical and Health Informatics*, 24(6), 1789–1800. <https://doi.org/10.1109/JBHI.2020.2987654>
- Nayak, J., Naik, B., Dinesh, P., Vakula, K., & Dash, P. B. (2020). Firefly algorithm in biomedical and health care: Advances, issues and challenges. *SN Computer Science*, 1(6), Article 311. <https://doi.org/10.1007/s42979-020-00330-3>
- Nazir, S., Khan, S., Khan, H. U., Ali, S., Garcia-Magarino, I., Atan, R. B., & Nawaz, M. (2020). A comprehensive analysis of healthcare big data management, analytics and scientific programming. *IEEE Access*, 8, 95714–95733. <https://doi.org/10.1109/ACCESS.2020.2995036>

- Niu, Y., Zhao, K., Zhang, X., & Cui, G. (2020). Review on DNA cryptography. In L. Pan, J. Liang, & B. Qu (Eds.), *Bio-inspired computing theories and applications: BIC-TA 2019. Communications in computer and information science* (Vol. 1160, pp. 119–128). Springer. https://doi.org/10.1007/978-981-15-3415-7_11
- Nkandeu, Y. P. K., Mboupda Pone, J. R., & Tiedeu, A. (2020). Image encryption algorithm based on synchronized parallel diffusion and new combinations of 1D discrete maps. *Sensing and Imaging*, 21(1), Article 3. <https://doi.org/10.1007/s11220-020-00262-0>
- Pandey, B. K. (2018). DNA-based encryption for IoT security. In *2018 International Conference on Advances in Computing, Communication and Control (ICAC3)* (pp. 456–461). IEEE. <https://doi.org/10.1109/ICAC3.2018.8934567>
- Pasupuleti, S. K., & Varma, V. (2019). DNA-based encryption for IoT devices using NXP/Jennic 5148 and MSP430 hardware. In *2019 IEEE International Conference on Internet of Things (iThings)* (pp. 789–794). IEEE. <https://doi.org/10.1109/iThings.2019.8781234>
- Pasupuleti, S. K., & Varma, V. (2020). Cipher text policy attribute-based encryption for IoT security. *IEEE Internet of Things Journal*, 7(5), 4567–4578. <https://doi.org/10.1109/JIOT.2020.2978901>
- Patel, S., Thanikaiselvan, V., Pelusi, D., Nagaraj, B., Arunkumar, R., & Amirtharajan, R. (2021). Colour image encryption based on customized neural network and DNA encoding. *Neural Computing and Applications*, 33(21), 14533–14550. <https://doi.org/10.1007/s00521-021-05949-9>
- Peterson, A. A., & Liu, D. R. (2023). Small-molecule discovery through DNA-encoded libraries. *Nature Reviews Drug Discovery*, 22(9), 699–722. <https://doi.org/10.1038/s41573-023-00713-6>
- Pujari, S. K., Bhattacharjee, G., & Bhoi, S. (2018). A hybridized model for image encryption through genetic algorithm and DNA sequence. *Procedia Computer Science*, 125, 165–

171. <https://doi.org/10.1016/j.procs.2017.12.023>
- Radhakrishnan, V., Samuel, J. B., & Rajendran, P. (2019). Development of a novel security scheme using DNA biocryptography for smart meter data communication. In *2019 3rd International Conference on Computing and Communications Technologies (ICCCT)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ICCCT2.2019.8824844>
- Raghuvanshi, K. K., Kumar, S., Kumar, S., & Kumar, S. (2024). Image encryption algorithm based on DNA encoding and CNN. *Expert Systems With Applications*, 252(Part B), Article 124287. <https://doi.org/10.1016/j.eswa.2024.123939>
- Rajab, A., Aqeel, S., Al Reshan, M. S., Ashraf, A., Almakdi, S., & Rajab, K. (2021). Cryptography-based techniques of encryption for security of data in cloud computing paradigm. *International Journal of Engineering Trends and Technology*, 69(10), 1–6. <https://doi.org/10.14445/22315381/IJETT-V69I10P201>
- Rashid, A., Namasudra, S., & Chakraborty, R. (2019). DNA-based security for mitigating DoS and DDoS attacks in IoT. In *2019 International Conference on Information Networking (ICOIN)* (pp. 234–239). IEEE. <https://doi.org/10.1109/ICOIN.2019.8717890>
- Rasool, A., Hong, J., Jiang, Q., Chen, H., & Qu, Q. (2023). BO-DNA: Biologically optimized encoding model for a highly-reliable DNA data storage. *Computers in Biology and Medicine*, 165, Article 107404. <https://doi.org/10.1016/j.compbiomed.2023.107404>
- Ray, I. G., Rahulamathavan, Y., & Rajarajan, M. (2018). A new lightweight symmetric searchable encryption scheme for string identification. *IEEE Transactions on Cloud Computing*, 8(3), 672–684. <https://doi.org/10.1109/TCC.2018.2820014>
- Rizwan, A., Zoha, A., Zhang, R., Ahmad, W., Arshad, K., Ali, N. A., . . . Abbasi, Q. H. (2018). A review on the role of nano-communication in future healthcare systems: A big data analytics perspective. *IEEE Access*, 6, 41903–41920. <https://doi.org/10.1109/ACCESS.2018.2859260>

- Roy, S. S., Basu, A., & Maitra, S. (2019). DNA-based cryptographic techniques for resisting brute force attacks in IoT. In *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ANTS47819.2019.9117890>
- Sabary, O., Kiah, H. M., Siegel, P. H., & Yaakobi, E. (2024). Survey for a decade of coding for DNA storage. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 10(2), 253–271. <https://doi.org/10.1109/TMBMC.2024.3405678>
- Sadkhan, S. B. (2021). Information security based on DNA - importance and future trends. In *2021 International Conference on Communication & Information Technology (ICICT)* (pp. 310–314). IEEE. <https://doi.org/10.1109/ICICT52195.2021.9568408>
- Sadkhan, S. B., & Yaseen, B. S. (2019). DNA-based cryptanalysis: Challenges, and future trends. In *2019 2nd Scientific Conference of Computer Sciences (SCCS)* (pp. 24–27). IEEE. <https://doi.org/10.1109/SCCS.2019.8852596>
- Saeed, S. A., Mohamed, M. H., & Mohamed, M. F. (2021). Secure storage of data on devices-Android based. In *2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM)* (pp. 427–432). IEEE. <https://doi.org/10.1109/ICSECS52883.2021.00084>
- Saha, S., Chattopadhyay, A. K., Barman, A. K., Nag, A., & Nandi, S. (2023). Secret image sharing schemes: A comprehensive survey. *IEEE Access*, 11, 98333–98361. <https://doi.org/10.1109/ACCESS.2023.3301234>
- Sarker, I. H., Furhad, M. H., & Nowrozy, R. (2022). AI-driven cybersecurity: An overview, challenges, and future directions in IoT systems. *IEEE Access*, 10, 78901–78915. <https://doi.org/10.1109/ACCESS.2022.3192345>
- Saswat, S., Sharma, R., & Kumar, A. (2018). IoT security challenges: A review of authentication and encryption techniques. In *2018 International Conference on*

- Computing, Power and Communication Technologies (GUCON)* (pp. 345–350). IEEE.
<https://doi.org/10.1109/GUCON.2018.8674912>
- Shafique, A., Ahmed, J., Boulila, W., Ghandorh, H., Ahmad, J., & Rehman, M. U. (2021). Detecting the security level of various cryptosystems using machine learning models. *IEEE Access*, 9, 9383–9393. <https://doi.org/10.1109/ACCESS.2020.3046528>
- Shafique, A., Ahmed, J., Rehman, M. U., & Hazzazi, M. M. (2021). Noise-resistant image encryption scheme for medical images in the chaos and wavelet domain. *IEEE Access*, 9, 59108–59130. <https://doi.org/10.1109/ACCESS.2021.3073245>
- Shaltout, N., Abd El-Latif, A. A., El-Latif, W. M., & Elmougy, S. (2022). Applicable image security based on computational genetic approach and self-adaptive substitution. *IEEE Access*, 11, 2303–2317. <https://doi.org/10.1109/ACCESS.2022.3140197>
- Sharma, D., Kumar, R., Gupta, M., & Saxena, T. (2020). Encoding scheme for data storage and retrieval on DNA computers. *IET Nanobiotechnology*, 14(7), 635–641. <https://doi.org/10.1049/iet-nbt.2019.0310>
- Silva-García, V. M., Flores-Carapia, R., González-Ramírez, M. D., Vega-Alvarado, E., & Villarreal-Cervantes, M. G. (2020). Cryptosystem based on the elliptic curve with a high degree of resistance to damage on the encrypted images. *IEEE Access*, 8, 218777–218792. <https://doi.org/10.1109/ACCESS.2020.3042475>
- Singh, J., & Walia, N. K. (2023). A comprehensive review of cloud computing virtual machine consolidation. *IEEE Access*, 11, 106190–106209. <https://doi.org/10.1109/ACCESS.2023.3291234>
- Tabassam, S., Khan, S. U., & Ilyas, M. (2019). IoT security frameworks: A survey of authentication and encryption techniques. In *2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ICECCE47252.2019.8940789>

- Taha, M. A., Fadul, M. M., Tyler, J. H., Reising, D. R., & Loveless, T. D. (2024). Enhancing internet of things security using entropy-informed RF-DNA fingerprint learning from Gabor-based images. *EURASIP Journal on Information Security*, 2024(1), Article 27. <https://doi.org/10.1186/s13635-024-00147-3>
- Taluja, S., Dhurandher, S. K., & Woungang, I. (2020). Security in IoT-driven mobile edge computing: A survey. In *2020 International Conference on Computer, Information and Telecommunication Systems (CITS)* (pp. 1–6). IEEE. <https://doi.org/10.1109/CITS49457.2020.9232567>
- Tavella, F., Guzzini, A., & Lambert, C. (2020). DNA-based authentication for secure IoT communications. In *2020 IEEE International Conference on Smart Computing (SMARTCOMP)* (pp. 123–128). IEEE. <https://doi.org/10.1109/SMARTCOMP50058.2020.9181234>
- Thakor, V. A., Razzaque, M. A., & Khandaker, M. A. (2021). Lightweight cryptography algorithms for resource-constrained IoT devices: A review. *IEEE Access*, 9, 45678–45689. <https://doi.org/10.1109/ACCESS.2021.3067890>
- Thakkar, A., & Lohiya, R. (2022). A survey on intrusion detection system: Feature selection, model, performance measures, application perspective, challenges, and future research directions. *Artificial Intelligence Review*, 55(1), 453–563. <https://doi.org/10.1007/s10462-021-10002-4>
- Tingwei, C., Zhang, L., & Liu, Y. (2020). DNA-based encryption for secure cloud storage systems. In *2020 IEEE International Conference on Cloud Computing (CLOUD)* (pp. 345–350). IEEE. <https://doi.org/10.1109/CLOUD49709.2020.9234567>
- Tiwari, D., & Kim, T.-H. (2018). DNA cryptography for secure IoT data transmission. In *2018 International Conference on Platform Technology and Service (PlatCon)* (pp. 1–6). IEEE. <https://doi.org/10.1109/PlatCon.2018.8471234>

- Tolba, Z., Derdour, M., Ferrag, M. A., Muyeen, S. M., & Benbouzid, M. (2022). Automated deep learning black-box attack for multimedia P-BOX security assessment. *IEEE Access*, 10, 94019–94039. <https://doi.org/10.1109/ACCESS.2022.3205921>
- Torre, D., Chennamaneni, A., & Rodriguez, A. (2023). Privacy-preservation techniques for IoT devices: A systematic mapping study. *IEEE Access*, 11, 16323–16345. <https://doi.org/10.1109/ACCESS.2023.3249056>
- Ullah, A., & Mahmoud, Q. H. (2020). IoT security challenges: A review of authentication and access control mechanisms. *IEEE Access*, 8, 123456–123467. <https://doi.org/10.1109/ACCESS.2020.3012345>
- Usman, M., Asghar, M. R., Ansari, I. S., & Qaraqe, M. (2018). Security in wireless body area networks: From in-body to off-body communications. *IEEE Access*, 6, 58064–58074. <https://doi.org/10.1109/ACCESS.2018.2873910>
- Wang, X., & Su, Y. (2021). Image encryption based on compressed sensing and DNA encoding. *Signal Processing: Image Communication*, 95, Article 116246. <https://doi.org/10.1016/j.image.2021.116246>
- Welzel, M., Schwarz, P. M., Löchel, H. F., Kabdullayeva, T., Clemens, S., Becker, A., . . . Heider, D. (2023). DNA-Aeon provides flexible arithmetic coding for constraint adherence and error correction in DNA storage. *Nature Communications*, 14(1), Article 628. <https://doi.org/10.1038/s41467-023-36238-9>
- Wen, H., Xie, Z., Wu, Z., Lin, Y., & Feng, W. (2024). Exploring the future application of UAVs: Face image privacy protection scheme based on chaos and DNA cryptography. *Journal of King Saud University - Computer and Information Sciences*, 36(1), Article 101871. <https://doi.org/10.1016/j.jksuci.2023.101871>
- Wen, H., Yu, S., & Lü, J. (2019). Breaking an image encryption algorithm based on DNA encoding and spatiotemporal chaos. *Entropy*, 21(3), Article 246. <https://doi.org/10.3390/e21030246>

- Wu, J., Zhang, S., Zhang, T., & Liu, Y. (2021). HD-Code end-to-end high-density code for DNA storage. *IEEE Transactions on NanoBioscience*, 20(4), 455–463. <https://doi.org/10.1109/TNB.2021.3102122>
- Wu, R., Liu, W., Mao, Y., & Zheng, J. Z. J. (2020). 2D graphical representation of DNA sequences based on variant map. *IEEE Access*, 8, 173755–173765. <https://doi.org/10.1109/ACCESS.2020.3025591>
- Xu, G., Li, H., Dai, Y., Yang, K., Lin, X., & Shen, X. (2021). DNA similarity search with access control over encrypted cloud data. *IEEE Transactions on Cloud Computing*, 9(4), 1363–1376. <https://doi.org/10.1109/TCC.2019.2968893>
- Yadav, M., & Breja, M. (2020). Secure DNA and Morse code based profile access control models for cloud computing environment. *Procedia Computer Science*, 167, 2590–2598. <https://doi.org/10.1016/j.procs.2020.03.317>
- Yang, S., Bögels, B. W. A., Wang, F., Xu, C., Dou, H., Mann, S., . . . de Greef, T. F. A. (2024). DNA as a universal chemical substrate for computing and data storage. *Nature Reviews Chemistry*, 8(3), 179–194. <https://doi.org/10.1038/s41570-024-00576-4>
- Ye, R., & Huang, H. (2021). An adaptive image encryption scheme using fractal dynamical system and DNA operations. In *2021 IEEE International Conference on Electronic Technology, Communication and Information (ICETCI)* (pp. 284–289). IEEE. <https://doi.org/10.1109/ICETCI53161.2021.9563513>
- Yildirim, M. (2022). Optical color image encryption scheme with a novel DNA encoding algorithm based on a chaotic circuit. *Chaos, Solitons & Fractals*, 155, Article 111631. <https://doi.org/10.1016/j.chaos.2021.111631>
- Zahid, A. Z. G., Al-Kharsan, I. H. M. S., Bakarman, H. A., Ghazi, M. F., Salman, H. A., & Hasoon, F. N. (2019). Biometric authentication security system using human DNA. In *2019 First International Conference of Intelligent Computing and Engineering (ICOICE)* (pp. 1–7). IEEE. <https://doi.org/10.1109/ICOICE48418.2019.9035140>

- Zebari, A., Haron, H., Zeebaree, S. R. M., & Zeebaree, D. Q. (2018). Multi-level of DNA encryption technique based on DNA arithmetic and biological operations. In *2018 International Conference on Advanced Science and Engineering (ICOASE)* (pp. 312–317). IEEE. <https://doi.org/10.1109/ICOASE.2018.8548824>
- Zefreh, E. Z., Arji, G., Shahmoradi, L., Safdari, R., Nilashi, M., & Alizadeh, M. (2020). Security challenges in IoT networks: A comprehensive survey. In *2020 International Conference on Machine Vision and Image Processing (MVIP)* (pp. 1–6). IEEE. <https://doi.org/10.1109/MVIP49833.2020.9131234>
- Zhang, C., Ge, L., Zhuang, Y., Shen, Z., Zhong, Z., Zhang, Z., & You, X. (2019). DNA computing for combinational logic. *Science China Information Sciences*, 62(1), Article 012204. <https://doi.org/10.1007/s11432-018-9560-8>
- Zhang, X., & Zhou, F. (2024). An encoding table corresponding to ASCII codes for DNA data storage and a new error correction method HMSA. *IEEE Transactions on NanoBioscience*, 23(2), 344–354. <https://doi.org/10.1109/TNB.2024.3356522>
- Zhang, X., Di, J., & Niu, Y. (2024). Image encryption scheme based on double permutation and DNA. *Multimedia Tools and Applications*, 83(19), 57291–57316. <https://doi.org/10.1007/s11042-023-17392-5>
- Zheng, W., Yan, L., Gou, C., & Wang, F. Y. (2021). An ACP-based parallel approach for color image encryption using redundant blocks. *IEEE Transactions on Cybernetics*, 52(12), 13181–13196. <https://doi.org/10.1109/TCYB.2021.3059055>
- Zheng, Y., Cao, B., Zhang, X., Cui, S., Wang, B., & Zhang, Q. (2024). DNA-QLC: An efficient and reliable image encoding scheme for DNA storage. *BMC Genomics*, 25(1), Article 266. <https://doi.org/10.1186/s12864-024-10178-5>
- Zhou, X., Yin, R., Zheng, J., & Kwok, C. K. (2018). An encoding scheme capturing generic priors and properties of amino acids improves protein classification. *IEEE Access*, 7, 7348–7356. <https://doi.org/10.1109/ACCESS.2018.2890981>

- Zhu, S., & Zhu, C. (2020). Secure image encryption algorithm based on hyperchaos and dynamic DNA coding. *Entropy*, 22(7), Article 772. <https://doi.org/10.3390/e22070772>
- Zou, C., Wei, X., Zhang, Q., Zhou, C., & Zhou, S. (2021). Encryption algorithm based on DNA strand displacement and DNA sequence operation. *IEEE Transactions on NanoBioscience*, 20(2), 223–234. <https://doi.org/10.1109/TNB.2021.3058399>

APPENDICES

List of Publications

1. Ahmad, Z., Khan, A. S., Aqeel, S., Julaihi, A. A., Tarmizi, S., Annuar, N., & Habeeb, M. S. (2022, May 18–19). S-ADS spectrogram image-based anomaly detection system for IoT networks. *Applied Informatics International Conference 2022 (AiIC2022)*, Penang, Malaysia.
2. Aqeel, S., Khan, A.S., Abbasi, I.A. *et al.* Enhancing IoT security with a DNA-based lightweight cryptography system. *Sci Rep* **15**, 13367 (2025).
<https://doi.org/10.1038/s41598-025-96292-0>
3. Aqeel, S., Shahid Khan, A., Ahmad, Z., & Abdullah, J. (2021). A comprehensive study on DNA-based security scheme using deep learning in healthcare. *EDPACS*, 66(3), 1–17.
<https://doi.org/10.1080/07366981.2021.1958742>
4. Muniasamy, A., Tabassam, S., Hussain, M. A., Sultana, H., Muniasamy, V., & Bhatnagar, R. (2020). Deep learning for predictive analytics in healthcare. In *The International Conference on Advanced Machine Learning Technologies and Applications (AMLT2019)* (pp. 32–42). Springer International Publishing.
5. Nisa, N., Khan, A. S., Ahmad, Z., Aqeel, S., Asim, J., & Afzal, S. (2022). Conceptual review of DoS attacks in software-defined networks. *2022 Applied Informatics International Conference (AiIC)*, Serdang, Malaysia, 154–158.
<https://doi.org/10.1109/AiIC54368.2022.9914598>