National College of Ireland

Secure Communication Using Quantum Key Distribution and Honey Encryption

MSc Research Project
Cyber Security

Taraka Raghavendra Sai Panchakarla
Student ID: x22150951

School of Computing
National College of Ireland

Supervisor: Dr Rohit Varma

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Taraka Raghavendra Sai Panchakarla
Student ID:	x22150951
Programme:	MSc in Cyber Security **Year:** 2023-2024
Module:	MSc Research Project
Supervisor:	Dr Rohit Varma
Submission Due Date:	25 April 2024
Project Title:	Secure Communication Using Quantum Key Distribution and Honey Encryption
Word Count:	6080 **Page Count** 25

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Taraka Raghavendra Sai Panchakarla

Date: 25 April 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Secure Communication Using Quantum Key Distribution and Honey Encryption

Taraka Raghavendra Sai Panchakarla
x22150951

Abstract

Nowadays security of presenting for eavesdropping, brute-force attacks, information, and spoofing etc are decreasing. To increase the protection, I also need to use a good combination of cryptography methods. In such a combination I am going to look at the combination of Quantum Key Distribution (QKD) and Honey Encryption (HE) are two advanced cryptographic mechanisms widely used in data security. The primary goal of this study is to critically assess the efficacy, implementation challenges, and practical applications of QKD and HE in terms of efficiency security against eavesdropping and brute force attacks. In this paper, I critically analyze their theoretical fundamentals, effectiveness, and their practical applications. I find out that the fundamental principles and mechanisms underlying the two cryptographic techniques contribute to their respective security advantages. The main contribution of this paper is to enlighten policymakers, industry leaders, and the public on the need to commit research resources and apply the smart cryptography option to protect the digital environment. This research seeks to make its contribution to the creation of stronger, more resilient, and quantum-immune digital environments.

1 Introduction

In this digital age of today, there is a rapid increase in internet and technological usage that experienced a tremendous amount of data growth and communication such as input validation, session management, authentication, and authorization, should be considered. The role of cryptosystems to protect and secure information is more indispensable than ever. Quantum computing has introduced entirely new threats in this arena. It signals an innovative approach to security, which may include different cryptographic ideas and implementations. The constantly changing security landscapes demands rethinking existing encryption technologies and producing new cryptographic techniques. As for the field of physics, there are the most prospect QKD (Quantum Key Distribution) and Honey Encryption that can be good for it. As the opening chapter of the discourse this part will lay a foundation for a detailed analysis of these technologies which will include their importance, working principles, positive contribution, or risks in the matter of increasing cybersecurity.

The primary goal of the chosen study will be to get deeply into the details of Quantum Key Distribution (QKD) and Honey Encryption discourse supported by the critical analysis of their theoretical fundamentals, effectiveness, and practical applications. Through this discussion, the extent of increasing cryptographic techniques to be malicious resistant in the present context quantum computing and impactful cyber-attacks will be within range. The investigation rests upon the understanding of the depth of the digital threats that these days, only complex countermeasures deployed can guarantee the security of confidential data, integrity plus other-related information.

1.1 Aim and Objectives

The primary aim of this study is to critically assess the efficacy, implementation challenges, and practical applications of Quantum Key Distribution (QKD) and Honey Encryption as advanced cryptographic mechanisms.

- To Understand the Principles Underlying QKD and Honey Encryption
- To Evaluate the Security Advantages on eavesdropping and brute force on using the combination of QKD and Honey Encryption

1.2 Research Questions

- How does the combination of QKD and Honey Encryption contribute in terms of efficiency security against eavesdropping and brute force attacks than other traditional cryptographic methods.

1.3 Rationale

QKD and Honey Encryption mechanism have become essential in the contemporary cryptography world. This study can be traced to the growing cry for cryptographic security to be faster and stronger in the digital world. This research pursues the understanding that the NIST may no longer be stronger than the advancement of computers with a quantum capability, especially in terms of their computational abilities. Hence, reinvestigation of QKD and Honey Encryption is concerned with not only timely but absolute importance, focusing on the prevention of the faults resulting from modern technology appearance.

Unlike traditional cryptographic systems, the fundamental building block of quantum cryptography's security idea is physics, not math. Quantum cryptography uses the intrinsic properties of quantum physics to secure and transmit data in an impenetrable way. QKD embodies a revolutionary approach to communication security, the quantum principles underlying it being the foundation for guaranteed secure and physically unattackable data transfer. The main feature of QKD is its capability of such keys that any disturbance during the quantum communication would alter the quantum state and so the communicating parties would be informed about any potential attacks (ERIK LIDBJÖRK, 2023).

In QDK I am going to use a protocol called BB84 which serves Heisenberg's uncertainty principle and was proposed by Charles Bennett and Gilles Brassard in 1984. By using the BB84 protocol, Alice can randomly transmit Bob a secret key by sending him a string of photons with the private key contained in their polarization. The no-cloning theorem states that Eve cannot measure these photons and communicate them to Bob without changing the photon's state in a way that can be observed. On the other hand, Honey Encryption Works as an additional method which targets data storage as opposed to the key used whereas the key was the target beforehand. Honey Encryption simulator provides detailed statistical analyses of attackers' behavior by mimicking incorrect decryption outputs for every save (Anusuya Devi V, 2021).

This, in fact, not only scandalizes offenders but also, shortens harm by easing information if the offenders get the ciphering key. The selection of QKD and Honey Encryption for this study is predicated on their potential to address two critical aspects of digital security: privacy channels, secure key distribution, and data encryption. The integration of QKD into the current digital security frameworks produces a completely new weapon in the fight against both by-product and quantum computational threats vulnerabilities. The study is striving to address the issues in a comprehensive way. To that end, it searches for the ways how QKD

and Honey Encoding can be integrated into the present-day systems and platforms and can be used in the future.

Through exploring their principal roots, analyzing their practical value, and overcoming the barrier in their implementation, this research seeks to make its contribution to the creation of stronger, more resilient, and quantum-immune digital environment. This investigation which not only strives to deepen the scholarly literature on the technologies but also to enlighten policymakers, industry leaders and public on the need to committing research resources and applying the smart cryptography option to protect the digital environment.

2 Literature Review

2.1 Related Works

Several researchers have attempted to encrypt communications by using novel architectures and methods. Researchers have explored many different approaches to determine which can better handle security-related concerns. Below, I critically review a few research publications by organizing them according to the technology that is used to secure communication.

2.1.1 Using Blockchain Technology

According to Kaoutar Elkhiyaoui, Romain Rouvoy, and Lionel Seinturier in Privacy and Security in Decentralized Energy Trading through Multi-signatures, Blockchain and Anonymous Messaging Streams it is said Both private person-to-person messaging and message broadcasting are supported by the PriWatt system. The system shields parties from passive eavesdropping by obscuring non-content data or by masking the identities of individuals communicating through the assignment of unique strings of 36 alphanumeric characters. Peers communicate with one another by forwarding messages based on their best efforts. All active nodes therefore receive all messages, and each node tries to decrypt each message using its private key. As it is not safe, I am not considering it to be used so much (N. Z. Aitzhan, 2016).

2.1.2 Using ECC, RSA, and AES Algorithms

By Seyyed Keyvan Mousavi, Ali Ghaffari in Data cryptography on the Internet of Things using the artificial bee colony algorithm in a smart irrigation system, told that an attacker can eavesdrop by network monitoring of the users, and the attacker can use Eq. Then, the attacker tries to obtain useful and meaningful information from the transmitted data. In other words, the eavesdropper tries to decrypt the ciphertext by ECC, RAS and AES using the private key. To do so, the attacker must extract the XOR from ECC, RAS and AES (Seyyed Keyvan Mousavi, 2021).

2.1.3 Using Quantum Cryptography

In the study conducted by Ralegankar, V.K. et al. (2022), the many aspects that contribute to communication security for mission-specific applications were analysed. Here, they leverage the benefits of quantum cryptography and go beyond its potential for boosting data throughput and data volume. BB84, an extremely secure quantum cryptography algorithm that is distinct from the currently employed conventional cryptographic algorithms, was specially adopted by them. This study is centred around the seven creative architectures that they offer to improvise communication. An overview of the state of QKD today, its applications, and possible future advancements like faster QKD are provided by author

Sasaki, M. (2018). Manage massive volumes of data in a long-lasting storage network system and apply security in a real QKD protocol implementation.

Parameter	Symmetric Encryption	Asymmetric Encryption	Hash Encryption	Quantum Key Distribution (QKD)
Security	Can be broken if the secret key is compromised	Can be broken if a private key is compromised	Not meant for encryption but for integrity check	Considered to be one of the most secure forms of encryption
Key Distribution	The secret key must be shared securely	Public key can be shared openly, the private key must be kept secret	Not applicable	Allows for secure key distribution without the need for a secure initial channel
Quantum-resistant	Not Quantum-resistant	Not Quantum-resistant	Not applicable	Quantum-resistant

Table 1: Comparison of using QKD over Symmetric Encryption, Asymmetric Encryption and Hash Encryption

The RSA scheme will become unsecured right away. This holds for any encryption method that an effective quantum algorithm can break. Even after a quantum computer is constructed, post-quantum algorithms classical methods thought to be impervious to efficient quantum computing continue to face increasing insecurity due to advancements in both hardware and software. Therefore, it is necessary to regularly update the key length that is deemed secure for applications relying on classical (and post-quantum) techniques like RSA. In contrast, QKD's security remains constant over time due to its information-theoretic security, which means it is independent of the capability of modern computers (Agarwal, 2023).

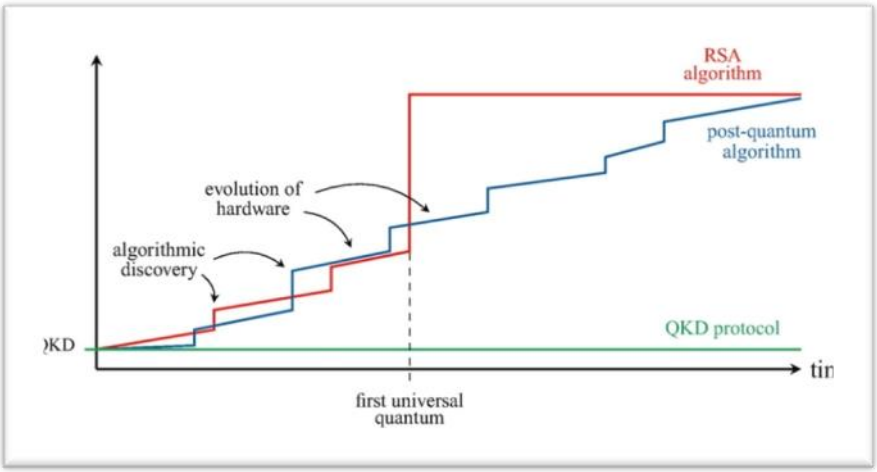

Figure 1: Comparing RSA, Post-QDK & QDK

2.2 Literature gap

The research background for introducing quantum key distribution (QKD) and Honey encryption (HE) to the field of cyber security is wide and changed, reflecting the eccentric, and quickly advancing nature of the security space. The origins of QKD can be traced back to the spearheading work of Cao et al. (2022), who proposed a strategy (BB84) to utilize quantum mechanical estimations for secure communications. These advancements represent a major departure from the literary tradition based on advanced rationale. The concept of QKD

is based on the law of vulnerability, which states that by measuring the estimated properties of a molecule, the shape of the particle changes its shape. These rules guarantee that assessments can be distinguished at the pre-market organization and will hence be shown in a secure and non-destructive way. For a long time, the advancement of QKD has appeared in distinctive patterns and changes, with analysts attempting to overcome practical challenges such as particular confinements and integration in communication regions.

Launched by Yu et al (2022), Honey Encryption represents a modern heading within the world of encryption. Unlike conventional encryption strategies, which produce deceptions through failed decryption endeavors, HE produces genuine but false data. This approach enormously progresses security by locking in the assailant and preventing leverage. The HE concept is based on the thought of hiding worldwide security information to permit translation. This strategy has been demonstrated to guarantee security in an assortment of applications including secure watchword databases and touchy individual data.

There is a large gap in the literature in the field of cryptography research, especially for quantum key distribution (QKD) and honey encryption (HE). The method focuses on the integration and practical application of two advanced cryptographic developments. Current thinking has explored the individual perspectives of QKD and HE, but their ability to integrate into a security framework has been understudied and less documented.

Quantum Key Distribution, founded on quantum mechanics, provides unbreakable security for key storage. Pioneering work by researchers such as Damasceno et al (2023) and later Huang et al. (2022) established the conceptualization of QKD. However, most of the literature focuses on theory, exploring the ideas of QKD and ignoring the problems of common sense related to a global context. Key issues such as integration with existing information infrastructure, flexibility, and cost-effectiveness of QKD systems have not yet been addressed. These areas highlight the need for research to move QKD from predictive authentication to a proper cybersecurity tool.

3 Research Methodology

The research approach section provides a thorough account of the specific steps taken to guarantee the research's high level of credibility and validity, where the strategies responsible for the collection, analysis, and interpretation of the data are elaborated. They outline how the samples were collected, with an emphasis on the methods that ensure the sample is representative of the population under study, and explain the techniques used to reduce bias and error. This section deals with either the theoretical framework or the model or both that form the basis of the study, which is aimed at showing how these models determine the study design and result interpretation.
In addition, the Research Design and Approach are also where the methodological novelties of the study are to be presented. It illustrates that the study breaks new ground by developing a fresh and original way of the matter or further develops an existing methodology. This can be done through the modification of age-old techniques to modern circumstances, the creation of unique tools for data collection, or the use of recent analytical methods.

3.1 Research Design and Approach

The Procedure and Methods Applied section is the heart of any scholarly study, yet it is an indispensable part of which should be taken into consideration because it presents a detailed description of the processes and approaches employed at the beginning of the research to the

data collection and analysis at the end. This careful documentation is critical for ensuring the transparency and reproducibility of research as it makes the methods used and the results obtained available for scrutiny by peers and future researchers, and hence, consolidates the validity and reliability of the findings. Using the BB84 algorithm, I generated a secret key for both encryption and decryption. The algorithm's authenticity, unpredictability, and security are increased by the mixed concept quantum and block cipher, making it more difficult for attackers to decode the original message (Agarwal, 2023).

In the Figure 2 it is showing the generation of the secret key from the plaintext stage in fist set the Plaintext will be made into bits and those bits are sent through in for of photons called bases theoretically. These bits and bases are converted into Qubits by using BB84 polarization scheme shown in Figure 6. These Qubits of the sender receiver and any other eaves are sent into the quantum channel to compare the qubits and work accordingly. If needed a secret key is generated based on the error correction level as per requirement. I am using Python software to implement the BB84 algorithms. Several scientific libraries are utilized in the BB84 protocol, which is used to implement the QKD, through Python programs such as Google Collaborator or Jupiter. To ensure that the system is as random as a true quantum system would be, the program was designed in a way that enables it to simulate a quantum environment. Identifying algorithms for eavesdropping and network monitoring can be used to secure data if Alice and Bob provide comparable sequences of measured values (NURUL T. ISLAM, 2017).

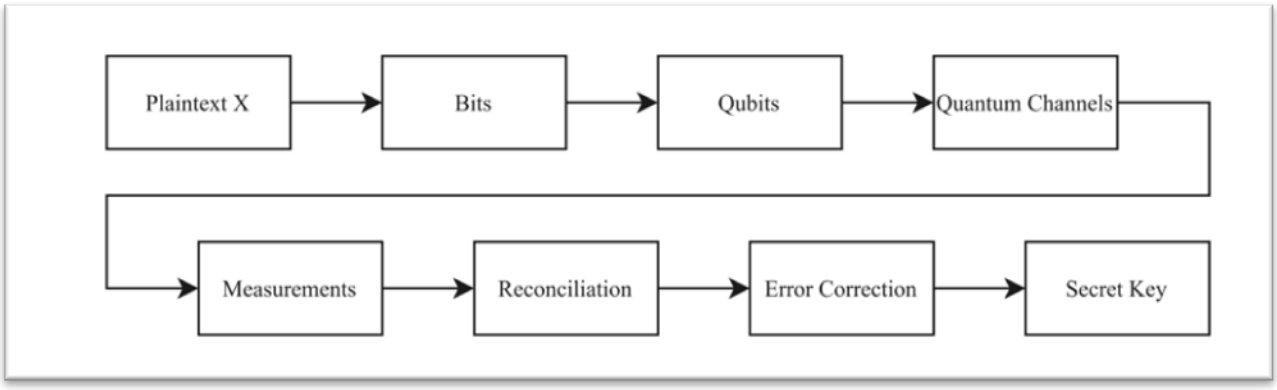

Figure 2: QDK Protocol

The goal is to create schemes that, even with all potential password and key combinations used, prevent attackers from recovering the secret key formed by using the QKD method above. I codify this strategy using honey encryption (HE), a novel cryptographic primitive. I used randomized key for encryption techniques to the Secret Key generated by QDK. HE decryption responds when a person tries to decrypt a ciphertext with the incorrect key will be where it diverges from traditional symmetric encryption methods. A plaintext that "looks" credible will be produced by decryption, as opposed to an error. HE can be used in some situations without giving attackers easy access to a user's public key of RSA. RSA-based client authentication, which permits access to a distant service via HTTPS or SSH, is a typical example. The client registers the matching public key with the distant service and keeps an RSA secret or private key (Thanda Win, 2018). Provide defence-in-depth if the client's system is passively infiltrated, practitioners advise encrypting the secret key beneath a password. However, an attacker can launch an offline brute-force assault against the encrypted secret key when using password-based encryption. In the similar way I wanted to you HE in combination of QKD which can also overcome the limitation which are said above related works.

Once the Secret Key has been generated using QKD process the Secret Key need to be given security by using honey encryption and decryption method. In this a password will be sent by the sender and will be encrypted by using functions like PBKDF2HMAC and SHA256 in my program once the receiver received the encrypted key it will be asked for the password if the receiver the access will be granted and the receiver can able to see the decrypted key sent by the sender if not the function will generate some random key and shares with user and generate alarm for sender (Wei Yin, 2017).

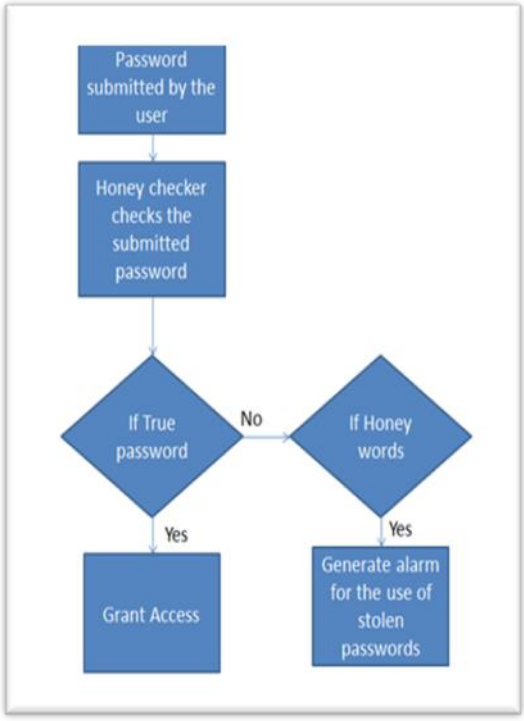

Figure 3: Honey encryption logic

3.2 Possibilities

There are 5 main possibilities of the program functioning based on the inputs given by receiver. They are as given below:

Possibility 1 - If the eavesdropping attack is true stop the generations of keys and alert the users.

Possibility 2 - If the eavesdropping attack is false continue with giving input of A and B bits and bases and generate Qubits if All Qubits of A and B are same B must receive the same directly.

Possibility 3 - If the eavesdropping attack is false continue with giving input of A and B bits and bases and generate Qubits if All Qubits of A and B are not match Users need to generate new qubits and A need to have alert for safety.

Possibility 4 - If the eavesdropping attack is false continue with giving input of A and B bits and bases and generate Qubits if All Qubits of A and B and only some Qubits matched A need to generate secret key and encrypt it by using password that need to be matched by the receiver password entry, if password is not matched an alarm will be sent, and a randomly generated key is displayed.

Possibility 5 - If the eavesdropping attack is false continue with giving input of A and B bits and bases and generate Qubits if All Qubits of A and B and only some Qubits matched A need to generate secret key and encrypt it by using password that need to be matched by the receiver password entry, if password is matched user can decrypt and know the key sent.

4 Design Specification

The design methodology in this research aims to use the BB84 protocol, Alice can randomly transmit Bob a secret key by sending him a string of photons with the private key contained in their polarization. The no-cloning theorem states that Eve cannot measure these photons and communicate them to Bob without changing the photon's state in a way that can be observed. Where the secret key is encrypted by using Honey Encryption technique and make sure that the Bob only can decrypt it my using password kept by Alice.

4.1 Proposed Algorithm of QKD

1. Check whether there is any eavesdropping attack on the device if true stop the generation of the key and alert the information sending user if false continue the below steps.
2. Two basis sequences are used (i) a rectilinear basis (y), (ii) diagonal basis (x)
3. Binary bit used are values 1 and 0.
4. Qubits are generated as per the comparison of the basis sequence and bits.
   - if the rectilinear basis (y) compared with bit value 0 it is called 0_p.
   - else if rectilinear basis (y) compared with bit value 1 it is called 45_p.
   - else if diagonal basis (x) compared with bit value 0 it is called 90_p.
   - else it is called as 135_p respectively as per the input values.
5. Now first let A generate both basis and bits of user length.
6. Next basis and bits are compared to get qubits.
7. Now perform same with B generate both basis and bits of user length.
8. And basis and bits are compared to get qubits as he guesses.
9. once both create their qubits of A and B as per the above qubit’s generation step, they are compared.
   - if all qubits of A and B are the same print as open the message.
   - else if no qubits are the same print do the generation of qubits again.
   - else if some quits are same and some or not same then perform an XOR operation between matched bits value and A remaining not matched bits.
10. The value of the XOR is secret quantum key which is a shared with B to know qubits.

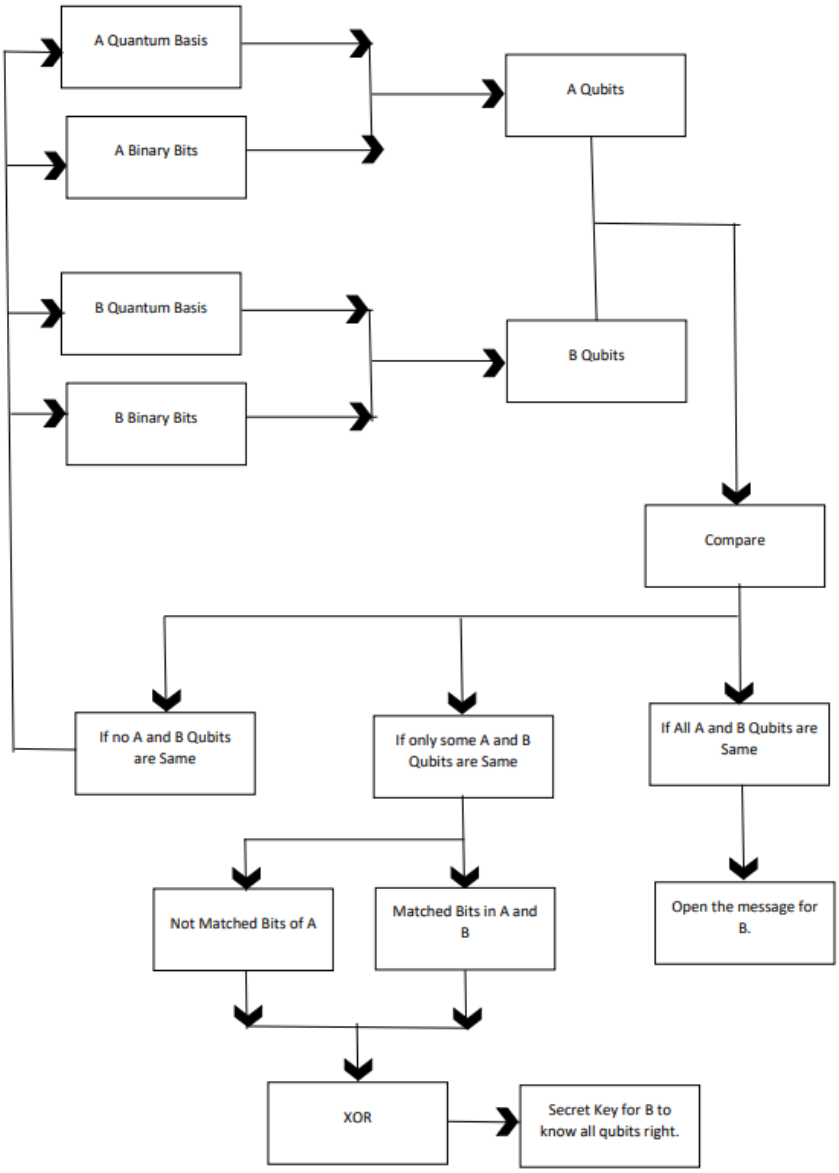

Figure 4: Flow Chart of QKD Algorithm

4.2 Proposed Algorithm of Honey Encryption

1. Secret quantum key for B is generated by using QKD algorithm.
2. Take a user password and generate a random key for encrypting the Secret quantum key which is generated.
3. The quantum key need to be encrypted and given encrypted message to B.
4. B need to enter the password kept by A
5. If password entered of B is not same as A throw an alert message of sender and generate a random key and display for the attacker deviation.
6. If password of A and B are same decrypt the encrypted message.
7. Once B get to know the Secret quantum key, he will get to know what the bits are he need to change regarding to open the actual message A wants to share.

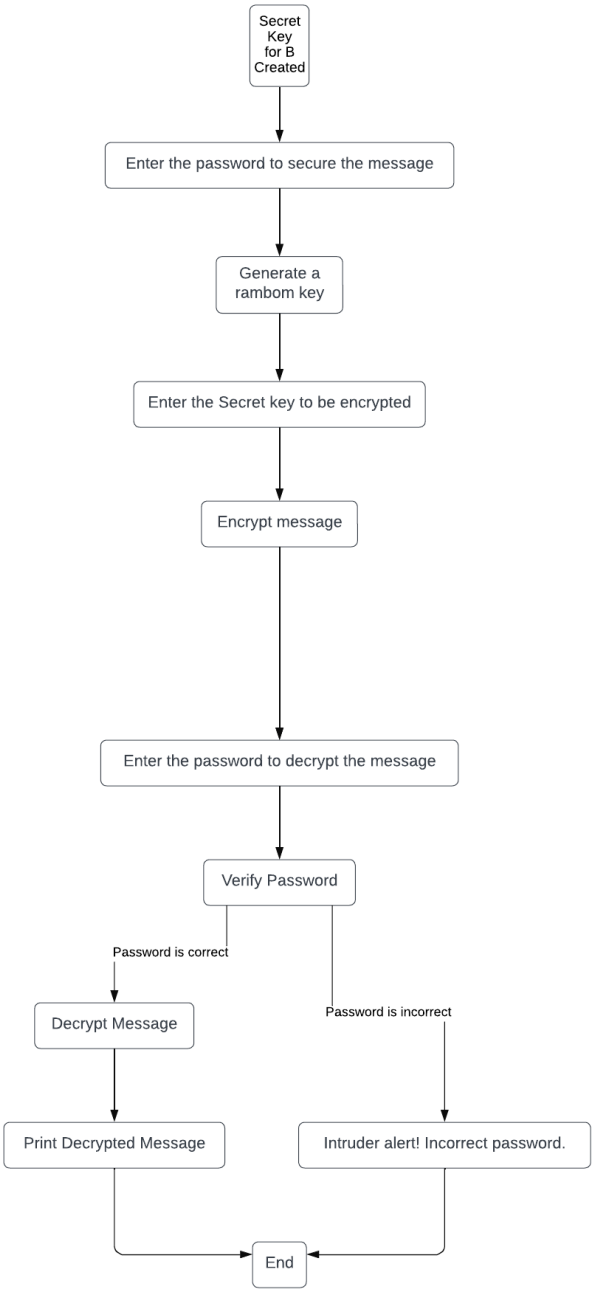

Figure 5: Flow Chart of Honey Encryption Algorithm

5 Implementation

This section discussed the implementation phase. A windows operating machine having the details of the following systems:

Processor	12th Gen Intel(R) Core (TM) i7-12700H 2.30 GHz
RAM	16.0 GB (15.7 GB usable)
System type	64-bit operating system, x64-based processor

Python 3.12.1 is used for coding. Pip version 24.0 is used. Jupyter notebook is used as a code editor. In the code there are six parts in total. The first three parts of generating Qubits, comparing Qubits, and generating key which is used to encryption of the secret key comes under performing QKD. Next fourth and fifth parts perform Honey Encryption and Decryption. To execute the above-mentioned parts, I need a main program which is the sixth part.

5.1 Generating Qubits

As per BB84 protocol Qubits are generated with bits and bases of the photons. By using those I need to generate Qubits of A and B (Anusuya Devi V, 2021).

BB84 Protocol Polarization Scheme		
Rectilinear Polarization Basis **(Y)**	0_p	45_p
Diagonal Polarization Basis **(X)**	90_p	135_p
Bit Value	0	1

Figure 6: BB84 polarization scheme

Pseudocode:

Algorithm 1 Qubit Generation

```
procedure GENERATEQUBITS(bits, basis)
    qubits ← []
    for each bit, base in zip(bits, basis) do
        if base equals 'y' then
            if bit equals '0' then
                append '0_p' to qubits
            else
                append '90_p' to qubits
            end if
        else if base equals 'x' then
            if bit equals '1' then
                append '135_p' to qubits
            else
                append '45_p' to qubits
            end if
        end if
    end for
    return qubits
end procedure
```

Figure 7: Pseudocode of Generating Qubits

5.2 Comparing of Qubits of A and B

The generated Qubits of A and B are compared to know if all Qubits of A and B are matching or else some only are matching else none are matching so that to continue with generation of secret key if needed (Anusuya Devi V, 2021).

Comparing of Qubits						
A basis	Y	X	X	Y	Y	X
A bits	1	1	0	0	0	1
A qubits	90_p	135_p	45_p	0_p	0_p	135_p
B basis	X	X	X	X	X	Y
B bits	1	1	1	0	1	0
B qubits	135_p	135_p	135_p	45_p	135_p	0_p
Comparing of A and B qubits		Matched bit (1)				

Figure 8: Comparing of Qubits

If only some Qubits in A and B are matched will perform an XoR operation between matched bits value and her remaining not matched bits of quantum key generation (Anusuya Devi V, 2021).

Step 1	Compare (Alice's Quantum Basis , Alice's Binary Bits) → Alice's Qubit
Step 2	Compare (Bob's Quantum Basis , Bob's Binary Bits) → Bob's Qubit
Step 3	If Alice's Qubit = Bob's Qubit Then the matched qubit's binary value will frame the secret key else Go to Step 1 and step 2 (repeately do the step 1 and step 2 process until matched qubit found)
Step 4	Matched bits (XOR) Alice's Not matched bit = Quantum Secret Key Value

Figure 9: Steps of generating Secret Key

Pseudocode:

Algorithm 2 Qubit Comparison

```
procedure CompareQubits(qubits_a, qubits_b)
    if length of qubits_a is not equal to length of qubits_b then
        return "Qubit lists have different lengths."
    end if
    matching_count ← 0
    secret_key ←''
    for each a, b in zip(qubits_a, qubits_b) do
        if a equals b then
            increment matching_count by 1
        else
            perform XOR operation on integers obtained by splitting a and b
by '_' and convert the result to string, append to secret_key
        end if
    end for
    if matching_count equals length of qubits_a then
        return "B have received the message."
    else if matching_count is greater than 0 then
        return "Secret Key: " concatenated with secret_key
    else
        return "Generate Qubits Again. Intruder Alert."
    end if
end procedure
```

Figure 10: Pseudocode of Comparing Qubits

Output of Qubits generation:

```
Enter the length of the bits:  4
Enter the basis for User A (e.g., 'xyxyxy'):  xyyx
Enter the bits for User A (e.g., '010101'):  1010
Enter the basis for User B (e.g., 'yyyyyy'):  yyyy
Enter the bits for User B (e.g., '110011'):  1111
User A Qubits: ['135_p', '0_p', '90_p', '45_p']
User B Qubits: ['90_p', '90_p', '90_p', '90_p']
```

Figure 11: Output of generating Qubits

5.3 Generating a Key to encrypt Secret key

In performing honey encryption to encrypt the secret I need to have a key and password through which the hashing of the secret key takes place in order to secure the info by making it to receive to the legitimate user.

Pseudocode:

Algorithm 3 Key Generation

```
procedure GENERATEKEY(password, salt)
    Initialize start_time
    kdf ← PBKDF2HMAC(algorithm=SHA256(), iterations=100000,
salt=salt, length=32, backend=default_backend())
    Start timer
    key ← base64_urlsafe_b64encode(kdf.derive(password.encode()))
    Stop timer
    Calculate key_generation_time = end_time - start_time
    return key, key_generation_time
end procedure
```

Figure 12: Pseudocode of generating Secret Key

5.4 Perform Honey Encryption and Decryption on Secret Key

A straightforward, all-purpose method for encrypting messages using low min-entropy keys—like passwords. When any of several erroneous keys are used to decipher the ciphertext produced by HE, fake plaintexts known as "honey messages" that appear genuine are produced. When there is insufficient entropy to survive brute-force attacks that attempt every key, HE offers security, which goes beyond traditional brute-force boundaries. This is one of the main advantages of HE (Ammar Abdul Majed Gharbi, 2021).

Encryption Pseudocode:

Algorithm 4 Message Encryption

```
procedure EncryptMessage(message, key)
    cipher ← Fernet(key)
    encrypted_message ← cipher.encrypt(message.encode())
    return encrypted_message
end procedure
```

Figure 13: Pseudocode for Honey Encryption

Decryption Pseudocode:

Algorithm 5 Message Decryption

```
procedure DecryptMessage(encrypted_message, key)
    cipher ← Fernet(key)
    decrypted_message ← cipher.decrypt(encrypted_message).decode()
    return decrypted_message
end procedure
```

Figure 14: Pseudocode for Decryption

Output of Encryption and Decryption:

```
Secret Key: 22190119
Enter the message to be encrypted with the secret key:  22190119
Secret Key: b'sqhYWZevgfj_jQKG54SYTVwWWmNewqXDPUXjIMRaxko='
Time taken to generate key: 0.031374216079711914 seconds

Encrypted Message: b'gAAAAABmC_s1LDRwhSOywUqjn-u_60YIF7s3UIhmeqdQg_BmZZeDGGNsHO1dpaqgyB0JTwGx9xO-94hTXcK89M7cYJte6tu7mQ=='
Enter the password to decrypt the message:  ram
Decrypted Message: 22190119
```

Figure 15: Output of Encryption and Decryption

6 Evaluation

An analysis of the suggested model is presented in this section. Evaluating the possibilities, Total execution time, Secret key generation time, encryption time, and decryption time are all calculated as part of the evaluation process. Calculating the throughput completes the performance study.

6.1 Evaluation of Possibilities

As I discussed above there are 5 possibilities mainly based on network between the sender and the receiver, inputs given by the receiver.

Possibility 1 - If the eavesdropping attack is true.

This possibility is based on the network channel between sender and the receiver. In theoretical practice the network is constantly monitored if in case of any malicious activities and eavesdropping the message will not be transferred and the sender will be alerted to be safeguarded without can loss. In the same way in our program as we cannot show the network, we are using a user given input to make eavesdropping attack there or not. If an eavesdropping attack is present the program needs to be stopped and must alert the sender. Which is shown in Figure 16.

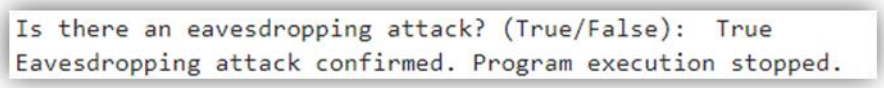

Figure 16: Output If the eavesdropping attack is true

Possibility 2 - If the eavesdropping attack is false and All Qubits of A and B are same.

If there are no signs of any attack of eavesdropping, then the bits and basis will be taken from users and qubits will be generated as shown in Figure 7. Once qubits of A and B are generated and compared, all the qubits are same the receiver need to receive the message as shown in Figure 17.

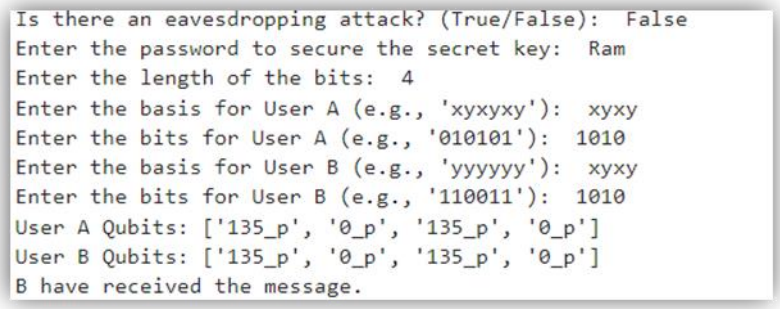

Figure 17: Output If the eavesdropping attack is false and All Qubits of A and B are same

Possibility 3 - If the eavesdropping attack is false and No Qubits of A and B are matched.

If there are no signs of any attack of eavesdropping, then the bits and basis will be taken from users and qubits will be generated as shown in Figure 7. Once qubits of A and B are generated and compared, if no qubit has been matched with the qubits of sender and the receiver there need to be an error message and tell sender and receiver to generate qubits again as shown in Figure 18.

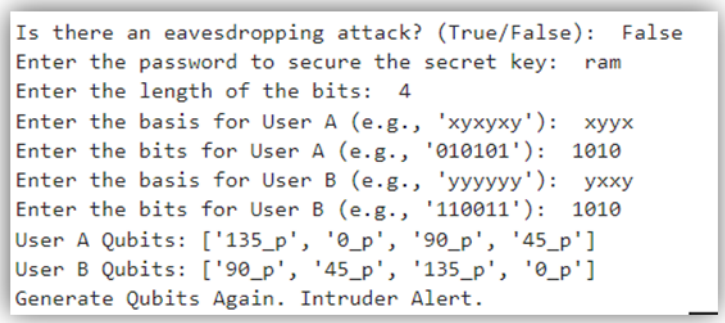

Figure 18: Output If the eavesdropping attack is false and No Qubits of A and B are matched

Possibility 4 - If the eavesdropping attack is false if A and B have only some Qubits matched A generated a secret key and encrypt it by using password if password is not matched.

If there are no signs of any attack of eavesdropping, then the bits and basis will be taken from users and qubits will be generated as shown in Figure 7. Once qubits of A and B are generated and compared, and if at least one qubit has been matched with the qubits of sender and the receiver. It needs to perform XOR operation between matched bits value and A remaining not matched bits and generate a secret key. Which is encrypted using honey encryption method for extra layer of security. To do so the sender will set a password and sends to encrypted message to the receiver. To decrypt it the receiver, need to enter the password. If the password does not match the sender the program will display some random key to make the user assume he got the key and alert as shown in figure 19.

Figure 19: Output If the eavesdropping attack is false if A and B have only some Qubits matched A generated a secret key and encrypt it by using password if password is not matched

Possibility 5 - If the eavesdropping attack is false if A and B have only some Qubits matched A generated a secret key and encrypt it by using password if password is matched.

If the password if the sender and receiver is matched, then the receiver can be able to decrypt the key and see the plain text of the key and know the qubits which he need to change to get the actual message the sender want the receiver to know as shown in Figure 20.

```
Is there an eavesdropping attack? (True/False):  Flase
Enter the password to secure the secret key:  Ram
Enter the length of the bits:  4
Enter the basis for User A (e.g., 'xyxyxy'):  xyxy
Enter the bits for User A (e.g., '010101'):  1010
Enter the basis for User B (e.g., 'yyyyyy'):  xxxx
Enter the bits for User B (e.g., '110011'):  1010
User A Qubits: ['135_p', '0_p', '135_p', '0_p']
User B Qubits: ['135_p', '45_p', '135_p', '45_p']
Secret Key: 4545
Enter the message to be encrypted with the secret key:  4545
Secret Key: b'DNmfxkHWYVO1pQ-iLSoCGoI4VvhQrl611QjqRFkWIiM='
Time taken to generate key: 0.0525670051574707 seconds

Encrypted Message: b'gAAAAABmDEfaWdo06zymcby77I_y7YHYyT7YtrK3RSTpjfwF6iQVYjHu8PLzSaVwhmZiUIRjfrcnN_Duhn9VLx3AJyBEVb1O-w=='
Enter the password to decrypt the message:  Ram
Decrypted Message: 4545
```

Figure 20: Output If the eavesdropping attack is false if A and B have only some Qubits matched A generated a secret key and encrypt it by using password if password is matched

6.2 Analysis of data bytes performance with respective to time

The graph in figure 21 will show the secret key bits generated per millisecond. It is evident to show the performance that the time taken for generating key bits.

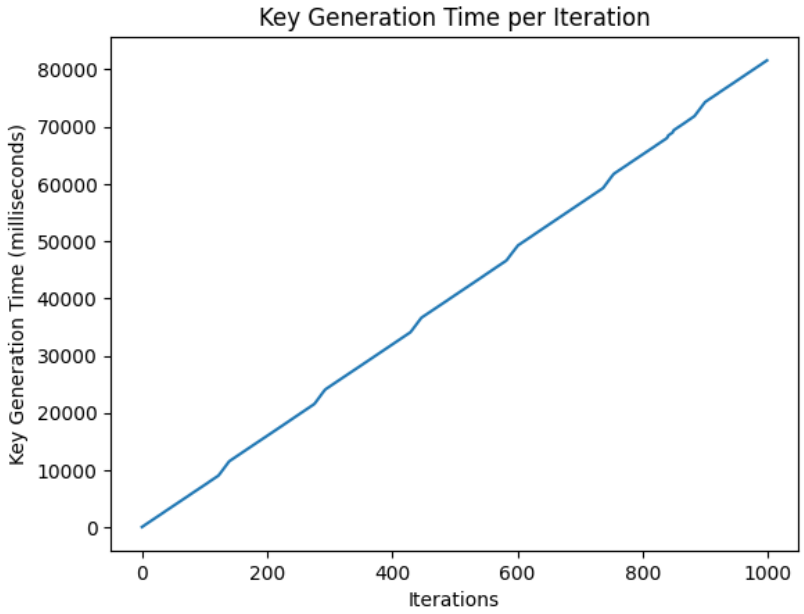

Figure 21 Key Bits Generated per millisecond

The graphs in Figure 22 and Figure 23 will show the time taken to the secret key to be encrypted and decrypted. The performance of the encryption and decryption times is clearly visible. It also demonstrates how much faster encryption and decoding are than with other conventional cryptography techniques like RSA.

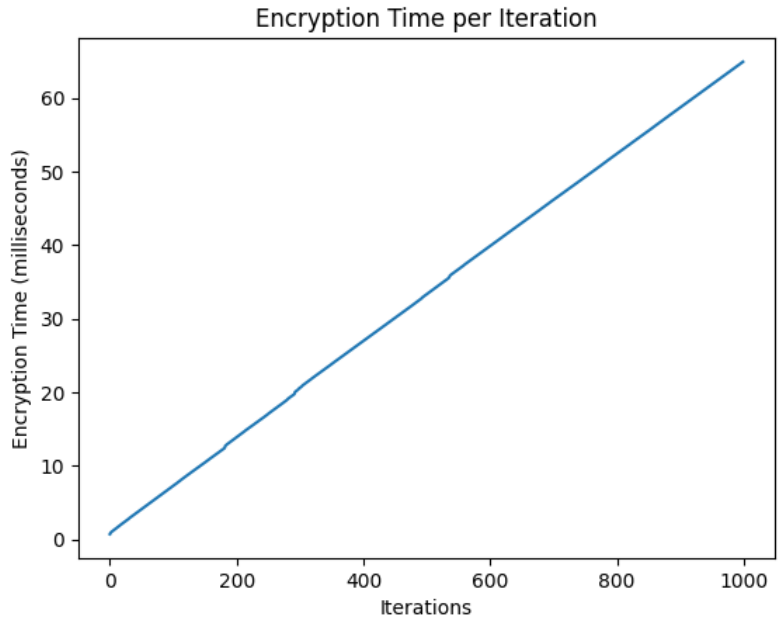

Figure 22 Time taken for encrypting

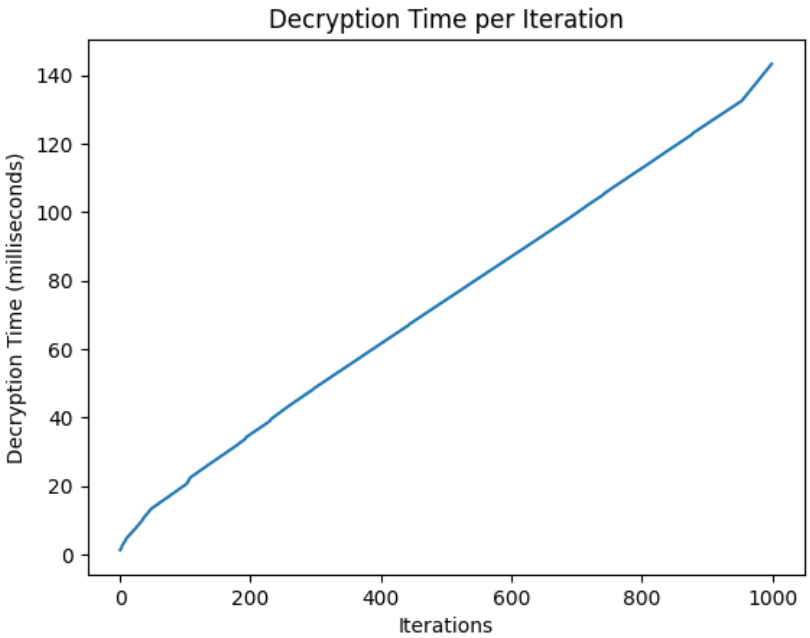

Figure 23 Time taken for decrypting

Now the analysis of the total execution size. Total Execution Time = Key Bits Generated Time + Encryption Time + Decryption Time + Message Retrieval. The graph in figure 24 shows the total execution time vs total bytes exchanged to the receiver.

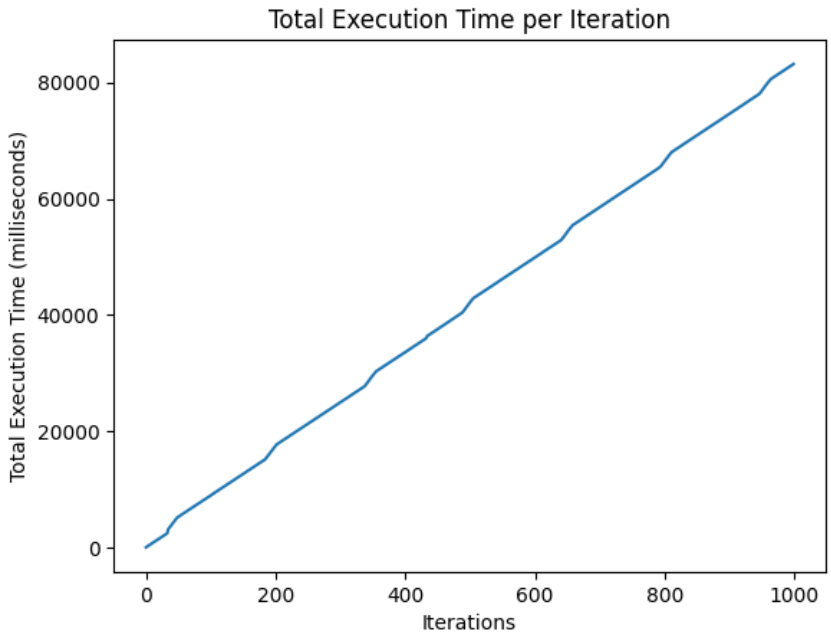

Figure 24 Total Execution Time vs Total bytes Exchanged

6.3 Analysis of DES, QKD and the proposed QKD in terms of Key Generation

Comparison between DES and QKD in terms of key generation is shown in figure 25. The graph is the performance indicator with respective to know how fast the method can generate keys.

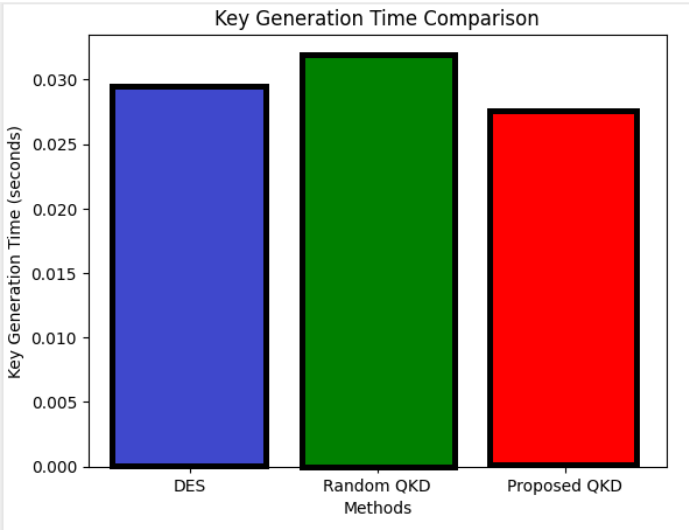

Figure 25 Comparison DES, QKD and proposed QKD in terms of Key Generation

6.4 Analysis of RSA and Honey Encryption in terms of Encryption and Decryption

Comparison between RSA Encryption and Decryption with Honey Encryption and Decryption is shown in Figure 26.

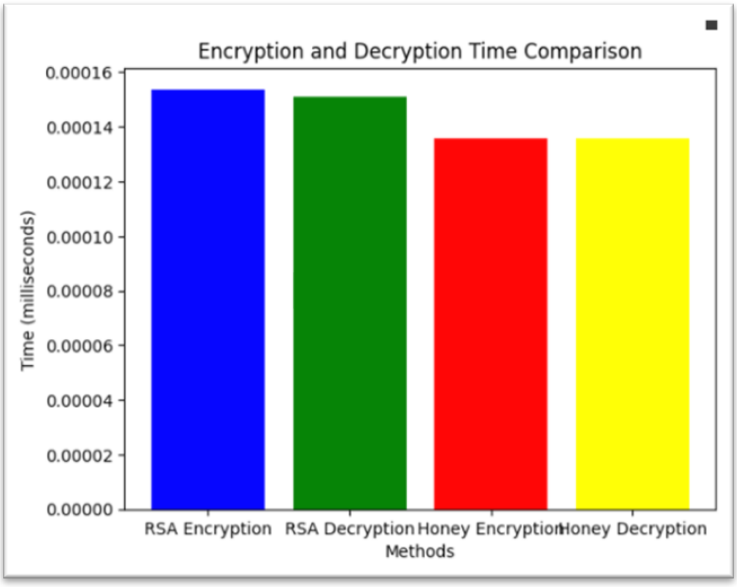

Fig 26 Comparison between RSA Encryption and Decryption with Honey Encryption and Decryption

6.5 Discussion

Asymmetric cryptography approaches currently employed in data security are vulnerable to attack by quantum computers. The quantum channel in the quantum key distribution scheme is intrinsically safe, while the classical channel is not. Postquantum cryptography techniques have addressed the security problems with the classical channel. They are comparing the security features and performance of the proposed integration with QKD classical to deliver superior performance. Resources are exposed to offline attacks when they are used in encryption. In situations like these, honey encryption can provide important extra security. Offline decryption attempts are insufficient to find the correct plaintext since HE produces plausible-looking plaintexts under decryption with invalid keys (passwords). Additionally, HE provides a well-behaved buffer against high min-entropy key partial disclosure. It is also more secure than current systems without sacrificing functionality. Among the many different types of data that are transported across many devices are text, image, and audio files. The preferred method among the various IoT security solutions put out is quicker encryption/decryption times. When encryption and decryption are faster, data may be stored and accessed more rapidly. The results showed when encryption and decryption should be done. The proposed model is by far the most efficient of all. Sufficient data protection is provided by the suggested approach.

7 Conclusion and Future Work

The current day and age of technological development foreshadows an advancement in cryptographic solutions. Technological innovations are the primary driver for addressing the existing shortcomings of QKD and Honey Encryption. Very possibly, the ability to develop effective and reliable quantum repeaters as well a satellite-based quantum communication systems would allow us to bypass any range limitation of QKD, making it possible to set up truly secure and long-distant communication networks. Machine learning algorithms can be applied for the purpose of examining the valid data compiled by Honey Encryption and faking decoy data to achieve even greater effectiveness against illegal deciphering efforts. Building QKD systems into existing crypto resources is a large obstacle and brings great potential for the next level of work. These incorporates both the technical aspects of the

implementation and the adaptation of the applications to various application scenarios, ranging from secure communications and ensured governmental data to the privacy of the data processed over the cloud. The most significant element should be the scalability and the quality of their adaptability in integrating these technologies into existing networks and systems, for their extended application and effectiveness.

The rise of QKD and Honey Encryption analogies has significant implications on how shaping laws and ethics. Future steps should focus on the design of a holistic approach to help forward ethical deployments of the very technologies as well as ensuring the protection of the public good and the privacy & security of individuals. Deliberating the hazard of misuse and the relevance of global cyber standards in promoting a safe world of cyberspace is a vitally important matter. The comprehensive adoption of QKD and Honey Encryption necessitates international cooperation, as well as entailment of a standard at the global level. It is essential that future activities build upon the lessons learned and seek to achieve more cooperation among countries, institutions of academic learning, and leaders of industry to exchange knowledge, share resources and improve the effectiveness of the existing practices. Creation of universal procedures and corresponding certifications is the key to developing the systems of crypto that will work across a broad range of areas and regions, which is why such systems would be more widely accepted by their users. To standardize would be the efforts to standardize would bring about security, interoperability, and regulation of QKD and Honey Encryption across various platforms and borders, which will further strengthen the chains.

Such a process should be inclusive of networking between government agencies, academics, and industry movers to get weaved on best practices and ethical deliberations. Other than this, building such partnerships between public and private is critical for quick adoption and application of these techniques. By concentrating on uniqueness of each sector and collectively performing different tasks, feasible innovations can be designed and implemented in real-world situations that help in a faster change process from theory to practice. Furthermore, being essential in raising awareness and education regarding the pros and cons of QKD and Honey Encryption also among policymakers, the industry professional, and the public make the process even more of a critical role. The concerned community should be highly knowledgeable of the appropriate uses of these technologies in order to positively influence their decision on their uptake and therefore the community security. Finally, the ethical deployment of Quantum Key Distribution (QKD) and Honey Encryption (HE) should be placed on the high priority list to translate these technologies into value-adding ones but regarding privacy and ethical issues. Over time, the digital world is expected to remain the hub for innovations, and these recommendations offer a good guide on using advanced cryptographic techniques to thwart off the intruding threats that keep on attacking the collective digital future.

8 Reference

Cao, Y., Zhao, Y., Wang, Q., Zhang, J., Ng, S.X. and Hanzo, L., 2022. The evolution of quantum key distribution networks: On the road to the qinternet. IEEE Communications Surveys & Tutorials, 24(2), pp.839-894.

Yu, X., Liu, Y., Zou, X., Cao, Y., Zhao, Y., Nag, A. and Zhang, J., 2022. Secret-Key Provisioning With Collaborative Routing in Partially-Trusted-Relay-based Quantum-Key-Distribution-Secured Optical Networks. Journal of Lightwave Technology, 40(12), pp.3530-3545.

Huang, X.J., Lu, F.Y., Wang, S., Yin, Z.Q., Wang, Z.H., Chen, W., He, D.Y., Fan-Yuan, G.J., Guo, G.C. and Han, Z.F., 2022. Dependency model for high-performance quantum-key-distribution systems. Physical Review A, 106(6), p.062607.

Damasceno, R.L.C., de Andrade, J.S. and Ramos, R.V., 2023. Applications of the Lambert–Tsallis W q function in QKD. JOSA B, 40(9), pp.2280-2286.

V, A.D. and V, K. (2021) Enhanced BB84 Quantum Cryptography Protocol for secure communication in Wireless Body Sensor Networks for medical applications, Personal and ubiquitous computing. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7971400/ (Accessed: 20 April 2024).

Agarwal, Neha & Verma, Vikas. (2023). Comparative Analysis of Quantum Key Distribution Protocols: Security, Efficiency, and Practicality. 10.1007/978-3-031-48774-3_10.

Noorunnisa, N.S. and Siddiqui, R.A. (2016) review on Honey Encryption Technique , International Journal of Science and Research. Available at: https://www.researchgate.net/publication/296806881_Review_on_Honey_Encryption_Technique (Accessed: 20 April 2024).

Campos, P.T. (2021) AES implementation in Python, Medium. Available at: https://medium.com/quick-code/aes-implementation-in-python-a82f582f51c2 (Accessed: 20 April 2024).

LIDBJÖRK, E. and NYLANDER, R.S. (2023) Cost and efficiency comparison of Quantum Key Distribution schemes, KTH Royal Institute of Technology . Available at: https://kth.diva-portal.org/smash/get/diva2:1779798/FULLTEXT01.pdf (Accessed: 20 April 2024).

Yin, W., Indulska, J. and Zhou, H. (2017) Protecting private data by Honey Encryption, Security and Communication Networks. Available at: https://www.hindawi.com/journals/scn/2017/6760532/ (Accessed: 20 April 2024).

N. Z. Aitzhan and D. Svetinovic, "Security and Privacy in Decentralized Energy Trading Through Multi-Signatures, Blockchain and Anonymous Messaging Streams," in IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 5, pp. 840-852, 1 Sept.-Oct. 2018, doi: 10.1109/TDSC.2016.2616861.
keywords: {Privacy;Peer-to-peer computing;Public key;Contracts;Online banking;Security and privacy;decentralized energy trading;blockchain technologies;smart grid systems},

Sangaiah, A.K. et al. (2021) Data cryptography in the internet of things using the artificial bee colony algorithm in a smart irrigation system, Journal of Information Security and Applications. Available at: https://www.sciencedirect.com/science/article/pii/S2214212621001605 (Accessed: 20 April 2024).

ISLAM, N.T. et al. (2017) Provably secure and high-rate quantum key distribution with time-bin qudits, Science . Available at: https://www.science.org/doi/10.1126/sciadv.1701491 (Accessed: 20 April 2024).

Win, Thanda & Moe, Khin. (2018). Protecting Private Data using Improved Honey Encryption and Honeywords Generation Algorithm. Advances in Science, Technology and Engineering Systems Journal. 3. 10.25046/aj030537.

Gharbi , A.A.M. and Nori, A.S. (2021) Honey Encryption Security Techniques: A Review Paper, RJCM. Available at: https://www.science.org/doi/10.1126/sciadv.1701491 (Accessed: 20 April 2024).

George, I., Lin, J. and Lütkenhaus, N., 2021. Numerical calculations of the finite key rate for general quantum key distribution protocols. Physical Review Research, 3(1), p.013274.

Kronberg, D.A., Nikolaeva, A.S., Kurochkin, Y.V. and Fedorov, A.K., 2020. Quantum soft filtering for the improved security analysis of the coherent one-way quantum-key-distribution protocol. Physical Review A, 101(3), p.032334.

Nadlinger, D.P., Drmota, P., Nichol, B.C., Araneda, G., Main, D., Srinivas, R., Lucas, D.M., Ballance, C.J., Ivanov, K., Tan, E.Z. and Sekatski, P., 2022. Experimental quantum key distribution certified by Bell's theorem. Nature, 607(7920), pp.682-686.

Juels , A. and Ristenpart, T. (2014) Honey encryption: Security beyond the brute-force bound, iarc.org. Available at: https://eprint.iacr.org/2014/155.pdf (Accessed: 20 April 2024).

Hoyul Choi, Hyunjae Nam and Junbeom Hur, "Password typos resilience in honey encryption," 2017 International Conference on Information Networking (ICOIN), Da Nang, Vietnam, 2017, pp. 593-598, doi: 10.1109/ICOIN.2017.7899565. keywords: {Encryption;Servers;Databases;Protocols;Decoding;password-based encryption;honey encryption;password typo;brute-force resilience},

Rani, Dr.N.U. et al. (2018) Honey Maze Encryption (home), SSRN. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3173517 (Accessed: 20 April 2024)

Zhang, Z.K., Liu, W.Q., Qi, J., He, C. and Huang, P., 2023. Automatic phase compensation of a continuous-variable quantum-key-distribution system via deep learning. Physical Review A, 107(6), p.062614.
Zhang, W., van Leent, T., Redeker, K., Garthoff, R., Schwonnek, R., Fertig, F., Eppelt, S., Rosenfeld, W., Scarani, V., Lim, C.C.W. and Weinfurter, H., 2022. A device-independent quantum key distribution system for distant users. Nature, 607(7920), pp.687-691.

Kapil, G. et al. (2020) Attribute based honey encryption algorithm for securing Big Data: Hadoop Distributed File System Perspective, PeerJ Computer Science. Available at: https://peerj.com/articles/cs-259/ (Accessed: 20 April 2024).

Jain, S. (2022) An enhanced hybrid encryption method for password and messages, norma.ncirl. Available at: https://norma.ncirl.ie/6523/1/somiljain.pdf (Accessed: 23 April 2024).