Eur. Phys. J. Special Topics

DOI: 10.1140/epjst/e2014-02118-x

THE EUROPEAN PHYSICAL JOURNAL SPECIAL TOPICS

Regular Article

Architecture of multicast centralized key management scheme using quantum key distribution and classical symmetric encryption

A.F. Metwaly[1,a], M.Z. Rashad[2], F.A. Omara[3], and A.A. Megahed[4]

[1] Information Technology Department, Al-Zahra College for Women, Oman
[2] Faculty of Computer and Information Sciences, Mansoura University, Egypt
[3] Faculty of Computer and Information Sciences, Cairo University, Egypt
[4] Faculty of Engineering, Cairo University, Egypt

Received 21 January 2014 / Received in final form 10 February 2014
Published online 19 March 2014

Abstract. Multicasting refers to the transmission of a message or information from one sender to multiple receivers simultaneously. Although encryption algorithms can be used to secure transmitted messages among group members, still there are many security aspects for designing a secured multicast cryptosystem. The most important aspects of Multicasting are key generation and management. The researchers have proposed several approaches for solving problems of multicast key distribution and management. In this paper, a secure key generation and distribution solution has been proposed for a single host sending to two or more (N) receivers using centralized Quantum Multicast Key Distribution Centre "QM_{KDC}" and classical symmetric encryption. The proposed scheme uses symmetric classical algorithms for encryption and decryption transmitted messages among multicast group members, but the generated keys which are used for authentication, encryption and decryption also play an important role for designing a secured multicast cryptosystem come from QKD protocols. Authentication verified using EPR entangled Photons and controlled-NOT gate. Multiple requests for initialization as well for transmitting sensitive information handled through priority and sensitivity levels. Multiple members' communication is achieved with full or partial support of QM_{KDC}.

1 Introduction

The computer networking cryptosystem consists of the encryption algorithm, decryption algorithm, and key management system [23]. On the other hand, multicast is the delivery of message or information to a group of users simultaneously using a single communication channel between source and destination users. Multicast has benefits in terms of bandwidth and optimized network performance compared to unicast transmission. Moreover, it is most commonly implemented in internet protocol (IP)

[a] e-mail: dr.ahmedfarouk85@yahoo.com

multicast, which is often employed in IP applications of streaming media, internet television, scheduled audio and video distribution, and file caching and distribution [1,4,25]. In order to securely transmit data packets in a multicast communication, messages are encrypted using a common cryptographic key for all valid members. This common key is often called by various names such as the session key, the traffic encrypting key, or the group key. Group key creation means that multiple communicators need to generate a shared secret to be used to exchange information securely. Group key management protocols can be almost organized into three classifications, specifically centralized, decentralized, and distributed [2,16–19]. Only group members who know the existing group key are capable of retrieving the original message using their own private keys.

Several approaches have been proposed to solve multicast key distribution problems [3,4,26–29]. Bennett and Brassard [5] introduced quantum key distribution (QKD) to solve key management problems. Quantum key distribution ensures the confidentiality and privacy of a cryptosystem by providing an unconditional security property. It provides this property by randomizing the preparation of bases based on quantum measurements to hide the message contents from intruders. "Randomizing" means that intruders try to guess how to measure the transmitted quantum state. If the measurement is guessed wrongly, the quantum state particles will be changed. In this case, two communicating parties will receive an alarm of intruder detection who listens to channel traffic [5,23]. The no-cloning theorem [6] forbids intruders or eavesdroppers to create identical replicas of unidentified random quantum state and forwards it without change as QKD, which has been invented and plays an important role in quantum cryptography and quantum private communication. Some QKD schemes and architectures have been practically implemented [7–14]. A simplified, general block diagram of a multicast QKD using a Quantum-Back-Bone Link Interface and Layered Architecture has been introduced in [24].

Currently, most of Centralized Group Key Management approaches use traditional Diffie-Hellman or Group Diffie-Hellman key distribution algorithms for generating and forming the group key. These approaches suffer from many serious problems as the computation and communication cost is very high due to many exponentiation operations as well authentication and confidentiality of transmitting messages between centralized server and multicast group members'. Other problems are performance bottleneck as Centralized Group Key receives requests from multiple members of the whole group concurrently which lead to transmission suspension or network breakdown, Centralized Group Key does not acknowledge the transmission and results in retransmission which consumes more energy and unreliable key distribution due to high packet drop ratio and man-in-middle attack [2,4,16,33,34].

In our proposed scheme, key generation and distribution for a multicast group implemented using quantum key distribution mechanism instead of Diffie-Hellman. With the purpose of protection against man-in-the-middle attack is developed as QM_{KDC} verifies member's identity using EPR entangled Photons and controlled-NOT gate. The confidentiality when transmitting a message contains quantum cryptographic Key among QM_{KDC} and communicating group members over a communication channel is achieved using quantum no-cloning and heisenberg uncertainty principle. Multiple members' communication is achieved with full or partial support of QM_{KDC}. Using full support of QM_{KDC}, as QM_{KDC} is responsible for decrypting received messages using received multicast group key and then decrypt it again with destined multicast group key. Using full support of QM_{KDC} as QM_{KDC} is responsible for creating a secured shared group key for communicated participants. In case of multiple initialization requests QM_{KDC} uses Dijkstra algorithm for computing distance and Min Heap Queue. As time sensitive multicast applications such as video conference, VOIP and On-line games become popular. When QM_{KDC} receives multiple transmission

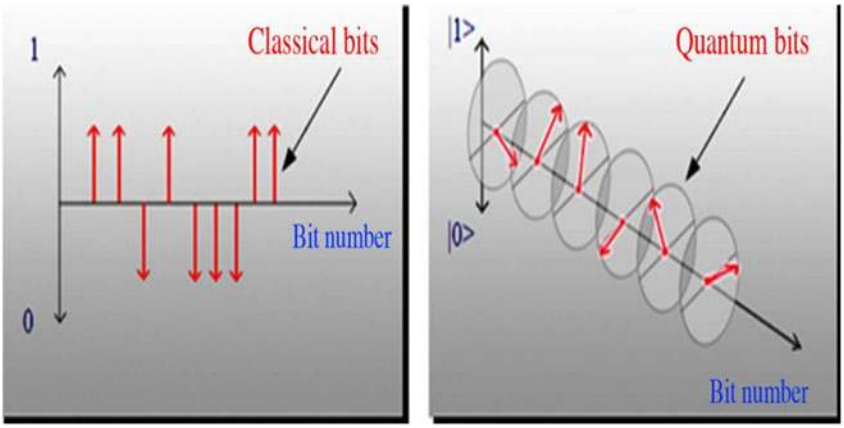

Fig. 1. Classical and quantum bits.

requests concurrently from different group members, different priorities will assigned based on sensitivity of transmitted information among multiple different members.

This paper is organized as; in Section 3.2, the definition of classical and quantum bit, symmetric and asymmetric cryptography, group key management categories and basic stages of quantum key distribution. In Section 3, our proposed scheme for designing a secured multicast cryptosystem using centralized quantum multicast key distribution centre and classical symmetric encryption is discussed in details with description of authentication, concurrent requests and synchronization. In Section 4 demonstrate security and performance analysis. Section 5 concludes the paper.

2 Preliminaries

2.1 Classical and quantum bits

In classical computer all information is expressed in terms of classical bit. Classical bit can be either 0 or 1 at any time. On the other hand quantum computer uses quantum bit rather than a bit. It can be in a state of 0 or 1, also there is usage of a form of linear combinations of state called superposition state. Quantum bit can take the properties of 0 and 1 simultaneously at any one moment as illustrated in Figure 1 [15].

Quantum bit definition is described as follow: Definition: A quantum bit, or qubit for short, is a 2 dimensional Hilbert space H_2. An orthonormal basis of H_2 is specified by $\{|0>, |1>\}$. The state of the qubit is an associated unit length vector in H_2. If a state is equal to a basis vector then we say it is a pure state. If a state is any other linear combination of the basis vectors we say it is a mixed state, or that the state is a superposition of $|0>$ and $|1>$ [30]. In general, the state of a quantum bit is described by (Eqs. (1), (2)) where $|\psi>$ is quantum state, α and β are complex numbers:

$$|\psi\rangle = \alpha|0\langle + \beta|1\rangle \tag{1}$$

$$|\alpha|^2 + |\beta|^2 = 1. \tag{2}$$

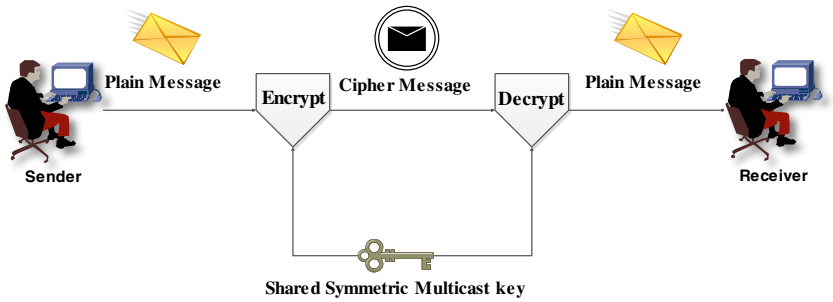

Fig. 2. Symmetric cryptography process.

2.2 Group key management categories

Group key management protocols can be almost organized into three classifications, specifically centralized, decentralized, and distributed.

In Centralized group key protocols a single member is responsible to manage the entire group and for re-keying, calculation and distributing group key to all group members. Centralized protocols are categorized into three approaches which are secure locks; Pairwise key and Hierarchy of keys [2,16–19,31].

In Distributed group key protocols, group members themselves participate to establish a group or session key. These members are similarly in charge of the re-keying and distribution of group keys. Distributed protocols are categorized into three approaches which are Ring based cooperation, Hierarchical based cooperation and Broadcast based cooperation [2,16–19,31].

In Decentralized group key protocols, the secured multicast group is divided into smaller groups or clusters; each sub-group is assigned by a local controller. Each local controller is accountable for security controlling of members and its subgroup. Decentralized protocols are categorized into two approaches which are distinguished as static and dynamic schemes [2,16–19].

2.3 Symmetric and asymmetric cryptography

Symmetric Cryptography is applied when the two communicating parties share a key before any encryption and decryption is done. The secret key should be circulated before transmission between communicated parties. The same key is take advantage of encryption and decryption data as shown in Figure 2. Symmetric key performance depends on size of used key. The longer used key, the harder to break. Most common symmetric cryptography algorithms are Data Encryption Standard (DES), Triple Data Encryption Standard (3DES), and Advanced Encryption Standard (AES) [20,21].

Asymmetric Cryptography is used when the two communicating parties use different keys for encryption and decryption as shown in Figure 3. Two separate keys are used; private and public keys. Public key is publicly available and used for encryption. Private Key is known only to the user and used for decryption. Neither key can perform both functions by itself. In contrast with symmetric key, there is no need for distributing keys before transmission between communicated parties. Most common asymmetric cryptography algorithms are Digital Signature Algorithm and RSA [20,21].

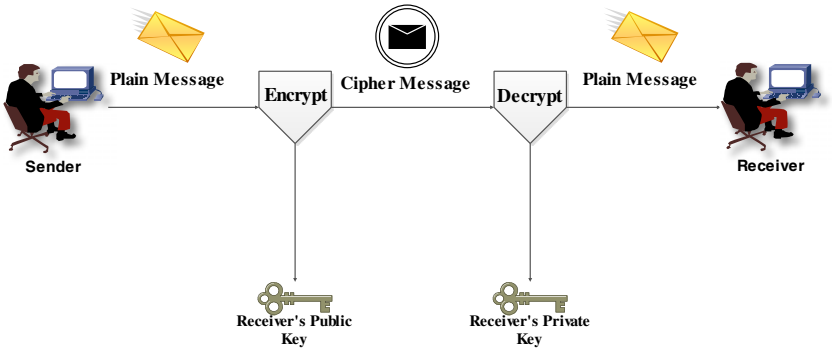

Fig. 3. A Symmetric cryptography process.

2.4 Basic stages of quantum key distribution

In general, a QKD scheme consists of three stages: quantum coding and transmission, raw key generation and eavesdropping detection [23]. The first stage is Quantum Coding and Transmission. It runs through a quantum channel. The Sender generates random bit string and encodes each with a quantum source. After encoding, qubits are transmitted actually from the sender to the receiver over a transmission channel. Clearly, a QKD system requires a transmission channel so that encoding qubits are transmitted from one communicator to another communicator through quantum carriers. Two popular types of transmission channels are optical fiber and open air often used for telecommunication networks and satellite communications respectively. The receiver generates measurements on received encoded qubits by selecting basis on his realized [21–23].

The second stage is Raw Key Generation. During transmission, communicators use different bases for measurements purpose. The objective of this step is to recognize and exclude those bit positions where communicators use different bases. These positions are then discarded by both communicators over a public channel [21–23].

The final stage is Eavesdropper Detection. During the transmission between communicators; eavesdropper might spy on quantum channel and retrieve potential secret key bits. Using quantum laws, eavesdropper operation on quantum channel can be detected. Eavesdropping is discovered as follows: An arbitrary subset of the raw key is agreed upon by communicators, and those bits are evaluated openly. If whichever two agreeing bits vary this specifies the existence of an eavesdropper and so communicators go back to stage1. Otherwise, the exchanged bits will be abandoned and the remainder of the raw key used as the final secret key [21–23].

3 The proposed secure key generation and distribution scheme

A cryptosystem is made up of encryption, decryption and key management. The confidentiality and privacy of cryptosystem are mainly based on key management and distribution. The basic idea of our proposed scheme is to use centralized QM_{KDC} and classical symmetric encryption to design a secure multicast cryptosystem. QM_{KDC} establishes preliminary connections between the sender and receivers in a multicast group. QM_{KDC} verifies member's identity using EPR entangled Photons and controlled-NOT gate. For each multicast group, QM_{KDC} generates two keys. The first key is group key and used for encrypting traffic between QM_{KDC} and a multicast group. The second key is shared symmetric key which shared among all members

Table 1. Abstract parameters and terminology used.

S_i^b	Random string of bits generated for member i
N	Total numbers of group members
m_i	A multicast group with ID i
$P_k(m_i)$	Private key of multicast group i
QM_{KDC}	Current centralized quantum multicast key distribution centre
P^e	Probability of key distribution error
Po	Agreed threshold of key distribution error
$M(G_k)$	group key Plain message
$C(G_k)$	group key Cipher message
$E(M)$	Encryption algorithm to encrypt plain message M
$D(G_M)$	Decryption algorithm to encrypt cipher message G_M
G_k	Group key generated by QM_{KDC}

in a multicast group as well used for encryption / decryption traffic within a multicast group members'.

If two members within same group needs to communicate, they communicate using group shared symmetric key. If two members in different groups need to communicate, they communicate using full or partial support of $\mathrm{QM_{KDC}}$. Using full support of QM_{KDC}, as QM_{KDC} is responsible for decrypting received messages using received multicast group key and then decrypt it again with destined multicast group key. Using full support of QM_{KDC}, as QM_{KDC} is responsible for creating a secured shared group key for communicated participants. In case of multiple initialization requests QM_{KDC} uses Dijkstra algorithm for computing distance and Min Heap Queue. As time sensitive multicast applications such as video conference, VOIP and On-line games become popular. When QM_{KDC} receives multiple transmission requests concurrently from different group members, different priorities will assigned based on sensitivity of transmitted information among multiple different members. Abstract parameters and terminology used for proposed scheme are defined in Table 1.

3.1 Initialization process

This is needed to establish preliminary connections between the sender and receivers in a multicast group. The initiator sends a request to a centralized quantum key distribution center with a list of members that will participate in the multicast group. The centralized quantum key distribution center broadcasts a request through the quantum network based on a fiber optic communication channel and sends a response back to the initiator. The initialization steps are illustrated in Figure 4 and are as follows:

(1) Initiator will send a request to QM_{KDC} with list of members in the multicast group.
(2) QM_{KDC} will send a broadcast message across the entire quantum network using fiber optic infrastructure, inviting the members on the initiator's list to join in.
(3) The members who wish to join will respond with ACK message to QM_{KDC}.
(4) Those who do not wish to join will respond with NACK message to QM_{KDC}.

3.2 Private key generation

For each multicast group, $\mathrm{QM_{KDC}}$ generates two keys. The first key is group key and used for encrypting traffic between $\mathrm{QM_{KDC}}$ and a multicast group. The second

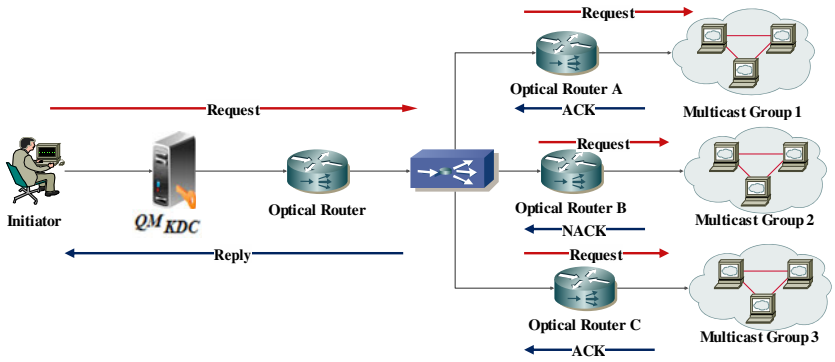

Fig. 4. Initialization process.

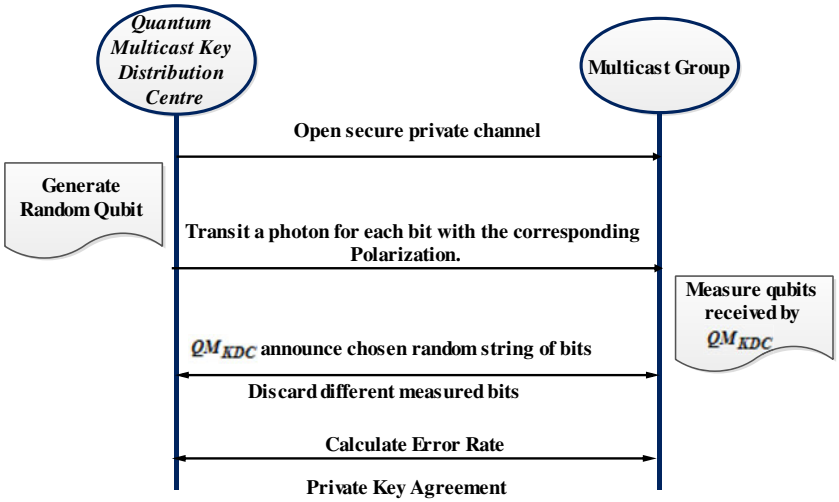

Fig. 5. Private key generation.

key is shared symmetric key which shared among all members in a multicast group as well used for encryption/decryption traffic within a multicast group members'. The private key is generated by choosing a random string of bits and transmitting it encoded with a series of polarized quantum-state photons. The multicast group measures each photon in one of numerous bases select at random. The measurements that have been made in a basis consistent with the initialization state of the photon are used for the key; different measured results will be discarded. The security and efficiency of the distributed key depend on eavesdroppers attempting to measure using the wrong basis. The quantum bit error rate is based on the ratio of errors within the distributed keys and is determined by comparing the error rate and a specified agreed-upon level. If the error rate is less than the agreed level, the communication process will continue; otherwise, the communication process terminates. A Figure 5 illustrates the flow of the process and the steps taken by QM_{KDC} and multicast group 1, respectively. The steps involved in the distribution of the private key are as follows:

(1) QM_{KDC} opens a secured private communication channel and broadcast initial configurations to the multicast group.

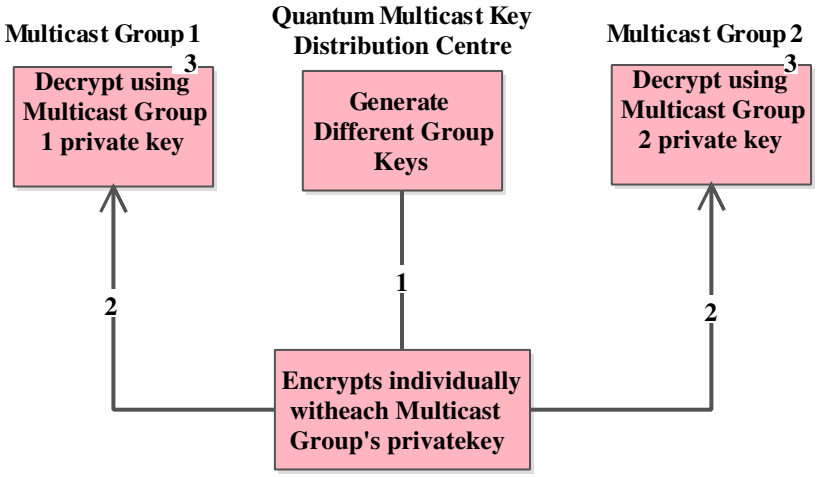

Fig. 6. Group key process.

(2) QM_{KDC} chooses a random string of bits (see Eq. (3)) to generate different private keys for the multicast group

$$S_b = \{S_i^b | i = 1, 2 \ldots \ldots N\} \; with \; S_i^b | \in \{0, 1\}. \tag{3}$$

(3) QM_{KDC} randomly chooses a basis to encode a string of bits for group and private keys. QM_{KDC} transmits a photon for each bit with the corresponding polarization.
(4) QM_{KDC} will then send the resulting qubits to the multicast group.
(5) After receiving the qubits, the multicast group measures them on the same basis as by QM_{KDC}. Then QM_{KDC} announces the random string of bits chosen, and the multicast group discards any different measured result.
(6) Eavesdroppers listening to the communication channel during the establishment process will be detected, as both QM_{KDC} and the multicast group have to publicly compare the randomly measured chosen string of bits to check error rate.
(7) If the error rate is less than the agreed threshold, the communication process will continue (see Eq. (4)). Otherwise, the processed protocol will be terminated.

$$P^e < p_o. \tag{4}$$

(8) Then, QM_{KDC} and the multicast group agree on shared secret key as the multicast group's private key (see Eq. (5))

$$P_k(m_i). \tag{5}$$

3.3 Group key distribution

The second sub-process is to distribute a group key to each multicast group. A group key is generated by randomly choosing a string of bits and encrypting it with each multicast group's private key. Encryption will be developed using classical symmetric algorithms. Each multicast group will retrieve a group key by decryption of the received message. The steps involving group key distribution for multicast group 1

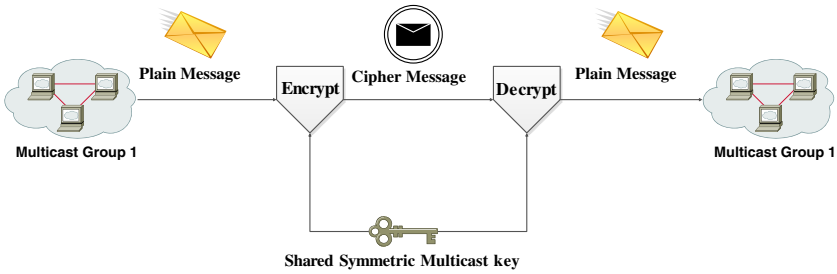

Fig. 7. Same multicast group members' communications.

and 2 are illustrated in Figure 6 and listed below.

(1) QM_{KDC} generates different group keys at random, one key for each multicast group and encrypts it with multicast group's private key. Encryption will be implemented using classical symmetric algorithm (see Eq. (6)):

$$C(G_k) = E(M(G_k),\ P_k(m_i)). \tag{6}$$

(2) QM_{KDC} sends the encrypted group key to each multicast group.
(3) Each multicast group retrieves group key by decrypting it using its own private key (see Eq. (7)):

$$M(G_k) = D(C(G_k),\ P_k(m_i)). \tag{7}$$

3.4 Same multicast group members' communications

If two members within same group needs to communicate, they communicate using group shared symmetric key. Sender encrypts message using group shared key. When receiver receives encrypted message, decrypts it using the same key. When member 1 in multicast group 1 needs to communicate with member 2 in same group, encryption/decryption process is achieved by multicast group 1 symmetric key. Member 1 encrypts message using multicast group 1 symmetric key. Message transmitted through fibre optic telecommunication infrastructure. Receiver decrypts it using same key. Now, receiver retrieves original message as illustrated in Figure 7.

3.5 Different multicast groups members' communication using partially QM_{KDC}

This process consists of the steps required if one member in a multicast group needs to connect with member in a different multicast group using partial operation of QM_{KDC}. In this process, QM_{KDC} is responsible for creating a secured shared group key for communicated participants. After, communicated parties use shared group key as shared symmetric encryption/decryption without help of QM_{KDC}. Figure 8 illustrates the flow processes to establish connection between Member 1 located in multicast group 1 and Member 2 in multicast group 2 and steps listed below.

(1) Initiated member sends the request along with its private key and the desired target member to QM_{KDC}.
(2) The group key, as well as configurations of the target member are generated by QM_{KDC} and forwarded to the initiated member.
(3) Initiated member retrieves group key by decrypting it using its own private key.

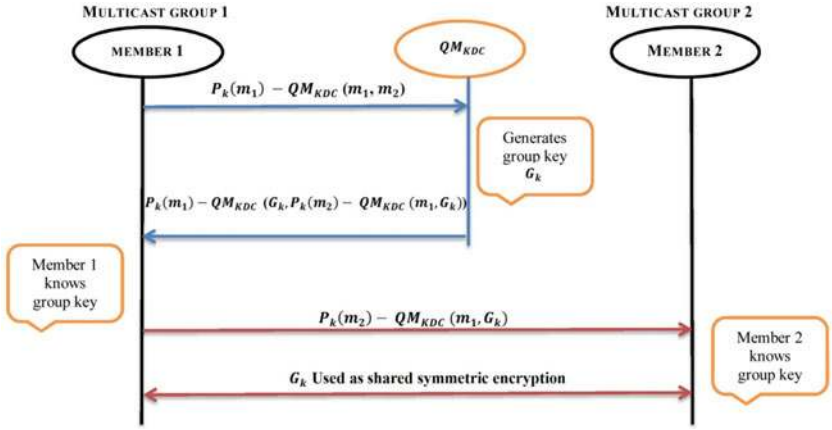

Fig. 8. Different multicast group members' communications with partially $\mathrm{QM_{KDC}}$.

Fig. 9. Different multicast group members' communications with fully $\mathrm{QM_{KDC}}$.

(4) Initiated member knows group key and will send the received configurations and group key to the desired target member.
(5) Target member retrieve group key by decrypting it using its own private key.
(6) Target member knows group key; initiated member and target member can now communicate using group key meant for shared symmetric encryption.

3.6 Different multicast groups members' communication using fully $\mathrm{QM_{KDC}}$

This process consists of the steps required if one member in a multicast group needs to connect with member in a different multicast group using full operation of QM_{KDC}. In this process, QM_{KDC} is responsible for decrypting received messages using received multicast group key and then decrypt it again with destined multicast group key as shown in Figure 9.

(1) Initiated member encrypts messages with its group key and sends it along with the desired target member to QM_{KDC}.
(2) QM_{KDC} retrieves original messages by decrypting it with initiated multicast group key.
(3) QM_{KDC} encrypts messages with destined multicast group key and forwards it.
(4) Target member decrypts messages with its group key.
(5) Now, Target member retrieve original messages sent by initiated member.

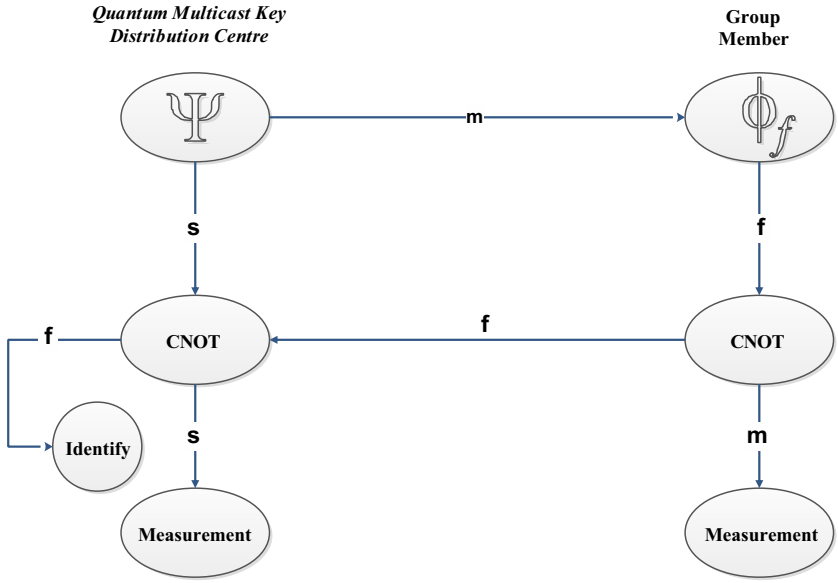

Fig. 10. Authentication process between QM_{KDC} and group members'.

3.7 Authentication process

QM_{KDC} authenticates and authorizes registered group members hence, they can communicate with other authenticated group members that are in the same group using the shared key that the group members have received from the key server or communicating with QM_{KDC}, in this case member's identity have to be verified. In this paper member's identity verified using EPR entangled Photons and controlled-NOT gate. Using the EPR method, Alice and Bob could potentially store the prepared entangled particles and then measure them and create the key just before they were going to use it, eliminating the problem of insecure storage as shown in Figure 10.

(1) QM_{KDC} and group member have a common key $K_C = \{K_1, K_2, \ldots\ldots\ldots K_N\}$ as authentication key.
(2) QM_{KDC} Prepares EPR pairs of polarized photons using two Particles s and m which related to QM_{KDC} and group member respectively as in Eq. (8). QM_{KDC} keeps s Particle at his side and send m Particle to the specified group member

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0_s 0_m\rangle + |1_s 1_m\rangle. \tag{8}$$

(3) When group member receives m particle, he encrypts it on photon through user's rectilinear or circular basis result a new particle f (information particle) as in Eq. (9) in the state

$$|\Phi_f\rangle = |\mathrm{K_{2i-1}} \oplus \mathrm{K_{2i}}\rangle. \tag{9}$$

Where $1 \leq i \leq N$ and $\oplus$ represent user's rectilinear or circular basis.
(4) Applying a quantum controlled-NOT gate on the m (group member Particle) and f (information particle) generates a tri-particle entanglement state as in Eq. (10) in the state

$$|\phi_e\rangle = \mathrm{C}_{NOT}(|\psi > \otimes |\Phi_f\rangle). \tag{10}$$

Table 2. Priority table.

Request Number	Multicast Group	Distance (QM_{KDC}, s)	Handling Seq.
1	1	5	2
2	2	4	1
3	3	7	3

(5) Group member maintains particle m and sends particle f to QM_{KDC}.
(6) QM_{KDC} decryption the state of particle f by performing controlled-NOT gate on the received particles s and $f$as in Eq. (11) in the state

$$|\phi_{\mathrm{e}}^{'}\rangle = \mathrm{C}_{NOT}\left(\phi_{\mathrm{e}}\right) = \mathrm{C}_{NOT}(|\psi\rangle \otimes |\Phi_f >). \tag{11}$$

(7) Start verifying group member identity so, QM_{KDC} measures particle f in the basis ∂_z by applying z state result can either 0 or 1. For an authentic user, the measurement must be $|\mathrm{K}_{2\mathrm{i}-1}, \mathrm{K}_{2\mathrm{i}}\rangle$. If the measurement result successfully for the first key, then key increased by one and recursively going back to step 1 until all keys processed. if all keys are authenticated then users identity is correct and communication process will going on else communication process will be aborted.

3.8 Multiple requests handling

In this point mechanism about how QM_{KDC} handle received multiple initialization requests simultaneously is proposed. Also how QM_{KDC} handles multiple received transmitting messages from different members.

3.8.1 Multiple initialization requests

In this case QM_{KDC} receives multiple initialization requests simultaneously from different members to forming a multicast group. If there are multiple initiators, then the distance is computed between QM_{KDC} and multiple initiators using Dijkstra algorithm. QM_{KDC} assigns priorities for multiple initiators according to result the nearer, the higher. After, QM_{KDC} creates min heap queue, so initiator with minimum distance will handle first. If two initiators have same distance, first come is first handle. As demonstrated in Figure 11, QM_{KDC} receive three different initialization requests concurrently, distance between QM_{KDC} and each inititor is calculated. In proportion to distance QM_{KDC} creates handling table as point out in Table 2. Table 2 states request number, initiator multicast group, computed distance and handling sequence. Handling sequence is processed as the nearer, the first. In case of distance equally, first come is first handled. So request from multicast group 2 will be handled first as its distance is the less, then request of multicast group 2 and finally multicast group 3.

3.8.2 Multiple transmission requests

As time sensitive multicast applications such as video conference, VOIP and On-line games become popular. In our proposed scheme, different priorities will assigned based on sensitivity of transmitted information among multiple different members. Sensitivity level is recognized into 4 levels. 00 is for None, 01 for Low, 10 for Moderate and 11 for High. When QM_{KDC} receives multiple transmission requests

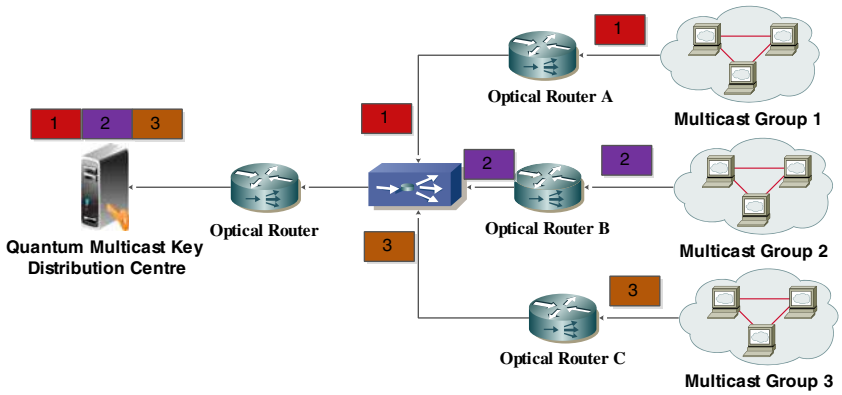

Fig. 11. Multiple initialization requests handling.

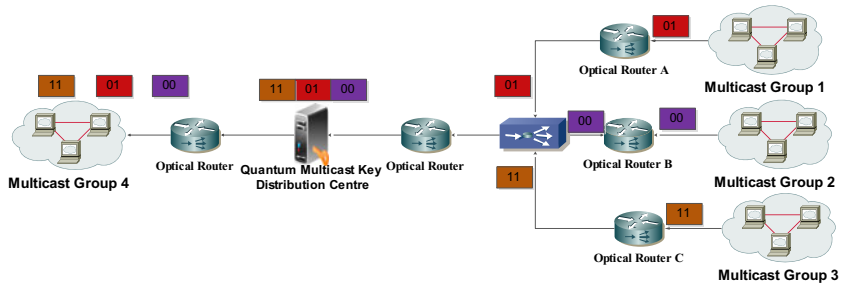

Fig. 12. Multiple transmission requests.

Table 3. Sensitivity level.

Priority Bits		Sensitivity Level
0	0	None
0	1	Low
1	0	Moderate
1	1	High

concurrently from different group members, QM_{KDC} extracts priority bits from each packet. Based on priority bits, QM_{KDC} will assign priority level. If two requests have same priority, then first come first served. As showing in Figure 12, QM_{KDC} receives three different multicast group transmitting messages to multicast group 4. QM_{KDC} extracts two- priority bits from each received packet header. According to priority table as illustrated in Table 3, QM_{KDC} determines the sensitivity level of transmitted messages. So firstly QM_{KDC} transmits messages from multicast group 3 as priority bits is 11 then multicast group 1 with 01 and lastly multicast group 2 with 00.

4 Analysis and demonstrations

4.1 Authentication

Obviously Using Diffie-Hellman key exchange doesn't authenticate either participant engaged in the exchange, so it vulnerable to man-in-the-middle attack. In our proposed scheme as demonstrated in Figure 13, with the purpose of protection against

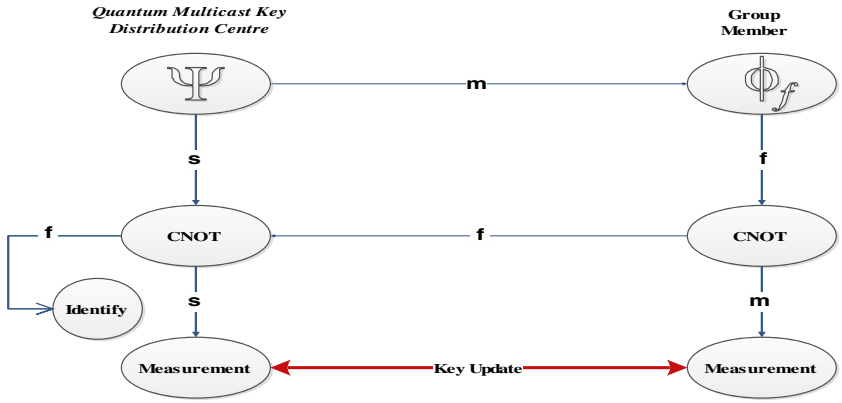

Fig. 13. Authentication security analysis.

man-in-the-middle attack is developed as QM_{KDC} verifies member's identity using EPR entangled Photons and controlled-NOT gate. After the identity authentication between QM_{KDC} and group member, $\mathrm{K_{2i-1}}, \mathrm{K_{2i}}$ are kept secretly in a maximally entangled state Ψ. QM_{KDC} and group member have to update authentication key $\mathrm{K'_{2i-1}}, \mathrm{K'_{2i}}$. The first key bit $\mathrm{K'_{2i-1}}$ can be achieved by measuring the state Ψ. The second key bit $\mathrm{K'_{2i}}$ can be achieved by first two bits of the previous key and $\mathrm{K'_{2i-1}}$. So Authentication is secured with unconditional property as in Eq. (12) in the state.

$$\mathrm{K'_{2i}} = \mathrm{K_{2i-1}} \oplus \mathrm{K_{2i}} \oplus \mathrm{K'_{2i-1}}. \tag{12}$$

4.2 Confidentiality

The confidentiality when transmitting a message contains quantum cryptographic Key among $\mathrm{QM_{KDC}}$ and communicating group members over a communication channel is achieved using quantum no-cloning and Heisenberg uncertainty principle. So the eavesdropper or even the intruder cannot achieve valid information or understand contents of transmitted keys. Instinctively, when $\mathrm{QM_{KDC}}$ sends a qubit $|0\rangle$ or $|1\rangle$, an eavesdropper measures it by applying operation E1 and E2 as demonstrated in Figure 14. Since eavesdropper has no any information about transmitted quantum cryptographic key, measuring output result will be either $|+\rangle$ or $|-\rangle$, so the eavesdropper operation intercept the quantum channel and create a result with erroneous probability 50%. After group member perform measurement on all received qubits which constitute quantum cryptographic key, in accordance with Heisenberg uncertainty principle an erroneous probability 25%. $\mathrm{QM_{KDC}}$ and group member detect eavesdropper if the erroneous probability more than the agreed threshold. If error rate is lower than agreed threshold then communication process will continue else processed protocol will be aborted by substitution in Eq. (4) then $25\% < p_o$.

4.3 Secure and sifted keys rate

$\mathrm{QM_{KDC}}$ determines secure key rate using Koashi's approach [35] afterward estimating parameters of single photon from decoy state's according to Rice and Harrington approach [36]. The secure key rate of $\mathrm{QM_{KDC}}$ and a multicast group is given by

$$R(QM_{KDC}) = [Q_1(1 - H(e_1) - Q_{FEC(e)}H(\mathrm{e}) + Q_0]/t. \tag{13}$$

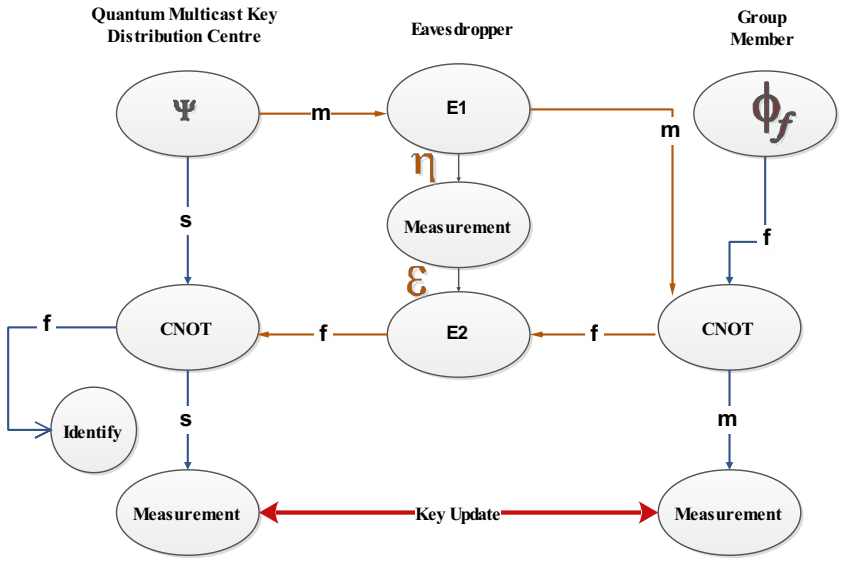

Fig. 14. Confidentiality security analysis.

Where Q_1 is the approximate number of sifted bits from single photon states of QM_{KDC} to a multicast group members', e_1 is the approximate number of errors of single photon states, Q is the total number of sifted bits between QM_{KDC} and a multicast group, FEC is error correction efficiency, e the sifted bits of QBER, Q_0 is the approximate number of sifted bits from 0-photon pulses, H(e) is the binary entropy function and t duration of session between QM_{KDC} and a multicast group. Fibre optic cable is shared concurrently among QM_{KDC} and group members with optical clock synchronization. Optical clock synchronization is essential for enhancement quantum states transmission, as the multicast group will not detect the quantum state properly without synchronization of the photon arrival. The data channels communicated between QM_{KDC} and group members are multiplexed using wavelength division multiplexers. With the purpose of improving operation between QM_{KDC} and a multicast group against quantum signal noise and distribution, the wavelength used is the shorter one with 1550 nm wavelength, additionally use of SSPD to improve stability. For eliminating optical modulators' from received group members Bloch sphere polar states' $|0\rangle$, $|0\rangle - i|1\rangle$, $|0\rangle + i|1\rangle$, and $|1\rangle$, are used as showing in Figure 15.

Figure 16 outlines $\mathrm{QM_{KDC}}$ secure key rate and sifted key measured in Kbits/s as a relation of fibre distance in km. The generated secure keys decreased as long fibre cable distance increased. The secured key rate over 35 km and 80 km are 993 and 82 Kbits/s respectively. According to fibre optics characteristics', the generated sifted key rate reduced as fibre distance increased. The sifted key rate over 35 km and 80 km are 1395.64 and 121.36 Kbits/s respectively.

Figure 17 outlines Quantum Bit Error Rate "QBER" as a function of fibre optic length. For fibre optic distance 50 km is showing 6.2% and 32.86 Kbits/s as QBER and number of errors respectively. For fibre optic distance 35 km is showing 3% and 28.24 Kbits/s as QBER and number of errors respectively.

Figure 18 outlines Quantum Bit Error Rate "QBER" and number of errors as a function of secured key rate. For secured key rate equal to 1500 Kbits/s is showing 1.4% and 21 Kbits/s as QBER and number of errors respectively. For secured key rate equal to 600 Kbits/s is showing 4.2% and 25.28 Kbits/s as QBER and number of errors respectively.

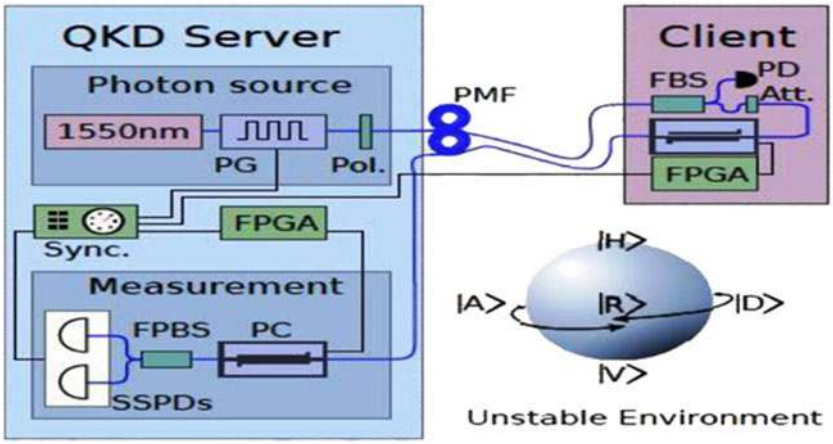

Fig. 15. Simulated connections between QM_{KDC} and group members'.

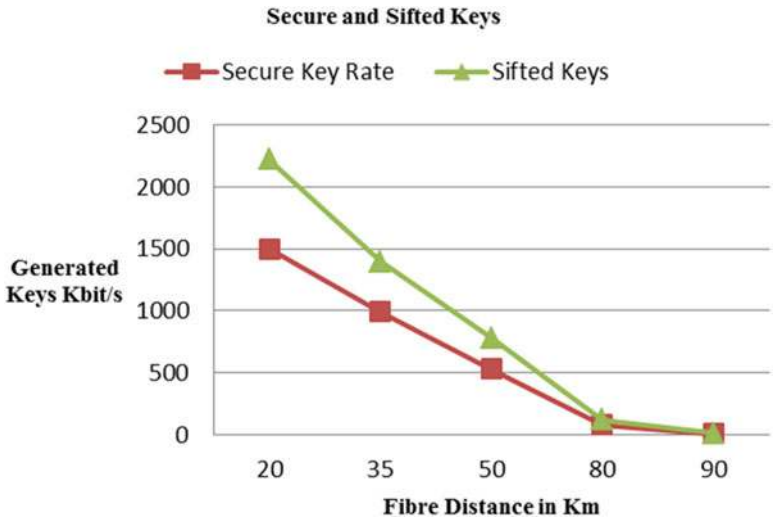

Fig. 16. QM_{KDC} secure and sifted keys as relation of fibre distance.

5 Conclusions

The Methods of key distribution and management which are widely used today to build internet security architecture may be penetrated at some points. Key distribution penetration leads to loss of capability to communicate securely within multicast network. Classical secret key approaches have suffered from different problems, specifically, insider hazards and the logistical difficulty of distributing keying information. We have presented model to secure key generation and distribution in one-to-many network using hybrid quantum key distribution properties and classical encryption algorithm. Our proposed scheme offers significant advantages in protection of keys and authentication using quantum physics laws. Authentication when QM_{KDC} delivering secret keys to multicast group members' achieved using EPR entangled Photons and controlled-NOT gate. The confidentiality when transmitting a message contains quantum cryptographic Key among QM_{KDC} and communicating group members over a communication channel is achieved using quantum no-cloning and Heisenberg uncertainty principle. Our future work will be focused on how to use quantum key

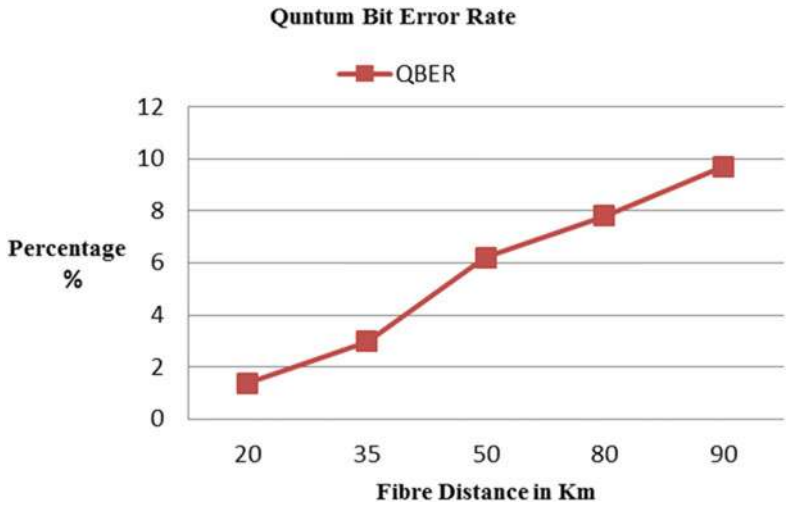

Fig. 17. QM_{KDC} quantum bit error rate percentage as relation of fibre distance.

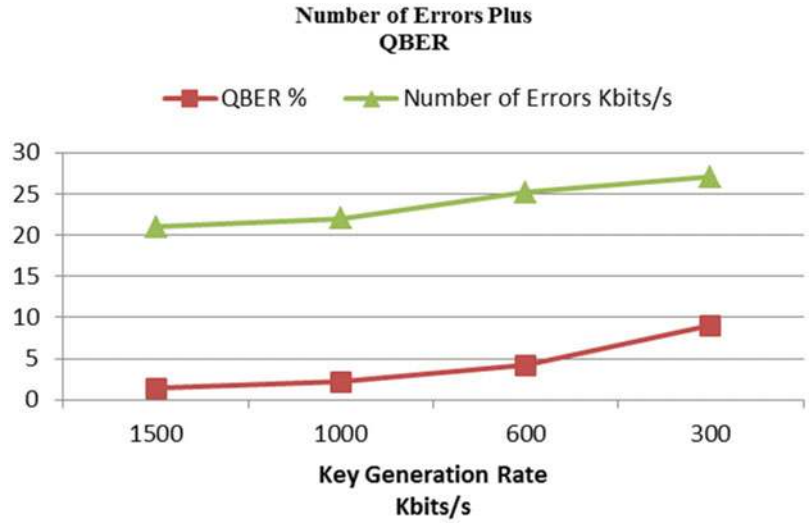

Fig. 18. QM_{KDC} quantum bit error rate percentage and number of errors as relation of key generation rate.

distribution and hashing-based authentication techniques for multicast network, as well, enhancement of multicast security will implemented by integrating quantum key distribution with IPsec and VPN.

References

1. S. Deering, Ph.D. thesis, Stanford University, 1991
2. S. Rafaeli, D. Hutchison, ACM Comput. Surv. **35**, 309 (2003)
3. D. Wallner, E. Harder, R. Agee, Internet Draft, RFC 2627, June (1999)
4. C.K. Wong, M. Gouda, S.S. Lam, IEEE/ACM Trans. Network. **8**, 16 (2000)
5. C.H. Bennett, G. Brassard, *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing* (Bangalore, India, 1984), p. 175

6. W.K. Wooters, W.H. Zurek, Nature **299**, 802 (1982)
7. C.H. Bennett, Phys. Rev. Lett. **68**, 3121 (1992)
8. A.K. Ekert, Phys. Rev. Lett. **67**, 661 (1991)
9. C.H. Bennett, F. Bessette, G. Brassard, L. Salvail, J. Smolin, J. Cryptology **5**, 3 (1992)
10. L. Goldenberg, L. Vaidman, Phys. Rev. Lett. **75**, 1239 (1995)
11. M. Koashi, N. Imoto, Phys. Rev. Lett. **79**, 2383 (1997)
12. H. Bechmann-Pasquinucc, Asher Peres, Phys. Rev. Lett. **85**, 3313 (2000)
13. D. Bruss, Phys. Rev. Lett. **81**, 3018 (1998)
14. G.H. Zeng, Z.Y. Wang, H.W. Zhu, The 5th *International Conference on Quantum Communication, Measurement and Computing* (Kluver Academic/Plenum Publishers, Capri, 2000), p. 2
15. A. Barenco, C.H. Bennett, R. Cleve, D.P. Di Vincenzo, N. Margolus, P. Shor, Phys. Rev. A **52**, 3457 (1995)
16. H.M.N. Dilum Bandara, A.P. Jayasumana, Peer-to-Peer Networking and Applications, Vol. 6 (Springer, 2013), p. 257
17. C.-J. Guo, Y.-M. Huang, Int. J. Innovative Comput. Inf. Cont. **8**, 5523 (2012)
18. H. Siramdasu, H. Krishna, Int. J. Eng. Trends Technol. (IJETT) **4**, 1367 (2013)
19. D.S. Devi, G. Padmavathi, Int. J. Comput. Sci. Inf. Security **7**, 218 (2010)
20. S. Cobourne, *Quantum Key Distribution Protocols and Applications. Technical Report, Department of Mathematics* (England: University of London, 2011)
21. Y. Kumar, R. Munjal, H. Sharma, Int. J. Comput. Sci. Manag. Studies **11**, 60 (2011)
22. H. Ansari, A. Parameswaran, L. Antani, B. Aditya, A. Taly, L. Kumar, *Network Security Course Project Report, Department of Computer Science and Engineering* (Mumbai: Indian Institute of Technology, Bombay, 2006)
23. G. Zeng, *Quantum Private Communication* (Springer Berlin-Heidelberg, 2010)
24. A.F. Metwaly, M.Z. Rashad, F.A. Omara, A.A. Megahed, *8th IEEE International Conference on Informatics and Systems* (Cairo, 2012), p. 25
25. S. Deering, Internet Draft, RFC 1112, August (1989)
26. S. Berkovits, *Advances in Cryptology, EUROCRYPT*, Lecture Notes in Computer Science, Vol. 547 (Springer-Verlag, Brighton, UK 1991), p. 535
27. G.H. Chiou, W.T. Chen, IEEE Trans. Software Eng. **15**, 929 (1989)
28. A. Fiat, M. Naor, *Advances in Cryptology*, Lecture Notes in Computer Science, Vol. 773 (Springer-Verlag, California, USA, 1993), p. 480
29. D.R. Stinson, Designs, Codes & Cryptography, Vol. 12 (Springer, 1997), p 215
30. M. Hirvensalo, *Natural Computing Series* (Springer, 2004)
31. S. Devaraju, P. Ganapathi, Int. J. Comput. Sci. Issues **7**, 30 (2010)
32. T. Jennewein, C. Simon, G. Weihs, H. Weinfurter, A. Zeilinger, Phys. Rev. Lett. **84**, 4729 (2000)
33. R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, B. Pinkas, Proc. INFOCOM'99 Conf. Comput. Comm. **2**, 708 (1999)
34. G. Caronni, K. Waldvogel, D. Sun, B. Plattner, Proceedings of the Seventh IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE '98) (1998), p. 376
35. M. Koashi, Efficient quantum key distribution with practical sources and detectors [`arXiv:quant-ph/0609180`] (2006)
36. P. Rice, J.W. Harrington, Numerical analysis of decoy state quantum key distribution protocols [`ArXiv:0901.0013`] (2009)