Advances in Networks
2026, Vol. 13, No. 1, pp. 9–30
https://doi.org/10.11648/j.net.20261301.12

Research Article

Design of a Quantum Resistant Cryptography Algorithm for Blockchain Network Using Modified Bit Flipping Key Encapsulation

Fagbohunmi Grifin Siji*

Computer Engineering Department, Abia State University, Uturu, Nigeria

Abstract

Due to the continuous development of Quantum computing, it is envisaged that in the near future, classical cryptography will no longer be considered secure, these will make systems which rely on them become outdated. One of such system is the Blockchain networks used in diverse applications, such as cryptocurrency, banks, supply chain management , just to mention but a fe .Current algorithms used to secure Blockchain networks such as Rivest-Shamir Adieman (RSA), Secure Hash Algorithm, 256 bit (SHA 256) and Elliptic Curve Cryptography (ECC) are prone to quantum attacks.. They all employ the traditional cryptography techniques which are prone to attack by Quantum computing algorithms It has been shown that Shor's and Grover's algorithm can easily compromise asymmetric encryption by increasing the rate at which brute force attack can be implemented. As a result of these development it becomes imperative to design an efficient quantum resistant blockchain algorithm capable of protecting decentralised systems like banks and corporate organizations from Quantum computing threats. The aim of this paper is to design a novel Quantum resistant cryptography algorithm (QRCA) that can protect systems from Quantum computing attacks. The algorithm consist of code-based Modified Bit Flipping key exchange (MBIKE) and Post-Quantum Pseudorandom Number Generators (PQ-PRNGs) to create Quantum resistant blockchain encrypting protocol. The traditional classical encrypting techniques is now replaced with a new code based cryptography protocol, MBIKE is a key Encapsulation Mechanism (KEM) which is usually regarded as a post quantum key distribution protocol and it can therefore be a good replacement for Quantum Key distribution (QKD). MBIKE is able to protect blockchain networks from eavesdropping, while PQ-PRNG provides randomness in cryptography key distribution, which aids in making the cryptographic keys difficult to compute by hackers. The code-based key generation helps to build a secure error detecting and correcting codes. The protocol designed in the paper improves the accuracy of encryption, secure key exchange and immunity to quantum hacking with only a little overhead. The operation of the protocol enhances network transparency as well protect the blockchain network from yet to be identified quantum threats. Experimental analysis show that the protocol guarantees 99.5% protection of blockchain networks against quantum attacks, thereby enhancing the economy of large decentralized networks used in multinational corporations.

Keywords

Key encapsulation mechanism, Blockchain Security, Modified Bit Flipping Key Exchange, Post-Quantum Cryptography, Post Quantum Pseudorandom Number Generation, Decentralized Ledger, code based cryptography

*Correspondence: Fagbohunmi Grifin Siji (fagbhume.griffn@abiastateuniversity.edu.ng)

Received: 30 December 2025; **Accepted:** 12 January 2026; **Published:** 5 June 2026

1. Introduction

A blockchain network can be defined as a decentralized and unchangeable digital shared record of transactions which take place among many computers in a peer-to-peer configuration. Here new nodes are admitted only when the entire participants in the blockchain network agree to its joining, this procedure is to provide secured communication in the network. Traditional techniques used in blockchain security use traditional cryptographic techniques such as Elliptic Curve Cryptography (ECC) and Rivest-Shamir-Adleman (RSA), Secure Hash and 256 bit (SHA 256) algorithm. These techniques can no longer be deemed secure with the present capabilities of Quantum Computing [1]. This is largely due to the fact that Grover's algorithm allows easy factoring of big data and quick generation of brute force attack makes traditional cryptographic hash function techniques used in blockchain networks vulnerable to quantum attacks [2]. The vulnerability of traditional cryptographic techniques in protecting blockchain network is further laid bare because Quantum computers can easily perform Shor's transforms algorithms [3].

Recent research employs cryptographic techniques that are resistant to quantum attacks to protect blockchain networks. Some researchers has come up with protocols that guarantee high level protection from quantum related attacks as well as high efficiency in computing [4]. In order to make blockchain network immune to quantum based attacks, new techniques such as Quantum Key Distribution (QKD) and Quantum Random Number Generation (QRNG) are employed. These techniques are based on quantum mechanics principles [5]. The methods used in securing most traditional blockchain networks can easily be compromised by quantum computing methods [6], this serves as the basis for the design of quantum resistant blockchain network proposed in this paper. This will help most organizations like banks and industries which rely on decentralized architectur sustain and protect their operations. Primitives like PoW and PoS used by Quantum computers can easily compromise blockchain networks using Shor's and Grover's algorithms. These two algorithms employ the same cryptographic hash functions used in public key encryption, hence they are able to capitalize on its weaknesses [7, 8]. The weaknesses in traditional cryptography is targeted by quantum computing attacks to compromise signature based encryption [9]. Some recent Post-Quantum Cryptography (PQC) techniques like hash based and lattice based encrypting are no longer practicable to prevent quantum attacks, these are due to the fact that these techniques are not scalable and requires high computational power, hence, they can only be configured on small sized networks and not applicable on resource constrained devices [10]. The applicability of these two techniques on blockchain networks are still in serious doubt due to the high computational power required by them which result in reduced throughput for these networks. Researchers are currently faced with the daunting task of designing efficient quantum secured protocols for blockchain networks suitable for large organizations like banks and industries [11]. Current blockchain networks cannot effectively manage efficient quantum cryptography mechanisms with other components of the decentralized network due to its computational requirements and scalability issues. Blockchain networks must therefore satisfy certain design metrics such as being energy efficient and scalable in order to be applicable for the long time operations required in decentralized network architectures.

The aim of this paper is to design a quantum resistant blockchain network protocol which contains quantum attack resistant optimization attributes. The protocol will also make the decentralized operations in the network quantum attack immune as well as being scalable. The model uses a code-based Modified Bit Flipping key Encapsulation (MBIKE) and Post-Quantum Pseudorandom Number Generators (PQ-PRNGs) to create Quantum resistant and decentralized blockchain network. It should be noted that blockchain networks are used on systems which operates over an extended period of time, so the platform should be operational over a long range of time.

The protocol designed in this paper takes cognizance of blockchain's network weakness to quantum attacks through incorporating quantum resistant mechanisms in addition to addressing security concerns. The protocol provides improved network security with the aid of a better preservation of data and an efficient decentralized system. The protocol is ideal for banks and large industries with large volumes of integrated systems. Other capabilities include:

Intelligent Quantum Resistant Method which include a Post Quantum Cryptography technique using Modified Bit Flipping Key Encapsulation algorithm.

Code based error correction technique which is an improvement over traditional key exchange methods aimed at providing resilience to channel eavesdropping through man in the middle attack.

Enhanced Post-Quantum Pseudorandom Number Generators (PQ-PRNGs) which increases entropy value and secures the blockchain network from brute force attacks and attacks that take advantage of the predictable nature of traditional cryptography key distribution methods.

The protocol has a little increased overhead, however this setback was more than made up for in its reliability and high resilience to present and possibly future quantum attacks.

This paper is organized as follows: section 1 provides the introduction, background of study and statement of problem, section 2 is review of related works, section 3 discusses certain quantum attacks that compromise traditional cryptography algorithm. Section 4 describes the implementation details (materials and methods) of the protocol proposed in this paper, section 5 shows results and analysis while section 6

concludes the paper.

2. Literature Review

In [12], the effects of quantum computing on traditional cryptography systems was discussed. The paper analysed the weaknesses within traditional cryptography systems which quantum computing takes advantage of, they then went further to discuss cryptography systems that are immune to quantum attacks. The paper discussed some fundamental concepts of quantum mechanics which underlies quantum computing these include superposition and entanglement. The paper reviewed Quantum encryption algorithms along the lines of explaining the significance of Quantum Key Distribution protocols and Post Quantum Cryptography techniques in designing quantum resistant cryptography protocols. The paper further highlighted the importance of designing a quantum attack proof cryptography algorithm to replace the traditional cryptography algorithms as the strength of quantum computers (that is currently being developed) will soon make the traditional cryptography algorithms obsolete and unsafe.

Byte Coin Network (BCN) uses blockchain in a decentralized network to keep track of transactions. The protocol used in byte coin guarantees that the identities of the parties involved in communication is not made public so the transaction is secure. A key feature of BCN is the use of ring signature which enable signing by multiple parties involved in initiating a transaction, this makes it impossible to identify to real sender. Local congestion mitigation (LCM) which is a technique designed to identify and mitigate a temporary congestion in network traffic benefits from the implementation of BCN. Researchers in [13] stated that BCN is vulnerable to quantum attacks, and since it is always employed in blockchain networks, the development of quantum computers present a great risk to the existence of blockchain networks designed with Public key Infrastructure. Their work further demonstrated techniques for the efficient management of network resources over many administrative environment. The paper combined byte coin with Post Quantum Cryptography techniques in order to design a secure and efficient network monitoring system. In the paper, BITRU was used in the analysis of the system. BITRU is a binary version of the NTRU public key cryptosystem, it uses algebraic algorithm to enhance security. The use of BITRU shows that Quorom outperforms both Ethereum and Hyper ledger in terms of average time to write metrics. The result of experimental analysis from the paper for Post Quantum cryptography algorithm together with byte coin's integration was demonstrated. The future direction of decentralized cryptographic system was discussed.

According to the paper in [14], the security of current cryptosystems in the face of recent developments in quantum computing was reviewed. The paper described software security in quantum computing. An encryption technique capable of protecting bank's sensitive information, medical equipment, military gadgets, cars, ships and navigation was developed. The capability of quantum computer evolution to cripple many cryptosystems was discussed. The 53-qubit Sycamore Processor was recently designed by Google, this design shows that many quantum computers will be present in the nearest future. Such developments will make current cryptosystems outdated because quantum computers can solve cryptographic algorithms in finite time. It therefore becomes pertinent in the face of these progressive developments in quantum computing to design cryptography that is resilient to quantum attacks. The main challenge in designing a quantum resistant cryptography algorithm for blockchain networks lie in meeting the joint requirements of security, usability and user satisfaction. Software efficiency refers to the ability of a given software to conserve memory usage and be able to optimally complete tasks correctly. The lifespan of a software is limited by the extent to which it remains relevant i.e. it doesn't have security flaws.

An improvement in multimedia data security was considered in [15], the data include pictures, audio and video gotten from IoT devices. The paper described novel techniques such as blockchain and quantum cryptography as possible ways to enhance multimedia data protection and privacy. One possible flaws of most traditional cryptography techniques and the internet are eavesdropping, man in the middle attack and channel interception. Alteration of transmitted data could have a catastrophic effects on a company's or bank's budget and image. Secure data communication amongst IoT devices rely on an efficient cryptographic key management and control. An effective key management system is needed to ensure optimum and secure network performance even with a fairly good security mechanism. With the current advancements in IoT networks, huge volumes of data are generated by the many IoT gadgets, this makes the requirement of efficient key management paramount in the design of this class of networks. The resource constraint of IoT devices makes them prone to many adversarial threats. The design of an efficient and secure network in this scenario calls for careful implementation design. Traditional cryptographic techniques are prone to quantum attacks, this compromises the integrity of blockchaim networks. Researchers have studied the weaknesses of traditional encryption algorithm and come to the conclusion that Post Quantum Cryptography and Quantum key distribution will be necessary for the design of a quantum resilient cryptography system. Despite the wide applicability of Blockchain networks due its transparency and decentralized configuration, its major weakness is the way it can be compromised in the face of quantum attacks, this review has discussed contributions from various researchers on techniques used in combating this shortcomings. However the shortfall of all these techniques are their inability to be use on resource constrained devices due to their high computational and power requirements. This paper intends to address this shortcoming.

In [4], a code based bit flipping key encapsulation (BIKE) algorithm was designed, this is among the fourth round NIST's candidate standardization process. BIKE is based on

extrapolation of the Monte Carlo simulations on the pattern of the iterative decoder The protocol was designed to address the vulnerabilities o classical cryptography algorithms to quantum attack as well as meet the requirements for NIST fourth round standardization requirements. In the work of the researchers in [5], they designed the LEDAcrypt which is a suite of post quantum asymmetric cryptography protocols designed using quasi-cyclic low parity density codes (LDPC) and moderate parity density codes (MDPC) both being a merger of the LEDAkem and the LEDApkc submitted to the fourth round of the NIST standardization. LEDAcrypt is based on a two-level bit flipping decoder, where the first iteration is combined with upper bound of the second iteration in order to compute the decoding failure rate in BIKE.

3. Problem Statement

It is envisaged that RSA, ECC and other traditional cryptography techniques may soon be outdated due to the continuous evolution of quantum computers which are capable of compromising these aforementioned techniques in finite time. Of most importance is Blockchain networks which relies on public key cryptography for its implementation. Primitives like PoW and PoS used by Quantum computers can easily compromise Blockchain networks using Shor's and Grover's algorithms. These two algorithms employ the same cryptographic hash functions used in public key encryption, hence they can capitalize on its weaknesses. It is therefore required that the underlying principles of public key cryptography be improved so as to combat this weakness. It has been shown through research that post quantum cryptography in conjunction with Quantum Key Distribution and Quantum Random number Generator amongst other techniques are required to build quantum resistant cryptography. Current blockchain network designs suffer the following shortcomings, high computational requirements, reduced scalability, inability to seamlessly integrate with original conventional blockchain networks [12]. Deploying post-quantum cryptographic components to blockchain applications are difficult due to the differences in implementation requirements of quantum computing and classical cryptographic techniques [15]. This paper aims to design a quantum resistant blockchain network using a combination of Modified Bit flipping Key Encapsulation (MBIKE), on top of Post Quantum Pseudorandom Number Generators (PRNGs), for the protection of bank's network resources from quantum attacks.

4. Materials and Methods

The model developed in this paper, is a combination of Key Distribution using Modified Bit flipping key Encapsulation (MBIKE) which is a Post Quantum Cryptography (PQC) protocol and Post-Quantum Pseudorandom Number Generators (PQ-PRNGs). The model designed will thereafter be tested using simulation in order to assess its ability to withstand current and future quantum attacks as well as its scalability in decentralized blockchain networks. The algorithm used for the model is shown in Figure 1.

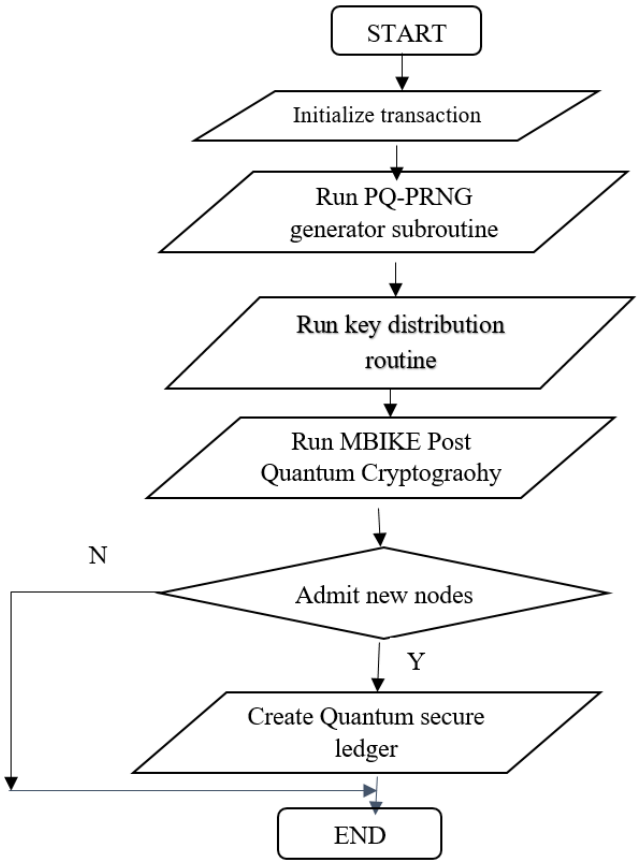

Figure 1. *Flowchart for secure transactional security.*

Figure 1 shows the quantum resistant pseudocode designed to protect blockchain networks. The processes include transaction initialization using Modified Bit Flipping Key Encapsulation (MBIKE) and Post-Quantum Pseudorandom Number Generators (PQ-PRNGs).

Function Post_Quantum PQ-PRNG1(Start_seed)
// Initializing
(Public_key, private_key) = Generate_Start_lattice_keys
Present_state = start_seed
while true
// Generate a pseudo random text message (e.g. from classical PQ-PRNG or a hash of presentt_state)
Text Message = GeneratePseudoRandomValue (present_state)
// Encapsuate the message using the public key encryption
(cipher, shared_secret) = Encapsulate (public_key, info)
// Use a part of the ciphertext or shared secret as the PQ-PRNG output
Pqprng_output = ExtractBits (ciphertext)
// State update for the next iteration

Present_state = Hash (shared_secret)
Produce pqprng_output

Figure 1 Generating distributed random keys using PQ-PRNG

The algorithm above shows how a post-quantum cryptographic primitive can be integrated into a Post-Quantum Pseudorandom Number Generators (PQ-PRNGs) to achieve quantum resistant blockchain algorithm. The processes involved in this algorithm include, transaction initialization using post-quantum encryption, use of a Post-Quantum Pseudorandom Number Generators (PQ-PRNGs), Post Quantum cryptographic key exchange and secure key generation using MBIKE. These are used to provide secure and quantum resistant key exchange between communicating parties. These processes can be further outlined as follows:

1) Download the Data

The blockchain dataset used for initializing the transaction was downloaded from Mempool archive dataset. Mempool is a queue of pending and unconfirmed data transaction for a blockchain network. Every node on the network maintains its mempool database, this implies that different nodes have different transactions. In order to view the blockchain data in real time, the mempool explorer is used. Mempool explorer enables the visualization of the transactions in the blockchain network. The mempool explorer from block native allows the downloaded data to exchange protocols and wallets when different formats are used within different transactions, Data can then be monitored and the transactions acted upon in real time. This enables administrators of the blockchain network deliver reliable and secure distributed system to quantum cryptographic attacks. The data downloaded from mempool.space is updated by the data from BigQuery, this is done every five minutes to provide regularly updated data to the blockchain network. The combination of the BigQuery data with mempool.space data was used in this paper to analyse the effect of Quantum key Distributuon for secure data transmission, Post quantum Cryptography and Post Quantum Random Number Generator on blockchain network allows for efficient random key generation for secure data communication. When Quantum cryptography is applied to processed mempool data, it results in enhanced transaction security and increased encryption stability, while the production of random keys makes it difficult for quantum threats decipher these keys. The real time update (very hour) of the mempool data together with its rich set of historical data coverage make it efficient in analysing quantum resistant networks.

2) Filtering out Private Transactions and Data Preprocesing

In the Mempool archive, private transactions refer to transactions with no pending event. This can be detected using the time pending column in Mempool lookup table. The time pending field computes the time used for the transaction in the mempool in milliseconds. This is shown in equation (1).

$$\text{Time pending} = \text{firstConfirmation} - \text{firstDetection} \quad (1)$$

The time pending attribute for all private transactions is 0. All transactions with a status of cancel or speedup will be eliminated from the analysis.

```
DEFINE TABLE PRIVATE_TRANSACTIONS AS
(
CHOOSE.*
FROM
MEMPOOL SAMPLE _ARCHIVE_
WHERE
timepending = 0
AND status != ('speedup', 'cancel')
)
```

After filtering out unwanted data from Mempool, the next stage is the pre-processing of unconfirmed data with non-zero time pending attribute. Pre-processing involves dealing with missing values and the extraction of functions that best defines the Mempool dataset and updated data. The steps involved in pre-processing is described below:

Cleaning Data: The steps required in filtering out unconfirmed transactions from the dataset comprise isolating and correcting wrong entries in database entries. Filtered data consists of data whose statistics do not conform with the rest of the dataset, together with damaged or repetitive data [16]. The validity of record statistics is verified by blockchain transaction authentication of each block's transaction details [17]. The downloaded dataset is processed by extracting wrong inputs with incorrect block hashes in addition to uncompleted transactions. For the encryption method to be quantum attack resistant, the processed dataset must be rid of all inconsistent inputs.

Normalization: Normalization of processed records is done by scaling the dataset different from expected values between zero and one. The value of dataset is computed from both its transaction details and measuring of timestamps [18]. The proper distribution of records around a value makes them consistent with cryptographic protocol requirements. The need for proper probability distribution of confirmed datasets is to conform to PQC and QKD standards, this will in the long run make for the protocol's integration with quantum computing requirements.

Alignment of Timestamp: Blockchain transaction's timestamp is done in real-time, the protocol update is done by MBIKE and Post Quantum Pseudo Random Number Generator (PQ-PRNG). In order to implement quantum security for blockchain networks, the time stamp must conform to the post quantum cryptographic protocol's time frame [19]. Blockchain transactions will be well integrated with all order nodes in a decentralized network so as to provide an accurate alignment or timestamp [20]. The inclusion of unpredictable pseudo random number generator allows the operation to proceed in line with quantum cryptography guidelines, these guarantees protection from time based attacks.

4.1. Assessing Quantum Cryptography Threats

This section assesses how secure blockchain's cryptography confirmed data are to quantum attacks. This paper aims

to design a blockchain cryptography that is immune to quantum based attacks. Shor's and Grover's algorithm presents a framework through which the resilience of blockchain cryptography protocols are tested against quantum attacks. These two algorithms analyse blockchain's data transactions with a view to validating the cryptography's resilience to quantum threats. The post quantum cryptography proposed in this paper helps to understand the vulnerability of pre-quantum cryptography to quantum attacks and what properties post quantum blockchain cryptography must have. The operational details of the post quantum cryptography threat assessment for blockchain network are presented below:

Quantum Attack Simulation: Virtual quantum attacks were simulated, and thereafter, quantum algorithms were used to analyse current cryptographic methods. Analysis of the shortcoming of blockchain's cryptographic protocols in the presence of quantum threats were carried out. The analysis were carried out on the following cryptography algorithms, ECDSA, RSA and SHA-256, BIKE and MBIKE.

Analysis of Quantum Attacks on Security: Quantum algorithms which represents real quantum attacks were simulated, these were incident on the above mentioned cryptography algorithms, with the aim of identifying the weaknesses in their security architecture when implemented on blockchain networks.

MBIKE Algorithm Implementation: An analysis of the weaknesses of ECDSA, RSA and SHA-256 (which uses public key encryption) to Shor's Algorithm was performed. It was observed that Grover's Algorithm compromised the SHA-256 within a short range of time. The implementation of these algorithms on top of blockhain network were tested with the aim of determining their weaknesses and possible resilience to quantum attacks.

Assessment of Shor's Algorithm on Quantum Cryptography: It was observed that RSA and ECDSA were easily compromised by Shor's Algorithm. This is because the algorithm can factorise large numbers in polynomial time. It should be observed that these techniques rely on the difficulty of classical cryptography to factorise numbers comprising of large primes. This is because it will take increasingly large key size using public key cryptography to factorize these numbers even in a very long time. It is also infeasible to continually increase classical cryptography key size due to its draining of network resources. Unfortunately this is not the case with quantum computer which can easily perform these factorisation and thereby compromising classical cryptography especially public key encryption.

The strength of Shor's Algorithm is its ability to factorise large primes within polynomial time, the equation used in Shor's algorithm is shown in equation (2).

$$\text{Find g Iterate } (g(x) = \text{number of iterations of } B^x \bmod N \quad (2)$$

Where N is the number (integer) to be factorized, mod N is the remainder when a given number is divided by N, The aim of the function is to identify the least positive integer t, so that the expression in equation 3 is satisfied.

$$g(x) = B^x \bmod N, \text{ then } g(a) = g(d + t) \quad (3)$$

The aim of the algorithm is to compute the prime factors f and g of N. the first step is computing f X g mod N if the value is 1 then f and g are the prime factors of N, otherwise in the next iteration f is replaced by the remainder of f X g mod N, this operation is repeated until the remainder is one. The final value of f and g are the prime factors of N.

Grover's Algorithm: This algorithm is used to search for an item in an unstructured (unsorted) data. Grover's algorithm uses quantum operators to speed of search for a marked element. The use of quantum operators make Grover's algorithm faster than classical search methods by a square factor, that is whereas it takes classical algorithm O (N) time to search for an item out of N elements, it would only take Grover's algorithm O ($\sqrt{N}$) time to do the same. Grover's algorithm tests the security of a blockchain network using SHA-256 cryptographic hash function. SHA-256 hash algorithm is used to secure blockchain networks, and since Grover's algorithm can quicken the search for a hash collision, it becomes possible for an attacker to modify the hash algorithm to obtain an alternate value which gives same hash output as another value [21]. It takes an infinite amount of time for classical algorithms to produce hash collision for a hash algorithm. It should be realized therefore that Grover's algorithm is used on blockchain network to find an image in which equal hash values are given by two different SHA-256 sources.

In a nutshell Grover's algorithm is capable of finding a preimage or hash collision in which another input, different from the original input produces a hash output equal to the original input by a square root factor of time it takes classical cryptography algorithm. The next section describes in detail the Modified Bit Flipping key Encapsulation algorithm.

4.2. The Modified Bit Flipping Key Encapsulation Algorithm

MBIKE is a code based Key Exchange Management (KEM) based on Quasi-Cycle version of the Moderate Density Parity Check (QC-MDPC) code. It uses the Fujisaki-Okamoto (FO) Chosen Cipher text Attack (CCA) transform [22, 23] together with Black Gray Flip (BGF) decoder [24] to test the Indistinguishability under Chosen-Cipher text Attack (IND-CCA) security on the QC-MDPC. The modified Bit Flipping Key Encapsulation scheme used in this paper contains the following subroutines:

System Setup (Input): This sets the level of quantum security (λ) desired,

System (Output) the following systems parameters (s, v, u and w) were used, where

s = s defines the length of the circulant block matrix (a square matrix in which each row is formed by rotating one

element to the right of the row at its top). For example for matrix L comprising of N x N blocks, A_0, A_1, …. A_{N-1}. The order of a circulant matrix defines its size. The number of circulant block in a row is referred to as the index

$$L = \begin{matrix} A_0 & A_1 & A_3 & .. & .. & A_{N-1} \\ A_{N-1} & A_0 & A_1 & .. & .. & A_1 \\ A_{N-2} & A_{N-1} & A_0 & .. & .. & A_2 \\ .. & .. & .. & .. & .. & .. \\ .. & .. & .. & .. & .. & .. \\ A_1 & A_2 & A_3 & & & A_0 \end{matrix}$$

$s = N/n_0$ where $n_0 = 3$, in modified BIKE. N = 3r where the length of redundant bits used for error detection and correction. Now r = N – K where N is the length of both the message bits and redundant bits (code word), while K is the length of the message bits. From the analysis in modified BIKE, bit length N = 3r or h = K. 2r, in addition, the lengths of v and u will be large enough to get a low decoding failure rate (DFR). The modified BIKE has three levels of security, 1, 3 and 5 which corresponds to AES-128, AES-192 and AES-256 security respectively. Different sets of system parameters (s, v, u and w) are required for achieving low DFR on the separate security levels. s is the Hamming weight of each parity check matrix row.

v = row weight, this defined an even integer where v/2 is odd, this also represents the Hamming weight of each parity check matrix row.

u = radius of decoding, this is a positive integer, this also represents the Hamming weight of the error vector.

w = shared secret size, this is also a positive integer, this also represents the size of computed symmetric key size.

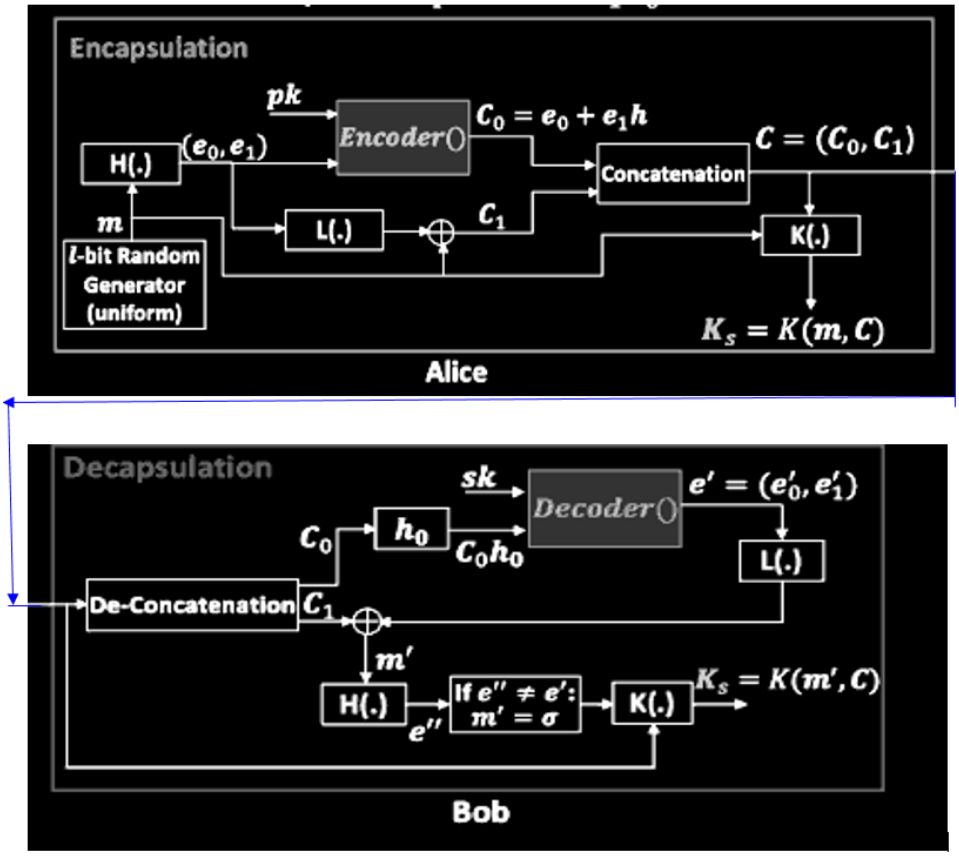

Figure 2. ***(previuos page)*** *MBIKE block diagram, Alice makes use of Bob's public key to produce a secured key ks. As soon as Bob receives ciphertext C, he, gets the same key ks that was generated by Alice (It should be noted that MBIKE does not receive a plaintext because it is a key encapsulation management scheme used by two nodes to generate secured shared key). At the end of the session, both nodes can use the shared key ks in communicating their message using Data Encryption Mechanism (DEM).*

Hash Functions: Three hash functions L, M, and N are used in MBIKE. These are randomly and uniformly selected. The input to L is l bits while the output is 2r bits (i.e. H: $\{0,1\}^l = \{0, 1\}^{\binom{2r}{t}}$), in the same vein, M and N can be represented as M: $\{0, 1\}^{2r} = \{0, 1\}^l$ while N: $\{0, 1\}^{2l + r} = \{0, 1\}^l$ respectively.

There are three subroutines in MBIKE, these include key generation, encryption and decryption. The procedure is shown in Figure 2. The end result of these subroutines result in symmetric key encryption between two agents. The operations of the subroutines are as follows:

Modelling Decoding Failure Rate

In order to model the decoding failure rate (DFR) of a decoder, the statistical distribution of the weight of its syndrome is characterized with the decoder input, this is repeatedly done for all other input in a stream of data. In order to obtain the decoding failure rate for all the input in a data stream, the law of total probability is combined with the DFR estimates of the syndrome weight distribution. The DFR can be estimated in two steps, the first step calculates the unsatisfied parity check (UPC) counts distribution in the first iteration, after which the distribution of the rightly flipped bits (r+) is derived from the error estimate e, together with the wrongly flipped bits (r-). During the second iteration, statistical distributions of r+ and r- is used to partition the bit value position into four divisions, depending on if the positions of the bit values were estimate correctly or not in the first iteration. The two level DFR iteration described above is used to calculate the approximate value of the decoding failure rate based on the overlap of the four division stated earlier.

The Syndrome Weight Distribution Model

Assume $(e, s)_h$ h ε {0, 1, 2, 3,….. v} denotes a pair of value which represents error vector e, with its corresponding syndrome s = He^v which are indexed by error vector weight h. The syndrome to be modelled together with its error vector of weight v is assumed to be the last pair in the series, (e, s(0, (e, s)1, (e, s)2, (e, s)3……(e, s)v, here (e, s)0 has a null error vector, while its syndrome is null. (e, s)h, l ≥ 1 represents a pair having error vector having equal asserted bits (bit 1) of an error vector $(e, s)_{h-1}$ placed uniformly and randomly among the n – (h – 1) positions.

The Hamming weight of each syndrome from the series stated earlier is modelled as a discrete random variable X_h which is relate to a probability mass function Pr (Xh = z), with h ε {0,….v}, z ε {0,….p}, this can be shown as an array $xr_{(h)} = [xr_{(h)0},….. xr_{(h)w},…..xr_{(h)p}]$.

The distribution of the weight of the null syndrome representing the first element in the series xr(0) = [1,0.////0] is first used, the random variable Xv linked to the weight of the syndrome in the series happens to be the same with the last state of of a discrete non homogenous Markov chains having p + 1 states. This type of Markov chain is explicitly defined by xr(0), while the transition matrices, $r(h) = [x,y,h]_{x,y\varepsilon\{0,…p\}}$.

The distribution of the random variable X_h can be computed by multiplying the vector matrix $xr_{(h)} = xr_{(h-1)}$. R (F), where F ε {1,…..v}, the transition probability in this case is given by $_{rx,y,h}$ = Pr ($X_h = z$ $_{Xh-1}$ = w is derived from the first and last weight of the syndrome as from from iteration step 1 described earlier.

The model derived below shows the amount of flips forced on a syndrome bit, the probabilities of flipping up a correct bit an flipping down an asserted bit (bit 1) of a syndrome is computed. Then the statistical distribution of a syndrome weight in addition to the transition probabilities $r_{x,y,h}$.

The probability mass function Fh is given by the following hypergeometric distribution ϕh (f,h) = Pr (Fh – f) = $\frac{\binom{x}{f}\binom{n-x}{h-f}}{\binom{n}{h}}$

From the bit 1 positions of the error vector, a single row is selected from the M rows. The selected row will form a syndrome bit. Anytime the position of one of the selected row contains bit 1 (asserted bit) amongst the x asserted bits from the n bits, the syndrome bit of the selected row is flipped. From the syndrome bit of the series, $(e, s)_{0,,,,,,}(e, s)_{h-1}$, the probability that a bit out of the syndrome bit is flipped is given by Pr (F_h = f + 1 F_{h-1} = f) = $\frac{x-f}{n-h}$, where the event, F_{h-1} means the syndrome bit is either 0 or 1, this depends on if F is even or odd respectively.

From the above, the probability $\pi \begin{matrix} h-1 & to & h \\ flip & to & 1 \end{matrix}$ (h) of flipping a bit during step 1 from any syndrome bit which was bit 0 during step h – 1, is dependent on the value of step 1, and this can be computed as in equation (4).

$$\pi \begin{matrix} h-1 & to & h \\ flip & to & 1 \end{matrix} (h) = \frac{\sum_f \Pr(F_h=f+1 \ ! \ F_{h-1}=f) \ \Pr(F_{h-1}= \ f)}{\sum_f \ \Pr(F_{h-1}=f)} \quad (4)$$

From equation (7), the range of f are the even values in {0,…….min (x, h} this equation is the model of the decoding failure rate in MBIKE.

Bob's public key Pk = $h_2h_1h_0$

Key Generation: The input to the KeyGen subroutine are the system parameters (s, v, u, w), the outputs are the public key h while the private keys (m_0, m_1, m_2, μ) are used in the asymmetric encryption,

ivate Key: Here m_0, m_1, m_2 = ℘², the three parameters have equal weight $m_0 = m_1 = m_2 =$ w/3, where ℘² is a cyclic polynomial ring F_2 [X] / (X^r – 1). In the same vein, m_0, m_1 and m_2 can represent r X 1 column vectors. A part ℧ of the message bits ℵ = {0, 1}^{l} are randomly selected. The private key is finally set as $s_k = (m_0, m_1, m_2, \mu)$.

Calculate public key (pk) = $m_2 \ \ m_1^{-1} \ \ m_0^{-2}$

Return (m_0, m_1, m_2, μ)

Encryption Subroutine: In the encryption subroutine, the public key pk is taken as input and an output cipher text C is generated. C = (C_0, C_1, C_2). The encrypted cipher text is sent to the recipient, the recipient uses a private key pk_s (asymmetric encryption) to decrypt the cipher text to obtain the plain text. This private key is only known to the recipient. The encryption subroutine has the following steps:

1) An l – bit vector n is uniformly selected at random from the message space M.
2) Calculate e_0, e_1, e_2 = ⩊(m) where e_0, e_1 and e_2 are error vectors having r bits, in such a way that $|e_0| + |e_1| + |e_2| = t$.
3) From the computed error vectors e_0, e_1 and e_2, calculate C = (C_0, C_1, C_2) = $e_0 + e_1$ h, m ⊕ Ⲏ (e0, e1) +
4) e_2 h, m ⊕ Ⲏ Ⲏ (e_0, e_1, e_2), these errors are sent to the recipient.
5) IV The secret key (asymmetric) is calculated as ks = Ⲏ(m, C).
6) Return (C, Ks)

Decryption Subroutine:

1) This takes both the private key pk_s (m_0, m_1, m_2, μ) and cipher text C as input to generate symmetric key or a failed symbol ϑ, while the output is key k.
2) Calculate the syndrome as $S = c_0h_0h_1$.
3) S is decoded using the Black-Gray-Flip decoder to get error vectors e_0', e_1' and e_2'. If $|e_0'| + |e_1'| + |e_2| \neq t$ or the operation is halted when decoding procedure is not ϑ.
4) Calculate m' = $C_0 \oplus h$ (e_0', e_1, e_2'). If $\mathfrak{H}(m') \neq (e_0', e_1', e_2')$, then m' = ∝.
5) Return Ks = $\mathfrak{H}$(m', C)

Using the above two subroutines, it will be possible for two nodes to communicate securely over a channel. In asymmetric key encryption, a public key is shared between the sender and receiver, the sender encrypts the message bits with the shared public key while the receiver decrypts the encrypted message (cipher text) using its private key. This will only be possible when only the authentic receiver having the correct private key encrypts the message, otherwise the transmission will be eavesdropped. With the subroutines described above, it will be impossible to eavesdrop the communication between the two parties because only the receiver has the private key required to decrypt the cipher text sent from the sender. This is because the retrieval of the private key from the receiver using the syndrome $C_0h_0h_1$ from a random Quasi-Cyclic code is proven to be NP-complete. However the authentic receiver can check the correctness of C_0 by confirming $\mathfrak{H}(m') = (e_0', e_1', e_2')$.

Details of the Modified BIKE: The encoding procedure of the modified BIKE is described as follows: The modified BIKE is developed around the Niederreiter's framework [25]. Parity check matrix is used to implement the encoding procedure. The systematic format is used for the parity check matrix. The encoding is depicted by equation (5).

$$C_0 = [e_0\ e_1\ e_2].\ H^T = [e_0\ e_1\ e_2] \begin{bmatrix} I_r \\ H_2\ \ H_1^{-1}H_0^{-2} \end{bmatrix} \quad (5)$$

Where H^T is a 3r x r matrix.

This is so as N = r + K = 3r, which is the dimension of $H_2H_1^{-1}H_0^{-2}$ represented as a (r x r x r) matrix. The resultant encoded vector is therefore as shown in equation (6).

$$C_0 = e_0 + e_1H_1H_0^{-1} + e_2H_2H_1^{-1}H_0^{-2} \quad (6)$$

It is possible to write equation (5) this way because H_2, H_1 and H_0 are circulant blocks which can be represented with their first row as shown in equation (7).

$$C_0 = e_0 + e_1h_1h_0^{-1} + e_2h_2h_1^{-1}h_0^{-2} = e_0 + e_1h + e_2h \quad (7)$$

The Reason why the Decoding Procedure is $C_0h_1h_2$: In the decrypting subroutine, $e_0e_1e_2$ has to be recovered from C_0 which is the received cipher text. In order to achieve this, it is assumed that the Quasi-Cycle code ℭ has parity check and generator matrices H = [$H_0H_1H_2$] and G = $[H_2^T H_1^T H_0^T]$ respectively. ℭ is a valid Quasi-Cycle code because these values of parity check and generator matrices satisfy the condition that $GH^T = 0$. If C has been used to encode a plaintext m = [$m_1m_2\ldots\ldots m_r$]. Note here k = r because $n_0 = 3$. The generated codeword (u.G) will be $[uH_2^T uH_1^T uH_0^T]$ which is a 1 x N column vector, where N = 3r. If an error e with length 3r and weight t is introduced into this codeword, with the added assumption that the error can be represented as e = [$e_0e_1e_2$], where e_0, e_1 and e_2 are 1 x r vectors, and $|e_0| + |e_1| + |e_2| = t$. If the introduction or the error results in a column vector of size r, then the resultant vector can be written as shown in equation (8).

$$s = u.G + e = [uH_2^T + e_0\ \ uH_0^T + e_1\ \ uH_0^T + e_2 \quad (8)$$

If the syndrome of the vector size r is calculated, the expression in equation (9) will be obtained. This is due to the fact that adding the transpose of the matrices give the same value as the transpose of the addition of three cyclic matrices. However, this condition can only be fulfilled if matrix D = [D_{ij}] i.e. the matrix is made up of smaller blocks such that $D^T = [D_{ji}^T]$.

$$S = H.\ r^T = [H_0\ H_1\ H_2]. \begin{bmatrix} (uH_2^T + e_0)^T \\ (uH_1^T + e_1)^T \\ (uH_0^T + e_2)^T \end{bmatrix} = H_0((uH_2^T)^T + e_0^T) + H_1((uH_1^T)^T + e_1^T) + H_2((uH_0^T)^T + e_2^T) \quad (9)$$

As H_0, H_1 and H_2 are circulant blocks, then $H_0H_1H_2 = H_2H_1H_0$, $(uH_2^T)^T = H_2u^T$, H_1H_0, $(uH_1^T)^T = H_1u^T$ and $(uH_0^T)^T = H_0u^T$, there the syndrome can be expressed as shown in equation (10).

$$S = H_0\ H_1\ H_2\ u^T + H_0\ H_1\ u^T + H_0e_0^T \quad (10)$$

The expression in equation (9) is arrived at due to the fact that modulo 2 addition was used for syndrome decoding. The syndrome above can be used on bit flipping base decoder to get e_0, e_1 and e_2. The encoded vector $C_0 = e_0 + e_1\ h_1\ \ h_0^{-1} + e_2\ h_2\ h_1\ \ h_0^{-1}$ obtained in the modified BIKE encryption subroutine will now be studied. When C_1h_1 is computed the solution will be equal to the syndrome S obtained in equation (9). It should be noted here that the circulant blocks H_0, H_1 and H_2 can be represented as h_0, h_1 and h_2 respectively, where h_0, h_1 and h_2 are r x 1 column vectors. This means that H_0, H_1 and H_2 are obtained column wise from h_0, h_1 and h_2 respectively.

This shows that e_0, e_1 and e_2 can be obtained from the syndrome $S = C_0h_0h_1$ by substituting it to a bit flipping decoder which is derived from H = [H_0 H_1 H_2]. It should be noted that the decryption subroutine has three input parameters h_0 h_1 and h_2 which was used to decode the syndrome $S = C_0h_0h_1$. In the original BIKE code, the parity check matrix has two parameters., H_0 and H_1, however in this modified version, three parameters H_0, H_1 and H_2 were used resulting in a lower decoding failure rate. Allowing for a lower decoding failure rate will

lower the rate of decoding errors in any number of decoding attempts. This will make the decoder more efficient which translates to higher model efficiency.

The algorithm below shows the Modified Bit Flipping Key Encapsulation algorithm

Algorithm: Modified Bit Flipping Key Encapsulation in Decentralized Bank Transactions

Input: t ε $F_3^r, H \ \varepsilon \ F_3^{r\ x\ n}$ *represents incoming* Customer transaction

ў = 0^n; w̄ = w

for j = 1 to N do

t = threshold (j, w̄, w)

for k = 0 to N – 1 do

If ℭj ≥ T then

w̄ = w̄ ⊕ 1

Ў = ў – col(G,k)

return w̄

qkd_keygen_exchange():

key = generate_pseudo-random quantum_key()

α = {0, 1} ^{256}

sk = (h_0, h_1, $h_{2)}$ = ℌ^3 where wgt(h_0) = wgt(h_1) = wgt(h_3) = w odd

h = h_2 h_1^{-1} h_2^{-2}

return (sk, α, h)

Encryption (C, K) = Encrypt (h)

Message Generation m= Ĵ

Error vector computation (e_0, e_1, e_2) = H(m, h) where wgt(e_0, e_1, e_2) = t and e_0, e_1, e_2 ε ℜ

Calculate the ciphertext C = (c0, c1, c2) = e_0 + e_1h, m⊕ L(e_0, e_1) + $e_2$hm ⊕L(e_1, e_2)

Calculate the shared key K = Ĵ(m, C)

decryption m = Decrypt(sk, α, h, C)

m' = Decrypt (sk, C) or ⊥ if fails to decode

If ((m' ≠ ⊥). AND. (C == ReEncrypt (m', h)))

Return K (m', C)

else

return K(α, C)

Function Generate Secure Pseudo Random Number using PQ-PRNG

define generate_secure_pseudo random_number():

return quantum_pseudo random_number_generator ()

Function Secure Transaction Signing

Define secure_transaction(transaction):

key = pq-prng_key_exchange()

data encrypted = pq-prng_encrypt(transaction, key)

signature = sign_transact (encrypted_data, generate_secure_pseudo random_number())

return signature

Function Blockchain Protocol Enhancement

define blockchain_protocol():

while true:

new_customer details = get_incoming_customer details()

signed_details

secure_transaction(new_customer details)

append_to_blockchain(signed_customer details)

Function Threshold (j, w̄, w)

T' ≤ f_1 ⎪s⎪

N = (e + 1)/3

If j = 1 then T = T'+ d̲

If j = 2 then T = (3T' + N)/3 + d̲

If j = 3 then T = (T' + 3N)/3 + d̲

If j = 4 then T = (3T' + 3N)/3 + d̲

If j = 2 then T = (3T' + N)/3 + d̲

If j ≥ 2 then T = T' + 2N + d̲

return maximum (f_1(⎪ў, T⎪))

Ft(y) = 0.006272. y + 11.102. d̲ = 4

Ctr (H, w̄, k) number of equations not satisfied by the position of k.

5. Result and Discussion

This paper described a post quantum cryptography for a blockchain system. The protocol is made up of two stages, these include (i) post quantum key distribution which was developed using the Modified Bit Clipping key Encapsulation Scheme (MBIKE) and (ii) the Post Quantum Pseudo Random Number Generator (PQ-PRNG) to generate the keys distributed during the communication. From the experiments carried out, the average transaction accuracy increased by 15.6% while the data protection increased by 64% when compared to traditional pre-quantum cryptography algorithms, which makes it better resistant to future enhanced quantum threats. The simulation was done on Python, due to its robustness in implementing cryptography algorithms together with its ability to incorporate the interconnected and decentralized security platforms in a blockchain network. The down side of the protocol is its reduced transaction speed due to the overhead incurred due to the two stages involved in the implementation of the protocol. However this reduction in transaction was more than made up for in its reliability and high resilience to present and possibly future quantum attacks.

5.1. Experimental Setting

This section provides details of experiments carried out using MBIKE on the blockchain network. Different performance metrics were used to test for the effectiveness of the proposed model. Table 1 shows the simulation parameters of MBIKE use in the blockchain system, while Table 2 shows the tools used to design the experimental setup for MBIKE based blockchain network. The framework used for the model was Hyperledger fabric, this was installed on a Linux operating system. Virtual machines were created using the Docker Engine, while one Hyperledger fabric were embedded on the virtual machines in a Docker image. Information about the blockchain network was gathered using the Hypeledger Caliper, which was used to gather data from the network (Hyperledger Caliper was embedded into the blockchain network). Apache CouchDB Q, an open source database was used to save the standard results from Hyperledger Fabric.

5.2. Performance Analysis

The MBIKE blockchain network proposed in this paper was analyzed using Hyperledger Caliper which is an open source tool. From the simulation carried out, in various experimental settings, there was reduction in the transaction speed in due to the increased network overhead in Modified Bit clipping key Encapsulation scheme The transaction speed in (MBIKE) was 47.5 TPS compare to 59 TPS, and 65 TPS in ECC-RSA blockchain model, The reduction in transaction speed of MBIKE was due to 17% increased overhead resulting from the more computationally involved MBIKE algorithm especially with the increase in vector size of the syndrome resulting from the introduction of higher size error vector into the generator matrix, the increased error vector size resulted in enhanced security. It should be noted that the transactions per second (TPS) in blockchain technology determines the extent to which the network is scalable. This means that MBIKE system is capable of higher level of scalability. The time needed to encrypt a string was about 2.4 ms to 2.7 ms, while the corresponding decryption time was between 4.8 ms to 6.4ms. The time required for data transmission from source to destination is almost immediate (i.e. no latency). The Cryptography algorithm (MBIKE) is resilient to quantum attacks, though at a little overhead. The simulation environment used for the MBIKE blockchain model is shown in Table 1. The table shows the parameters used in configuring the model. The parameters listed here was used in order to optimize the operation of the MBIKE blockchain network. Table 2 shows performance evaluation comparison between MBIKE, BIKE and traditional blockchain cryptography algorithm.

Table 1. *Simulation Environment for the MBIKE model.*

Components	Specifications
Docker Engine	21.11.18
Docker Composer	1.30.2
CPU	Intel i3 core 3.7 GHz CPU, 1 TB, HDD 16 GB RAM
Operating System	Linux Mint Debian edition
Node SDK	Node.js
Blockchain Technology	Hyperledger fabric
Programming language	Java 8
Database	Apache CouchDB Q

Table 2. *Parameters for the Experimental setup for MBIKE blockchain network.*

Parameters	Values	Description
Size of block	257024	Block maximum size
Block Interval	300,350,400	Creation time for new block
TSOs	1	Transmission system operation for the network
DSOs	5	Distribution system operator for the network
Internal Database	CouchDB	Storage for Ledger data
External Database	Azure Blob Storage	Storage for off-chain data
Ordering Service	Quantum Byzantine Fault Tolerance	Responsible for ordering transaction into the block
Policy for Endorsing	Aurora Postgre SQL	Determines policy which members must agree in order for new nodes to be added

Table 3. Performance Evaluation.

Metrics	Traditional Blockchain ECC, RSA	BIkE Blockchain	MBIKE Blockchain
Speed	65 TPS	59 TPS	47.5 TPS
Computational Cost	1.06	1.26	1.30
Time for Encryption	2.6ms	2.85ms	2.91ms
Time for Decryption	5.1ms	6.8ms	6.93ms

5.3. MBIKE's Performance Analysis in a Blockchain Network

The combination of Post Quantum Pseudo Random Number using (PQ-PRNG) and MBIKE enhanced the security in blockchain network due to its use of NP-complete code based cryptography as explained in section III, the Modified Bit Flipping Key Encapsulation Scheme (MBIKE) is a code based Key Exchange Management (KEM) based on Quasi-Cyclic Moderate Density Parity Check (QC-MDPC) code. This is shown in Figure 4. It employed an increased syndrome size compared to the original BIKE in order to enhance the security of the code. The cyclic polynomial size was also increased in order to incorporate a higher bit error vector size, this will make it even more difficult for an eavesdropper to detect locations of errors in the code word, thereby improving the security of the code. The QKD incorporated into the MBIKE resulted in secure key exchange with Quantum Bit Error Rate (QBER) ≤ 1.5%. Due to the introduction of higher bit error vector size, the entropy of the generated keys was improved from 0.97 entropy in BIKE and 0.82 in traditional cryptography to 0.995 in MBIKE. This makes it even more difficult for adversarial nodes (eavesdropper) from predicting the code word. The increased size of the error vector and cyclic polynomial size introduced additional overhead in MBIKE so the cryptography speed (TPS) was reduced to 47.5 TPS as compared to 59 TPS in BIKE and 65 TPS in traditional cryptography, as shown in Figure 3, however this trade off was more than compensated for by the improved resilience to quantum attacks. The cyclic polynomial size was increased in order to incorporate a higher bit error vector size, this has the effect of making a higher bit size error to be introduced into the encrypted data, which will make it even more difficult for an eavesdropper to detect locations of errors in the code word, thereby improving the security of the code. The setup of MBIKE's seamless interconnection / interoperability with blockchain network in addition with its fairness in mining makes it easily deployable for blockchain network.

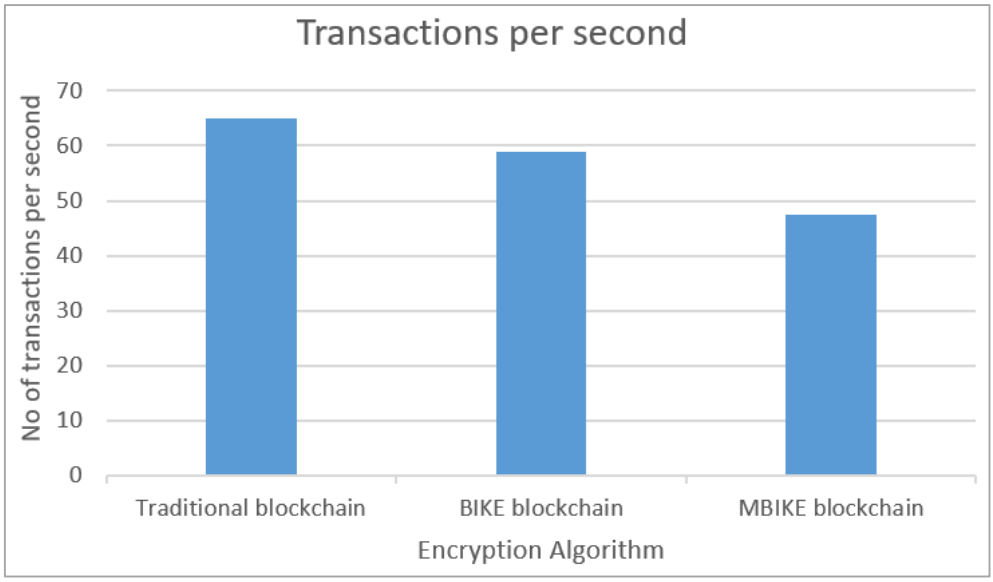

Figure 3. Transaction speed comparison.

Figure 4 shows a chart which compared MBIKE's blockchain security using (i) hash rate (ii) Nakamoto coefficient and (iii) number of active nodes with BIKE and classical cryptography. Hash rate measures the computational power needed in mining and securing the network. Nakamoto coefficient determines the number of different entities such as validators, mining pools, token holders needed to collude the network so as to disrupt its operations. This metric measures the level of decentralization of the blockchain network. Number of distributed nodes measure the percentage of total nodes required to be captured by an adversarial node before the network can be compromised. From the graph (Figure 4), it can be noticed that MBIKE outperformed the classical cryptography and BIKE methods in terms of hash rate by 37% and 24% respectively, in terms of Nakamoto coefficient by 24% and 30% respectively in terms on number of active nodes by 22% and 28% respectively. Figure 5 shows security comparison in key exchange of MBIKE with that of BIKE and classical cryptography. From the graph, it was observed that MBIKE outperformed the classical cryptography and BIKE by 56% and 21% respectively. MBIKE introduced improved randomness of 23% and 11% to classical cryptography and BIKE cryptography respectively, MBIKE also interfaces seamlessly to blockchain network, which further aids its performance and operational fairness. From Figure 6, it can be observed that MBIKE reduced the quantum bit error rate (QBER) from 0.04 in traditional cryptography and 0.021 in BIKE to 0,012. In terms of smart contract security (shown in Figure 6) which is used to protect operations on a blockchain network, MBIKE outperforms traditional blockchain technology and BIKE by 21% and 8% respectively. Smart contract usually refer to the protection of blockchain source codes from intrusion and other software malfunctions. In terms of mining fairness (shown in Figure 7) MBIKE outperforms traditional blockchain technology and BIKE by 25% and 7% respectively. It should be noted that mining fairness refers to the fact that various nodes in blockchain transactions should receive number of block reward in direct proportion to the computational hash rate they contributed to the network. This means that a node with higher computational requirement (hash rate) should have faster access time than that with smaller transaction percentage. This scenario is usually violated in blockchain technology due to the presence of blockchain forks resulting from multiple miner in a network locating a valid block around the same time.

5.4. Comparison of MBIKE's Key Exchange and Encryption Security

Blockchain security is based on 256-bit encryption using RSA, SHA-256 and ECDSA in classical blockchain environment while MBIKE blockchain is based on 512 bit code based cryptography. Classical cryptography can be compromised (eavesdropped) in a blockchain network using quantum computers, however this is impossible with MBIKE. MBIKE'S QBER is $\leq 1.5\%$ as shown in Figure 6 which is a confirmation of its secure key exchange. Though key exchange with classical cryptography's pseudo randomness has entropy of around 0.85, the key exchange using pseudo randomness in MBIKE is 0.995. This is due to the NP-complete nature of the McEliece cryptosystem public key encryption used in MBIKE. These properties make MBIKE code based cryptography have enhanced blockchain security affording optimum resilience, integrity and confidentiality to existing and predicted future quantum threats with increased power of quantum computers in view.

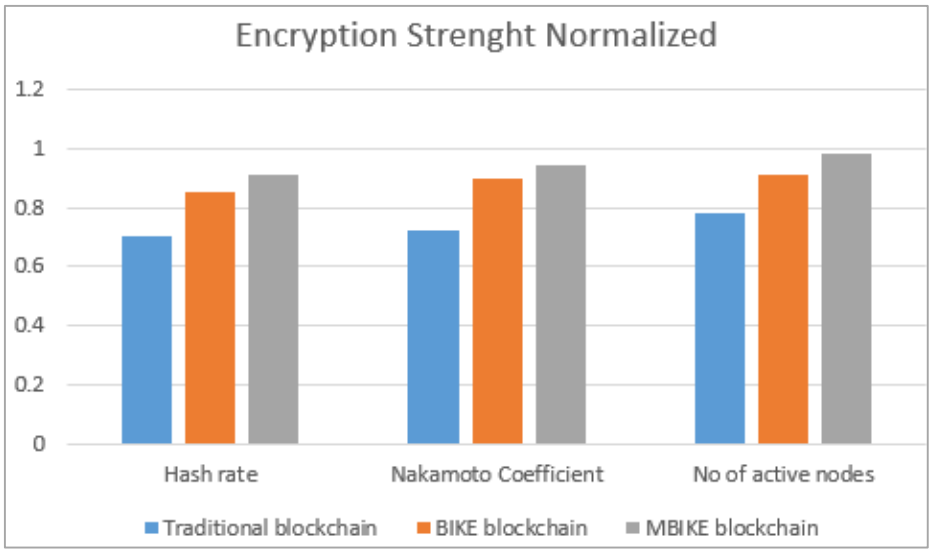

Figure 4. *Blockchain security comparison.*

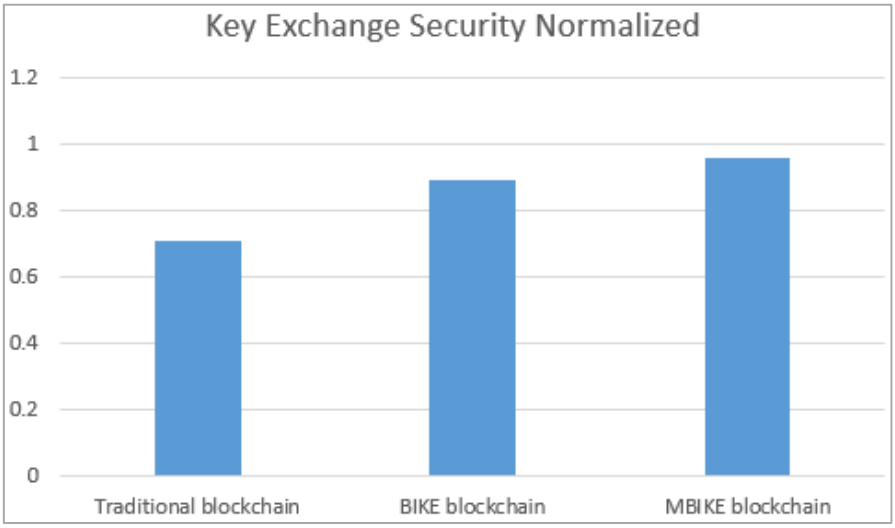

Figure 5. *Key Exchange security comparison.*

From Figure 8, it can be observed that while classical cryptography can easily be compromised by quantum threats, Quantum based MBIKE is resilient to quantum threats, its improved encryption efficiency / resiliency compared to classical cryptography is 100%, while it outperforms BIKE by 16%. It also improves key exchange confidentiality in comparison to classical cryptography and BIKE by 1125% an 25% respectively. The QBER in MBIKE is lower to both classical cryptography and BIKE cryptography by 65% and 30% respectively. The entropy value, i.e. unpredictability of the Post Quantum Pseudo Random Number (PQ-PRNG) was increased in comparison to classical cryptography and BIKE by 17% and 4% respectively. These results validate the improvement of MBIKE's blockchain technology with respect to the different performance metrics when compared to the traditional and BIKE's blockchain systems.

Table 4. *Comparison of traditional and BIKE blockchain with MBIKE blockchain network.*

	Transaction per second (TPS)	**Transaction Finality time**	**Level of Smart contract security**	**Mining Fairness**	**Interoperability Security**
Traditional Blockchain	65	10	4	3	2
BIKE Blockchain	59	12	6	5	4
MBIKE Blockchain	47.5	15	8	7	6

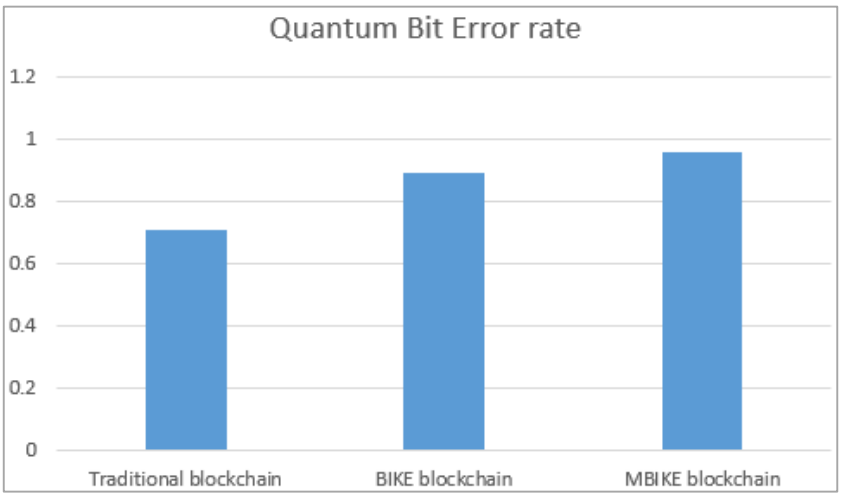

Figure 6. *Quantum Bit Error Rate comparison.*

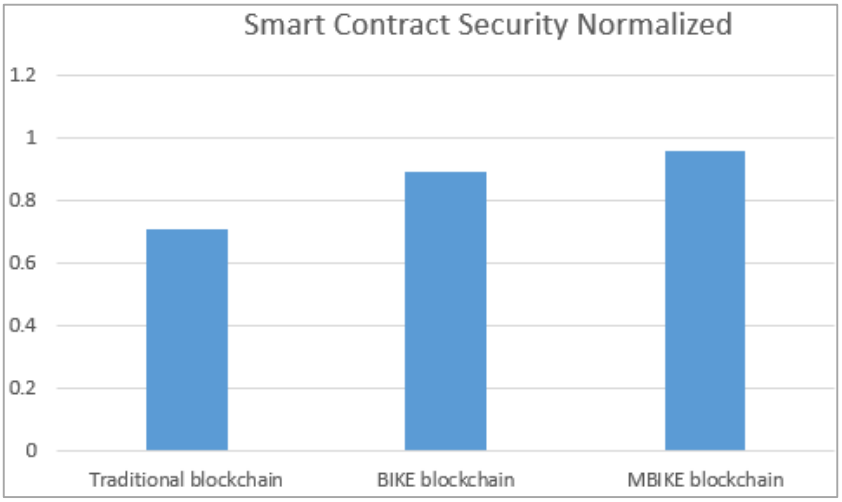

Figure 7. *Smart Contract security comparison.*

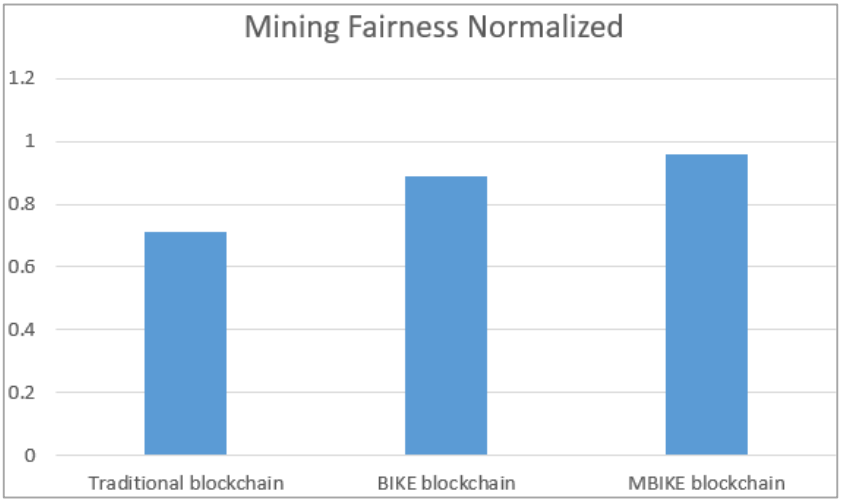

Figure 8. *Comparison of Mining Fairness.*

5.5. MBIKE's Analysis on Blockchain Transaction and Security

The combination of Post Quantum Pseudo Random Number Generator (PQ-PRNG) and MBIKE resulted in improvement in various performance metrics related to transaction security from the experiments conducted in this paper. From Table 3, the transaction per second (TPS) was reduced from 65 TPS in classical cryptography to 47.5 TPS in MBIKE. This is due to the increased overhead i.e. syndrome size used in MBIKE compared to both the original BIKE and classical cryptography algorithm, this was done in order to enhance the security of the code. The cyclic polynomial size was also increased in order to incorporate a higher bit error vector size, this has the effect of making higher bit size error introduced to the encrypted data, and this will make it even more difficult for an eavesdropper to detect locations of errors in the code word, thereby improving the security of the code. In the same vein, the transaction finality time was increased from 10 seconds in classical cryptography to 15 seconds in MBIKE, however this increase was more than made up for in terms in increased quantum threat protection.

The use of Post Quantum Pseudo Random Number Generator (PQ-PRNG) improved the unpredictability of the key exchange mechanism (KEM), the size of the error bit was increased in MBIKE as compared to BIKE so as to make the decryption of the encrypted data from the source difficult for the eavesdropper. The PQ-PRNG is an NP-complete system. The improved KEM enhances blockchain transaction security across different blockchain platforms. MBIKE can easily be incorporated to many blockchain platforms due to its enhanced interoperability feature as shown in Table 4, this ultimately leads to efficient and secure communications across various blockchain networks.

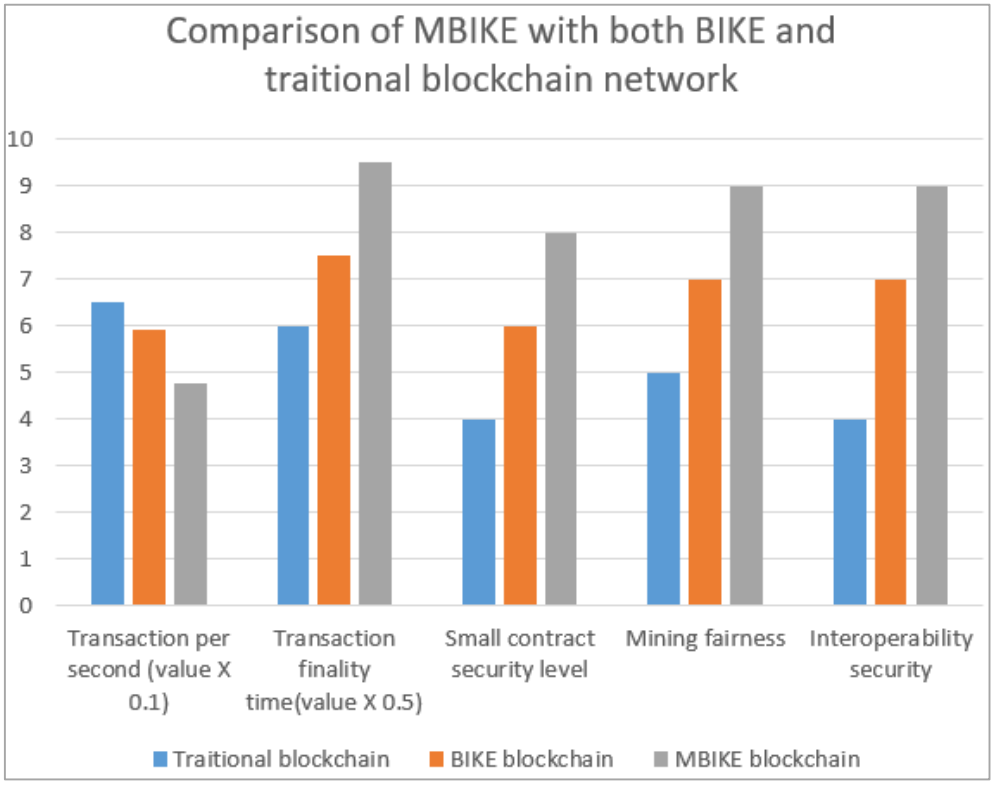

Figure 9. *MBIKE; s comparison with both BIKe and traitional blockchain network performance.*

It can be noticed from Table 6 that the performance of post quantum cryptography algorithm (MBIKE) outweigh that of classical cryptography and BIKE cryptography algorithms in the Blockchain system. Classical cryptography has 4% and 6% better performance in terms of transaction speed compared to BIKE and MBIKE respectively, In terms of transaction finality time (i.e. time taken to complete communication between sender and receiver) in a blockchain network, classical cryptography outperforms BIKE an MBIKE by 3% and 5% respectively. However in terms of operation transparency (mining fairness also shown in Figure 8), MBIKE outperforms BIKE and classical cryptography by 65% and 12% respectively, while in terms of protection against quantum threats (interoperability security), MBIKE outperforms BIKE and classical cryptography by 60% and 9% respectively. In terms of resilience to smart contract security, MBIKE outperforms BIKE and classical cryptography by 57% and 10% respectively.

5.6. Analysis of MBIKE's Resilience to Quantum Attack

It was earlier emphasized that Shor's algorithm can compromise RSA cryptosystem in polynomial time, while Grover's algorithm can quicken brute force attack on SHA-256 hash code s in polynomial time. Classical key exchange is vulnerable to Man In the Middle (MITM) attack, however, MBIKE is resilient to all these attacks due to its NP-complete code based nature, i.e. it is difficult to decode random linear codes with the addition of random errors in the code words by quantum computers, in order words its key exchange mechanism is quantum attack resilient The ability of the combination of MBIKE which is a Post Quantum Cryptography protocol and Post Quantum Pseudo Random Number Generator (PQ-PRNG) to mitigate the weakness to quantum attacks observed in other classical cryptography algorithms is shown in Figure 9.

The Combination of MBIKE and Post Quantum Pseudo Random Number Generator (PQ-PRNG) improve the security of blockchain network when compared to both traditional blockchain systems and BIKE systems. It should be noted that RSA-2048, ECDSA and SHA-256 are the encryption techniques used in traditional blockchain systems, however as stated earlier these cryptographic techniques can be compromised by quantum computing algorithms such as Shor's algorithm and Grover's algorithm. From the results in the experiments carried out in this section and as shown in Figure 10, it was observed that MBIKE is resilient to quantum attacks. The Man in the Middle attack cannot compromise MBIKE due to its NP-complete code based cryptography nature, this also makes it secure against upcoming and future quantum threats. The Post Quantum Pseudo Random Number Generator (PQ-PRNG) method used also increases the difficulty of adversarial nodes in identifying the randomness of the key distribution mechanism thereby making it impossible to decrypt the cipher text from the sender. The only shortfall is the increased transaction per second (TPS) and coherence time resulting from the

overhead i.e. syndrome size used in MBIKE is higher compared to both the original BIKE and classical cryptography algorithm. This was done in order to enhance the security of the code. The cyclic polynomial size was also increased in order to incorporate a higher bit error vector size, this has the effect of making higher bit size error introduced to the encrypted data, and this will make it even more difficult for an eavesdropper to detect locations of errors in the code word, thereby improving the security of the code. On the long run, the deployment of the MBIKE on a blockchain system will enhance fairer mining and improved security to upcoming and future quantum attacks.

Table 5. *Comparison of traditional and BIKE blockchain network security with MBIKE in the face of Quantum threats.*

	Encryption Strength (Normalized)	Key exchange Vulnerability	Quantum Bit Error rate (QBER)	Key Generation randomness (Entropy) Normalized
Traditional Blockchain	0.78	10	15	0.85
BIKE Blockchain	0.89	100	10	0.99
MBIKE Blockchain	0.96	125	5	0.995

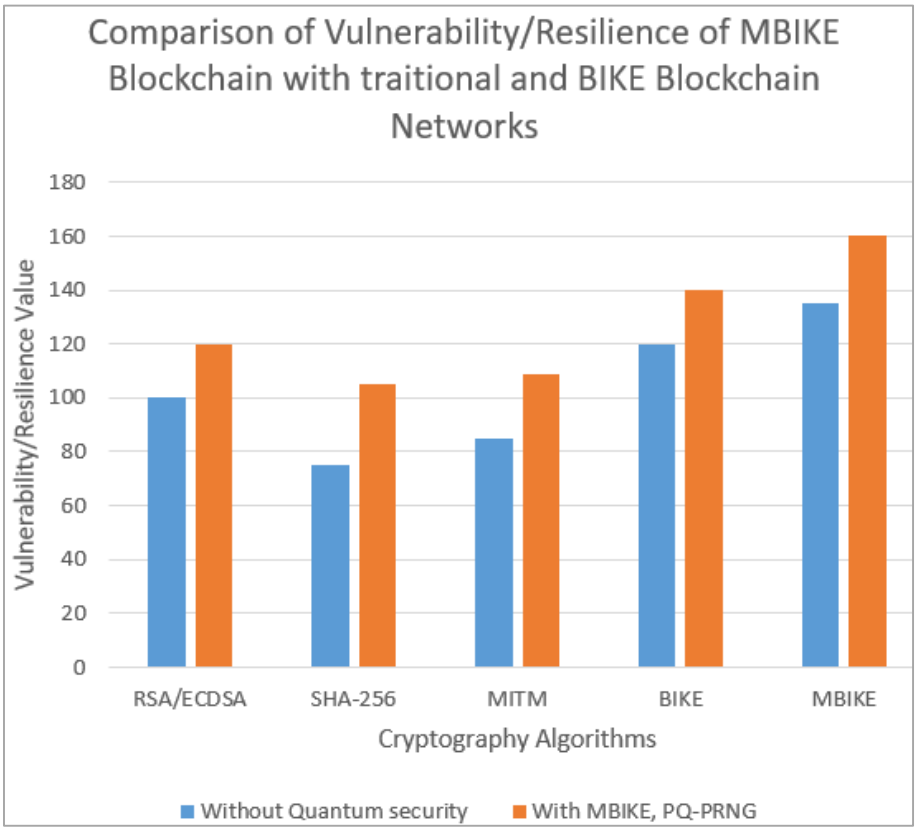

Figure 10. *Comparison of Vulnerability/Resilience of MBIKE Blockchain with traitional and BIKE Blockchain networks.*

From Table 5, it can be seen that MBIKE in combination with PQ-PRNG show remarkable resilience to quantum attacks in a blockchain network as opposed to the classical blockchain system. In terms of encryption strength, MBIKE outperforms BIKE and the classical blockchain system by 14% and 23% respectively, in terms of key exchange security, MBIKE outperforms BIKE and the classical blockchain system by 14% and 35% respectively, in terms of key generation randomness, MBIKE outperforms BIKE and the classical blockchain system by 10% and 33% respectively, in terms of quantum bit error rate, MBIKE outperforms BIKE and the classical blockchain system by 22% and 38% respectively,

Table 6. Comparison of MBIKE to both BIKE and traditional blockchain network in terms of various network security performance metrics.

Performance Metrics	ES	KES	RKG	QBER	TPS	TFT	MS	IS	RSA	RGA
Cryptography Algoritms	Security an Performance values									
Traditional Blockchain	7	8	7	0	65	10	6	9	2	4
BIKE Blockchain	12	13	13	12	59	14	13	14	17	16
MBIKE Blockchain	18	20	19	19	47.5	20	19	20	25	24

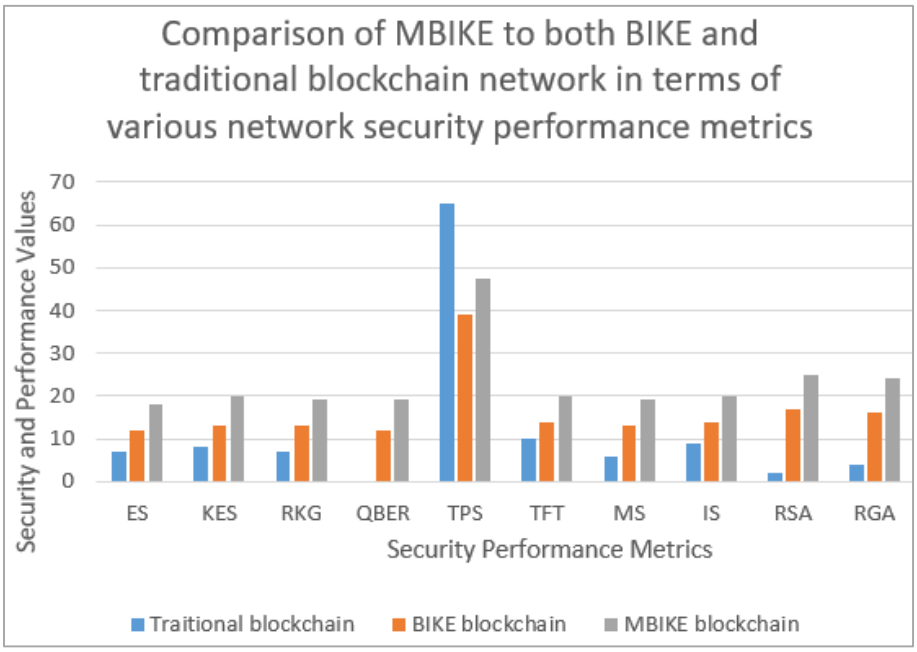

Figure 11. Comparison of MBIKE to both BIKE and traditional blockchain network in terms of various network security performance metrics.

Table 6 and Figure 11 show the general MBIKE blockchain network performance with both BIKE and traditional blockchain network using different network performance metrics. From the table it can be seen that MBIKE outperforms both BIKE and the traditional blockchain network in terms of encryption strength by 50% and 112% respectively. In terms of Key exchange security, MBIKE outperforms both BIKE and the traditional blockchain network by 45% and 120% respectively. In terms of randomness in key generation, MBIKE outperforms both BIKE and the traditional blockchain network by 40% and 110% respectively. In terms of Quantum bit error rate, it will be noticed that the traditional blockchain network has a value of 0 as it is not resilient to Quantum attacks, here MBIKE outperforms BIKE by 55%. In terms of transaction speed, the increased overhead in MBIKE as discussed earlier made it incur a lower transaction speed per second, here MBIKE has a lower TPS compare to both BIKE an traditional blockchain network by 13% an 27% respectively. In terms of transaction finality time, MBIKE outperforms both BIKE and the traditional blockchain network by 60% and 100% respectively. In terms of mining security, MBIKE outperforms both BIKE and the traditional blockchain network by 40% and 120% respectively. In terms of interoperability security, which etermines the flexibility of the algorithm to iffernf blockchain network configuration, MBIKE outperforms both BIKE and the traditional blockchain network by 40% and 110% respectively. In terms of resistance to Shor's algorithm, MBIKE outperforms both BIKE and the traditional blockchain network by 45% and 210% respectively, while in terms of resistance to Grover's algorithm, MBIKE outperforms both BIKE and the traditional blockchain network by 50% and 240% respectively.

5.7. Analysis of MBIKE's Blockchain Security Metrics with and Without Implementing Quantum Security

MBIKE in combination with the embedded QKD and PQ-PRNG show improved security and error free ledger transaction. Table 7 shows the improvements to security offered by MBIKE in combination with the embedded QKD and PQ-PRNG in a blockchain system. The following results were

highlighted (i) MBIKE resulted in increased ledger transaction accuracy from 84% in traditional blockchain to 99.95% in MBIKE. (ii) MBIKE outperforms the traditional blockchain system in terms of resilience to quantum threats by 36%. As a result of the NP-complete code based nature of MBIKE, the man in the middle attack (MITM) was completely repelled. (iii) Entropy in the key generation was increased from 0.85 in traditional blockchain system to 0.995 in MBIKE blockchain system. This is due to both use of Post Quantum Pseudo Random Number (PQ-PRNG) which increases the randomness and increase in the cyclic polynomial size used in incorporating a higher bit error vector size into the algorithm, thereby increasing the unpredictability in identifying position of bit errors in the input codeword. Overall, the resistance of MBIKE blockchain network to quantum attack was increased from 40% in traditional blockchain system to 99%, while (iv) mining fairness was improved by 34%. In MBIKE blockchain system compared to traditional blockchain network, it therefore provides a more decentralized and secure platform for banking and cryptocurrency. The only downside was the reduction in number of transactions per second (TPS) in MBIKE compared to the traditional encryption system due to the increased overhead due to both syndrome and cyclic polynomial size. The reduction in TPS was 17% in comparison to tradition encryption system.

Table 7. *Blockchain security before and after applying MBIKE algorithm.*

	Without Quantum security	**With MBIKE and PQ-PRNG**	**Improvement in accuracy**
Accuracy in transaction encryption	89%	99.98%	10.98%
Security of key exchange	72%	99.88	17.88%
Entropy value for key generation (Normalized)	0.88	0.995	0.115
Resilience to Quantum attacks	44%	99%	55%
Mining Fairness	77%	99.2%	22.2%

It has been shown by the work of researchers in [26] that the QC-MDPC variant of McEliece is not resistant to a specific reaction attack which takes advantage of the decoding failures which can prevent recovering the private key. This shortcoming is addressed in MBIKE with the increase in randomness of the codeword generated by the sender, it should be noted that MBIKE incorporate randomness into the key generation procedure used in decrypting the plaintext. In other words a public private key pair is generated for each cipher text. However instead of generating unique public/private key pair for each transaction, MBIKE uses the same private key for a full session of transaction, while the PQ-PRNG is used to repeatedly generate new public key for same private key for every transaction. With this approach, the reaction based key recovery cannot be realistic since different private/public key pair are generated for each ciphertext which would require the reaction based attack make several observations of the randomness introduced by the PQ-PRNG code.

5.8. Security of MBIKE Against Side Channel Attack

Side channel attacks cripples a cryptosystem by observing the implementation requirements (performance metrics) of the system [27], in other words there are many categories of side channel attacks, targeting different performance metrics of a system, i.e. timing attacks, power consumption attack, latency attacks. For example in the timing side channel attack, the threat attempts to obtain the time taken to complete every complete transfer from source to destination. In MBIKE, the system parameters are not fixed in dimension due to the randomness in error bit size introduced into the codeword generated from the sender as well as the sizes of the syndrome and cyclic polynomial size. These adjustable parameters bring about uncertainty into value of the performance metrics thereby making it very difficult for side channel attack to intrude.

A protection for the differential power analysis (DPA) is given below. Whenever any value from the data captured by a side channel attack is modified, the power consumption is depleted as shown in equation (11).

$$P = a \text{ X } t_0 + P \text{ X } t_0 + V \tag{11}$$

Where a represents the power consumed by the bit an attacker wants to capture, P X t_0 represents the power consumed due to bit t_0 and other bits (which the attacker isn't interested in) in the channel being checked by the attacker, while V represents the power consumed by bits not related to t_0.

When an attacker captures an event (bit) from a channel, it is represented in 3n bytes, where n denotes the size of each

word data. The attacker will have a higher probability of modifying bits in the channel if he is able to capture adequate information from his measurement to lower the number of attempts to 2^8 tries. When the hamming weight of captured data is either very low or very high, the attacker will be able to capture enough data from the data in order to compromise the data transmission, and in the extreme case that the hamming weight is zero, the attacker can capture all data in the channel. However, if it was assumed that the secret values are uniformly distributed, the probability distribution of the information available to the attacker can be calculated. Experiments were conducted using the experimental setup in section? to calculate the probability distribution for 64 bits and 128 bits words. From the results obtained from the experiments, it was deduced that the probability that an attacker captures enough information from the channel in order to compromise it is about 2^{-238}.

The probability distribution for different word size and security size was computed, the probabilities for every event that gives the attacker a higher likelihood above the desired security level was computed. The probabilities are shown in Table 8. From the table, it can be realized that the probability of channel data capture is lower than the desired security level.

Table 8. *Probability of attack as a result of hamming weight.*

Strength of Security	Size of word	Probability of leakage
128	32	2^{-225}
	64	2^{-240}
192	32	2^{-335}
	64	2^{-355}
256	32	2^{-443}
	64	2^{-472}

5.9. Analysis Discussion

The results obtained from the different experiments carried out in this paper attests to the fact that MBIKE which is a code base post quantum cryptography technique had overwhelming performance upgrade to the cryptography algorithm used in traditional blockchain systems. As discussed earlier, it had performance increase in its resilience to quantum attack due to its NP complete code based error correcting algorithm, improved mining transparency, and improved Time to live (TTL) of the ledger operations. With the embedded QKD, the system achieves efficient and optimum key distribution mechanism, while its increased syndrome and cyclic polynomial size almost make it impossible to detect locations of errors in the codeword making it almost impossible for attackers to decrypt the cipher text. The use of PQ-PRNG increases the randomness (entropy) in key distribution thereby making it difficult for attackers to guess the key used for decryption in any round of cryptography operation. All these preserves the integrity of data transmission and transaction confidentiality in the blockchain system.

6. Conclusion and Future Work

This paper prosed a Quantum Resistant Blocchain network using Modified Bit Flippping Key Encryption, the protocol comprised of MBIKE in combination with the embedded QKD and PQ-PRNG in a blockchain system. The protocol is capable of withstanding both current and expected quantum attacks. The MBIKE protocol is actually ideal for the decentralized nature of blockchain networks which can be used for the protection of customer's records/ledgers in banks as well as in cryptocurrency. Blockchain networks can be ideal for banks due to the fact that new ledgers (records) can only be admitted onto the network only when the entire participants in the blockchain agree to its joining, hence the security of the network or system is guaranteed. From the results gathered from the different experiments conducted, it was shown that MBIKE improved the resilence of a blockchain system using various performance metrics, some of which include encryption strength, key exchange security, randomness of distributed random keys and overall transaction transparency and optimal performance of the decentralized ledger in blockchain system. Results show that the proposed protocol improved quantum attack resilience of the blockchain system, while the precision of the encryption was 99.95%, with a reduced transaction per second due to its increase overhead. However this setback was more than made up for in quantum attack resilience. The protocol was able to address many shortcomings in traditional cryptographic system such as integrity, side channel attack, reaction attacks and confidentiality in the face of quantum attacks, thereby making it ideal for deployment in future blockchain system. The future direction in this work is to reduce the overhead in the protocol design so as to enhance TPS, together with inclusion of quantum machine learning algorithms for real time anomaly detection in data processing.

Abbreviations

ES	Encryption Strength
KES	Key Exchange Security
RKG	Randomness in Key Generation
QBER	Quantum Bit Error Rate
TPS	Transaction Per Second
TFT	Transaction Finality Time
MS	Mining Security
IS	Interoperability Security
RSA	Resistance to Shor's Algorithm
RGA	Resistance to Grover's Algorithm

Acknowledgments

The author acknowledges Department Computer Engineering, Abia State University for making available the use of their laboratories.

Author Contributions

Fagbohunmi Grifin Siji is the sole author. The author read and approved the final manuscript.

Funding

This section is not applicable. The author used his funds to get the materials and used the facility of the faculty.

Data Availability Statement

The Data for this work can be gotten from the author.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] D. Herman (2023) "Quantum computing for finance," Nat. Rev. Phys., vol. 5, no. 8, pp. 450–465, 2023.

[2] D. Gurung, S. R. Pokhrel, and G. Li, (2023) "Performance Analysis and Evaluation of Post Quantum Secure Blockchain Federated Learning," ArXiv Prepr. ArXiv230614772, 2023.

[3] M. K. Hadap, (2024) "LDQKDPB: Unbreakable Network Security via Long- Distance Quantum Key Distribution Enhanced by Post-Quantum Techniques and Blockchain," Commun. Appl. Nonlinear Anal., vol. 31, no. 2s, pp. 561–571, 2024.

[4] S. Dhar, A. Khare, A. D. Dwivedi, and R. Singh, (2024) "Securing IoT devices: A novel approach using blockchain and quantum cryptography," Internet of Things, vol. 25, p. 101019, 2024.

[5] S. Bhimajiyani, (2024) "Quantum-Resilient Self-Evolving Blockchains: AI- Driven Consensus and Autonomous Security Upgrades," International Journal of Innovative Science Resolution. Technology IJISRT.

[6] J. Gomes, S. Khan, and D. Svetinovic, (2023) "Fortifying the blockchain: A systematic review and classification of post-quantum consensus solutions for enhanced security and resilience," IEEE Access, vol. 11, pp. 74088–74100, 2023.

[7] G. Nkulenu, (2024) "Quantum Computing: The Impending Revolution in Cryptographic Security," 2024.

[8] J. J. Tom, N. P. Anebo, B. A. Onyekwelu, A. Wilfred, and R. Eyo, (2023) "Quantum computers and algorithms: a threat to classical cryptographic systems," Int J Eng Adv Technol, vol. 12, no. 5, pp. 25–38, 2023.

[9] R. A. Jowarder and S. Jahan, (2024) "Quantum computing in cyber security: Emerging threats, mitigation strategies, and future implications for data protection," World J. Adv. Eng. Technol. Sci., vol. 13, pp. 330–339, Sep. 2024, https://doi.org/10.30574/wjaets.2024.13.1.0421

[10] L. R. Desai, P. Malathi, R. R. Bandgar, H. Joshi, A. S. Kore, and R. Y. Totare, (2025) "Advanced Techniques in Post-Quantum Cryptography for Ensuring Data Security in the Quantum Era," 2025, https://doi.org/10.52783/pmj.v35.i1s.2097

[11] A. Al Sadawi, M. S. Hassan, and M. Ndiaye, (2021) "A survey on the integration of blockchain with IoT to enhance performance and eliminate challenges," IEEe Access, vol. 9, pp. 54478–54497, 2021.

[12] S. K. Sahu and K. Mazumdar, (2024) "State-of-the-art analysis of quantum cryptography: applications and future prospects," Front. Phys., vol. 12, 2024.

[13] E. Zeydan, J. Baranda, and J. Mangues-Bafalluy, (2022) "Post-quantum blockchain-based secure service orchestration in multi-cloud networks," IEEE Access, vol. 10, pp. 129520–129530, 2022.

[14] H. Alyami (2022) "Analyzing the data of software security lifespan: quantum computing era," Intell. Autom. Soft Comput., vol. 31, no. 2, pp. 707–716, 2022.

[15] D. Harinath, M. Bandi, A. Patil, M. Murthy, and A. Raju, (2024) "Enhanced Data Security and Privacy in IoT devices using Blockchain Technology and Quantum Cryptography," Journal of Systematic Engineering Electron. ISSN NO 1671-1793, vol. 34, no. 6, 2024.

[16] G. BigQuery, "Bitcoin Blockchain Historical Data." 2019. [Online]. Available: https://www.kaggle.com/datasets/bigquery/bitcoin-blockchain

[17] V. Ganti and A. D. Sarma, Data Cleaning. Springer Nature, 2022.

[18] S. Rouhani and R. Deters, (2021) "Data trust framework using blockchain technology and adaptive transaction validation," IEEE Access, vol. 9, pp. 90379–90391, 2021.

[19] I. Izonin, R. Tkachenko, N. Shakhovska, B. Ilchyshyn, and K. K. Singh, (2022) "A two-step data normalization approach for improving classification accuracy in the medical diagnosis domain," Mathematics, vol. 10, no. 11, pp. 1942 - 1958, 2022.

[20] C. Xu, F. Su, B. Xiong, and J. Lehmann, (2022) "Time-aware entity alignment using temporal relational attention," in Proceedings of the ACM Web Conference 2022, pp. 788–797.

[21] Z. Yang, H. Alfauri, B. Farkiani, R. Jain, R. Di Pietro, and A. Erbad, (2023) "A survey and comparison of post-quantum and quantum blockchains," IEEE Communication Survey Tutor., vol. 26, no. 2, pp. 967–1002, 2023.

[22] D. Hofheinz, K. Hövelmanns, and E. Kiltz, (2019)''A modular analysis of the Fujisaki–Okamoto transformation,'' in Proceedings Theory of Cryptography Conference Cham, Switzerland: Springer, 2019, pp. 341–371.

[23] N. Drucker, S. Gueron, D. Kostic, and E. Persichetti, (2021) ‘‘On the applicability of the Fujisaki–Okamoto transformation to the BIKE KEM,’’ International Journal of Computing Mathematics and Computer System Theory, vol. 6, no. 4, pp. 364–374, Oct. 2021.

[24] N. Drucker, S. Gueron, D. Kostic, J. Ding, and J. Tillich, (2022) ‘‘QC-MDPC decoders with several shades of gray,’’ in Proceedings Post Quantum Cryptography, vol. 12100, 2022, pp. 35–50.

[25] H. Niederreiter, (1996) ‘‘Knapsack-type cryptosystems and algebraic coding theory,’’ Problems Control Information Theory, vol. 15, no. 2, pp. 159–166, 1996.

[26] Q. Guo, T. Johansson, and P. Stankovski, (2020) ‘‘A key recovery attack on MDPC with CCA security using decoding errors,’’ in Proceedings International Conference Theory Application of Cryptology Information Security Cham, Switzerland: Springer, 2020, pp. 789–815.

[27] V. Dragoi, T. Richmond, D. Bucerzan, and A. Legay, (2022) ‘‘Survey on cryptanalysis of code-based cryptography: From theoretical to physical attacks,’’ in Proceedings 9th International. Conference of Computer Communication Control (ICCCC), May 2022, pp. 215–223.