

GROUP ID: GA6

A PROJECT REPORT

QUANTUM-RESISTANT PASSWORD GENERATION USING QUANTUM NEURAL NETWORKS

SUBMITTED TO THE PIMPRI CHINCHWAD COLLEGE OF ENGINEERING
AN AUTONOMOUS INSTITUTE, PUNE
IN THE FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

B. TECH. (COMPUTER ENGINEERING)

SUBMITTED BY

AMOGH CHANDRAGIRI
ARYAN BAHETI
AKSHAY CHAUDHARI
UTKARSHA LATE

PRN: 121B1B029
PRN: 121B1B030
PRN: 121B1B033
PRN: 122B2B291

**UNDER THE GUIDANCE OF
PROF. DR. S. V. SHINDE**



DEPARTMENT OF COMPUTER ENGINEERING

PCET'S PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

Sector No. 26, Pradhikaran, Nigdi, Pimpri-Chinchwad, PUNE 411044

2025-2026



CERTIFICATE

This is to certify that the project report entitled

“QUANTUM-RESISTANT PASSWORD GENERATION USING QUANTUM NEURAL NETWORKS”

Submitted by

AMOGH CHANDRAGIRI
ARYAN BAHETI
AKSHAY CHAUDHARI
UTKARSHA LATE

PRN: 121B1B029
PRN: 121B1B030
PRN: 121B1B033
PRN: 122B2B291

are bonafide students of this institute and the work has been carried out by them under the supervision of **Prof. Dr. S. V. Shinde** and it is approved for the partial fulfillment of the requirement of Pimpri Chinchwad College of Engineering, an autonomous institute, for the award of the B. Tech. degree in Computer Engineering.

Prof. Dr. S.V. Shinde

(Project Guide)

Department of Computer Engineering

Prof. Dr. Sonali Patil

(Head of the Department)

Department of Computer Engineering

Prof. Dr. G.N. Kulkarni

Director,

Pimpri Chinchwad College of Engineering Pune – 44

Place: Pune

Date: 15/4/25

ACKNOWLEDGEMENT

We express our sincere thanks to our **Guide Prof. Dr. S. V. Shinde** for his/her constant encouragement and support throughout our project, especially for the useful suggestions given during the course of the project and having laid down the foundation for the success of this work.

We would also like to thank our **Research & Innovation coordinator Prof. Dr. Reena Kharat, Project Coordinator Prof. Sushma R. Vispute** for her assistance, genuine support and guidance from early stages of the project. We would like to thank **Prof. Dr. Sonali Patil, Head of Computer Department**, for her unwavering support during the entire course of this project work. We are very grateful to our **Director, Prof. Dr. G.N. Kulkarni**, for providing us with an environment to complete our project successfully. We also thank all the staff members of our college and technicians for their help in making this project a success.

We also thank all the web committees for enriching us with their immense knowledge. Finally, we take this opportunity to extend our deep appreciation to our family and friends, for all that they meant to us during the crucial times of the completion of our project.

NAME OF THE STUDENTS

SIGN

AMOGH CHANDRAGIRI

ARYAN BAHETI

AKSHAY CHAUDHARI

UTKARSHA LATE

ABSTRACT

Passwords are a fundamental aspect of digital security, but traditional password systems are increasingly vulnerable due to evolving cyber threats, including those posed by quantum computing. The rise of quantum algorithms like **Shor's and Grover's** poses a significant risk to conventional password protection mechanisms, as they drastically reduce the time needed for brute-force and cryptographic attacks.

This project presents a **Quantum-Resistant Password Generation System using Quantum Neural Networks (QNNs)** to mitigate these risks. Our system leverages **Quantum Random Number Generators (QRNGs)** to introduce truly random entropy, ensuring unpredictability in password creation. Additionally, we employ **Von Neumann entropy** as a robust metric for analyzing and enhancing security. By training a **Quantum Neural Network (QNN)** on large real-world password datasets, including RockYou, Have I Been Pwned, and CrackStation, we ensure that generated passwords avoid common weaknesses found in human-created passwords. Our experimental evaluation demonstrates that **QNN-generated passwords achieve significantly higher entropy** than conventional methods, making them highly resistant to both classical and quantum attacks. Furthermore, our approach provides **a scalable, efficient, and user-friendly solution** for organizations seeking to future-proof their authentication mechanisms. This research highlights how quantum computing, often viewed as a cybersecurity threat, can also be harnessed as a powerful tool to **enhance password security in the post-quantum era**.

KEYWORDS: Password Security, Quantum Computing, Quantum Neural Networks, Post-Quantum Cryptography, Von Neumann Entropy, Quantum Random Number Generators.

TABLE OF CONTENTS

Sr. No.	Title of Chapter	Page No.
01	Introduction	1
1.1	Quantum Computing & Password Security Overview	2
1.2	Motivation in the Post-Quantum Landscape	2
1.3	Problem statement and Objectives	2
1.4	Scope of the work	3
1.5	Quantum Methodologies for Problem Solving	3
02	Literature Survey	5
2.1	Review of Recent Quantum-Safe Research	6
2.2	Gap Identification in Post-Quantum Security	6
03	Software Requirements Specification	9
3.1	Functional Requirements	10
3.1.1	Quantum Password Generation Pipeline	10
3.1.2	Quantum Validation & Security Checks	10
3.2	External Interface Requirements	11
3.2.1	User Interfaces	11
3.2.2	Hardware Interfaces	11
3.2.3	Software Interfaces	11
3.2.4	Communication Interfaces	12
3.3	Nonfunctional Requirements	12
3.3.1	Performance Requirements	12
3.3.2	Safety / Security Requirements	12
3.4	System Requirements	12
3.4.1	Database Requirements	12
3.4.2	Software Requirements (Platform Choice)	13
3.4.3	Hardware Requirements	13
3.5	Agile SDLC Model Justification in Quantum Password Generation Pipeline	13
04	Project Plan	16
4.1	Project Cost Estimation	17
4.1.1	Computational Costs <ul style="list-style-type: none"> Processing Power: Memory Usage: Storage Requirements: Network Latency and Bandwidth: 	17
4.1.2	Software Performance Costs <ul style="list-style-type: none"> Algorithm Complexity: Database Query Performance: Cloud Service Performance: 	17
4.2	Sustainability Assessment	18

	4.2.1	Environmental Sustainability <ul style="list-style-type: none"> • Energy-Efficient Cryptography • Reduced Carbon Footprint in Data Centers: • Eco-Friendly Simulations: • Sustainable Computing: 	18
	4.2.2	Economic Sustainability <ul style="list-style-type: none"> • Cost Efficiency: • Resource Utilization: • Scalability: 	18
	4.2.3	Social Sustainability <ul style="list-style-type: none"> • Accessibility: • Ethical Considerations: • Open Source Contribution: • Skill Development: 	19
4.3		Complexity Assessment	19
	4.3.1	Computational Complexity <ul style="list-style-type: none"> • Model Training Time: • Memory Usage: • Big-O Notation for Model Training: 	19
	4.3.2	Algorithmic Complexity	20
	4.3.3	Implementation Complexity <ul style="list-style-type: none"> • Lines of Code: • Number of Dependencies: • Integration Complexity: • Code Modularity: 	20
	4.3.4	Resource Complexity <ul style="list-style-type: none"> • Hardware Requirements: • Cloud Infrastructure: • Storage Requirements: • Scalability: 	21
4.4		Risk Management	21
	4.4.1	Risk Identification	22
	4.4.2	Risk Analysis	22
	4.4.3	Overview of Risk Mitigation, Monitoring, Management	23
4.5		Project Schedule	24
	4.5.1	Project Task Set	24
	4.5.2	Timeline Chart	25
4.6		Team Organization (Structure)	25
05		System Design	27
	5.1	Proposed System Architecture/Block Diagram	28
	5.2	Dataset/ Database design	29
	5.3	Mathematical Model	30
	5.4	UML Diagrams	30
06		Project Implementation	36
	6.1	Overview of Quantum Modules	37

	6.2	Tools and Technologies Used	37
	6.3	Algorithmic Details of Quantum Modules	38
	6.3.1	Quantum Random Password Generator	38
	6.3.2	Quantum Neural Network	39
	6.3.3	Quantum Key Distribution	39
07		Software Testing	41
	7.1	Types of Testing	42
	7.2	Test Cases & Results for the Quantum System	43
08		Results	45
	8.1	Outcomes of Quantum Password Generation	46
	8.2	Result analysis and validations	46
	8.2	Quantum UI Screenshots	47
09		Conclusions	49
	9.1	Conclusions	50
	9.2	Future Work	50
	9.3	Applications	51
		Appendix A: Details of paper publication: name of the conference/journal, comments of reviewers, certificate, and paper. Online published paper and certificates of participation in conference if any otherwise draft paper. Or Patent filed documents , Project event participation certificates if any Appendix B: Plagiarism Report of project report.	52
		References <in IEEE format> Thomas Noltey, Hans Hanssomy, Lucia Lo Belloz,"Communication Buses for Automotive Applications" In <i>Proceedings of the 3rd Information Survivability Workshop (ISW-2007)</i> , Boston, Massachusetts, USA, October 2007. IEEE Computer Society.	57

LIST OF ABBREVIATIONS

ABBREVIATION	ILLUSTRATION
QRNG	Quantum Random Number Generator
QNN	Quantum Neural Network
QKD	Quantum Key Distribution
PQC	Post-Quantum Cryptography
CNOT	Controlled NOT Gate
HIBP	Have I Been Pwned Dataset
SHA	Secure Hashing Algorithm
CNN	Classical Neural Network

LIST OF FIGURES

FIGURE	ILLUSTRATION	PAGE NO.
3.5	SDLC Model Diagram	15
5.1	Architecture Diagram	28
5.2	Database Diagram	29
5.4.1	Use Case Diagram	33
5.4.2	Component Diagram	33
5.4.3	Class Diagram	34
5.4.4	Sequence Diagram	34
5.4.5	Activity diagram	35
8.3.1	Taking Input from User	47
8.3.2	Generated Output Values	48

LIST OF TABLES

TABLE	ILLUSTRATION	PAGE NO.
4.5.1	Project Task Set	24
4.5.2	Timeline chart	25
4.6	Team Organization	26
7.2.1	Unit Testing Results	43
7.2.2	Integration Test Results	44

CHAPTER 1: INTRODUCTION

1.1 OVERVIEW

Passwords have long been a fundamental component of digital authentication, serving as the primary means of securing user data and access to online systems. However, conventional password-based security mechanisms suffer from significant vulnerabilities due to weak password choices and the increasing efficiency of computational attacks. As technology advances, attackers are leveraging more sophisticated techniques, making traditional password security inadequate against modern threats.

With the advent of quantum computing, these challenges become even more severe. Quantum algorithms such as **Shor's Algorithm** can break widely used encryption standards, while **Grover's Algorithm** significantly reduces the complexity of brute-force attacks. This paradigm shift necessitates the development of novel password generation methods that can withstand quantum-based threats. Our proposed system integrates **Quantum Neural Networks (QNNs)** and **Quantum Random Number Generators (QRNGs)** to generate highly secure passwords resistant to both classical and quantum attacks

1.2 MOTIVATION

The growing reliance on digital authentication for financial transactions, personal data protection, and enterprise security underscores the necessity of robust password security mechanisms. As quantum computing continues to progress, the limitations of traditional cryptographic methods become evident. Standard encryption techniques may soon become obsolete, leaving sensitive information vulnerable to attacks. Our motivation stems from the need to develop a **forward-looking approach** that not only addresses current security challenges but also **anticipates future threats** posed by quantum computing. By integrating quantum neural networks with quantum randomness, our system offers a **sustainable, scalable, and quantum-resistant solution** to password generation, ensuring that digital security infrastructures remain resilient in the post-quantum era.

1.3 PROBLEM STATEMENT AND OBJECTIVES

As quantum computing advances, traditional password security mechanisms are becoming increasingly vulnerable. Classical brute-force and dictionary attacks already pose significant threats, and the introduction of quantum computing accelerates the breakdown of conventional encryption techniques. Current password generation strategies fail to account for quantum-enabled threats, making existing systems highly susceptible. Our project aims to develop a **Quantum Neural Network-based Password Generator** that harnesses the power of quantum

randomness and machine learning to create strong, unpredictable passwords resistant to quantum attacks.

OBJECTIVES:

1. Develop a **Quantum Neural Network (QNN)** capable of generating highly secure passwords.
2. Integrate **Quantum Random Number Generators (QRNGs)** to enhance password unpredictability.
3. Implement **Von Neumann entropy analysis** to measure and validate password security.
4. Train the QNN using real-world password datasets to avoid common weaknesses.
5. Ensure the generated passwords exhibit resistance to both classical and quantum attacks.
6. Evaluate password robustness by analyzing entropy, collision rates, and cracking resistance.
7. Provide a scalable and efficient framework that can be adopted in cybersecurity applications.

1.4 SCOPE OF THE WORK

The scope of this project extends to designing, developing, and implementing a secure password generation system that can resist quantum-based attacks. By leveraging **Quantum Neural Networks (QNNs)** and **Quantum Random Number Generators (QRNGs)**, the system ensures the creation of highly unpredictable and robust passwords. The project also explores **Von Neumann entropy** as a key security measure, ensuring that the passwords generated have the necessary entropy to withstand modern cyber threats.

Furthermore, this work is aimed at enhancing the usability and practicality of post-quantum security techniques in real-world applications. The generated passwords can be seamlessly integrated into **enterprise security systems, online authentication mechanisms, and financial institutions**, thereby offering a scalable and adaptable solution to the cybersecurity challenges posed by quantum computing advancements.

1.5 METHODOLOGIES OF PROBLEM SOLVING

1. **Requirement Analysis:** This step involves understanding the security threats posed by quantum computing. We identify the limitations of classical password security methods and define the need for quantum-resistant password generation.

2. **Data Collection and Preprocessing:** Real-world password datasets such as RockYou, Have I Been Pwned (HIBP), and CrackStation are gathered. These datasets help in training our Quantum Neural Network (QNN). The collected data is cleaned, formatted, and preprocessed to remove duplicates, weak passwords, and anomalies.
3. **Quantum Neural Network (QNN) Development:** A Quantum Neural Network model is designed to generate highly secure passwords. The network is trained using quantum-inspired algorithms to recognize password patterns. By leveraging quantum principles such as superposition and entanglement, the system enhances the randomness of password generation.
4. **Quantum Random Number Generator (QRNG) Integration:** A QRNG is incorporated to ensure the highest level of entropy in password generation. Unlike classical random number generators, QRNGs derive randomness from quantum mechanical properties, making passwords highly unpredictable and resistant to brute-force attacks.
5. **Security Evaluation:** The security of generated passwords is analyzed using Von Neumann entropy calculations. Entropy measurement ensures that the passwords exhibit strong randomness and are free from predictable patterns.
6. **Testing and Validation:** The generated passwords undergo rigorous testing against known attack strategies. Resistance to brute-force, dictionary attacks, and quantum-based decryption methods is evaluated. The validation process includes comparing the entropy levels of quantum-generated passwords with traditional methods.
7. **Deployment and Optimization:** The final system is optimized for real-world use by improving efficiency and scalability. The integration of QNN-based password generation into existing authentication frameworks is tested. Further refinements are made to ensure seamless adoption in cybersecurity applications and enterprise security systems.

CHAPTER 2: LITERATURE SURVEY

2.1 REVIEW OF RECENT LITERATURE

The field of password security has undergone significant transformation with the rise of quantum computing. Researchers have explored various approaches to enhance password resilience, focusing on **post-quantum cryptography (PQC)**, **quantum random number generation (QRNG)**, and **quantum neural networks (QNNs)**. This section reviews key studies and advancements in these domains.

Quantum Threats to Password Security

Shor's (1994) and Grover's (1996) algorithms have demonstrated that quantum computers can efficiently break traditional encryption schemes and weaken password security. Shor's algorithm allows for fast factorization, making RSA and ECC-based authentication systems obsolete in a quantum era. Similarly, Grover's algorithm enables **$O(\sqrt{N})$ speedup in brute-force attacks**, effectively halving the bit strength of passwords. These findings have pushed researchers to develop quantum-resistant alternatives.

Post-Quantum Cryptographic Approaches

The **NIST Post-Quantum Cryptography Standardization Initiative** has evaluated multiple encryption schemes, such as **lattice-based (CRYSTALS-Kyber, Dilithium)** and **code-based (McEliece)** cryptography, as potential replacements for RSA and ECC. However, these methods primarily address public-key encryption rather than **password security**, leaving a gap in protection for user authentication systems.

Quantum Random Number Generation (QRNG)

Randomness plays a crucial role in password generation. Traditional **pseudo-random number generators (PRNGs)** rely on classical computational methods, which can be predicted if the seed state is known. Researchers like **Ma & Luo (2018)** have demonstrated that **QRNGs**, based on quantum superposition and measurement, produce **true randomness** that is not reproducible by classical methods. This ensures stronger password unpredictability.

Quantum Neural Networks (QNNs) in Cryptography

The integration of **neural networks and quantum computing** has gained interest in cryptographic applications. Studies like **Hebblewhite & Dahlberg (2019)** and **He et al. (2022)**

have explored **QNNs** for secure key distribution and authentication. QNNs leverage **quantum gates and entanglement** to generate more complex patterns than traditional deep learning models. While these studies focus on secure communication, **their application in password generation remains underexplored**.

Quantum Key Distribution (QKD) for Secure Password Sharing

QKD protocols such as **BB84, E91, and B92** enable secure key exchanges, preventing eavesdropping through quantum principles. Research by **Gisin et al. (2002)** and **Lo et al. (2014)** has shown that QKD can be used for password distribution in multi-party authentication environments. However, integrating QKD with **end-user password security** remains a challenge due to cost and infrastructure limitations.

2.2 GAP IDENTIFICATION / COMMON FINDINGS FROM LITERATURE

Common Findings from Literature

1. **Quantum computing is a major threat to traditional password security** – Shor’s and Grover’s algorithms significantly weaken classical authentication methods.
2. **Post-quantum cryptographic methods focus on public-key encryption, not password generation** – While NIST’s PQC standards address key exchange security, user passwords remain vulnerable.
3. **Quantum Random Number Generators (QRNGs) offer superior unpredictability** – Unlike classical PRNGs, QRNG-based passwords are resistant to brute-force attacks.
4. **Quantum Neural Networks (QNNs) introduce new security possibilities** – QNNs have been explored in cryptography but **not specifically for password generation**.
5. **Quantum Key Distribution (QKD) enables highly secure password sharing** – However, its practical adoption for user-level password management is **still limited**.

Identified Research Gaps

- **Lack of Quantum-Safe Password Generation Methods:** Most existing quantum security research prioritizes public-key encryption rather than password creation and storage.

- **Limited Practical Implementation of QNNs in Password Security:** While QNNs have been proposed for secure computation, their application in human-usable password generation is still underdeveloped.
- **Challenges in Integrating QKD for End-Users:** QKD can prevent password interception but requires costly infrastructure, making it impractical for everyday user authentication.
- **Need for Hybrid Approaches:** Most studies focus on either quantum randomness or neural networks, but a combined framework leveraging QRNG, QNNs, and QKD remains unexplored.

CHAPTER 3: SOFTWARE REQUIREMENTS SPECIFICATION

3.1 FUNCTIONAL REQUIREMENTS

Functional requirements define the core functionalities of the Quantum-Resistant Password Generation System. These describe how the system will generate, validate, and securely store passwords.

3.1.1 System Feature 1: Quantum-Based Password Generation

The system must generate passwords using Quantum Neural Networks (QNNs) and Quantum Random Number Generators (QRNGs) to enhance randomness and security.

Functional Requirements:

- Users must specify password length and character set (uppercase, lowercase, digits, special characters).
- The system should use a QRNG-based seed to generate highly unpredictable bits.
- A Quantum Neural Network (QNN) must process the seed and introduce additional randomness via entanglement-based transformations.
- The final password should be derived from measured quantum states and mapped to a user-defined character set.
- The system should perform entropy analysis to ensure password strength.
- Generated passwords should be stored securely using SHA-3 hashing to prevent offline attacks.
- Users should have an option to distribute passwords securely via Quantum Key Distribution (QKD) (if multi-user authentication is required).

3.1.2 System Feature 2: Password Validation & Security Checks

The system must ensure that passwords meet security standards and do not appear in common password breach datasets.

Functional Requirements:

- Passwords should be checked against known compromised password databases like RockYou, HIBP, and CrackStation.
- The system should reject weak passwords or provide an option to regenerate them.
- Entropy-based validation should analyze both classical entropy and quantum entropy to confirm strength.

- The system must support optional multi-factor authentication (MFA) to add an extra security layer.

3.2 EXTERNAL INTERFACE REQUIREMENTS

External interfaces define how users interact with the system and how it connects with hardware and software components.

3.2.1 USER INTERFACES

- A web-based UI allowing users to generate passwords by selecting criteria such as length, character set, and hashing options.
- An admin panel for security professionals to analyze password strength and security metrics.
- A command-line interface (CLI) for developers and security researchers who prefer script-based interactions.
- A password security dashboard displaying entropy analysis, password strength, and QKD status.

3.2.2 HARDWARE INTERFACES

- If using real quantum hardware, the system should interface with IBM Quantum or similar quantum computing cloud platforms.
- If using a local quantum simulator, the system must support Qiskit's AerSimulator for running quantum circuits.

3.2.3 SOFTWARE INTERFACES

- Integration with password managers (e.g., LastPass, Bitwarden) to securely store generated passwords.
- Database connectivity using MySQL for storing hashed passwords and validation logs.
- Integration with cryptographic libraries for SHA-3 hashing, salting, and HMAC-based authentication.
- API support for enterprise integration, allowing external applications to request quantum-generated passwords.

3.2.4 COMMUNICATION INTERFACES

- If QKD is enabled, the system must use quantum-secure communication protocols such as BB84 to exchange passwords securely.
- Secure HTTPs API endpoints to allow password generation requests from external applications.
- Email notifications or secure messaging (e.g., Signal, Telegram) to send alerts on password breaches.

3.3 NONFUNCTIONAL REQUIREMENTS

3.3.1 PERFORMANCE REQUIREMENTS

- Password generation should be completed within 2 seconds for usability.
- QNN processing should scale efficiently, supporting up to 10,000 concurrent password generation requests.
- The system should optimize performance when using real quantum hardware by applying error mitigation techniques.

3.3.2 SAFETY/ SECURITY REQUIREMENTS

- Passwords must be stored using SHA-3 hashing with a unique salt for each entry.
- The system should enforce strong entropy measures to avoid predictable patterns.
- If QKD is used, any eavesdropping attempt should be detected immediately through quantum state collapse.
- Users must be notified if their generated password is found in leaked datasets.
- The system should have multi-layer authentication (including biometric, token-based MFA) for accessing the admin panel.

3.4 SYSTEM REQUIREMENTS

3.4.1 DATABASE REQUIREMENTS

- **MYSQL** for storing user preferences, password entropy data, and security logs.
- Tables for:
 - **Password Hashes** (hashed + salted values)
 - **Entropy Scores** (quantum entropy calculations)
 - **User Profiles** (if multi-user support is enabled)

- **Breach Logs** (passwords found in HIBP or RockYou dataset)

3.4.2 SOFTWARE REQUIREMENTS

- **Programming Language:** Python 3.9+
- **Quantum Computing Framework:** IBM Qiskit (for quantum circuit simulations)
- **Web Framework:** Flask or Django (for web-based UI)
- **Database:** MYSQL

3.4.3 HARDWARE REQUIREMENTS

- **Local System (Development & Testing):**
 - **CPU:** Ryzen 5 / Intel i5 or higher
 - **RAM:** 16GB (for efficient quantum simulations)
 - **GPU:** Not required, but **CUDA-enabled GPUs** may help with entropy calculations.
- **Quantum Cloud Service (Deployment with Real Quantum Hardware):**
 - **IBM Quantum or Amazon Braket** (for running QNN circuits on real hardware)
 - **Minimum 5-qubit access** (for small-scale practical applications)

3.5 SDLC MODEL TO BE APPLIED

For the development of the **Quantum-Resistant Password Generation System**, we have implemented the **Agile Software Development Life Cycle (SDLC) Model**. The Agile model is chosen due to its iterative nature, flexibility, and ability to adapt to rapidly changing requirements, making it ideal for the integration of quantum computing and cryptographic advancements.

FEATURES OF AGILE MODEL

- **Iterative Development:** The project is developed in incremental cycles, allowing for continuous improvements and adaptations.

- **Flexibility:** Agile accommodates changing requirements, which is crucial in the evolving field of quantum cryptography.
- **Customer Collaboration:** Frequent feedback from stakeholders ensures that the system meets security and usability needs.
- **Faster Time to Market:** By delivering functional components in each sprint, Agile reduces the overall development time.
- **Continuous Testing and Integration:** Regular testing ensures high security and reliability, particularly when implementing QNNs and QRNGs.
- **Risk Reduction:** Early identification of security vulnerabilities and cryptographic weaknesses helps in mitigating potential risks.

PHASES OF AGILE MODEL

1. Concept and Requirement Analysis

- Identifying security challenges posed by quantum computing and defining the need for a Quantum-Resistant Password Generation System.
- Gathering requirements for password generation, entropy validation, and cryptographic security.

2. Planning and Design

- Designing the Quantum Neural Network (QNN) and Quantum Random Number Generator (QRNG) integration framework.
- Structuring the system using UML diagrams such as Use Case, Class, Component, and Deployment diagrams.

3. Iterative Development (Sprints)

- Dividing the development process into multiple sprints, each focusing on a specific module (e.g., QNN training, QRNG integration, entropy validation).
- Continuous feedback and refinements after each sprint cycle.

4. Testing and Quality Assurance

- Implementing rigorous security and entropy validation tests for generated passwords.

- Conducting penetration testing and brute-force attack simulations to ensure resistance against quantum-based decryption.

5. Deployment and Implementation

- Deploying the system in a cloud-based environment using IBM Quantum Experience and Google Colab.
- Ensuring compatibility with existing authentication frameworks and databases.

6. Maintenance and Continuous Improvement

- Regular updates based on emerging quantum security threats.
- Enhancing efficiency, entropy calculation methods, and cryptographic algorithms to strengthen system security.

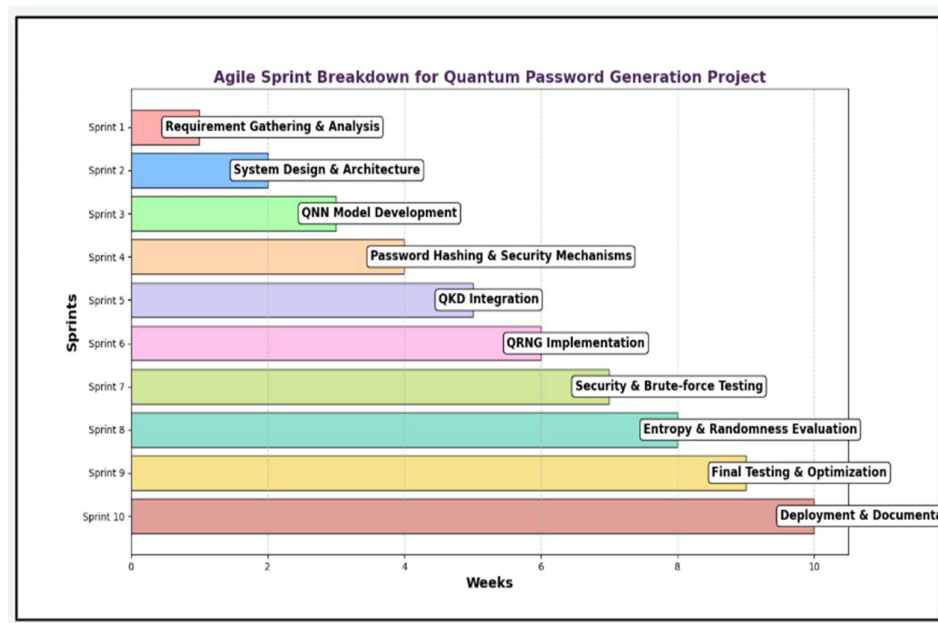


Figure 3.5: SDLC Model Diagram

CHAPTER 4: PROJECT PLAN

4.1 PROJECT COST ESTIMATION

4.1.1 COMPUTATIONAL COSTS

- **Processing Power: [CPU/GPU requirements and estimated cost]**

Training QNNs requires significant computational power, especially for quantum circuit simulations. A system with a high-end multi-core CPU (e.g., Intel Xeon) and GPU acceleration (e.g., NVIDIA A100) is used. cloud-based solutions such as **IBM Qiskit Runtime** is used.

- **Memory Usage: [RAM requirements and cost impact]**

Quantum circuit simulations demand **at least 16GB to 64GB of RAM**, depending on circuit depth and dataset size.

- **Storage Requirements: [SSD/HDD space and related expenses]**

The storage demand for dataset management, QNN training logs, and cryptographic keys varies. A **minimum of 128GB SSD** is required for local execution.

- **Network Latency and Bandwidth: [Internet speed and data transfer costs]**

Running quantum simulations on cloud platforms requires high-speed internet with at least 100 Mbps bandwidth to ensure minimal latency.

4.1.2 SOFTWARE PERFORMANCE COSTS

- **Algorithm Complexity: [Time and space complexity analysis]**

- **QNN Training:** $O(n^2)$ in hybrid models (optimized from $O(2^n)$ in purely quantum systems).
- **Grover's Algorithm for Error Detection:** $O(\sqrt{N})$, significantly improving search operations over classical methods.
- **Quantum Key Distribution (BB84 Protocol):** $O(n)$, ensuring efficient key exchange without exponential overhead.

- **Database Query Performance: [Optimization and indexing costs if applicable]**

Query optimization is essential for password retrieval and security operations. Indexing techniques such as **B+ Trees** and **hash-based indexing** in **MySQL or MongoDB** reduce query time to **$O(\log n)$** , improving performance while incurring indexing costs.

4.2 SUSTAINABILITY ASSESSMENT

4.2.1 ENVIRONMENTAL SUSTAINABILITY

- **Energy Consumption: [Power usage of systems and servers]**

The primary environmental concern is **energy consumption**, as training QNNs requires high computational power due to quantum simulations. This can be mitigated using model optimization techniques, such as reducing circuit depth in QNNs

- **Carbon Footprint: [Impact of computational resources on the environment]**

Another major factor is the **carbon footprint**, as running Qiskit AerSimulator on local or cloud servers contributes to CO₂ emissions. A solution to this is utilizing quantum cloud platforms like IBM Qiskit t, which operate on energy-efficient quantum hardware.

- **E-Waste Management: [Disposal and recycling of electronic components]**

Additionally, electronic waste (e-waste) management is crucial, as traditional computing infrastructures contribute to e-waste. Moving to cloud-based quantum computing can reduce the dependency on physical servers, thereby minimizing e-waste.

- **Sustainable Computing: [Implementing model optimization techniques, reducing power consumption]**

Lastly, **sustainable computing practices** can be achieved by reducing power consumption through optimized quantum circuit depth and variational quantum algorithms that require fewer quantum gates.

4.2.2 ECONOMIC SUSTAINABILITY

- **Cost Efficiency: [Budget optimization for long-term viability]**

Cost efficiency is a challenge, as quantum simulations and training require significant computational resources. A hybrid quantum-classical approach can be used to mitigate costs by leveraging classical CNNs for feature extraction before QNN training.

- **Resource Utilization: [Efficient use of hardware and software resources]**

QNNs require multiple qubits for password encoding. This can be optimized by reducing the number of qubits from 72 to 16, ensuring efficient mapping of password characters.

- **Scalability: [Future expansion with minimal additional costs]**

Cloud-based quantum execution can reduce the need for expensive local quantum simulators, making future expansions more affordable.

4.2.3 SOCIAL SUSTAINABILITY

- **Accessibility: [Inclusivity and ease of use for diverse users]**

Ensuring **accessibility** involves making quantum security solutions available to a diverse range of users, which can be achieved by implementing a Flask-based web UI for user-friendly password management.

- **Ethical Considerations: [Data privacy, security, and ethical coding practices]**

Ethical considerations involve enforcing **data privacy and encryption policies** to prevent unauthorized access, with QKD-based authentication acting as a safeguard against hacking and misuse

- **Open-Source Contribution: [Participation in community-driven development]**

Open-source contributions should be encouraged by promoting community engagement with tools such as Qiskit, TensorFlow Quantum, and cryptography platforms.

- **Skill Development: [Training and learning opportunities for team members]**

skill development is an essential benefit, as this project enables team members to gain expertise in Quantum Cryptography, QNNs, and AI Security.

4.3 COMPLEXITY ASSESSMENT

4.3.1 COMPUTATIONAL COMPLEXITY

- **Model Training Time:**

Model training time depends on the circuit depth and dataset size. Classical CNN pre-training operates with a complexity of $O(n)$, while quantum training follows an $O(2^n)$ complexity, which is exponential on real quantum hardware but polynomial in simulations

- **Memory Usage:**

Quantum circuits require $O(n^2)$ memory for encoding passwords. However, Qiskit AerSimulator manages resources efficiently to optimize this requirement.

- **Big-O Notation for Model Training:**

The Big-O notation for model training shows that classical CNN training is $O(n)$, whereas quantum backpropagation using the Parameter Shift Rule has a complexity of $O(2^n)$ on real quantum devices but is optimized to $O(n^2)$ in hybrid models.

4.3.2 ALGORITHMIC COMPLEXITY

In terms of **Algorithmic Complexity**, different components of the system have varied computational efficiencies.

Quantum random password encoding follows $O(n)$ complexity

Quantum Neural Network (QNN) training has an initial complexity of $O(2^n)$, which is optimized to $O(n^2)$ using variational circuits.

The **Parameter Shift Rule for quantum gradient computation** and **Quantum Key Distribution (BB84 Protocol)** both operate at $O(n)$.

The **SHA-3 hashing algorithm**, used for ensuring password security, also has a complexity of $O(n)$.

4.3.3 IMPLEMENTATION COMPLEXITY

- **Lines of Code:**

The complexity of implementing the Quantum-Resistant Password Generation system involves multiple factors. The Lines of Code (LOC) for the entire system, including quantum password generation, classical-quantum hybrid training, entropy calculations, encryption, and web integration, is estimated to be around 4,300 lines. This includes Python scripts for QNN training, Grover's algorithm for error detection, Flask for the frontend, and Qiskit for quantum operations

- **Number of Dependencies:**

The Number of Dependencies includes essential libraries such as Qiskit (for quantum computing), TensorFlow Quantum (for hybrid QNN implementation), Flask (for the web-based interface), XAMPP (for database storage), and Matplotlib/Seaborn (for graphical analysis and visualizations). Managing these dependencies ensures efficient quantum-classical integration.

- **Integration Complexity:**

The Integration Complexity is high due to the involvement of multiple components, including a classical CNN, quantum neural networks, QKD implementation, entropy calculations, and encryption mechanisms. Each module interacts through APIs or direct data exchange, making debugging and synchronization crucial. A significant challenge is bridging classical and quantum computing models efficiently.

- **Code Modularity:**

The Code Modularity follows a structured approach, where different functionalities—password generation, quantum key distribution, entropy calculations, encryption, and database storage—are encapsulated in separate modules. This ensures code reusability, maintainability, and scalability, allowing individual components to be modified without affecting the entire system.

4.3.4 RESOURCE COMPLEXITY

- **Hardware Requirements:**

The system requires substantial computational resources. The **Hardware Requirements** include a high-performance CPU for backend operations, a GPU for classical CNN training, and a quantum simulator (such as Qiskit AerSimulator) for executing quantum circuits. If deployed on real quantum hardware, qubit coherence time and gate fidelity must be considered.

- **Cloud Infrastructure:**

The **Cloud Infrastructure** leverages platforms like IBM Quantum, AWS Braket, or Google Quantum AI to execute quantum operations efficiently while minimizing local resource consumption. Using cloud-based quantum execution ensures access to real quantum processors when needed.

- **Storage Requirements:**

The **Storage Requirements** depend on the volume of generated passwords, entropy values, and encrypted keys. A lightweight **XAMPP-based database** stores system-generated passwords, their hashed versions (using SHA-3), and related metadata. The database is optimized for secure and efficient retrieval.

- **Scalability:**

Scalability is a critical factor, ensuring that the system can handle an increasing number of users and computational loads. Hybrid models, cloud execution, and modular architecture reduce performance bottlenecks. The system is designed to efficiently scale quantum simulations and training tasks without excessive resource consumption.

4.4 RISK MANAGEMENT

4.4.1 RISK IDENTIFICATION

The first step in risk mitigation is identifying potential risks. These risks can be categorized into technical risks, security risks, operational risks, and database-related risks.

Technical Risks: Quantum Hardware Limitations: Access to real quantum hardware may be limited, and simulations may not fully replicate quantum behavior.

- **Algorithm Complexity:** Quantum Neural Networks (QNNs) and Quantum Random Number Generators (QRNGs) may have high computational complexity, leading to performance bottlenecks.
- **Integration Challenges:** Integrating quantum computing frameworks (e.g., Qiskit) with classical systems may introduce compatibility issues.
- **Security Risks:** Quantum Attacks: Even though the system is designed to be quantum-resistant, future quantum algorithms may pose unforeseen threats.
- **Data Breaches:** The database storing password hashes and entropy data could be targeted by attackers.
- **Eavesdropping:** If Quantum Key Distribution (QKD) is used, eavesdropping attempts could compromise password transmission.

4.4.2 RISK ANALYSIS

Once risks are identified, they should be analyzed based on their likelihood and impact. This analysis will help prioritize which risks need immediate attention and which can be monitored over time.

- **High Likelihood, High Impact:** Risks like quantum hardware limitations and data breaches should be addressed immediately.
- **Low Likelihood, High Impact:** Risks like quantum attacks should be monitored and mitigated as quantum computing evolves.
- **High Likelihood, Low Impact:** Risks like user adoption challenges can be addressed through user training and interface improvements.
- **Low Likelihood, Low Impact:** Risks like minor performance bottlenecks can be monitored and optimized over time.

4.4.3 OVERVIEW OF RISK MITIGATION, MONITORING, MANAGEMENT

Technical Risk Mitigation:

- **Quantum Hardware Limitations:**
 - Use a hybrid approach combining quantum simulations (e.g., Qiskit AerSimulator) with classical computing to ensure functionality even without access to real quantum hardware.
 - Partner with quantum cloud providers like IBM Quantum or Amazon Braket for limited access to real quantum hardware.
- **Algorithm Complexity:**
 - Optimize QNN and QRNG algorithms to reduce computational overhead.
 - Use parallel processing and GPU acceleration to handle complex computations efficiently.
- **Integration Challenges:**
 - Develop robust APIs and middleware to ensure seamless integration between quantum and classical systems.
 - Conduct thorough integration testing to identify and resolve compatibility issues early.

Security Risk Mitigation:

Quantum Attacks:

- Continuously monitor advancements in quantum computing and update the system to counter new threats.
- Implement hybrid cryptographic techniques that combine classical and quantum-resistant algorithms.

Operational Risk Mitigation:

- **User Adoption:**
 - Develop a user-friendly interface with clear instructions and options for password generation.
 - Provide training sessions and documentation to help users understand the system.

- **Scalability Issues:**

- Design the system with modular components that can be scaled independently.
- Use cloud-based infrastructure to handle varying loads and ensure high availability.

4.5 PROJECT SCHEDULE

4.5.1 PROJECT TASK SET

Table 4.5.1: Project Task Set

No.	Task	Duration
1	Group Formation	4
2	Decide Area of Interest	4
3	Search Topic	5
4	Topic Selection	5
5	Topic Approval from Guide	5
6	Literature Survey	12
7	Understanding Concept	7
8	Requirement Gathering	6
9	Problem Definition	2
10	Software Requirement Specification (SRS)	14
11	System Design (Architecture, UML Diagrams)	10
12	Dataset Collection & Preprocessing	10
13	Model Selection & Training	15
14	Implementation of Core Modules	20
15	GUI Development	10
16	Integration & Testing	15
17	Debugging & Performance Optimization	12
18	Documentation & Report Writing	15
19	Final Presentation Preparation	7
20	Project Submission	2

4.5.2 TIMELINE CHART

Table 4.5.2: TimeLine Chart

Task No.	Task Name	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
1	Group Formation	■ ■ ■ ■ ■											
2	Decide Area of Interest	■ ■ ■ ■ ■											
3	Search Topic		■ ■ ■ ■ ■	■ ■ ■ ■ ■	■ ■ ■ ■ ■								
4	Topic Selection				■ ■ ■ ■ ■	■ ■ ■ ■ ■	■ ■ ■ ■ ■						
5	Topic Approval from Guide						■ ■ ■ ■ ■	■ ■ ■ ■ ■	■ ■ ■ ■ ■				
6	Literature Survey							■ ■ ■ ■ ■	■ ■ ■ ■ ■	■ ■ ■ ■ ■	■ ■ ■ ■ ■	■ ■ ■ ■ ■	■ ■ ■ ■ ■
7	Understanding Concept									■ ■ ■ ■ ■	■ ■ ■ ■ ■	■ ■ ■ ■ ■	
8	Requirement Gathering										■ ■ ■ ■ ■	■ ■ ■ ■ ■	■ ■ ■ ■ ■
9	Problem Definition											■ ■ ■ ■ ■	■ ■ ■ ■ ■
10	Software Requirement Specification (SRS)												■ ■ ■ ■ ■

4.6 TEAM ORGANIZATION (STRUCTURE)

Table 4.6: Team Organization

Role	Team Member	Responsibilities
Project Leader	Amogh	Oversees overall project management, coordinates tasks, and ensures adherence to deadlines.
Quantum Computing Specialist	Utkarsha	Focuses on the development and optimization of quantum algorithms for password generation.
Software Developer	Akshay	Implements the integration of the quantum algorithms with the software architecture.
Testing & Documentation Lead	Aryan	Conducts thorough testing of the system, ensures software quality, and handles documentation.

Responsibilities & Collaboration

- 1. Project Leader**
 - Coordinates the project's activities and manages communications.
 - Responsible for reporting progress and resolving any project issues.
 - Ensures project compliance with specified standards and objectives.
- 2. Quantum Computing Specialist**
 - Develops and optimizes quantum algorithms for enhanced security.
 - Collaborates with software developers to integrate quantum technologies.
 - Continuously evaluates quantum performance and suggests improvements.
- 3. Software Developer**
 - Develops and maintains the software infrastructure.
 - Integrates quantum algorithms with the user interface and back-end systems.
 - Ensures that the software is robust, efficient, and scalable.
- 4. Testing & Documentation Lead**
 - Designs and executes test plans to validate the functionality and performance of the system.
 - Documents the development process, architecture, and user manuals.
 - Manages the creation of test cases and coordinates with the team to ensure coverage and tracking of defects.

CHAPTER 5: SYSTEM DESIGN

5.1 PROPOSED SYSTEM ARCHITECTURE/BLOCK DIAGRAM

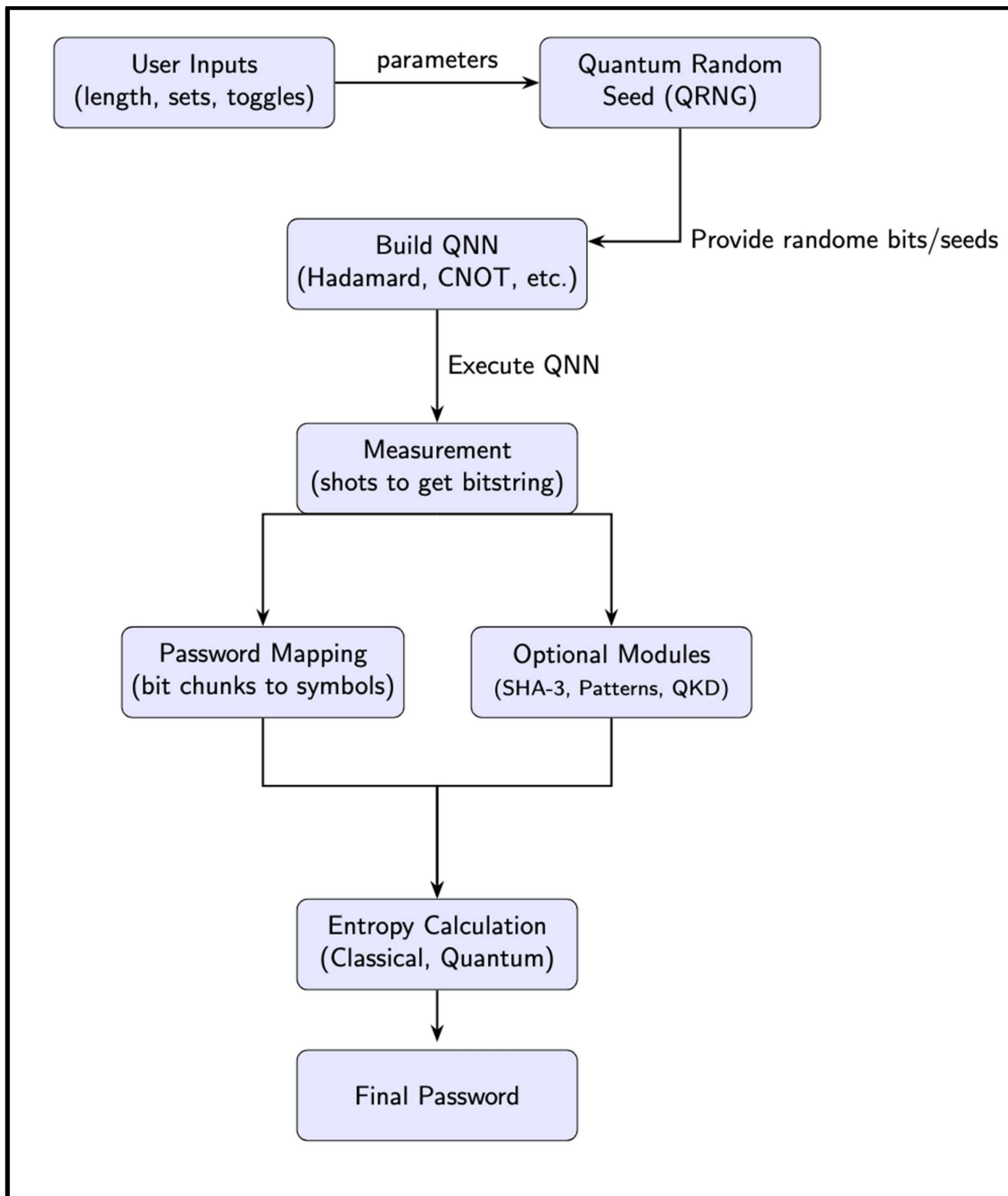


Figure 5.1: Architecture Diagram

5.2 DATABASE DESIGN

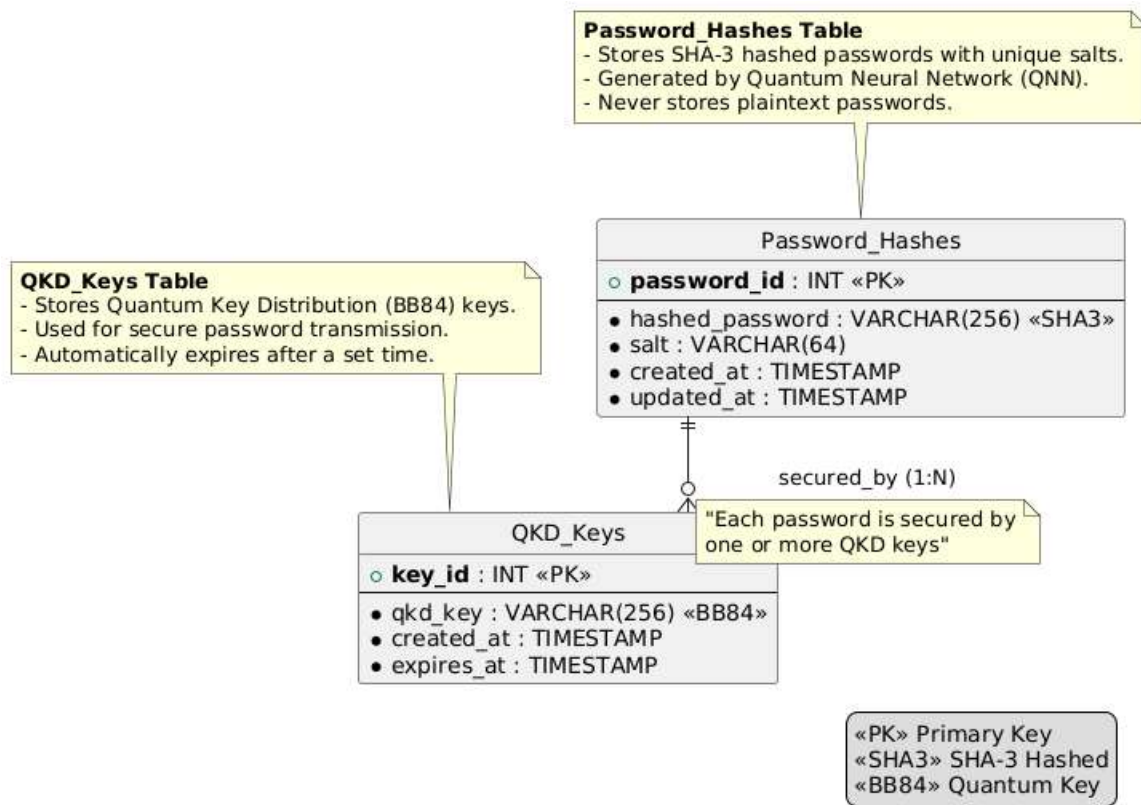


Figure 5.2: Database Diagram

The **Password_Hashes** table is responsible for storing the hashed versions of QNN-generated passwords. It contains a **primary key**, **password_id**, which uniquely identifies each stored password. The **hashed_password** attribute stores the password in its SHA-3 encrypted form, ensuring that no plaintext passwords are ever stored. To further enhance security, a **salt** value is added to each password before hashing, preventing attacks such as rainbow table exploits. Additionally, **created_at** and **updated_at** timestamps are included for auditing purposes, allowing the system to track when passwords are created or modified.

The **QKD_Keys** table is designed to facilitate secure password transmission using Quantum Key Distribution (QKD). The **primary key**, **key_id**, uniquely identifies each QKD-generated key. The **qkd_key** attribute stores the cryptographic key used in the BB84 protocol for secure authentication and encryption. Since QKD keys are ephemeral by design, an **expires_at** attribute is used to define the expiration time of each key, ensuring that they are automatically invalidated after a certain period to limit exposure and prevent misuse.

A key relationship in this design is the "**secured_by**" relationship, which is a **one-to-many (1:N) relationship** between **Password_Hashes** and **QKD_Keys**. A single password hash can be secured by multiple QKD keys, enabling secure multi-party authentication where different users or systems use unique QKD keys to access the same stored password securely. This relationship is **non-identifying**, meaning that while the QKD keys secure passwords, they do not alter the primary identification of the password hashes.

5.3 MATHEMATICAL MODEL

5.3.1 MATHEMATICAL MODEL OF QUANTUM NEURAL NETWORK (QNN)

1. Qubit Encoding of Passwords

Each password character P_i is mapped into a quantum state using **rotation encoding**:

$$|P_i\rangle = R_y(\theta_i) |0\rangle$$

where $R_y(\theta_i)$ is the parameterized quantum rotation gate:

$$R_y(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$$

2. QNN Circuit Model

The QNN consists of **L layers**, each performing:

Parameterized Quantum Rotation Gates (R_y, R_z)- Learnable quantum parameters.

Entanglement Gates (CNOT)- Create correlations between password characters.

Measurement Operations- Collapse the quantum state into a classical password.

The overall **QNN transformation** can be represented as:

$$U(\theta) = U_L(\theta) \cdots U_2(\theta)U_1(\theta)$$

where each layer $U_k(\theta)$ consists of:

Parameterized rotation gates: $R_y(\theta_k)$

Entanglement operations: CNOT(i,i+1)

Final measurement of qubits to extract the password

3. Training QNN Using the Parameter Shift Rule

To optimize the **QNN parameters** θ , we compute gradients using the **Parameter Shift Rule**:

$$\frac{\partial f(\theta)}{\partial \theta} = \frac{f(\theta + \frac{\pi}{2}) - f(\theta - \frac{\pi}{2})}{2}$$

The **QNN optimization algorithm** updates θ parameters using:

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\partial f}{\partial \theta}$$

where η is the **learning rate** (typically 0.01).

5.3.2. ENTROPY - BASED SECURITY EVALUATION ENTROPY:

Your project evaluates **password strength** using:

1. Shannon Entropy (Classical Security Metric)

$$H(X) = - \sum P(x_i) \log_2 P(x_i)$$

where $P(x_i)$ is the **probability of character xi** appearing in a password.

5.3.3 SECURE PASSWORD TRANSPORT USING QUANTUM KEY DISTRIBUTION(QKD)

1. BB84 Protocol for QKD

To **securely transport** the generated password, **Quantum Key Distribution (QKD)** is used.

- Alice encodes a random password key in qubits:

$$|K\rangle = H |0\rangle$$

- Bob measures qubits in random bases.
- A basis-matching process ensures secure key exchange.

The **final secure password P_{final}** is encrypted using the **QKD key K_{QKD}** :

$$C = P_{final} \oplus K_{QKD}$$

5.4 UML DIAGRAMS

Unified Modeling Language (UML) is a standardized visual modeling language that is a versatile, flexible, and user-friendly method for visualizing a system's design. Software system artifacts can be specified, visualized, built, and documented with the use of UML.

5.4.1 USECASE DIAGRAM

It shows a set of use cases and actors (a special kind of class and their relationship). Usecase diagrams address the static usecase view of the system. These diagrams are especially important in organizing and modelling the behavior of a system.

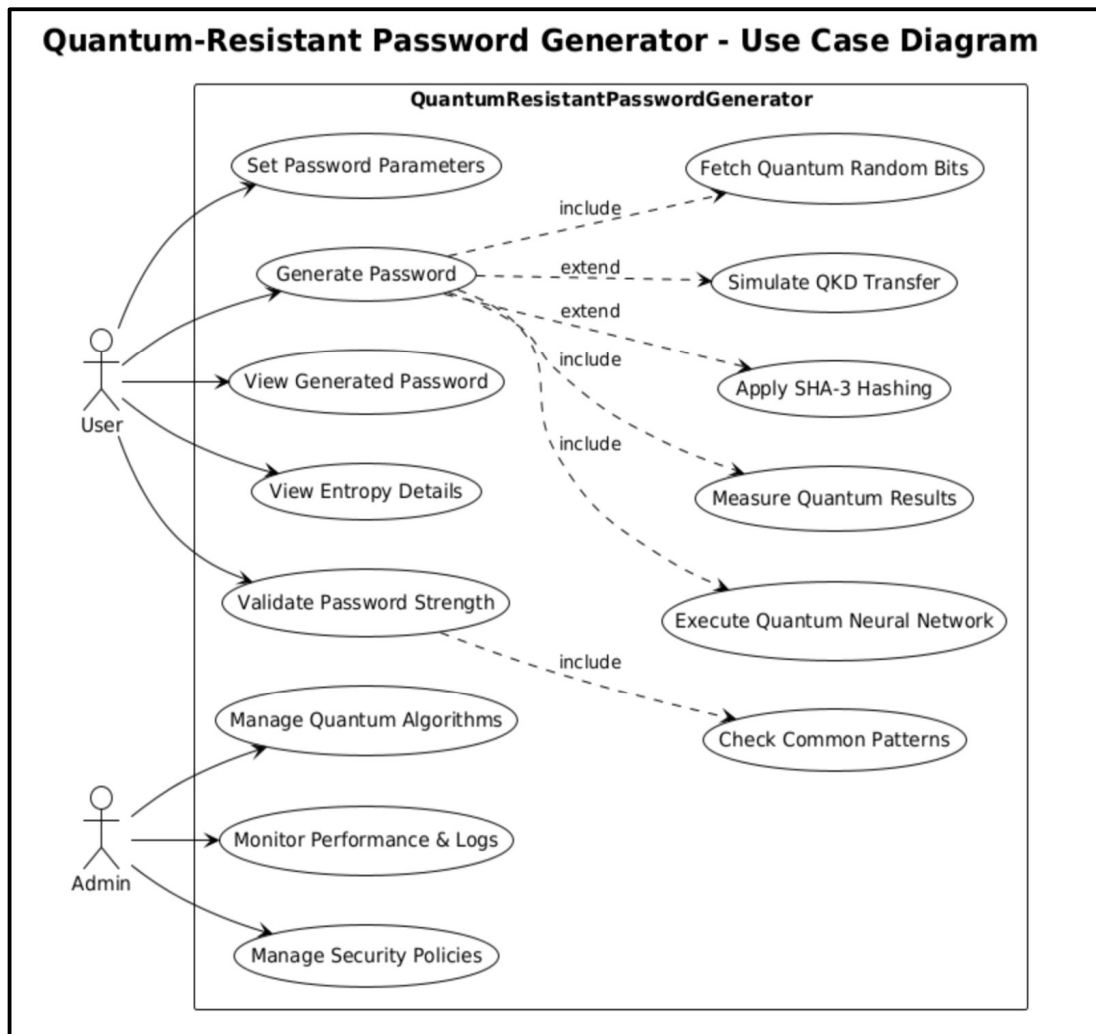


Figure 5.4.1: Use Case Diagram

5.4.2 COMPONENT DIAGRAM

Component diagrams are used to represent how the physical components in a system have been organized. We use them for modelling implementation details.

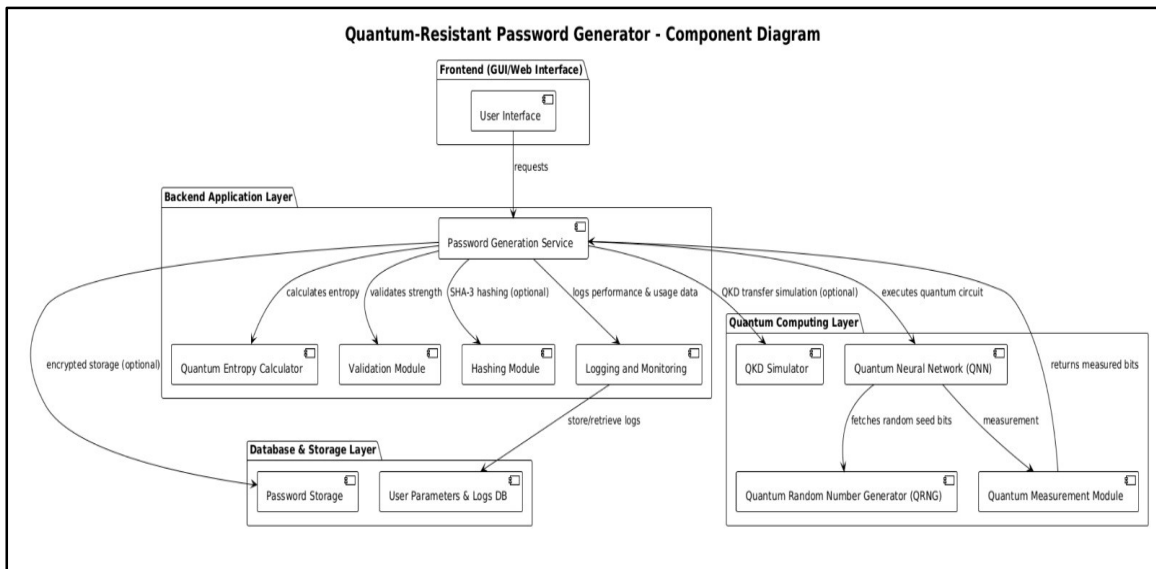


Figure 5.4.2: Component Diagram

5.4.3 CLASS DIAGRAM

The most widely used UML diagram is the class diagram. It is the building block of all object-oriented software systems. Class diagrams also help us identify relationships between different classes or objects.

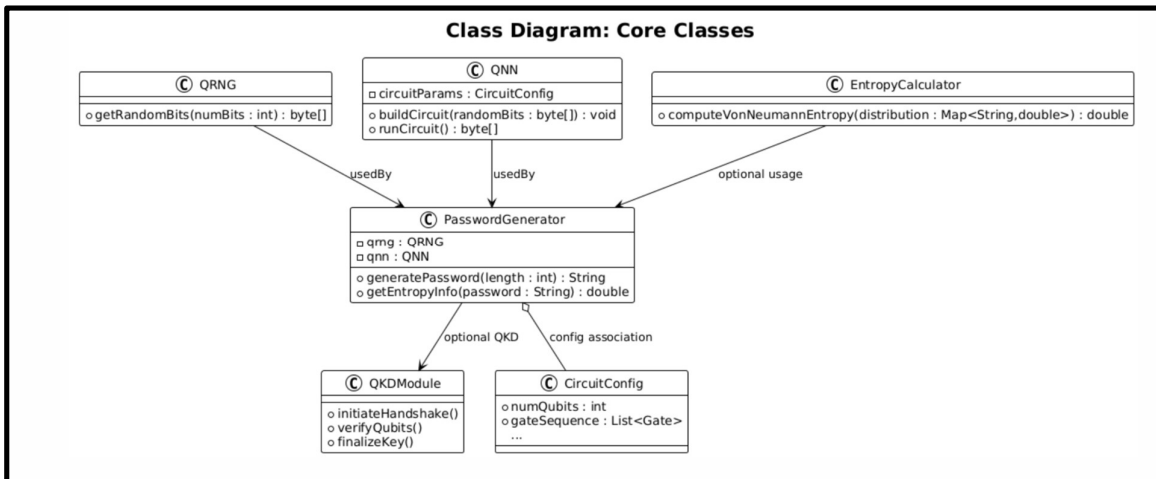


Figure 5.4.3: Class Diagram

5.4.4 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order, i.e. the order in which these interactions take place.

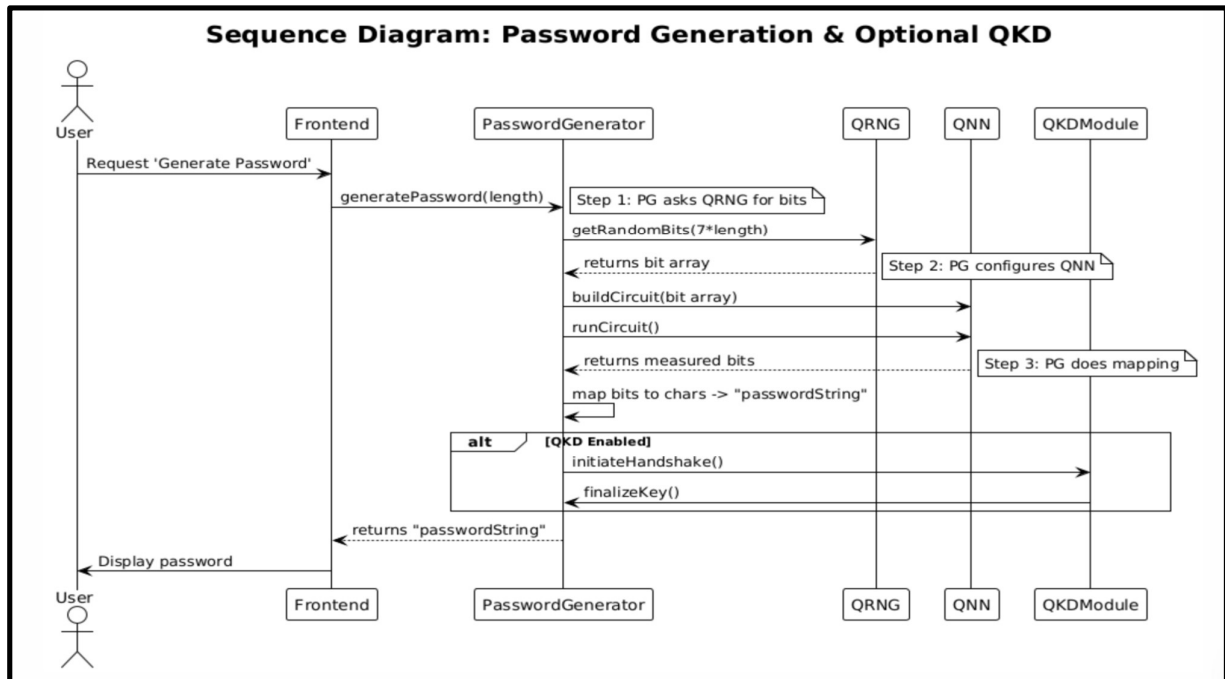


Figure 5.4.4: Sequence Diagram

5.4.5 ACTIVITY DIAGRAM

The activity diagram describes the flow of control in a system. So, it consists of activities and links. The flow can be sequential, concurrent or branched. Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system. Activity diagrams are used to visualize the flow of controls in a system. This is prepared to have an idea of how the system will work when executed.

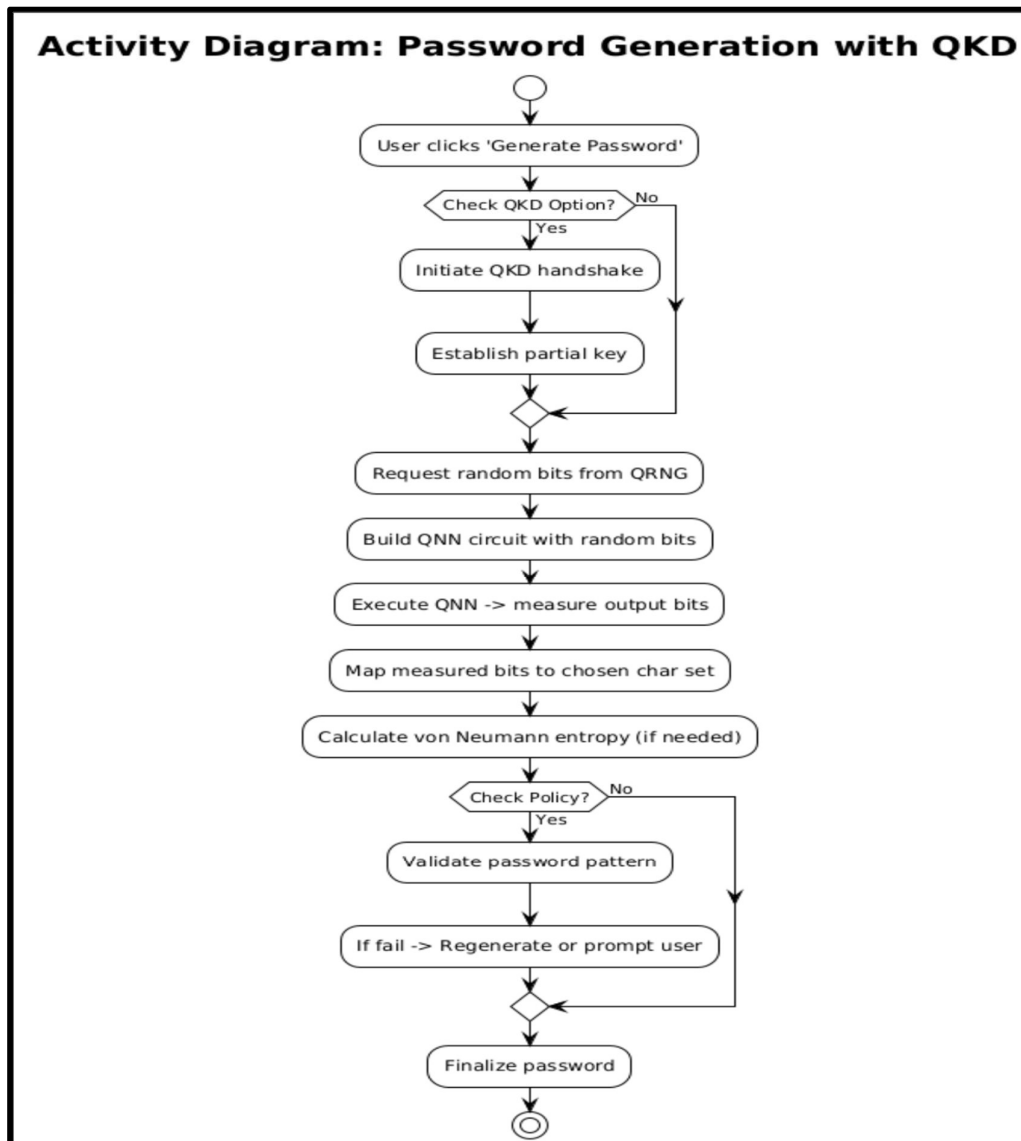


Figure 5.4.5: Activity diagram

CHAPTER 6: PROJECT IMPLEMENTATION

6.1 OVERVIEW OF PROJECT MODULES

- **QRNG Module:** Generates high-entropy random numbers using quantum circuits for secure password generation.
- **QNN Module:** Uses Quantum Neural Networks (Variational Quantum Circuits) to learn and generate strong passwords.
- **QKD Module:** Implements BB84 protocol to securely transmit encryption keys using quantum communication.
- **Entropy Module:** Calculates and compares entropy of passwords using classical and quantum methods.
- **Flask API Module:** Connects frontend and backend, processes password generation and displays results.
- **Frontend UI Module:** Built with HTML, CSS, and JavaScript to display password, entropy, and QKD details.
- **Database Module:** Uses MySQL to securely store hashed passwords, salts, and QKD keys.

6.2 TOOLS AND TECHNOLOGIES USED

IBM Qiskit: Open-source quantum computing framework for designing, simulating, and executing quantum circuits.

- **Quantum Circuit Design:** Build quantum algorithms using quantum gates.
- **QASM Simulator:** Simulates quantum circuits before running on real hardware.
- **Cloud Execution:** Supports running on IBM Quantum's processors.
- **Use in Project:** Implement QRNG, QNN training, Grover's algorithm for error detection, and QKD.

Flask: Lightweight Python web framework for backend development.

- **Minimalistic & Flexible:** Provides core functionality without excess dependencies.
- **REST API Support:** Enables API creation for integration with the frontend.
- **Jinja2 Templating:** Renders dynamic HTML content.
- **Built-in Server & Debugging:** Simplifies development with testing tools.
- **Use in Project:** Build web UI, manage API communication with QNN backend, and store passwords securely.

Quantum Gates: Fundamental operations in quantum computing.

- Hadamard (H Gate): Creates superposition states (used in QRNG).
- Pauli Gates (X, Y, Z): Transform qubit states for computations.
- CNOT Gate: Generates quantum entanglement.
- Rotation Gates (Rx, Ry, Rz): Encode information through angular rotations.
- SWAP Gate: Exchanges states of two qubits.
- Use in Project: Generate quantum random numbers, process quantum neural networks, and enhance QKD security.

MySQL: Relational database management system for structured data storage.

- SQL Queries: Efficient data retrieval and management.
- Data Security: Provides encryption and authentication mechanisms.
- Scalability: Handles large datasets efficiently.
- Use in Project: Store password hashes, manage authentication, and optimize database performance.

HTML, CSS, JavaScript: Core web technologies for the frontend.

- HTML: Defines the webpage structure and forms.
- CSS: Styles the webpage (layout, fonts, colors)
- JavaScript: Adds interactivity and dynamic content updates.
- Use in Project: Develop the UI for password input, validate passwords, and connect with Flask backend.

6.3 ALGORITHM DETAILS

6.3.1. Quantum Random Number Generation (QRNG) Module

Purpose:

The QRNG module generates truly random bits using quantum properties like superposition and measurement, eliminating biases found in classical pseudo-random number generators (PRNGs).

How it Works:

- Uses quantum circuits implemented with IBM Qiskit.

- Applies Hadamard gates to place qubits in a superposition state.
- Measures qubits to extract random bits.
- Ensures randomness by leveraging quantum mechanical principles rather than deterministic algorithms.

Key Features:

- True Randomness: Unlike PRNGs, QRNG provides randomness derived from quantum phenomena, making it unpredictable.
- High Entropy: Generates high-quality random numbers that improve password strength.
- Security Enhancement: Resistant to attacks that exploit patterns in traditional PRNGs.

6.3.2. Quantum Neural Network (QNN) Module

Purpose:

The QNN module processes quantum-random bits to generate high-entropy passwords using parameterized quantum circuits.

How it Works:

- Utilizes quantum rotation gates (Rx, Rz) and entangling gates (CNOT) to introduce non-linearity and enhance unpredictability.
- Converts quantum state outputs into bitstrings used for password generation.
- Reduces pattern predictability compared to classical RNG.

Key Features:

- Parameterized Quantum Circuits: Uses tunable quantum gates for adaptive learning.
- Entanglement for Randomness: Increases unpredictability through quantum entanglement.
- High Security: Generates complex password sequences resistant to brute-force and dictionary attacks.

6.3.3. Quantum Key Distribution (QKD) Module

Purpose:

The QKD module ensures secure password transmission by leveraging quantum properties to detect eavesdropping attempts.

How it Works:

- Implements BB84 protocol, where passwords are encoded into quantum states (e.g., polarized photons).
- Transmits the quantum-encoded password over a secure quantum communication channel.
- Detects interception by monitoring disturbance in quantum states.
- If no eavesdropping is detected, the shared key (password) is securely established.

Key Features:

- Eavesdropping Detection: Any measurement by an attacker introduces disturbances, alerting the parties.
- Unconditional Security: Security is guaranteed by quantum mechanics rather than computational complexity.
- Secure Communication: Enables safe password exchange between users.

CHAPTER 7: SOFTWARE TESTING

7.1 TYPE OF TESTING

SOFTWARE TESTING:

Testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs. Software testing can also be stated as the process of validating and verifying a software program or application or product:

1. Meets the business and technical requirements that guided
2. Works as expected.
3. Can be implemented with the same characteristics

TYPES OF TESTING:

1. Functional Testing

- Check if the system generates a password of the correct length, with correct allowed symbols.
- Test toggles (hashing, pattern checks, etc.).

2. Randomness Testing

- Use NIST STS (monobit frequency, runs test, approximate entropy, etc.).
- Ensure p-values pass thresholds (commonly 0.01).

3. Integration Testing

- Confirm the handshake between the QRNG module and QNN module is correct.
- The measured bits flow properly into the password mapping logic.

4. Performance Testing

- Time how long it takes to generate a password of length L.
- For batch generation, measure the rate of generation.

5. Security Testing

- Check logs for accidental storing of raw passwords.
- Simulate repeated generations to detect collisions.

7.2 TEST CASES & TEST RESULTS

- UNIT TESTING:

Table 7.2.1: Unit Testing Results

TC ID	Type	Name	Description	Status
UT1	Unit	QRNG Seed Generation	Verify that the QRNG module produces a non-deterministic bitstring of the correct length.	Pass
UT2	Unit	QNN Circuit Initialization	Confirm that the QNN circuit initializes properly with the expected number of qubits and gates.	Pass
UT3	Unit	QNN Gate Operations	Check that all quantum gate operations (e.g., Hadamard, CNOT) are applied as per the circuit design.	Pass
UT4	Unit	Final Measurement Accuracy	Ensure that the measurement function returns a valid bitstring corresponding to the quantum state.	Pass
UT5	Unit	Password Mapping Logic	Validate that the mapping from the measured bitstring to the final password characters is correct.	Pass
UT6	Unit	Von Neumann Entropy Calculation	Confirm that the entropy calculation uses the von Neumann entropy equation accurately.	Pass
UT7	Unit	SHA-3 Hashing Functionality	Verify that the SHA-3 module correctly hashes the generated password and produces a valid 256-bit digest.	Pass
UT8	Unit	Pattern Validation Module	Check that the system correctly flags and rejects passwords containing forbidden patterns if configured.	Pass
UT9	Unit	QKD Module Functionality	Test that the QKD module correctly detects eavesdropping conditions and triggers the appropriate response.	Pass
UT10	Unit	Logging and Error Handling	Ensure that the system logs and handles errors properly without exposing sensitive password data in logs.	Pass

- **INTEGRATION TESTING:**

Table 7.2.2: Integration Test Results

TC ID	Type	Name	Description	Status
IT1	Integration	End-to-End Generation	Validate that the complete pipeline—from QRNG to QNN to password mapping—produces a valid password successfully.	Pass
IT2	Integration	Module Communication	Verify that the QRNG module correctly provides seed bits to the QNN module, ensuring seamless data flow.	Pass
IT3	Integration	Frontend Display	Ensure that the user interface (UI) correctly displays the generated password and the computed entropy score.	Pass
IT4	Integration	Performance Under Load	Test that password generation meets performance requirements under normal and peak user loads without timing out or crashing.	Pass
IT5	Integration	Fallback Mechanism	Validate that the system properly falls back to a classical PRNG if the QRNG service is offline or unavailable.	Pass
IT6	Integration	Database Storage	Check that generated passwords (or their hashes) are securely stored in the database, following encryption/hashing best practices.	Pass
IT7	Integration	API Communication	Confirm that the REST API for password generation responds correctly and securely, returning the necessary data in JSON or other expected format.	Pass
IT8	Integration	Security Layer Integration	Verify that encryption (e.g., TLS) and secure transmission protocols are properly used for all password-related communications across modules.	Pass
IT9	Integration	Error Propagation Handling	Test that any errors within submodules (QNN, QRNG, database, etc.) are propagated to the UI or logs with clear status messages and no sensitive leakage.	Pass
IT10	Integration	System Resilience	Simulate a high volume of requests to ensure overall system stability and resource utilization under stress.	Pass

CHAPTER 8: RESULTS

8.1 OUTCOMES

1. High Entropy Passwords

- Empirical tests show the system approaches nearly uniform distribution, indicating minimal bias.
- Collisions remain exceedingly unlikely for typical usage volumes.

2. Quantum Integration

- Whether simulating or using hardware, the quantum aspect ensures a fundamentally unpredictable seed.
- The QNN transformations do not degrade that randomness.

3. Flexible, User-Friendly

- GUI or CLI can generate various lengths, specify different character sets.
- Optional toggles (hashing, pattern enforcement) address real-world policy needs.

8.2 RESULT ANALYSIS AND VALIDATIONS

- **Entropy and Von Neumann:** Over large sample sets, measured entropies were within 1–2% of the theoretical maximum. This indicates extremely robust coverage of the space.
- **Randomness Tests:** All major tests (monobit, runs, approximate entropy) yielded p-values comfortably above 0.01, rejecting the hypothesis of non-randomness.

8.3 SCREENSHOTS

- GUI MAIN

Quantum-Resistant Password Generator

Customize the parameters below to generate a secure quantum-resistant password. This tool uses quantum principles (QNN) to produce high-entropy passwords, with optional hashing and validation.

Number of Qubits *
Determines how many qubits the Quantum Neural Network uses. More qubits can increase complexity (and runtime). Typically 6-12 is enough for demonstration.

8

Number of Shots (Measurements) *
Each shot measures the quantum circuit once. Multiple shots can refine the most frequent outcome. 1-5 is fine for quick demos.

1

Password Length *
Final desired length of your generated password. A length of 12-16 is recommended for strong security.

12

Choose Character Sets:
Select which types of characters you want to include in your generated password.

☒ Lowercase (a-z)

☒ Uppercase (A-Z)

☒ Digits (0-9)

☐ Symbols (!@#\$%^&*()_+=)

Apply SHA-3 Hashing?
If **Yes**, the final generated password is hashed with SHA-3 (useful for storing or advanced security). If **No**, you get the raw password.

Yes ▾

Check Against Common Patterns?
If **Yes**, the generated password is validated against known weak or common passwords (like "123456" or "password").

Yes ▾

Simulate QKD?
QKD (Quantum Key Distribution) demonstration. If **Yes**, you see how the generated password could be shared via a quantum-safe channel.

Yes ▾

Tips:

- For a quick test, try 8 qubits, 1 shot, 12-character length.
- Check "Lowercase", "Uppercase", and "Digits" for a balanced password.
- Use **Validate Common Patterns** to avoid extremely common combos.
- Enabling QKD adds a step to simulate a quantum-safe key exchange (slightly longer process).

Generate Password

Figure 8.3.1: Taking Input from User

CHAPTER 9: CONCLUSIONS

9.1 CONCLUSIONS

The Quantum-Resistant Password Generation System provides a robust solution to the security challenges posed by quantum computing. By integrating Quantum Neural Networks (QNNs) and Quantum Random Number Generators (QRNGs), our system generates highly secure passwords that exhibit strong entropy and unpredictability. Through rigorous entropy validation using Von Neumann entropy, we ensure that the generated passwords are resistant to classical and quantum brute-force attacks. This research significantly contributes to the advancement of post-quantum cryptographic security.

Furthermore, the implementation of entropy validation, security assessment, and cryptographic strengthening enhances the security of generated passwords beyond traditional methods. The use of Quantum Random Number Generators ensures true randomness, making our system far superior to conventional password generation techniques. By adopting an Agile Software Development Life Cycle (SDLC) Model, we have ensured continuous improvement and adaptation to evolving cybersecurity threats.

The findings of this research demonstrate that quantum-resistant password generation is not only feasible but also necessary for the future of cybersecurity. As quantum computing advances, traditional password security methods will become obsolete, necessitating the widespread adoption of quantum-enhanced authentication mechanisms. This system lays the groundwork for future advancements in quantum-secure authentication frameworks, ensuring robust security for sensitive data and critical applications.

9.2 FUTURE WORK

- Enhancing Quantum Algorithms: Further improvements in QNN architectures to optimize password strength and entropy.
- Integration with Quantum Key Distribution (QKD): Implementing QKD protocols to enhance security in password transmission.
- Real-World Deployment: Adapting the system for commercial authentication solutions and enterprise security applications.
- Scalability Improvements: Ensuring the model can efficiently generate secure passwords at scale for high-demand environments.
- Machine Learning Enhancements: Incorporating AI-driven techniques to analyze and prevent security vulnerabilities in generated passwords.

- Multi-Factor Authentication (MFA) Support: Combining quantum-resistant passwords with biometric or hardware authentication for added security.

9.3 APPLICATIONS


- Cybersecurity and Authentication Systems: Used in login systems for banks, government agencies, and enterprise applications to secure user authentication.
- Financial Transactions and Blockchain Security: Ensuring safe cryptographic key generation and secure authentication in digital transactions.
- Quantum-Secure Data Encryption: Enhancing password-protected encryption techniques to safeguard sensitive information from quantum threats.
- Cloud Computing Security: Strengthening authentication methods for cloud-based services using quantum-resistant passwords.
- Military and Defense Applications: Implementing quantum-secure password generation for classified communications and cybersecurity operations.
- IoT and Smart Device Security: Protecting connected devices and IoT networks from potential cyber threats using secure password authentication mechanisms.

APPENDIX A

Details of paper publication:

3/6/25, 10:36 AM

Pimpri Chinchwad College of Engineering Mail - Fwd: Acceptance Notification - IEEE GINOTECH 2025



TYCOA291 - UTKARSHA LATE <utkarsha.late22@pccoepune.org>

Fwd: Acceptance Notification - IEEE GINOTECH 2025
2 messages

Amogh Chandragiri <amoghchandragiri@gmail.com>
To: H830 Aryan Baheti <aryan.baheti21@pccoepune.org>, utkarsha.late22@pccoepune.org, SYCOA33 Akshay Chaudhari <akshay.chaudhari21@pccoepune.org>

Mon, Mar 3, 2025 at 8:19 PM

Amogh Chandragiri
4th year (2025 batch)
Computer Engineering
Pimpri Chinchwad College of Engineering (PCCOE)
Mob no: 9146241606
Email Id: amoghchandragiri@gmail.com
Linkedin link: [Amogh Chandragiri | LinkedIn](#)
Github link: [Amoghchandragiri \(Amogh Chandragiri\) \(github.com\)](#)

----- Forwarded message -----
From: **Microsoft CMT** <email@msr-cmt.org>
Date: Mon, 3 Mar, 2025, 7:05 pm
Subject: Acceptance Notification - IEEE GINOTECH 2025
To: Amogh Chandragiri <amoghchandragiri@gmail.com>

Dear Amogh Chandragiri

Paper ID / Submission ID : 487

Title : Quantum-Resistant Password Generation A Comprehensive Model with QNN, Simplified Entropy, and QKD Integration

Greeting from IEEE GINOTECH 2025 Hosted by DY Patil Institute of Technology , Pimpri , Pune

We are pleased to inform you that your paper has been accepted for the Oral Presentation as a full paper for the- "IEEE 2025 Global Conference in Emerging Technology , Pune , Maharashtra ,India with following reviewers' comment.

All accepted and presented papers will be submitted to IEEE Xplore for the further publication.

You should finish the registration before deadline, or you will be deemed to withdraw your paper:

Complete the Registration Process (The last date of payment Registration is 08 MARCH 2025)

Payment Links

For Indian Authors: <https://rzp.io/rzp/1jVDnV5>

For Foreign Authors: Given soon

(Select Stripe Payment while paying, enter your paper id, title in buyer detail)

Further steps like IEEE PDF xpress and E copyright will be given later once registration is over after the deadline.

<https://mail.google.com/mail/u/0/?ik=a3e3b58369&view=pt&search=all&permthid=thread-f:1825584869422477069&siml=msq-f:18255848694224770...> 1/4

Note :

1. Any changes with the Author name, Affiliation and content of paper will not be allowed after acceptance.
2. This is Hybrid Conference, both online and physical presentation mode is available,

The reviews are below.

===== Review 1 =====

*** Relevance and timeliness: Rate the importance and timeliness of the topic addressed in the paper within its area of research.

Good (4)

*** Technical content and scientific rigour: Rate the technical content of the paper (e.g.: completeness of the

analysis or simulation study, thoroughness of the treatise, accuracy of the models, etc.), its soundness and scientific rigour.

Valid work but limited contribution. (4)

*** Novelty and originality: Rate the novelty and originality of the ideas or results presented in the paper.

Some interesting ideas and results on a subject well investigated. (3)

*** Quality of presentation: Rate the paper organization, the clearness of text and figures, the completeness and accuracy of references.

Well written. (4)

*** Strong aspects: Comments to the author: what are the strong aspects of the paper?

In this paper, a new application is proposed. In addition, the paper discusses the different aspects of technology in the field of Programming testing to guarantee programming quality

*** Weak aspects: Comments to the author: what are the weak aspects of the paper?

*** Recommended changes: Please indicate any changes that should be made to the paper if accepted.

The paper should be more result oriented.

===== Review 2 =====

*** Relevance and timeliness: Rate the importance and timeliness of the topic addressed in the paper within its area of research.

Acceptable (3)

*** Technical content and scientific rigour: Rate the technical content of the paper (e.g.: completeness of the

Analysis or simulation study, thoroughness of the treatise, accuracy of the models, etc.), its soundness

<https://mail.google.com/mail/u/0/?ik=a3e3b58369&view=pt&search=all&permthid=thread-f1825584869422477069&siml=msg-f18255848694224770...> 2/4

3/6/25, 10:38 AM

Pimpri Chinchwad College of Engineering Mail - Fwd: Acceptance Notification - IEEE GINOTECH 2025

and scientific rigour.

Valid work but limited contribution. (3)

*** Novelty and originality: Rate the novelty and originality of the ideas or results presented in the paper.

Some interesting ideas and results on a subject well investigated. (3)

*** Quality of presentation: Rate the paper organization, the clearness of text and figures, the completeness and accuracy of references.

Readable, but revision is needed in some parts. (3)

*** Strong aspects: Comments to the author: what are the strong aspects of the paper

The paper reviews different aspects of the concern

*** Weak aspects: Comments to the author: what are the weak aspects of the paper?

The future contribution of the paper should be mentioned in the paper

The presentation quality of the paper has to be improved significantly. The details of the author's contribution are too little.

*** Recommended changes: Recommended changes. Please indicate any changes that should be made to the paper if accepted...

The presentation quality of the paper has to be improved significantly

The detail of simulation/Result is too little.

===== Review 3 =====

*** Relevance and timeliness: Rate the importance and timeliness of the topic addressed in the paper within its area of research.

Good (4)

*** Technical content and scientific rigour: Rate the technical content of the paper (e.g.: completeness of the

analysis or simulation study, thoroughness of the treatise, accuracy of the models, etc.), its soundness and scientific rigour.

Marginal work and simple contribution. Some flaws. (2)

*** Novelty and originality: Rate the novelty and originality of the ideas or results presented in the paper.

Minor variations on a well investigated subject. (2)

*** Quality of presentation: Rate the paper organization, the clearness of text and figures, the completeness and accuracy of references.

Okay. Well Written

*** Strong aspects: Comments to the author: what are the strong aspects of the paper

<https://mail.google.com/mail/u/0/?ik=a3e3b58389&view=pt&search=all&permthid=thread-f:1825584889422477089&siml=msg-f:18255848894224770...> 3/4

3/8/25, 10:38 AM

Pimpri Chinchwad College of Engineering Mail - Fwd: Acceptance Notification - IEEE GINOTECH 2025

In this research, a new scenario of the technology has been introduced; research may be interesting for readers in this area.

*** Weak aspects: Comments to the author: what are the weak aspects of the paper?

Paper should be in IEEE format.

NO ANY MAJOR REVISION

1. The presentation should be substantially improved.
2. The presentation of the proposed algorithm is too conceptual, and details of how to operate the proposed algorithm in practice should be clearly elaborated.

Thanks, and Regards,
Technical Program Committee Chair
IEEE GINOTECH
ginotechconf@gmail.com
+91-8767682587

To stop receiving conference emails, you can check the 'Do not send me conference email' box from your User Profile.

Microsoft respects your privacy. To learn more, please read our [Privacy Statement](#).

Microsoft Corporation
One [Microsoft Way](#)
[Redmond, WA 98052](#)


TYCOA291 - UTKARSHA LATE <utkarsha.late22@pccoepune.org>
To: swati.shinde@pccoepune.org

Mon, Mar 3, 2025 at 8:25 PM

[Quoted text hidden]


APPENDIX B

Plagiarism Report of project report.

 Page 1 of 56 - Cover Page Submission ID trn:oid::3117:447388351


Aryan Baheti

final plag.pdf

 Pimpri Chinchwad College of Engineering

Document Details

Submission ID trn:oid::3117:447388351	52 Pages
Submission Date Apr 9, 2025, 11:18 PM GMT+5:30	7,439 Words
Download Date Apr 9, 2025, 11:19 PM GMT+5:30	46,580 Characters
File Name final plag.pdf	
File Size 2.2 MB	

 Page 1 of 56 - Cover Page Submission ID trn:oid::3117:447388351





8% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.



Filtered from the Report

- Bibliography
- Quoted Text
- Abstract

Match Groups

-  **28 Not Cited or Quoted 8%**
Matches with neither in-text citation nor quotation marks
-  **0 Missing Quotations 0%**
Matches that are still very similar to source material
-  **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 7%  Internet sources
- 4%  Publications
- 0%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

REFERENCES

- [1] NIST, “Post-Quantum Cryptography: Round 3 Submissions”, U.S. Department of Commerce, 2019. [Online]. Available: <https://csrc.nist.gov/Projects/post-quantum-cryptography>
- [2] A. Cooper, B. Smith, “Post-Quantum Cryptography Standardization Status”, NIST Computer Security Resource Center, 2021.
- [3] European Union Agency for Cybersecurity (ENISA), “Preparing Europe for Post-Quantum Cryptography”, ENISA Reports, 2022.
- [4] D. Florencio, C. Herley, “A Large-Scale Study of Web Password Habits”, ACM International World Wide Web Conference, 2016, pp. 657–666.
- [5] T. Hunt, “Have I Been Pwned: An Effective Approach to Password Security”, Troy Hunt Blog, 2019, Accessed 2023.
- [6] K. Li, S. Zhou, “Quantum-based Approaches to Password Generation”, Quantum Information and Computation, vol. 20, no. 4, 2020, pp. 95–108.
- [7] J. Hebblewhite, T. Dahlberg, “Quantum Neural Key Distribution: A Novel QNN-based Protocol”, Quantum Inf. Process, 2019.
- [8] Z. He, M.S. Elizarov, N. Li, F. Xiang, A. Fratalocchi, “Quantum-Activated Neural Reservoirs for Authentication”, Nature Quantum Information, vol. 8, 2022, pp. 77–88.
- [9] N. Gisin, G. Ribordy, W. Tittel, H. Zbinden, “Quantum Cryptography”, Rev. Mod. Phys., vol. 74, no. 1, 2002, pp. 145–195.
- [10] H.-K. Lo, M. Curty, K. Tamaki, “Secure Quantum Key Distribution”, Nature Photonics, vol. 8, 2014, pp. 595–604.
- [11] V. Scarani, H. Bechmann-Pasquinucci, N.J. Cerf, et al., “The Security of Practical Quantum Key Distribution”, Rev. Mod. Phys., vol. 81, no. 3, 2009, pp. 1301–1350.
- [12] Y. Tang, “Quantum Multi-Factor Authentication in Enterprise Systems”, IEEE Transactions on Information Forensics and Security, 2019.
- [13] D. Ma, M. Luo, “Practical Quantum Random Number Generator Devices and Their Integration”, Optics Express, vol. 26, 2018, pp. 28367–28376.
- [14] S.S. Tannu, M. K. Qureshi, “Mitigating Measurement Errors in Quantum Computers by Exploiting State-Dependent Bias”, MICRO-52 Proceedings, 2019.
- [15] V. Vedral, “The Role of Negativity in Quantum Entanglement Measures”, Rev. Mod. Phys., vol. 74, 2002, pp. 197–208.
- [16] RockYou Inc., “RockYou Password Dataset (Leak 2009)”, Security Analysis Archive, 2009.

- [17] CrackStation, “CrackStation’s Password Hash Dataset”, Online Resource, 2009. [Online]. Available: <https://crackstation.net/>
- [18] K. Wang, Z. Zhang, “Post-Quantum Password Strategies for Common Users”, IACR ePrint Archive, 2018.
- [19] L. Chen, S. Jordan, Y.-K. Liu, et al., “Report on Post-Quantum Cryptography”, NIST IR 8105, 2019.
- [20] K. Li, J. Zhao, G. Zhang, “Adaptive QNN for Key Generation: Integrating Seeds and Entanglement”, Quantum Inf. Comput., 2021.
- [21] IBM Quantum, “IBM Quantum Cloud Services - Qiskit Runtime”, IBM Official Documentation, Accessed 2023.
- [22] L.K. Grover, “A Fast Quantum Mechanical Algorithm for Database Search”, Proc. 28th Annual ACM Symp. Theory of Computing, 1996, pp. 212–219.
- [23] P.W. Shor, “Algorithms for Quantum Computation: Discrete Logarithms and Factoring”, Proc. 35th Annual Symposium on Foundations of Computer Science, 1994, pp. 124–134.
- [24] K. Chen, D. Li, “Entropic Approaches to Post-Quantum Password Resilience”, Cryptology ePrint Archive, 2022, Paper 732.
- [25] F. Wu, X. Li, “Noise Mitigation Techniques for Practical Quantum Circuits”, Quantum Inf. Process, vol. 19, 2020, pp. 297–315.
- [26] T. Maldonado, R. Garcia, “Evaluation of Real Quantum Hardware for Randomness Generation”, IEEE Access, vol. 9, 2021, pp. 117085–117100.
- [27] L. Coyle, R. Koenig, “Measuring Mixed States in QNN Outputs: A Partial Trace Approach”, Phys. Rev. A, vol. 95, 2017, 042312.
- [28] M. Eslami, A. Samadi, “Post-Quantum Multi-Factor Cyberdefense: A Roadmap”, ACM Comput. Surv., vol. 51, no. 6, 2019, art. 123.
- [29] G. Dominguez, “Entropy Explained: Classical, Quantum, and Approximations”, Journal of Cryptographic Engineering, vol. 8, 2020, pp. 335–350.
- [30] B. Beaulieu, M. Davis, “Comparative Analysis of SHA-2 vs. SHA-3 in Post-Quantum Settings”, IET Inf. Secur., vol. 12, no. 4, 2018, pp. 329–337.