Quantum digital signature based group key management in e-commerce

UDAYABHANU N P G RAJU 1,
DR. R.VIVEKANANDAM 2,
1,2 SSSUTMS- SEHORE BHOPAL MADHYA PRADESH

July 9, 2018

Abstract
The Internet development is high over the past decades and there are lot of cloud computing technology is used in business and other purpose. This leads to the major trend to the security in the cloud and secure communication is much required. The encryption technology is used to communicate with the member of the group and not seen by other member, using a shared key technology. There are few methods provides the security to user and still lacking in the secure communication. Then the quantum cryptography is used to communicate and this provide better security to the user than the previous method. Quantum Key Distribution (QKD) is one of the most common method used to provide better encryption. The main focus of this research is that provide secure communication from the group keying. The Quantum Digital Signature based Group Key management (QDSGKM), this combine QKD with the Fibonacci, Lucas and Fibonacci-Lucas matrices, which provides the quantum digital signature checks. This method gives the technique to verify the integration of information received by the participants, to authenticate the identity of the participants and to improve the verification of the signing-verification. The experimental result shows that the proposed system provides the security with less delay when compared to QKD system.
Keywords: Encryption technology, Quantum cryptography, Quantum Digital Signature based Key Management, Quantum Key Distribution, Secure communication, Fibonacci-Lucas matrices.

1 Introduction

The fast development of the internet and cloud computing improves the significant of the research of the quantum cryptography [1]. The cloud computing has the significant impact on both scientific and business in information technology. In order to prevent the message sent in the group communication from other members to read, the encryption method is needed [2]. The applications are able to communicate with each other in a secure way from the help of shared key in a secure manner. The important aspect that consider is that quantum key distribution is perfectly safe for the quantum key distribution [3]. The data owners store the encrypted data into cloud, in which that user want to access. The decryption key and digital certificates can be distributed to the user by the data owners if the access control policy permits the user [4]. Then the user shows the certificate to the cloud and can get the access to the encrypted data. It is well known that once the large quantum computer is built and the existing encryption system is efficiently broken [5], [6]. Cloud computing grows fast and has quite expensive in application for instance e-commerce, which is the new type of transaction that brings the consumer, logistic and enterprise into a compressive network era. In theoretically, quantum computer is unbreakable and this is highly needed in quantum computer era. Classical communication is the submission of data in the unsecure connection, for example internet [7].

The several algorithm is presented to prevent the leak in the communication and also to withstand some attacks [8]. These algorithms are considered as secure until the quantum computer is publicly available. On the other hand, quantum cryptography provides the secure for the large information for the shared data and communication, even after the quantum computer is found [9], [10]. Then the quantum attack of natural noise gives the major challenge for the quantum encryption. These challenge is prevented by using ro-
bust secret key builder that is made by group key. The QKD is the mechanism of two user initiate the secret shared key (SSK). This key helps to encrypt and decrypt the data in the secure connection. The various QKD protocol was proposed like KMB09 protocol, Coherent One Way (COW) protocol, EPR protocol, SARG04 protocol, and B92 protocol. The payment service is the most important link in the whole transaction and the mobile payment is highly used. In this method, the QKD is improved with Fibonacci, Lucas and Fibonacci-Lucas matrices. The quantum signature is verified from the Charlie and authenticate to the user then the information is provided. The proposed method compared with the QKD method and it has the low delay.

2 Literature review

Mohajer and Eslami [11], shows that the scheme is not secure against attack when there is no trust is present between the participants and also provide the method to improve protect against the attack. Improved each partly send his share to others and each party is bind to his contribution of the final secret key, which help to track the cheaters. The protocol is claimed that it is secure against the inside and outside attackers and attack was proposed that rejects its claim for inside attackers. All participants retrieve the final key simultaneously and there is no guarantee that all the participants received the final key.

Mogos [12], used the twelve orthogonal states in a four-state system in the Quantum key distribution of software implementation. For twelve states, the sender encodes the two bits classical information in one particles and secrete key is distributed to the receiver. This method explores the quantum alternatives to the traditional key distribution protocols, and involves implementations of quantum key distribution protocol on two cases: without cyber-attacks and with cyber-attacks. The error rate of the method is high.

Qiu, [13]. Devoted this paper to investigate the quantum cryptography in access control and for the key distribution, identity authentication and digital certification, three quantum protocol had been designed. The protocol had been analyzed with the graphical language of categorical quantum mechanics. These protocols are
secure and implemented by the current technologies. The method is still vulnerable for man-in-the-middle attack.

Tanizawa, et al [14]. Proposed a method for the secure communication that select between standard cryptography-based security and QKD-based automatically according to the current value of QKD key by QKD. This method provides the secure communication with QKD key and in the shortage of QKD key, this provide the provide the standard cryptography-based secure communication function to reduce the communication delay. This provide the secure communication along with the small delay than the standard cryptography method. This method still has the delay, so it is applicable for only some applications.

Metwaly, et al [15]. Centralized Quantum multicast key distribution and classic symmetric encryption is used to provide the secure communication and distribution method for the one sender to one or more receiver. In this method, the encryption, decryption and messaging among multicast group members by the symmetric classic encryption algorithm and the key is generated from the QKD protocols, which are used for authentication, encryption and decryption plays an important role cryptosystem. This method has the advantages in the protection of keys and authentication using quantum physics laws. This technique need to have quantum key distribution and hashing-based authentication helps to implemented on the multicast network.

3 Proposed method

The matrices are integrated in the QKD to provide the secure connection is cloud. The Charlie is used to verify the signature of the user and provides the information if the signature matches the Al-ices signature. The matrices used to combine the QKD is Fibonacci, Lucas and Fibonacci-Lucas matrices.

3.1 Quantum Key Distribution

The quantum key distribution protocol of the Simon et al.s and also shows that how protocols can be improved in coding efficiency of entangled states, while use the Fibonacci, Lucas or Fibonacci-Lucas matrices.
3.1.1 Fibonacci Matrices:

Fibonacci number $F_n$ are an infinite sequence of integers defined in the following recursion:

$$F_n = F_{n-1} + F_{n-2}, n \geq 2$$  \hspace{1cm} (1)

Where the first two elements of the sequences are $F_0 = 0$ and $F_1 = 1$. Taking the first three integers $F_0, F_1, F_2$ of the Fibonacci sequence, construct a $2 \times 2$ Fibonacci matrix:

$$Q_1 = \begin{pmatrix} F_0 & F_1 \\ F_1 & F_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$  \hspace{1cm} (2)

Where $\det(Q_1) = F_0F_2 - F_1^2 = -1$. Using recursion (1), compute the $n^{th}$ power of the Fibonacci matrix $Q_1$ as follows:

$$Q_1^n = \begin{pmatrix} F_{n-1} & F_n \\ F_n & F_{n+1} \end{pmatrix}$$  \hspace{1cm} (3)

Since $\det(A^n) = (\det(A))^n$, $\det(Q_1^n) = (\det(Q_1))^n = (-1)^n$. This shows that Fibonacci matrices

$$Q_1^{2k} = \begin{pmatrix} F_{2k+1} & F_{2k} \\ -F_{2k} & F_{2k-1} \end{pmatrix}, \text{ for } n = 2k;$$  \hspace{1cm} (4)

$$Q_1^{2k+1} = \begin{pmatrix} F_{2k+1} & F_{2k+1} \\ -F_{2k+1} & F_{2k} \end{pmatrix}, \text{ for } n = 2k + 1;$$  \hspace{1cm} (5)

Construction of $Q_p$. The new class of Fibonacci matrices $Q_p$, where $p=2,3$, and $Q_1$ is given in equation (2). The class satisfies the following relation:

$$Q_p = \begin{bmatrix} Q_1 & Q_1 & \cdots & Q_1 & Q_1 \\ 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & I \end{bmatrix}$$

Where $O$ is the $22$ matrix has the zero entries and $I$ is $22$ Identity matrix. It is easy to prove that matrices $Q_p^n$ satisfy the following properties in terms of (3):

$$\det(Q_p^n) = (\det(Q_p))^n = (-1)^pn.$$  \hspace{1cm} (6)
$Q^n_p$ (where $p=1,2,3,$) is invertible and its inverse can be calculated using (4) and (5) is shown as follows

$$Q^{-1}_1 = \begin{bmatrix} Q^{-1}_1 & -I & \cdots & -I & -I \\ 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \ddots & 0 \\ \cdots & \cdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & \cdots & I \end{bmatrix} \text{ where } p = 1, 2, 3, \ldots \quad (7)$$

3.1.2 Lucas matrices:

Lucas numbers $L_n$ are an infinite sequence of integers, defined by the following recursion holds

$$L_n = L_{n-1} + L_{n-2}, n \geq 2, \quad (8)$$

Where the integers $L_0 = 2$ and $L_1 = 1$ start the sequences and $n=1,2$, Lucas and Fibonacci numbers share the following conjugate relation:

$$L_n = F_{n-1} + F_{n-1} \quad (9)$$

Let us define a 22 matrix $R_1$ as

$$R_1 = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix} \quad (10)$$

According to (1) and (3), the $n'th$ is define the power of $R_1$ as

$$R^n_1 = \begin{pmatrix} L_{n-1} & L_n \\ L_n & L_{n+1} \end{pmatrix} = Q^n_1 \times \begin{pmatrix} -1 & 2 \\ 2 & 1 \end{pmatrix}, \quad (11)$$

$$\text{det}(R^n_1) = \text{det}(Q^n_1 \times \begin{pmatrix} -1 & 2 \\ 2 & 1 \end{pmatrix}) = 5 \times (-1)^{n+1} \quad (12)$$

This imply that $R^n_1$ is invertible and its inverse matrix $R^{-n}_1$ can also be derived using the properties of Lucas sequences. They are

$$R^{-2k}_1 = \begin{pmatrix} \frac{L_{2k+1}}{L_{2k}} & -\frac{L_{2k}}{L_{2k+1}} \\ -\frac{L_{2k}}{L_{2k+1}} & \frac{L_{2k+1}}{L_{2k}} \end{pmatrix} \text{ for } n = 2k \quad (13)$$
The matrix $R_1$ is used to build the new class of Lucas matrices $R_p$ that satisfy the following relation:

$$R_p = \begin{bmatrix}
R_1 & R_1 & \cdots & R_1 & R_1 \\
0 & I & 0 & \cdots & 0 \\
0 & 0 & I & \ddots & 0 \\
\vdots & \vdots & \ddots & \ddots & 0 \\
0 & 0 & \cdots & 0 & I
\end{bmatrix}$$

The matrices $R_p^n$ stratify the following properties:

$$\det(R_p^n) = (\det R_p)^n = (-1)^{p(n-1)}5^p$$ \hspace{1cm} (14)

The $R_p^n$ are invertible and inverse is calculated is shows as

$$R_p^{-1} = \begin{bmatrix}
R_1^{-1} & -I & \cdots & -I & -I \\
0 & I & 0 & \cdots & 0 \\
0 & 0 & I & \ddots & 0 \\
\vdots & \vdots & \ddots & \ddots & 0 \\
0 & 0 & \cdots & 0 & I
\end{bmatrix}$$

### 3.1.3 Fibonacci-Lucas Matrices:

Fibonacci and Lucas sequences can be jointly to create a new class of matrices, which we call them Fibonacci-Lucas matrices. They are consecutive power of $T_1$ and are defined according to the following recursion:

$$T_1^n = \begin{bmatrix}
F_{n-1} & F_n \\
L_{n-2} & L_{n-1}
\end{bmatrix}$$ \hspace{1cm} (15)

Where the first Fibonacci-Lucas matrix $T_1$ is

$$T_1 = \begin{bmatrix}
F_1 & F_2 \\
L_0 & L_1
\end{bmatrix} = \begin{bmatrix}
1 & 1 \\
1 & 1
\end{bmatrix}$$ \hspace{1cm} (16)

Lucas and Fibonacci numbers satisfy the relation $L(n-1) = F_n + F(n-2)$, thus $T_1^n$ can be written as

$$T_1^n = \begin{bmatrix}
F_{n-1} & F_n \\
L_{n-1} + F_{n-3} & F_n + F_{n-1}
\end{bmatrix}$$ \hspace{1cm} (17)

The determinant of Fibonacci-Lucas matrices. The following transformation are self-explanatory
\[
\det(T^n_1) = \det \begin{pmatrix}
F_{n-1} & F_n \\
L_{n-1} + F_{n-3} & F_n + F_{n-2}
\end{pmatrix} = \det \begin{pmatrix}
F_{n-1} & F_n \\
F_{n-1} & F_n
\end{pmatrix} + \\
\det \begin{pmatrix}
F_{n-1} & F_n \\
F_{n-3} & F_{n-2}
\end{pmatrix} = (-1)\det \begin{pmatrix}
F_{n-3} & F_{n-2} \\
F_{n-1} & F_n
\end{pmatrix}
\]

(18)

The properties of Fibonacci matrices

\[
\det(T^n_1) = (-1)^{n-3}, n = 4, 5, ...
\]

(19)

The inverse matrix \(T^n_1 - n\) is

\[
\begin{pmatrix}
L_{2k+1} & -F_{2k} \\
-L_{2k} & -F_{2k}
\end{pmatrix}
\]

(20)

for \(n\) even

\[
\begin{pmatrix}
-L_{2k+1} & -F_{2k} \\
L_{2k} & -F_{2k}
\end{pmatrix}
\]

(21)

for \(n\) odd

The matrix \(T^n_p\) can be used to produce matrices of higher dimensions \(T^n_2, T^n_3, T^n_p\).

Construction of \(T^n_p\), The matrix \(T_1\) to build a new class of Fibonacci-Lucas matrices \(T^n_p\) that satisfy the following relation:

\[
T_p = \begin{pmatrix}
T_1 & T_1 & \cdots & T_1 & T_1 \\
0 & I & 0 & \cdots & 0 \\
0 & 0 & I & \ddots & 0 \\
\vdots & \vdots & \ddots & \ddots & 0 \\
0 & 0 & \cdots & 0 & I
\end{pmatrix}
\]

It is easy to prove that matrices \(R^n_p\) satisfy the following properties

\[
\det(T^n_p) = (\det(T^n_p))^n = (-1)^{pn}
\]

(22)

The \(R^n_p\) are invertible and their inverses can be calculated, where \(p = 1, 2, 3, \ldots\). Their inverse as follows
3.1.4 Matrix Encryption

Consider a message that is a sequence of integers \( m_i (i = 1, 2, \ldots) \). Integers of the message can be packed into a square \( l \times p \) matrix \( M \). The arrangements of message in \( M \) can be to some extent arbitrary as integers can be determined by selecting odd or even number of digits. For instance, assume we have message 489165723489625471635, then we can create a 34 matrix

\[
\begin{pmatrix}
48 & 91 & 65 & 723 \\
4 & 89 & 6 & 25 \\
47 & 16 & 3 & 5
\end{pmatrix}
\]

Given a matrix \( K \) matrix encryption can be defined as follows

\[
E = M \times K
\]

The decryption can be done using the inverse matrix \( K^{-1} \)

\[
M = E \times K^{-1}
\]

Where \( K \) can be either \( Q^n_2 \) or \( R^n_2 \) or \( T^n_p \). For instance, consider again the message 489165723489625471635 and the key matrix

\[
K = Q^n_2 = \begin{pmatrix}
8 & 13 & 8 & 13 \\
13 & 21 & 13 & 21 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

Then the matrix encryption is

\[
E = M \times K = \begin{pmatrix}
48 & 91 & 65 & 723 \\
4 & 89 & 6 & 25 \\
47 & 16 & 3 & 5
\end{pmatrix} \times \begin{pmatrix}
8 & 13 & 8 & 13 \\
13 & 21 & 13 & 21 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

Decryption is

\[
T_P = \begin{pmatrix}
T^{-n} & -I & \ldots & -I & -I \\
0 & I & 0 & \ldots & 0 \\
0 & 0 & I & \ldots & 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots \\
0 & 0 & \ldots & 0 & I
\end{pmatrix}
\]
\[ E = M \times K^{-1} = \begin{pmatrix} 48 & 91 & 65 & 723 \\ 4 & 89 & 6 & 25 \\ 47 & 16 & 3 & 5 \end{pmatrix} \times \begin{pmatrix} 8 & 13 & 8 & 13 \\ 13 & 21 & 13 & 21 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} -21 & 13 & -1 & 0 \\ 13 & -8 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 48 & 91 & 65 & 723 \\ 4 & 89 & 6 & 25 \\ 47 & 16 & 3 & 5 \end{pmatrix} \]

The symmetric encryption is given in the encryption method and symmetric cryptography needs a secure channel to distribute secret keys between two communicating parties. This is the linear encryption and this can be easily breakable in a chosen plaintext attack. Note that if the key matrix K is either \( Q_p \) or \( R_p \) or \( T_p \), then the matrix K can be determined after two elements of the matrix are known. This is due to the recursive nature of the matrix and the crucial point is that we use the matrix K to encode quantum states. The Fibonacci sequences to prepare entangled states and two communicating parties can detect the Fibonacci values for the entangled states with the designated sorters. More importantly, \( Q_p \) or \( R_p \) or \( T_p \) can be used just one time and their order is determined by quantum random generators in Alices, Bobs and Charlies side. Considering these, the quantum matrix encryption is secure.

### 3.1.5 Re-encryption QKD

The ReEnc-QKD is constructed by using meta proxy re-encryption scheme and the scheme is shown to be group key indistinguishable.

**Building Blocks:** The proxy re-encryption is the encryption technique that allows a proxy to re-encrypt a cipher text under encryption key \( e_k_A \) into other cipher text under another encryption key \( e_k_B \) by using the re-encryption key \( r_k(A \rightarrow B) \) from A to B without revealing the plain text. A pre-scheme is unidirectional if \( r_k(i \rightarrow j) \) cannot be used to compute \( r_k(j \rightarrow i) \). It is multi-hop if the cipher text can be re-encrypted many times in sequences. A Pre-scheme consists of six poly time algorithm.

1. \( \text{setup} ()(MK,sp) \). This setup algorithm generates the master...
secrete MK and the public parameter sp by inputting the
security parameter \( \lambda \).

2. KG(MK,i) \( \rightarrow \) \((ek_i, dk_i)\). This key generates the algorithm encrypt the plain text \( m \) into a cipher text \( c_i \).

3. RKG\((dk_i, ek_j) \rightarrow rk_{ij}\). This encryption key generate the algorithm generate a re-encryption key \( rk_{ij} \).

4. ReE\((rk_i \rightarrow j, c_i) \rightarrow c_j\). This re-encryption algorithm re-encrypt the cipher text \( c_i \) into \( c_j \).

5. DEC\((dk_i, c_i) \rightarrow m\). This decryption algorithm decrypt a cipher text \( c_i \) as a plain text \( m \).

This helps to build the group keying, re-encryption is made by this process to improve the security of the cloud.

3.1.6 Simons et al.s QKD protocol

The major idea behind Simon et al.s QKD protocol [16] is the use of a Vogel spiral and this allows either Alice or Bob to prepare a source of entangled Fibonacci-valued orbital angular momentum (OAM) states. The spiral is left after the Fibonacci-valued entangled pairs and enter the down-conversion crystal. The down-conversion breaks each Fibonacci value into two lower OAM values and in both Alices and Bobs laboratories, there is a beam splitter directing some regular proportion of the beam to two different types of OAM sorters L and D. The entangled photons are randomly transmitting the beam splitters to either the L or D sorters. The L sorter allows Fibonacci-valued entangled photons to reach at the single photon array detectors only. The D sorter allows diagonal superposition in the form \( \frac{1}{\sqrt{2}} (|F_n\rangle \times |F_{n+2}\rangle) \) and filter out any non-Fibonacci entangled photons. Namely, the entangled photon is sent to

1. L by the beam splitters in both Bobs and Alices side,

2. D and L by the beam splitters in Bobs and Alices side respectively,

3. L and D by the beam splitters in Bobs and Alices side respectively,
4. D by the beam splitters in both Bobs and Alices side.

For (1) to (3) cases are only available for the key establishment.

### 3.1.7 Improved QKD protocols

The classes of Fibonacci, Lucas and Fibonacci-Lucas matric coding, used to improve Simon et al.’s QKD protocol. Alice randomly prepares m two-photon entangled states $|\varphi\rangle_1, |\varphi\rangle_2$, which are shown in the equation (1).

\[
\Sigma_n(|F_{n-1}\rangle_s | F_{n-2}\rangle_i + |F_{n-2}\rangle_2 | F_{n-1}\rangle_i) ............(1)
\]

\[
\Sigma_n(|F_{n+1}\rangle_s | F_{n-1}\rangle_i + |F_{n-1}\rangle_2 | F_{n+1}\rangle_i) ............(2)
\]

Where the subscripts of s and i” represent the signal photon and the idler photon, respectively. For (1), the Alice have the one entangled photon and Charlie have the other through the unauthenticated quantum channel. For (2), one half entangle photo gives to the Alice and other goes to the Bob, respectively through insecure quantum channel. The entangled photons are received in both for Bob and Charlie in the state of either (1) or (2).

Each entangled photon goes to any one of the three sorters L, $D_1, D_2$ in the party sides randomly and independently and these party record the obtained outcome. Their states represent Fibonacci values, then L allows photons to arrive at the arrays of single-photon detectors. There possible result obtain: (1) $D_2$ got the both entangled photons; (2) $D_1$ got the both entangled photons; (3) $D_1$ got one entangled photons and $D_2$ got one entangled photons. The data in the parties are discarded and keep the entangled photons left. The Alice and Bob announces the set of states, whether he or she chooses via authenticated classic channels. The measurements are compared between the parties with the two entangled states. The information is exchanged among themselves using authenticated classic channels and they also detect Fibonacci or Lucas values used in the relevant matrices.

For instance, if 8 is the detected as the Fibonacci value, then the key matrix can be constructed as follows:
If 11 is the detected value, the key matrix can be constructed as follows:

\[
\begin{pmatrix}
8 & 13 \\
13 & 21
\end{pmatrix} = Q_1^7,
\begin{pmatrix}
8 & 13 & 8 & 13 \\
13 & 21 & 13 & 21 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix} = Q_2^7
\]

If the detected Fibonacci and Lucas values are \(F_n = 8 \equiv 0 \pmod{2}\) and \(L_n = 11 \equiv 1 \pmod{2}\), the Fibonacci-Lucas matrix should be

\[
\begin{pmatrix}
5 & 8 \\
11 & 18
\end{pmatrix} = T_1^4,
\begin{pmatrix}
5 & 8 & 5 & 8 \\
11 & 18 & 11 & 18 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix} = T_2^4
\]

### 3.2 Proposed Quantum Digital signature

The proposed Quantum Digital signature includes five steps: setup, key distribution, message blinding, signing and verification. Consider that there are authenticated classic channels and insecure quantum channels in the Alice, Bob and Charlie. Every pair of parties have the different quantum key matrices \(K_{AB}, K_{AC}\) and \(K_{BC}\) respectively is shown in figure (1). The key matrices \(K_{AB}, K_{AC}\) and \(K_{BC}\) are generated from the Simon et al.s QKD algorithm, which are of the form \(Q_n, p, r, m, o, T_n, p, n\).
3.2.1 Setup

From this method, consists of 3 participants: (1) the owner of the message, Alice transfer the message into an n-square matrix (n=2,3,) and blinds the matrix, (2) Bob the signer who signs blind messages, (3) Charlie the verifier who checks if a signature matches a message.

3.2.2 Key distribution

Alice and Charlie, Alice and Bob and Bob and Charlie have the pairwise quantum key matrices $K_{AB}, K_{AC}$ and $K_{BC}$ respectively. The QKD protocol are used by parties and generate their pairwise key matrices $K_{AB}^1, K_{AB}^2, K_{AB} = K_{AB}$ between Alice and Bob; $K_{AC}^1, K_{AC}^2, K_{AC} = K_{AC}$ between Alice and Charlie; and $K_{BC}^1, K_{BC}^2, K_{BC} = K_{BC}$ between Bob and Charlie as shown in table 2. The key generation order is determined by Alices, Bobs and Charlies quantum random generators.

3.2.3 Message blinding

Alice message is transformed into matrices $(M_1, M_2, ..., M_n) = M$ where $M_k = (m_{ij})_{m \times n}, K \epsilon (1, 2, ..., \alpha), t, j \epsilon (1, 2, ..., n)$. Then the key is blinds the message matrix M using the key $K_{AC}$ and obtains the blind message.
\[ M'_k = M_k \times K^{k}_{AG'}, k \epsilon (1, 2, ...a) \] ...............(3)

Then the blind message \( M' \) is encrypted in the Alice side with the key \( K_{AB} \) as follows:

\[ M''_k = M'_k \times K^{k}_{AB'}, k \epsilon (1, 2, ...a) \] ...............(4)

and \( M_k, M_k, \) and \( M'_k \). Finally, Alice sends \( (M'', det(M'_k)) \) to Bob, and \( det(M_k) \) to Charlie.

Table 2 An example for key distribution and digital signature

<table>
<thead>
<tr>
<th>( V_v )</th>
<th>( S_1 )</th>
<th>( S_2 )</th>
<th>( V'_v )</th>
<th>( R_k )</th>
<th>( M_k )</th>
<th>( M_k \times M_k' )</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>L</td>
<td>L</td>
<td>8</td>
<td>2</td>
<td>( \begin{pmatrix} 8 &amp; 13 \ 13 &amp; 21 \end{pmatrix} )</td>
<td>( \begin{pmatrix} 2 &amp; 1 \ 1 &amp; 1 \end{pmatrix} )</td>
</tr>
<tr>
<td>11</td>
<td>L</td>
<td>D_3</td>
<td>11</td>
<td>2</td>
<td>( \begin{pmatrix} 4 &amp; 2 \ 11 &amp; 21 \end{pmatrix} )</td>
<td>( \begin{pmatrix} 1 &amp; 0 \ 3 &amp; 6 \end{pmatrix} )</td>
</tr>
<tr>
<td>8,11</td>
<td>D_2</td>
<td>L</td>
<td>8,11</td>
<td>4</td>
<td>( \begin{pmatrix} 8 &amp; 13 &amp; 8 &amp; 13 \ 11 &amp; 19 &amp; 11 &amp; 18 \ 0 &amp; 0 &amp; 1 &amp; 0 \end{pmatrix} )</td>
<td>( \begin{pmatrix} 1 &amp; 2 &amp; 0 &amp; 7 \ 0 &amp; 3 &amp; 6 &amp; 0 \ 10 &amp; 4 &amp; 1 &amp; 1 \end{pmatrix} )</td>
</tr>
<tr>
<td>8</td>
<td>D_2</td>
<td>D_3</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

1. \( V_v \): the value for the entangled state.
2. \( S_1 \): the other participants sorter.
3. \( V'_v \): the value for the recovered entangled state.
4. \( R_k \): the rank of key matrix.
5. \( M_K \): the key matrix.
6. \( M_k \): the \( k^{th} \) message matrix.

### 3.2.4 Signing

Bob signs message \( M \) blindly by creating a signature for the message \( M' \) and this means that Bob does not know the contents of \( M \). The execution of the following steps to receive the message:

1. It checks the authenticity of \( (M''_k, det(M'_k)) \). First the message \( M''_k \) with the key \( K^{k}_{AB} \) and obtains

\[ M'_k = M''_k \times (K^{k}_{AB})^l - 1, \] ...............(5)

Where \( (K^{k}_{AB})^l - 1 \) denotes the inverse matrix of \( K^{k}_{AB} \). If the determinant of \( M_k \) recovered by Bob is not equal to the value...
of the determinant obtained from Alice, Bob aborts this communication. Otherwise, he performs the next step.

2. Bob signs the blind message \( M'_k \) using \( K_{BC}^k \). The signature is
\[
S^k = M'_k \times t_{BG}^k \quad \text{.................(6)}
\]

3. Sends the signature \( S = (S^1, S^2, ..., S) \) to Charlie.

### 3.2.5 Verification

Charlie verifies the signature provided by Bob and he uses the key \( K_{AC} \) and the determinant \( \det(M) \). It executes the following steps.

1. Having received the signature \( S \), Charlie decrypts it using \( K_{BC} \) and obtains the blind message \( M' \) with \( K_{AC} \) and obtains \( M \).

2. Charlie checks if the determinant of \( M \) recovered from the signature is the same as \( \det(M) \) obtained from Alice. If the check holds, he verifies the following equations:

\[
\det(S^k) = \det(M'_k K_{BG}^k) = \det(M'_k) \times \det(T^n_p) = (-1)^n det(M'_k) = (-1)^{2n} \det(M_k) \quad \text{.................(7)}
\]

If the verification holds as well, Charlie accepts \( S^k \). Otherwise, he aborts this communication.
4 Experimental Result

In this section, the evaluation of the proposed system in the cloud computing along with the exiting QKD method. The specification of the computer used to evaluate the performance, are intel i5 processor with 4GB DDR3 RAM of 500GB storage space and have the network of 5x Gigabit internet. The tool used measure the performance of this method in this paper are NetBeans.

<table>
<thead>
<tr>
<th>Method</th>
<th>Condition</th>
<th>Overhead delay</th>
<th>Handshake delay</th>
<th>Total delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>QKD</td>
<td>QKD 160 byte</td>
<td>240.4</td>
<td>13.7</td>
<td>254.1</td>
</tr>
<tr>
<td></td>
<td>QKD 16 byte</td>
<td>443.7</td>
<td>13.6</td>
<td>457.3</td>
</tr>
<tr>
<td></td>
<td>QKD 512 byte</td>
<td>1101.2</td>
<td>13.8</td>
<td>1115</td>
</tr>
<tr>
<td></td>
<td>RSA 1024 bit</td>
<td>7.1</td>
<td>2</td>
<td>9.1</td>
</tr>
<tr>
<td>QDSKM</td>
<td>QKD 160 byte</td>
<td>220.2</td>
<td>12.6</td>
<td>232.8</td>
</tr>
<tr>
<td></td>
<td>QKD 16 byte</td>
<td>404.5</td>
<td>12.7</td>
<td>417.2</td>
</tr>
<tr>
<td></td>
<td>QKD 512 byte</td>
<td>946.2</td>
<td>12.4</td>
<td>958.6</td>
</tr>
<tr>
<td></td>
<td>RSA 1024 bit</td>
<td>7</td>
<td>2</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 2. Comparison of different methods in delay.
There are three methods are compared to evaluate the performance of the QDSKM and the establishment of the delay. The two method used to compare the functions are: QKD and QDSKM. The QKD method uses the QKD-QKD-AES256-SHA cipher suite with 3 different size of the global key. The delay of the method is calculated and compared with each other as shown in table 2. This clearly shows that the QDSKM performs well when it is compared with QKD method. The delay of this method is less than other methods. In the quantum cryptography method, this QDSKM gives better result than the QKD. This clearly shows that this method can be applied to the cloud communication, which provides the secure communication.

The graph is plotted for the performance of the three methods, as represent in figure 3. This value is evaluated by using the different keys and delay is calculated. The delay of QKD is more than the QDSKM. The high delay is recorded in the QKD while generating the QKD512 byte as 1115 msec delay. There are two types of delay is calculating handshake delay and overhead delay.

Table 3. Time taken for accessing directory.
The performance of the QDSKM is calculated in the range of time taken for the methods for the directory access as shown in table 3. The evaluation is done on the different types of conditions and the calculated the performance. This clearly shows that the QDSKM requires the less time to access the directory.

<table>
<thead>
<tr>
<th>Condition</th>
<th>QKD 160 byte</th>
<th>QKD 16 byte</th>
<th>QKD 512 byte</th>
<th>RSA 1024 bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>QKD</td>
<td>2.2</td>
<td>2.3</td>
<td>2.3</td>
<td>2.2</td>
</tr>
<tr>
<td>QDSKM</td>
<td>2.1</td>
<td>2.1</td>
<td>2.2</td>
<td>2</td>
</tr>
</tbody>
</table>

The performance of the QDSKM is calculated in the range of time taken for the methods for the directory access as shown in table 3. The evaluation is done on the different types of conditions and the calculated the performance. This clearly shows that the QDSKM requires the less time to access the directory.

![Figure 4. Directory access for the different conditions.](image)

Figure 4. Directory access for the different conditions.

The performance of the system is evaluated for accessing the directory and shown in the figure 4. This shows the time taken for the two methods to access the directory and this method requires the less time for directory access. The QDSKM is implemented to access the directory with very less time.

Table 4. Time taken for the checking key amount delay (msec)

![Table 4](image)
From Table 4, time taken for the two methods to checking the key amount is shown. This is one of the processes to implement the encryption in cloud communication. These are evaluated for the different condition for the two techniques. This only requires the less time for the process and both methods are taken the less time to check the key amount. The QDSKM performs better than the QKD method.

Table 5. Time taken for the method selection

<table>
<thead>
<tr>
<th>Condition</th>
<th>QKD 160 byte</th>
<th>QKD 16 byte</th>
<th>QKD 512 byte</th>
<th>RSA 1024 bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>QKD</td>
<td>4.2</td>
<td>4.2</td>
<td>4.1</td>
<td>4.1</td>
</tr>
<tr>
<td>QDSKM</td>
<td>4.1</td>
<td>4</td>
<td>3.8</td>
<td>3.7</td>
</tr>
</tbody>
</table>

Table 6. Time taken for the key synchronization

<table>
<thead>
<tr>
<th>Condition</th>
<th>QKD 160 byte</th>
<th>QKD 16 byte</th>
<th>QKD 512 byte</th>
<th>RSA 1024 bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>QKD</td>
<td>233.2</td>
<td>436.5</td>
<td>1094</td>
<td>-</td>
</tr>
<tr>
<td>QDSKM</td>
<td>210.4</td>
<td>384.3</td>
<td>942</td>
<td>-</td>
</tr>
</tbody>
</table>
The performance of the two systems is measured for the method selection and key synchronization. The time taken for selecting the method is given in Table 5 and time taken for the key synchronization is given in Table 6.

### Table 7. Total Overhead delay

<table>
<thead>
<tr>
<th>Condition</th>
<th>QKD 160 byte</th>
<th>QKD 16 byte</th>
<th>QKD 512 byte</th>
<th>RSA 1024 bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>QKD</td>
<td>240.3</td>
<td>443.7</td>
<td>1101.1</td>
<td>7</td>
</tr>
<tr>
<td>QDSKM</td>
<td>217.3</td>
<td>391.1</td>
<td>948.7</td>
<td>6.4</td>
</tr>
</tbody>
</table>
The total overhead delay is given in the table 6 and this is the combination of the direct access, checking key amount, method selection and key synchronizing. This is taken for the 4 different condition and QDSKM provide low delay compared to the QKD. The QKD 512 byte have the more delay in the two process and QDSKM have 948.7 msec delay. The delay of the QKD 160 byte is 240.3 in QKD and 217.3 for the proposed method. The graphical representation of the total overhead delay is shown in figure 6. The experimental result clearly shows that the QDSKM provides better result compared to the QKD method. This method can be applied in the e-commerce application with more efficiency and with the secure transaction.

5 CONCLUSION

Quantum cryptography is one of important method for the secure communication and QKD is the most common technique is used in the cryptography. The quantum cryptography is method that is theoretically unbreakable and this is much needed in the secure communication. Quantum cryptography provides the better security in the cloud for the shared data and communication. The shared key is helps in sharing the information between the user without data leaking. In this method, the technique is used to provide the security in the communication from the QKD in the cloud.
The QKD is taken from the Simon et al. research and Fibonacci, Lucas and Fibonacci-Lucas matrices are used to improve the security. Charlies are used in this method, which checks the signature of the Bob and provides the information. This helps to provide the security in the cloud and this also improve the performance. It method can used in the e-commerce application like cloud storage, Mobile networks, online transaction etc. The result shows that it has low delay, while compared to the other method. The future work of this method is that it is some attacks has to applied to this method and check its security performance.

References


