Journal of Advanced Zoology

ISSN: 0253-7214
Volume ***44*** *Issue* ***S-5 Year 2023*** *Page* ***2347:2364***

A Study on "Security of Cyber-Physical Systems in the Cloud"

P. Ravindra[1*], S. Vasu[2], V. Suresh[3], K. Bangaru Lakshmi[4]

[1,2,3]K.B.N. College (Autonomous), Vijayawada-520001, Andhra Pradesh, India

**Corresponding author's: P. Ravindra*

Article History

Received: 06 June 2023
Revised: 05 Sept 2023
Accepted: 14 Nov 2023

Abstract

The existing security models are built with certain assumptions. The solutions like distributed accountability, provable data possession (PDP), Third Party Auditing (TPA) and so on are secure as long as the assumptions hold true. To ensure fool proof security for cloud storage security little research has been made on quantum key cryptography. Since the quantum key distribution is unconditionally secure, we propose a new scheme known as Cloud QKDP (Quantum Key Distribution Protocol for Cloud Computing) which exploits the benefits of quantum mechanisms to secure cloud storage and data dynamics. We consider a case study in which three parties such as cloud server, data owner and trusted client have provably secure communications with our proposed scheme which uses random oracle model. Our empirical study revealed mixture of success and failure rates with private and public clouds respectively.

Keywords: *Cyber physical systems, cloud computing, cryptography, quantum cryptography*

1. Introduction

Need for Quantum Cryptography

When compared to traditional cryptography, Quantum Key Distribution (QKD) has properties that can make is unconditionally secure. The former is based on computational complexity of mathematical problem while the latter is based on laws of quantum mechanics. Cryptanalysis has been around which paves the way for breaking security of public key cryptography due to the availability of quantum computers in future. It does mean that quantum computers provide sufficient power to break the computational complexity in the mathematical problem used by public key cryptography. Therefore, it is indispensable to use quantum key distribution along with best possible classical cryptographic primitives. As cloud users have concerns about outsourcing their data to remote cloud servers, cryptography plays a vital role in securing data transmission. Quantum cryptography when succeeded to be used for cloud storage and retrieval, it will be a paradigm shift in protecting data with unbreakable security.

Quantum Key Distribution Protocols

Transport Layer Security (TLS) and IPSec are widely used applications for Internet security. The TLS is based on Secure Sockets Layer for secure communication while the IPSec is a suite of protocols meant for ensuring that the communications over Internet Protocol (IP) are secure. According to Arkko and Nikander [2] the current policy mechanisms of IPSec are inadequate with respect to authorization. Oracle [3] states that the TLS has drawbacks such as inability to provide end-to-end solution. Mink, Frankel and Perlner [1] integrated QKD into the security applications such as TLS and IPSec using an additional support layer that helps in communication between QKD and those security applications. Authenticated Key Establishment (AKE) is the take pertaining to cryptography which is achieved by QKD. QKD has been proved to be secure against adversaries using future computational improvements. Mosca, Stebila and Ustaoglu [4] described BB84 QKD protocol which is then integrated with traditional AKE models. Their experiments proved that QKD can withstand future advances in computing arena. They used both classical cryptography and QKD and tested long-term and short-term security of BB84.

Shih, Lee and Hwang [5] proposed two three party QKD protocols and claimed that they were efficient. However, later, Gao et al. [6] proved that those QKDPs are susceptible to dense-coding attack. The problem with these protocols is that eavesdroppers can use entangled quits in order to obtain session keys without introducing errors in ongoing communications. Cotler and Shor [7] proposed a new QKDP that works faster than the existing such protocols. The protocol increases key generation rate by using a single photon's spatio-temporal modes effectively. Fiber optic and line of sight channels were used to demonstrate the proof of concept.

According to Zeng and Wang [8] improved QKD that can verify identity of communicator and distribute quantum secret key concurrently. However, their QKDP has a distinct problem such as common key reservation. Chuan et al. [9] proposed a new QKDP with pulsed homodyne detection that makes use of weak coherent states. This protocol was proved to be robust to attacks such as Trojan-horse and intercept-resend. Huang et al. [10] proposed and applied a novel QKDP to Wireless Sensor Network (WSN). It was an agent-oriented implementation of quantum communication for Wi-Fi network. With this the QKD could handle multiple users in the network. Brougham et al. [11] proposed a high dimensional QKDP that makes use of Franson interferometers. However, they concluded that usage of single Franson interferometer is not enough to have adequate security. Instead, multiple interferometers cloud be a better solution for high – dimensional QKD.

Lim et al. [12] proposed a new device independent quantum key distribution mechanism that is compatible with Bell's theory with respect to inequalities between two parties. Thus, they could overcome the problem of detection loophole attack. Dianati and All´eaume [13] described transport layer protocols used for QKD for the implementation of European project known as "Secure Communication Based on Quantum Cryptography".

Threats to Cloud Computing Security

According to Ted Samson the Cloud Security Alliance (CSA) identified nine top threats to cloud computing security. Data breaches are the first threat which causes a Virtual Machine (VM) to gain access to the cryptographic keys of another VM with ease. A single breach of security in one application can cause damage to all clients. Encryption can be used to avoid data breaches but when the cryptographic keys are compromised, the whole security is lost. Second threat is data loss which might be due to attacks launched by hackers to delete your data. In the process if the encryption keys are lost, it should be the worst case. Service traffic hijacking is the third security threat. When an adversary gains access to credentials, it could lead to hijacking of user's requests to illegal web sites that make use of the credentials. Insecure interfaces and API is the fourth threat for cloud security. The APIs that are vulnerable can expose applications to cloud security issues such as integrity, confidentiality, availability, and accountability. The fifth threat which is more frequent is denial of service attack which proves costly to cloud users as they are given services in pay per use fashion.

Malicious insiders are the sixth security issue that is difficult to address as the malicious insiders have legal access to data and services rendered. They can also misuse the keys stored in cloud storage. Cloud abuse is the seventh security problem that is practiced by hackers to break cloud security in order to launch various kinds of attacks such as sharing pirated software, propagating malware and so on. The eighth threat to cloud computing security is the lack of knowledge of cloud computing and security keys on the part of cloud users. Extensive knowledge when acquired can help cloud users to overcome this problem. Shared technology vulnerabilities are the very important threat to cloud security. When the vulnerabilities are shared, that causes havoc to the whole cloud computing phenomenon.

Secure Storage Solutions for Cloud

Cloud computing, a new model of computing, has become a reality which facilitates data owners to outsource their data to cloud besides providing various other services. However, the cloud servers are treated "untrusted" by cloud users as their valuable data is stored in remote servers. There are many security concerns over the outsourced data and communications between the cloud server and cloud users. Many solutions came into existence in order to curb this problem. Lin and Tzeng [14] proposed a threshold proxy re- encryption scheme that secures outsourced data. Their security architecture is facilitated by number of storage servers and key servers. The storage servers store data while the key servers act as access nodes. The

scheme supports encoding, encryption and forwarding. Each storage server and key server independently performs encoding and re-encryption and partial decryption respectively.

Provable Data Possession (PDP) is technique used to ensure integrity of outsourced data. Many PDP schemes came into existence such as PDP [15], SPDP [16], DPDP – I and DPDP – II [17], CPOR – I and CPOR – II [18]. These schemes tried to make the data provably secure. However, recently, Zhu et al. [19] presented a cooperative PDP scheme in a distributed and multi-cloud environment. The scheme is provably secure which is based on hash index hierarchy and verifiable response. The scheme is also efficient in terms of minimizing computational costs and communication overheads. Proof of data integrity is another scheme proposed by Kumar and Saxena [20] which provide data integrity proofs besides supporting Service Level Agreements (SLAs) that can have mutual agreements between the service provider and service consumer.

Wang et al. [21] focused on cloud storage security by implementing a security scheme known as "Third Party Auditing" which audits data for integrity verification. The scheme supports batch auditing besides supporting data dynamics which can't be done easily with cryptographic systems.

Sundareswaran, Squicciarini and Lin [22] proposed a decentralized information accountability framework for cloud storage security. They made use of JAR programmable features in order to encapsulate user's data and security policies in JAR files and that possess mechanisms for distributed accountability. In all the cloud computing solutions there was more importance to data integrity rather than providing end to end security.

Quantum Cryptography and BB84 Protocol

Quantum cryptography is based on quantum mechanics where the qubit used in key distribution cannot be altered without the possibility of making changes to the original state. In order to exchange a sequence of bits randomly two parties such as Alice and Bob make use of quantum channel to ensure security in communication using one-time pad. When any adversary such as Eve attempts to eavesdrop, detection of it is possible with high probability. The BB84 protocol supports quantum cryptography where quantum channel is used by two parties to send qubits. However, the classical channel which is also used by them is insecure. Quantum states can be represented using different polarizations. The BB84 protocol for secure communication between Alice and Bob works as described here.

1. The random sequence of bits sent Alice are encoded and sent to Bob.
2. Bob is supposed to receive photons and decode them randomly.
3. Both parties compare some bits that have same basis. In the process the test is considered successful if the estimated error rate is less.
4. At the end, Alice and Bob can obtained a secret key using other bits after subjecting them to privacy amplification and error correction.

The communication process with respect to secure key distribution using BB84 protocol is as presented in Table 4.1.

Table 4.1. BB84 protocol

Alice's String	1	1	0	1	0	0	1	0	1	1	1	1	0	0
Alice's basis	+	+	+	X	x	+	x	X	X	x	+	+	+	+
Alice sends	-	-	|	\	/	|	\	/	\	\	-	-	|	|
Bob's basis	+	x	+	+	x	+	x	+	X	x	+	+	+	+
Bob's string	1	0	0	1	0	0	1	1	1	1	1	1	0	0
Same basis?	Y	N	Y	N	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
Bits to keep	1		0		0	0	1		1	1	1	1	0	0

Test	Y		N		N	Y	N		N	N	N	Y	Y	N
Key			0		0		1		1	1	1			0

Design Of Secure Key Management Model For E-Governance

Conceptual Overview of the Proposed Model

The proposed secure key management model is a comprehensive solution to e- Governance in India. The proposal encompasses end to end security among different layers involved in the e-Governance applications. E-governance applications are highly sensitive and they are to be protected from unauthorized access and also from all kinds of adversaries. Towards this end, in this sub section, a conceptual overview is provided for the proposed model. There are many communication hurdles due to internal and external attacks in the real-world communication networks. Therefore, this proposal is aimed at providing a comprehensive model that can protect the interests of all stake holders of e-Governance. Secrecy and effective communication are given importance while designing the framework. Since the e-Governance applications involve many parties, they are to be protected under a secure domain. Towards this end, the proposed conceptual model is as shown in Figure 1. Various custom protocols were proposed to realize the model. For highly secure efficient key management, a technique is proposed that exploits quantum cryptography. Quantum device provided by "Quantum in the Cloud" [15], a quantum test bed, of University of Bristol is used for experiments.

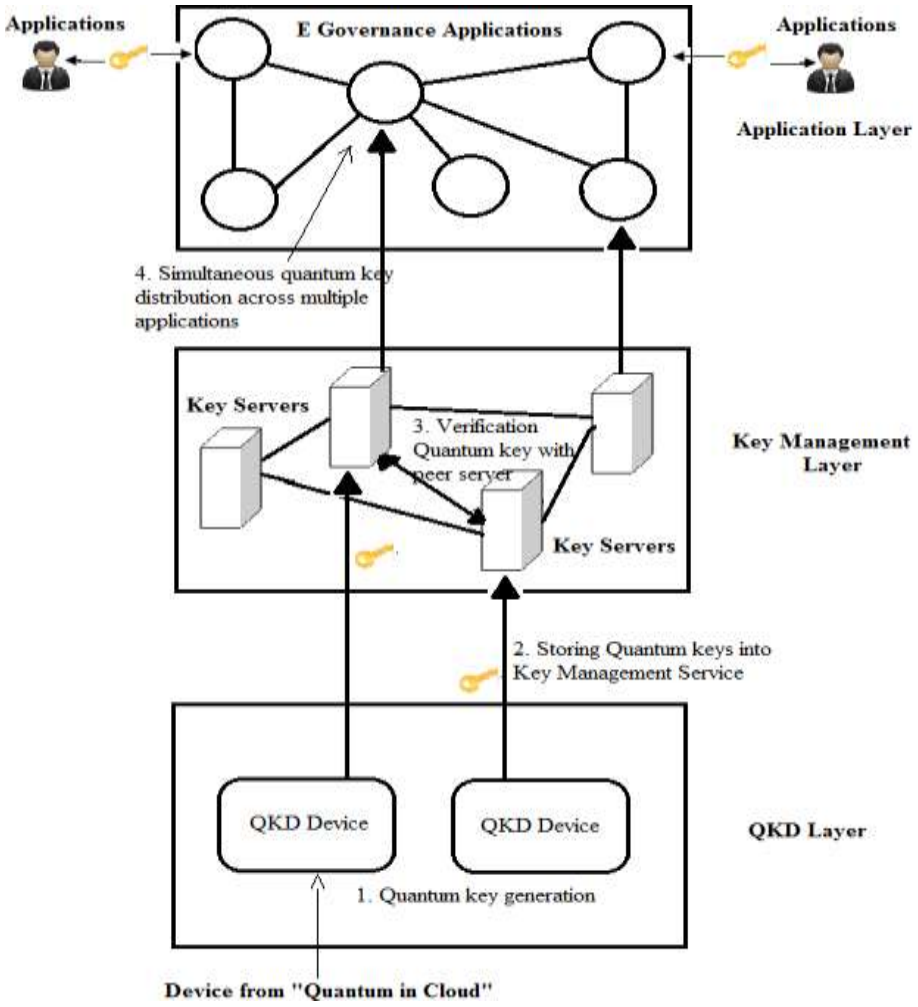


Figure 1.1 – Conceptual framework with quantum network for proposed e- Governance applications

As shown in Figure 1.1, there are three layers in the proposed framework namely Quantum Key Distribution (QKD) layer, key management layer and application layer. The QKD layer makes use of quantum device provided by "Quantum in Cloud" of University of Bristol. This layer is responsible to generate a shared random secret key that can be used by the parties involved. Pool of such keys is maintained by servers of the key management layer. Ultimately the application layer consumes the keys as and when required. Key management plays vital role in privacy and security of any communication network [12]. There is key

management interoperability protocol named Key Management Interoperability Protocol [13] which has important role to play in distributed applications for key management. This protocol was introduced in key management layer is Organization for the Advancement of Structured Information Standards (OASIS) [14]. The security mechanism is described here. First of all, the web interface provided by "Quantum in Cloud" device is used to generate quantum keys. The keys are then handed over to key management service which is crucial for secure communications. The quantum key verification is done among the peer servers that are part of key management service. Then simultaneous quantum key distribution is made across different e-Governance applications that ensure the underlying operations to be made in highly secure fashion.

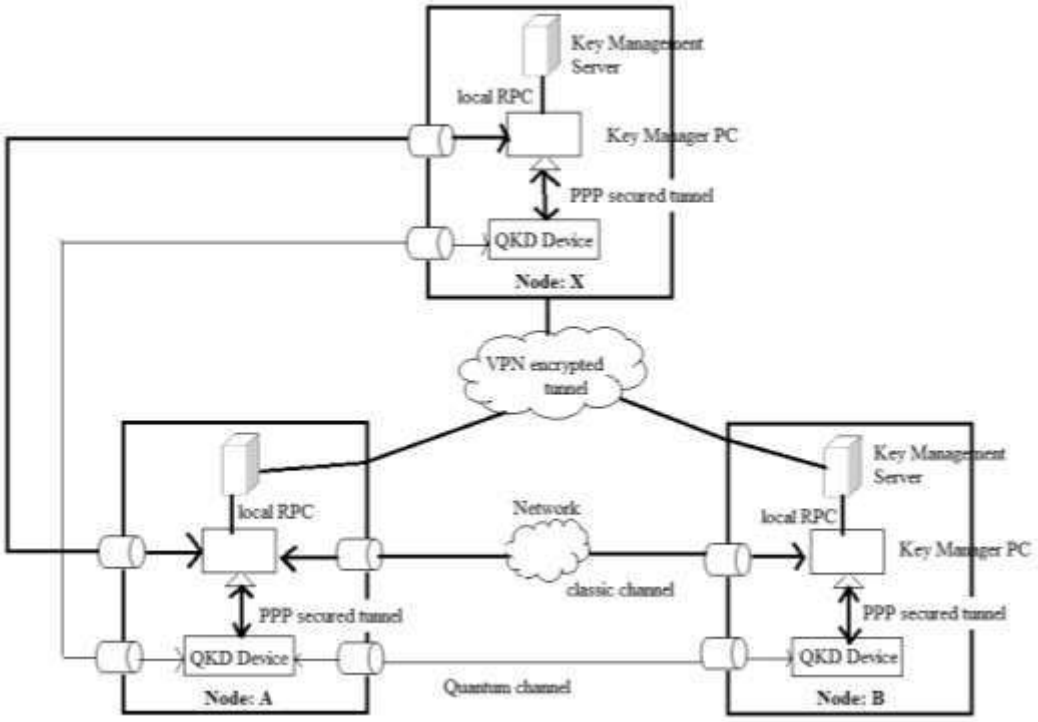


Figure 1.2 – Network infrastructure with quantum cryptography among different nodes

As shown in Figure 1.2, there are three nodes and two links that show effective and secure communication. The information passed through VPN is encrypted to avoid eavesdropping possibilities and other attacks. QKD devices form QKD layer that takes care of generation of random shared keys. The Key manager PC, key management server does have a Remote Procedure Call (RPC) done locally. Between node A and node B there are two channels established. The classic channel is meant for transferring data while the quantum channel is meant for sharing key in secure fashion. They key management server is responsible to manage keys and provide them access to a group of privileged users. There is user management service that takes care of privileges being assigned to genuine users and tracking them from time to time. Since there are different devices and communication requirements are involved, it is essential to have many customized protocols to realize the proposed framework.

Proposed QKDP

In our previous paper we implemented a protocol that helps secure communities in e- governance applications. In this paper a part of that protocol is reused in the framework we proposed for cloud. The proposed protocol is named QKDP. QKDP is the underlying protocol in the framework proposed in Figure 1.

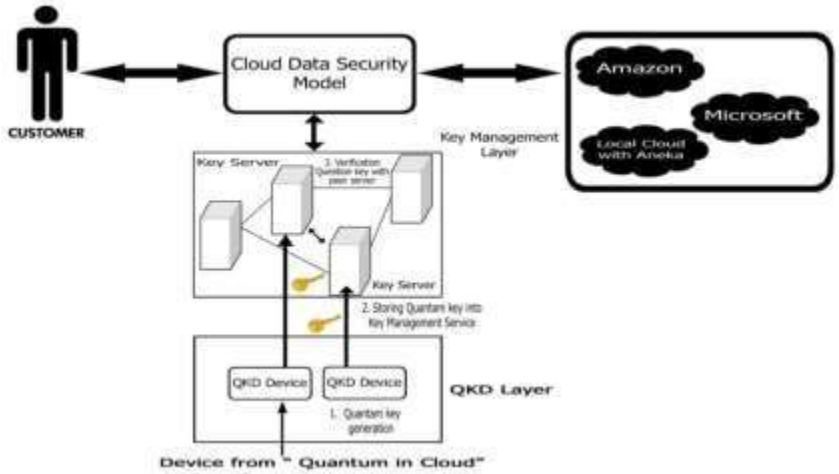


Figure 1.3 – Proposed framework for QKDP implementation

As can be seen in Figure 1.3, it is evident that the proposed framework has different layers. They are QKD layer, key management layer, cloud data security layer and cloud layer. The cloud layer is responsible to provide cloud services. The cloud data security layer is responsible to take care of encryption and decryption procedures using quantum and traditional cryptography. The traditional cryptography is for securing data while the quantum is to distribute keys in secure fashion. The QKD layer is responsible to produce quantum keys. We used devices for real quantum key generation using "Quantum in Cloud" platform. The generated keys are maintained by key servers which are located in key management layer. The Quantum Cloud infrastructure is depicted in Figure 2. The cloud infrastructure includes application server and license server in which the application server is connected to various nodes. Quantum device QD is installed in application server where as the key generation and key distribution process is managed by the license server. The Quantum key distribution is taken place through the quantum channel in the form of Qubit's and the shared key is distributed through the classical channel across the clouds.

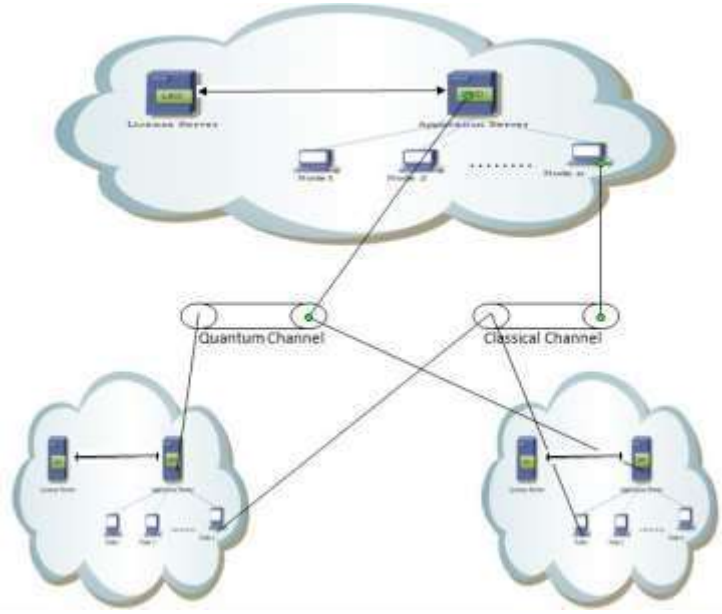


Figure 1.4: Quantum Cloud Infrastructure

The process is initiated at the cloud user end while sharing the document. The document is encrypted using 3-DES schema and transmitted through IP-Multicast using QKD phenomenon. The key transmitted through

the quantum channel where it is converted in to qubit and is transmitted based on various phases of polarization to the receiver's end.

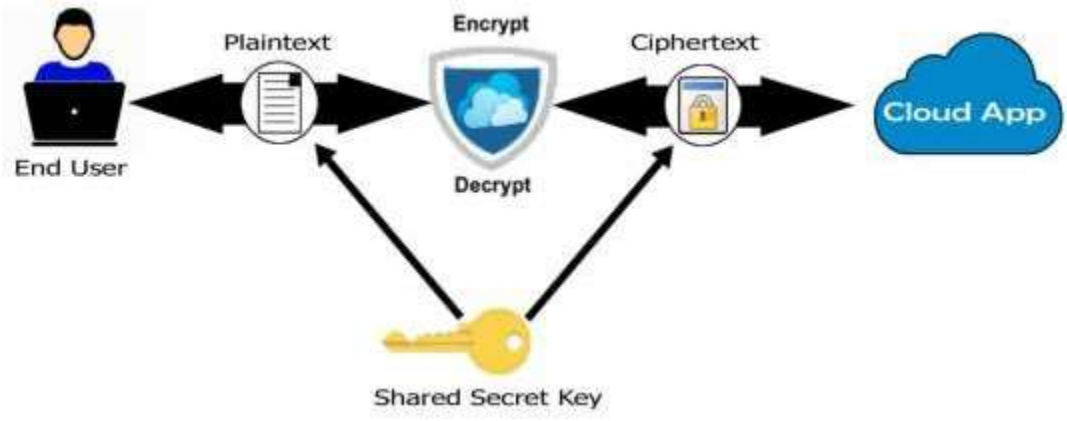


Figure 1.5 – Cloud data security model

This layer is in the proposed framework. It is elaborated here. It takes data from cloud user and encrypts it using Triple DES algorithm before sending it to cloud. In the same fashion, the data which comes from cloud is decrypted. However, in the proposed framework the key distribution is done using quantum channel for highly secure cloud communications. The process of communication within the Quantum channel is depicted in Figure 4.6.

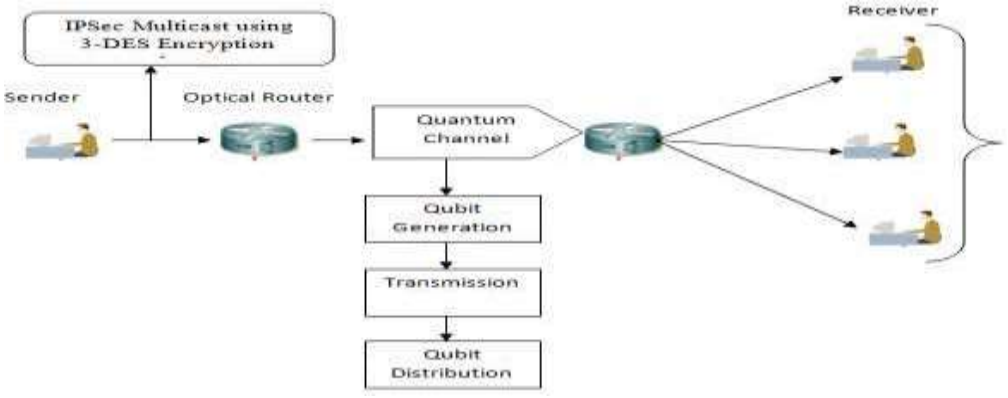


Figure 1.6: Quantum key distribution through quantum channel

The process of key distribution includes Qubit generation, Transmission of Qubit across the clouds and distributing it. The process of Qubit distribution is managed by Quantum key manager with the help of license server. Local host cache stores the generated qubits and they are transmitted across the cloud.

Notations Used

NOTATION	DESCRIPTION
application	A single e-Governance application
EGA	E-Governance Applications that work in distributed environment
Kb	Key block
kds	Key distribution service
kms	Key management server
Lcs	Local caching service
osb	Ordered secret bits
OSB	Pool of ordered secret bits
qcd	Quantum cloud device

qk	Quantum key
rpc	Report procedure call
timeout	Timeout value
Tt	Timeout threshold
Txid	Transaction id
VPN	Virtual Private Network
S	Secure key
Sh	Shared key
APS	Application server
LS	Licence server

QKDP Protocol

1 Initialize ***EGA***
2 APS starts qcd
3 ***qcd*** generates **qk** and communicates **to LS**
4 ∀ **qk in LS** ∃ **Qubit**
5 ***lcs*** stores ***Qubits***
6 ***lcs*** (***Qubits)*** ***s e c u r e c h a n n e l kms***

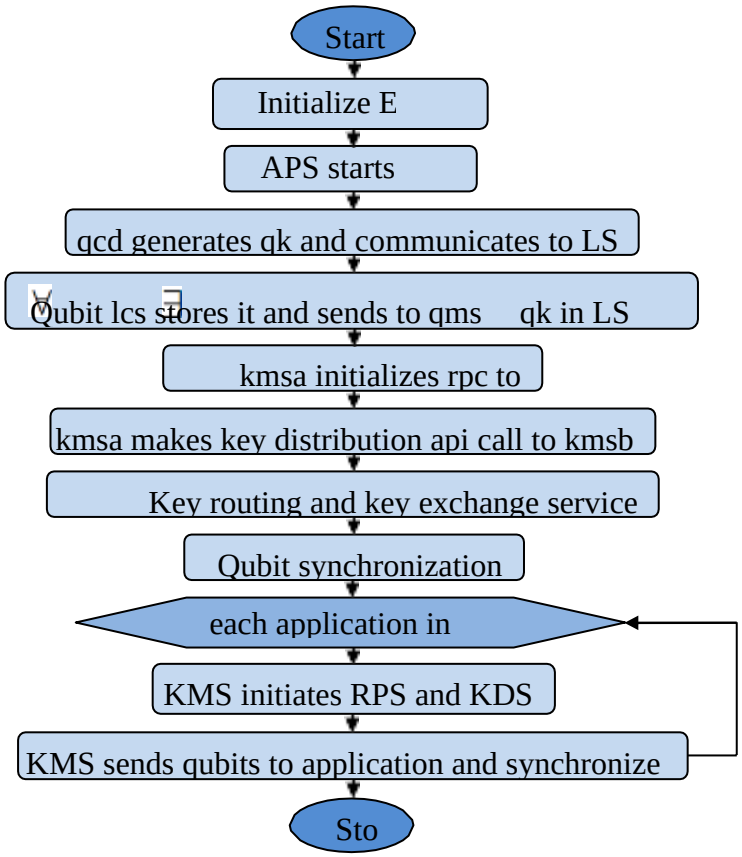
Managing quantum keys across peer key management servers

7 ***kmsa*** initializes ***rpc*** to ***kmsb***
8 ***kmsa*** makes key distribution api call to ***kmsb***
9 key routing initiated
10 key exchange service initiated 11 kmsb (***Qubit)V P N kmsb***
12 ***qubit*** synchronization

Simultaneous quantum key distribution to EGA

13 For each application EGA
14 14 ***kms*** initiates rpc
15 ***kms*** initiates ***kds***
16 ***kms*** (***qubit)***→ application
17 17 ***qubit*** synchronization

18 End For



List 1: Flow of CloudQKDP

Flow chart 1.1: Flow of Cloud QKDP

The purpose of this protocol is to have end-to-end security among cloud-based e- Governance applications that run in distributed environment. Quantum and traditional security are provided to communications appropriately. Quantum keys are generated by quantum devices provided "Quantum in Cloud". These keys are initially stored in local caching service. From caching service secure channel is used to send them to key management servers. The keys are shared among the key management servers available. Between two key servers, remote procedure call is initiated, key routing and key exchange services work as part of the protocol to complete key sharing successfully. Virtual Private Network (VPN) is established between servers to have sharing of quantum keys. Once exchange is carried out, the key synchronization is done to ensure consistency. They key managed by key management servers are given simultaneous access to e-Governance applications with appropriate synchronization. Such keys are used by the applications in order to have quantum keys to leverage the level of security of applications that involve in sensitive communications.

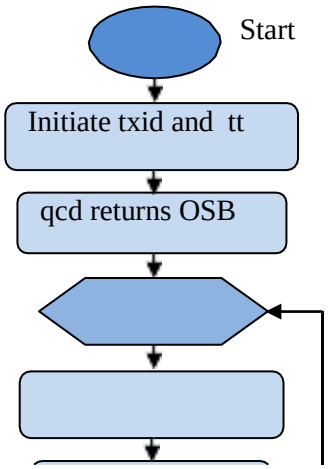
Key Caching Algorithm (KCA)

1 Initiate ***txid***
2 Initialize ***tt***
3 ***qcd*** returns ***OSB***
4 For (; ; kb++) ***{***
5 assign new ***txid***
6 track ***timeout***
7 [(***timeout***<= ***tt)→(kb*** ***e n c r y p t e d t r a n s f e r kms)]&[~*** (***timeout***<= ***tt)→(kb***

synchronization)]

8}

Algorithm1: Key Caching Algorithm (KCA)



Assign new txid

Flow chart 1.2: Key Caching

The key cashing algorithm is responsible to ensure that the quantum keys provided by quantum key distribution devices are cached and distributed to key management servers. When quantum key device returns pool of ordered secret bits (OSB), the caching algorithm is supposed to take the OSB and securely send to key management servers in timely fashion. There is timeout threshold that is employed to control the flow of OSB. For each key block, a transaction id is maintained in order to track secure and timely exchange of quantum key blocks to key management servers.

Secure Key Transfer Algorithm (SKT)

1 ***A*** (s) →***E***

2 ***A*** =E (***kb*** ,sh)

3 ***A(***E(***kb*** ,sh) → ***B***

4 ***B*** =D(kb, sh)

5 ***B*** =E(kb ,sh)

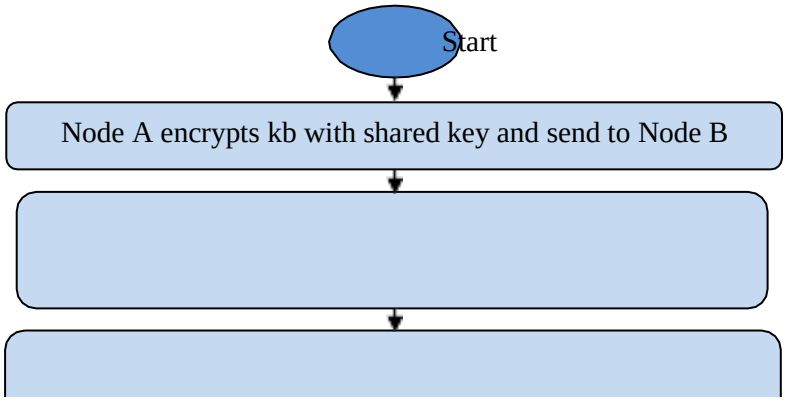
6 ***B*** (E(kb ,sh) → ***C***

7 ***C*** =D(kb, sh)

8 Repeat this process hop by hop

9 ***E*** ←D(kb***)***

Algorithm 2: Secure Key Transfer (SKT)



Node B decrypts with A Shared key and encrypts kb with

shared key and sends to Node C

Node C decrypts with B Shared key and encrypts kb with

Flow chart 1.3: Secure Key Transfer (SKT)

The key transfer algorithm is to have hop by hop process in order to transfer quantum keys securely. The keys are encrypted and transferred to destination node through intermediate nodes. End to end secure transfer is made at each hop until it reaches the destination where it is subjected to decryption and secure storage and usage. Algorithm 2 (SKT) shows that A □E the key blocks are transferred securely.

Modeling QKD system for Cloud

The design of the QKD system initially consists of two phases that includes initialize communication over Quantum channel and post dispensation over classical channel. The performance of the system is analyzed with the help of following parameters:

a. Secured key rate (Skr)

b. Qubit error rate (Qer)

The Secured key rate (Skr) is notated as **Skr = vBP.** **(1)**

Where 'v' is considered as pulses per second from the source and 'BP' is the bit rate per pulse.

Analysis to calculate the Qubit error rate

To evaluate the Qubit error rate initially the bit rate per pulse is calculated with the help of protocol inherent efficiency ℕi and βdas the mean detected signal per pulse with the help of the detectors at the Bob's end.

BP = ℕi βd (2)

Mean detected signal per pulse is calculated as follows

βd = β*signal* + β*dark* - β*signal* . β*dark* (3)

where β*signal* is the probability of photon emmitance of the Alice identified by Bob detectors and β*dark* is the probability of the false count in the signal. The overall probability of the false count of the photon signal for the experimental setup is given as:

β*dark* = ToD * Fd (4)

Here ToD is the total number of the detectors and 'Fd' is the probability of detecting false count per detector. Overall probability of bob receiving a photon is calculated as follows:

$$\beta signal = \sum_{k=0}^{\infty} \mathbb{N}i\ \beta(\mu, k) = \sum^{\infty} \frac{\mu k^{\mathbb{N}i}}{} e^{-\mu} \quad (5)$$

$k!$ $k=0$

In Practical QKD transmission, the ideal photons are replaced with weak coherent μ as shown above. It is assumed that the photons are the independent sources and it is substituted as follows

$$\beta signal = \sum_{k=0}^{\infty} \frac{\mu k \mathbb{N}i}{k!} e^{-\mu} = 1 - e^{-\mathbb{N}\mu} \quad (6)$$

Overall detection probability of Bob is given as follows:

$$\beta d = \beta signal + \beta dark = 1 - e^{-\mathbb{N}\mu} + \beta dark \quad (7)$$

βd and β*signal* contribute to Qubit error rate that is given as ratio of probability βerror is considered as the bit error δ

$$\delta = \beta \text{ fault rate} = \frac{1\beta dark + efaultrate.\ \beta signal}{\beta d} = \frac{4}{1 - e^{-\mathbb{N}\mu} + \beta dark} \quad (8)$$

In the above equation the term $^{1}\beta dark$specifies the random occurrence of the fault 4

counts of the photons and $efaultrate$represents the alignment of the experimental setup for the polarization of photons.

Protocol Evaluation

design of the protocol is based on the phenomenon of quantum entanglement, which plays a vital role in various applications Quantum Secret sharing (QSS), Quantum Key Distribution (QKD) and Quantum secure direct communication (QSDC).

We adopt the numeric searching program of Borras et. al. found the maximal BPB state that is represented as follows:

1 = [(000000 > +|000011 > +|111100 >

√32

+|000101 >+|111010 >+|000110 >+|111001 >+|001001 >

+|110110 >+|001111 >+|110000 > +|010001 >

+|111100 > +|101110>+ |010010>+ |101101>

+ |011000>+ |100111>+ |011101>+ |100010>)

− (|010100>+ |101011>+ |010111>+ |101000>

+ |011011>+ |100100>+ |001010>+ |110101>

+ |001100>+ |110011>+ |011110>+ |100001>)] 123456 ..... (1)

the above stated six qubit state is denoted as $\varphi 6qb$ from equation (1) we observe that $\varphi 6qb$

consists of 32 terms and each term have equal and even coefficient |0> .

To illustrate the entanglement property of $\varphi 6qb$ Equation (1) could be written as follows

1 [|000 > $|\alpha^1>$ + |001 > $|\alpha^1>$ + |010 > $|\alpha^1>$ + |011 > $|\alpha^1>$

$\varphi 6qb$ =

2 2 3 4

√8

− |100 > $|\alpha^1>$ − |001 > $|\alpha^1>$+|110 > $|\alpha^1>$ + |111 > $|\alpha^1>$]123456 .....(2)

5 6 7 8

where $\{|\alpha^1|i = 123 \ldots .8\}$ generally form an orthogonal basis.

The overall bell states $|\partial^{\pm}> and |\varphi^{\pm}>$ are represented in the form of

$|\partial^{+}>$ = 1 $|\varphi^{-}>$ = 1 (|00 > +|11 >) = 1

√2 √2 √2

$|\partial^{-}>$ = 1 (|00 > −|11 >) = 1

√2 √2

$|\varphi^{+}>$ = 1 (|01 > + |10 >) = 1

√2 √2

(|01 > −|10 >) = (| + +>
1 +|−−>).....(3)

√2

(| + −> +|− +>)
.....(4) (| + +>
−|−−>) … … … . (5) (| −
+> −|+ −>) .....(6)

the encoding of the bell states are represented as follows

$|\partial^{+}> \rightarrow 00$

$|\partial^{-}> \rightarrow 01$

$|\varphi^{+}> \rightarrow 10$

$|\varphi^{-}> \rightarrow 11$

The above equations allow us to design a multi-party quantum compromise protocol in which multiple participants can analyze the confidentiality level of their information.

3. Results and Discussion
Experimental Result

The Proposed framework is evaluated through the experimental analysis conducted by simulating 3DES and BB84 protocols while they are integrated with Quantum cloud architecture as shown in figure 2. Initially, the plain text in the uploaded file is encrypted by using 3DES algorithm. The sample plain text shown as follows:

Message Quantum Key Distribution is used for providing secure transmission. It enables two parties to produce a shared random bit string known only to them, which can be used as a key to encrypt and decrypt messages

Shared secret keys and the encrypted cipher text generated by 3DES are

DES Symmetric key1 = å€)XdÕ”C
DES Symmetric key2 = RÐÇì sþ%
DES Symmetric key3 = ÖÈ þÙzm

Shared Secret Key

oG\ØðÉ$#„šçÚùÐ3^1?ü"AÁÆ·O‰ Ô1ï'ž]±‰¥Áð **Encrypted message** ¾UQÚï;- ‰ ;¢ÿ¸0PìÒ>C@cSÂ¦Afµ•]0Ixpñ>d6åªÁE0Y;?q£n"í™~Ÿîiöq@F??Et,aC:> y”½-GŽ»= Pðx¼§R÷ðfWù%Kfª]É%¢‰6±¯D8–äÔ âÛ¬Åih*í]…=ïÄ¾iGw£øäkdVY- ª-hjh?gþ.˜J‹]où¾Ì?Xˆr®¥aßkôìØ‰+

Encrypted Cipher Text

Here, the shared key is transferred through the quantum channel, whereas the cipher text is transferred through the public channel. For this purpose the simulation ob BB84 protocol is adopted to generate qubits for the shared secret keys and transfer them to the users in the distributed scenario. The process in the quantum channel is illustrated as follows

The Alice sent the flow of content to bob is:

|||−>−>−>>||<−>|−−><<<>−<|||>||−>>−|><>||<<−−>>|>|−|−<|−|<|−|>−−|||−−−|><−<−>>||<>><
>−−<−|−>−−>|>−|>−<|−<−−>>|<>−>>>>>><−>|−>−>−−<<|>||>>|−−<<<<||>−<−−<>|−−|>|>
>|<−<|>|><|><|>|>−|>−|>><><|||−<><−|−>><>−>||−>−−−<|>|<>−><|<−−>−−−|<<−><<<|<>
>−>|>−>|><−>−−>|<>

The content sent by Alice is taken by Eve and decides randomly to take each photon in circular or rectangular measurements:

+O+++OOO++O++++OOO+O++OOOOO+OO++OOO+O+OOOOOOO+OOOO++++OOO
++O++O++++O++O++O++OOOO++O++++OO+O+O+OO++OO++OOOOOOOOOOO++
+OO+OOO++OO++OO++++O+++OO+OOOO++OO+OO+O++++O++O++O++O++OOO
++OOOO+++++O++++O+OO+O++O+O+O+O+OOO+++O++O++++OO++OO+O+O+++
+O+++O+OO++++++O+OO++OO+

The probability of each measurement is 0.1 in which the detector fails to detect the photons. Eve's measurement results are:

|<|− <>>||>||−|><><|−>><><|><|−>><|>−>><<< |><>>− − ><> |>|−<−|||<−−<|−<−−>><>−|>
−−<<|<|<−><>>−|<<<>>><> |−|><><|−><|−><||−|<|− ><|<<<<||<<−<<|<−−| >|−<−−<||<−
<><|−>>>− |−|<|||><>< −<|>|<|>>>−−−>−|<−−−|><− >>−>|<−−−|<−|−>|<> | ||−<− <−−><|

The content sent by Alice is taken by Bob and decides randomly to take each photon in circular or rectangular measurements:

+++OO+OO+OOO+++OO+O+++OOO++++O+++O+O+OO++OOO++OOO+OOO+O+OO
+OOO+O+O++O+++O++OOO++O++OOOO+O+++OOOO++++O+++OO++++O+OOO+
O++++O+++++OO+O+O+++O+O+O+OO+OOO++OO++O++OOOOO++++O+OO+OO++
+O+OOOO+OOO++O+++O+OOOO+O++++OO+++O+++++++++OOO++O+++O+OOO+
++++O++OOO+OO++++OOO+O

The probability of each measurement is 0.1 in which the detector fails to detect the photons. Eve's measurement results are:

||>>|>>>> |><|>| −>><−− −<|−|>−>− >−− <<−|><>| <>>|><><<−<|>||<−

|>|−>>>||>|−<><<|<|||<<><−−−−<||−<<−|||<<<<|>−|−|>|−−||>>|>|> ||>| |<<<−<> |−<<|−<

|>><> −−||<|> −><|−|>−><<<−<<<||>|−|>−<<>>− ||−>>|−−>−−− −−−|><<−|

−−|<−><>−−|−− ||><<| |−−−><>−>

The type of measurements successfully made by Bob through public channel is said to Alice as:

++OO+OO OO +OO+O+ +OOO++ +O+++O+O+ O++ OO++OOO+ OO O+OO OOO+O+O++O+ +O++OOO++O++OOOO+O+++OOOO++++O+++OO++++O OOO+O++++O+++++OO+O+O ++O+ +O OO+OO ++OO++O +OOOO ++++O+O

+OO+++O+OOOO+OOO++O+++O+OOOO+ +++OO+++O+++ ++++OOO++

+++O+OOO+++++ ++OOO+ ++++OOO+O

The correct measurements then said to Bob by Alice are:

.. ..

. . ..

The measurement made by Bob is also made by Alice in half of time and the probability the detector fails to detect each photon is 0.1.We expect remaining 115.6 shared bits out of 256.At this time 121 usable bits are generated.

To know any eavesdropping, Alice and Bob compare publicly 50% of shared digits. Due to random subset no eavesdropper can predict which digits are checked and to be avoided by messing.

Alice reveals 50% of shared digits by refining the previous answer:

|| >−> . −> . <|. |. .. >.| < . >>| − . .− . |− − <− >>.|< .. .− > .. ... − |. >> . ..|.

. ..> . .| .. .<| > − < . .>>. − .>|. |>.| .>− >><. . |. >> − . .. . − . − .−|< .. | − . . − >

.

Bob's corresponding check digits are said to Alice:

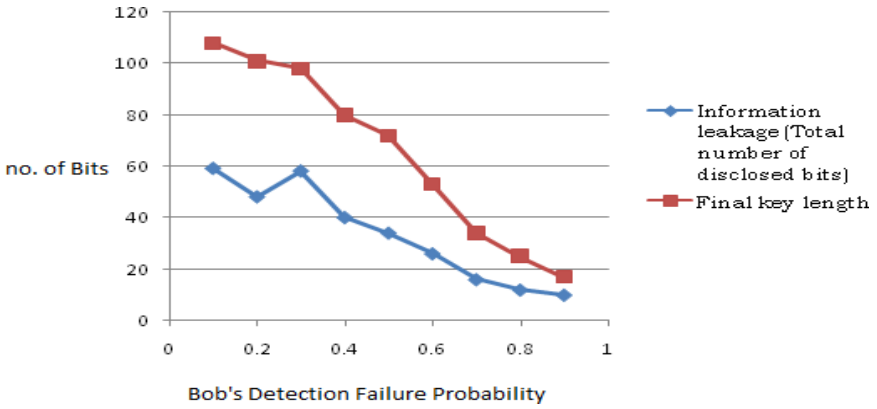
|| >|> . |> . <–. |. .. >.– < . >>| | . .– . || – >| >>.|> .. .| > .. ... – |. << . ..|. .
..> . .| .. .<– > – < . .><. – .<|. –>.| .>– <<<. . |. >> | . .. . – . – .–|< .. | | . . – < .

Alice and Bob got the confirmation that someone has listening to their exchange because 21 out of 61 check digits were gone wrong. This process is iterated with different bias values till

0.9 and the results are depicted as follows:

Table 1.2: Performance evaluation to detect information leakage

Initial number of qubits	BoB's basis selection bias	Information leakage (Total number of disclosed bits)	Final key length
256	0.1	59	108
256	0.2	48	101
256	0.3	58	98
256	0.4	40	80
256	0.5	34	72
256	0.6	26	53
256	0.7	16	34
256	0.8	12	25
256	0.9	10	17



The above graph depicts the detection failure probability at the receiver end and the required message is decrypted with optimum key length and provides high level confidentiality when compared with the existing and traditional cryptographic methods for data security.

4. Conclusion

This paper presents a number of theoretical solutions aimed at facing the challenges of the new cloud computation era. Addressing the problem of security and privacy in cloud environment two effective solutions were illustrated and it is observed through performance evaluation that the proposed solutions outperform various security algorithms previously proposed for securing cloud. Firstly the work demonstrates a working model to authenticate the users in cloud using quantum cryptography and further for the experimental analysis the BB84 protocol is simulated using QKD simulator that establishes a secured quantum channel. In the second contribution a prototype is modeled to ensure the secured data exchange between various clients and centralized cloud server for this purpose we integrated 3DES and BB84 protocols that enable multilevel security based on the proposed key management framework. Further this could be extended to minimize the computation process of cyber physical systems.

References:

1. Jian Wang, Quan Zhang, Chao-jing Tang. (2006). Quantum key distribution protocols using entangled state. *IEEE*. p1355-1358
2. T. Hwang and K.-C. Lee. (2007). EPR quantum key distribution protocols with potential 100% qubit efficiency. *IEEE*. 1 (1), p43-45.
3. Shirantha Wijesekera, Dr. Sajal Palit and Dr. Bala Balachandran . (2007). Software Development for B92 Quantum Key Distribution Communication Protocol. *IEEE*. p1-5.
4. Fábio A. Mendonça, Daniel B. de Brito, João B. R. Silva, George A. P. Thé, and Rubens V. Ramos. (2007). Experimental Implementation of B92 Quantum Key Distribution Protocol. *IEEE*. p712-717.
5. Q. Wang1,2W. Chen2G. Xavier1M. Swillo1, S. Sauge1, M. Tengner1, T. Zhang2, Z. F. Han2G.
6. C. Guo2, A. Karlsson1. (2008). Robust decoy-state quantum key distribution with heralded single photon source. *ieee*, 1-2.
7. Yang Han, Ming Gao, Ping-xing Chen, and Cheng-zu Li. (2008). Novel Robust Quantum Key Distribution Protocol with Symmetry. *IEEE*. p865-869.
8. Dazu Huang and Zhigang Chen. (2008). Quantum Key Distribution Based on Multi- qubit Hadamard Matrices. *IEEE*. p333-337.
9. Mohamed ElboukhariMostafa Azizi, Abdelmalek Azizi and. (2010). Quantum Key Distribution In Practice: The State Of Art. *IEEE*.p1-4.
10. Ivan Capraro and Tommoso Occhipitni. (2007). Implementation Of A Real Time High Level Protocol Software For Quantum Key Distribution. *IEEE*. p704-707.
11. Nabeel A. Siddiqui. (2007). QUANTUM-GRID INFINITESIMAL BIT CRYPTOSYSTEM. *ieee*, 1-4.
12. Mario Pivk,Christian Kollmitzer. (2009). SSL/TLS with Quantum Cryptography. *ieee*. 0 (0), 1-6.
13. Qing Xu, Manuel Sabban, Philippe Gallion and Francisco Mendieta. (2008). Quantum Key Distribution System using Dual-threshold Homodyne Detection. *IEEE*. p1-8.
14. Marko Hölbl, Tatjana Welzer. (2009). An Improved Authentication Protocol Based on One-Way Hash Functions and Diffie-Hellman Key Exchange. *ieee*,1-5.
15. David Elkouss, Jesus Martinez-Mateo, Alex Ciurana and Vicente Martin. (2013). Secure Optical Networks Based on Quantum Key Distribution and Weakly Trusted Repeaters. *IEEE*. 5 (4), p316-328.
16. Zhihao Liu1, Hanwu Chen1, Wenjie Liu1, 2 and Xiling Xue1. (2009). Mutually Authenticated Quantum Key Distribution Based on Entanglement Swapping. *ieee*, 1- 4.
17. WANG Yong WANG Huadeng, LI Zhaohong and HUANG Jinxiang. (2009). Man- in-the-middle Attack on BB84 Protocol and its Defence. *IEEE*. p438-439.
18. Mahboobeh Houshmand Monireh Houshmand, Habib Rajabi and Mashhadi. (2010). Game Theory based View to the Quantum Key Distribution BB84 Protocol. *IEEE*. p332-336.
19. Yuanyuan Zhou, Xuejun Zhou, Jun Gao. (2009). Scarani-Acin-Ribordy-Gisin Decoy-State Protocols in Quantum Key Distribution with a Heralded Single Photon Source. *IEEE*. p751-756.
20. L. Moli-S´anchez, A. Rodr´ıguez-Alonso, G. Seco-Granados. (2009). Performance Analysis of Quantum Cryptography Protocols in Optical Earth-Satellite and Intersatellite Links. *ieee*. 27 , 1-9.

21. Guoqing Cui, Yuan Lu, Guihua Zeng. (2009). A new scheme for Quantum Key Distribution in Free-space. *IEEE*, 1-4.
22. Lihua Liu,Zhengjun Cao. (2006). Improvement of One Password-Based Authenticated Key Exchange Protocol. *IEEE*, 1-4.
23. Mahboobeh Houshmand Monireh Houshmand, Habib Rajabi and Mashhadi. (2010). Game Theory based View to the Quantum Key Distribution BB84 Protocol. *IEEE*. p332-336.
24. Yuanyuan Zhou, Xuejun Zhou, Xiaoqiang Li. (2010). Performance of Scarani-Acin- Ribordy-Gisin Protocol in Quantum Key Distribution. *ieee*. 2 , 1-5.
25. Anand Sharma, VibhaOjha and Prof. S.K.Lenka. (2010). Security of Entanglement Based Version of BB84 protocol for Quantum Cryptography. *IEEE*. p615-619.
26. Vladimir Kurochkinl, Yury Kurochkin2. (2010). Quantum Cryptography Security Improvement with Additional States. *IEEE*, 1-3.
27. Xiaofang Xu and Xiaoyu Chen . (2010). Simulating B92 Protocol in Depolarizing Channel. *IEEE*. p1-3.
28. Masakazu YOSHIDA, Takayuki MIYADERA† and Hideki IMAI∗†. (2010). On the security of the quantum key distribution using the Mean King Problem. *ieee*, 1-5.
29. Mohamed Elboukhari1, Abdelmalek Azizi1,2,Mostafa Azizi. (2010). QUANTUM KEY DISTRIBUTION IN PRACTICE: THE STATE OF ART. *IEEE*, 1-4.
30. Patcharapong Treeviriyanupab , Paramin Sangwongngam , Keattisak Sripimanwat and Ornlarp Sangaroon . (2010). Performance of ½-Rate Convolutional Code on Winnow Protocol for Quantum Key Reconciliation.*ieee*, 1-4.
31. Yingkai Tang, Xiaoyu Chen. (2010). Simulating BB84 Protocol in Amplitude Damping Channel. *IEEE*, 1-4.