A SIMPLE POST QUANTUM SCHEME FOR HIGHER KEY RATE MULTIPARTY QUANTUM KEY DISTRIBUTION

Abudhahir Buhari[1], Zuriati Ahmad Zukarnain[1], Shamala K. Subramaniam[1] ,
Hisham Zainuddin[2] and Suhairi Saharudin [3]

[1]FSKTM, University Putra Malaysia, Serdang, Malaysia
toabu@hotmail.com, zuriati@fsktm.upm.edu.my,shamala@fsktm.upm.edu.my
[2] INSPEM, University Putra Malaysia, Serdang, Malaysia
hisham@fsas.upm.edu.my
[3] MIMOS, Technology Park Malaysia, Kuala Lumpur, Malaysia
Suhairi.sh@mimos.my

ABSTRACT

We propose a multi-party quantum key distribution protocol which enables all the receivers can convert their respective private shared key into common secret key without use of entanglement. The main component of our protocol is a simple post quantum scheme for achieving the higher secret key rate. Efficiency of the extracted key rate is almost 100%. We assume that sender established the pre-shared private secret keys and a common secret number with the receivers. Our proposed scheme sends n strings of number to n receivers in the public channel to convert their respective shared secret key into common secret key in the presence of Eve. We also analyze the complexity of attack by the adversary to guess the secret key.

KEYWORDS

Network security, quantum cryptography, secret key distribution, multi-party QKD

1. INTRODUCTION

Communication security is become a critical aspect of today's hi-speed world. Today's transmission security relies on the unproven computational security. There have been so many intrigue researches on providing better security. Under the Quantum Information Science (QIS), quantum cryptography (QC) researches promise to provide the unconditional security. QC protocols are relied on the principles of quantum mechanics or laws of nature. From the ground breaking research from Bennet and Brassard [1] until recent second generation QC called QLE-1, QC undergoes various changes and an active target for both code developers and code breakers. QC consist of various domains i.e. quantum key distribution (QKD), quantum secret sharing key (QSSK), quantum authentication and so on. QKD is one of the matured fields in QIS. There have been many researches on QKD attributes include environment, hardware, encoding methods, algorithms to achieve low cost and efficient key distribution system.

Multiparty QKD (MQKD) is a key distribution protocol in which the same key is distributed to different parties using quantum mechanism. In the jargon of digital cryptography, MQKD can refer as a conference key distribution protocol (CKDP) establishes a common key among a number of users forming a conference. To achieve the practical feasibility and simplicity in MQKD is the hot research. In the history of QC after the QKD research, QSSK is much concentrated work because it aligns with nature of quantum mechanism. In other words, we can

DOI : 10.5121/ijnsa.2012.4501

refer as stochastic or non-deterministic nature. While for MQKD or conference key is much lesser than compare to QSSK. Most researches have used entanglement property to achieve multiparty system.

The aim of this paper is to propose a simple protocol without use of entanglement to achieve high efficient key rate and attack resilient MQKD.

2. RELATED WORKS

Fundamentally, QC protocols can be classified into two major kinds' namely single photon and entangled photon. Furthermore teleportation, quantum computer and other derivatives of quantum mechanics family supports development of QC protocols. Entanglement based protocols are still in state-of-art while fainted laser or single photon concept protocols have been presented in the market since some solid years.

The main goal of this paper is to achieve MQKD without use of entanglement. Therefore, we have only a small recap on the entanglement based protocols. The entanglement has different states like Einstein-Podolsky-Rosen (EPR) and Greenberger-Horne-Zeilinger (GHZ) and Calderbank-Shor Steane CSS). Cabello extended his entanglement swapping concept for MQKD and QSSK [2]. His protocols are efficient in both data transmission and eavesdropper detection. However, the protocol is impractical due to difficult in differentiation of complete Bell state and GHZ. Hong et al. proposed an enhanced MQKD using entanglement swapping and a distributor [3]. Nishioka expressed the possibility of constructing multiparty architecture by extending one to many link architecture and looped networks [4]. He demonstrated circular type interferometer system for two parties. Zeng et al. proposed group of key establishment protocols suitable for small or medium-sized groups by using semi-trusted servers [5]. Some of the protocols are able to achieve authentication among the groups. Sing et al. presented a MQKD using EPR and group theory [6]. They proves unconditional security using mechanism of bipartite key distribution and also convert the n-KD problem into 2-KD. Ramzan et al. proposed a MQKD protocol using GHZ sate and decoding matrix [7]. They also proved the security in the presence of Eve with intercept and resend attacks. Nihira et al. proposed two parties with multi-level using mixed entangled state generate a secret key by taking advantage of residual entanglement of the reduced density matrix [8]. This protocol also explained the practical realization of photon orbital angular momentum. Chen et al. utilized wide class of distillation schemes for multi-partite entangled states that are CSS-states and bipartite graph [9]. They also highlighted the actualization.

As a prior relevant research, Matsumoto proposed a first protocol without use of entanglement to achieve MQKD which enables three parties agrees at once on a shared common random bit strings in presence of eavesdroppers [10]. The main difference between proposed protocol and Matsumo's protocol is that proposed protocol allows n parities to share a common secret key after the establishment of secret key among the parties. Furthermore our protocol utilizes one way public communication (post processing) to share a final secret key. Here, Matsumoto's protocol requires three way post processing effectively. All the parties are required participate in the calculation. On contrast, our proposed protocol needs only the sender to transmit a public message to the parties. As long as, the public channel is authenticated and unedited by Eve then our proposed protocol proves unconditional security. Moreover we use simple post-processing technique to share a common secret key among the parties. The disfavour of our protocol, all parties are required to establish a secret key among the parties. However, proposed protocol is higher efficiency in terms of extract common secret key. Our protocol has the feature which allows using the same key repeatedly and differently.

3. MOTIVATION

One of the deficiencies of the QC protocols is not deterministic. On the other hand, this stochastic nature is the center of providing unconditional security. Sending efficiently the same data to different users by using quantum protocols is still challenging goal to the researchers.

To achieve a higher key rate MQKD using simple algorithm without entanglement, one-way public communication and resilient to sophisticated attacks are our motivation. To accomplish this task, we propose a protocol in this paper as a first of many steps. So far, QKD can establish secret key between the parties. Is there any way to convert the shared secret keys into common secret key in the presence of Eve? We would like to answer this question by following puzzle game.

This is a multiplayer puzzle game. Players are organized as team. Each player in the team is given with a row matrix of 0's and 1's in a random order. The aim of the game is to find out first team is able to convert their players' different matrix into common matrix. The rules of the game are players are not allowed to discuss the value and the team members are not allowed to view other members' matrix except the leader. Each team leader has got an equal time to see their players' matrix. After the time over, leader are separated, they have to give voice commands about the position or index of the matrix to his team players for converting their respective matrix into common matrix. How it would be possible?

Firstly, leader has to determine the common secret matrix's elements arrangement or order of 0's and 1's in a row matrix. This can be done by two ways; first, he can create a new matrix or simply use his given matrix as a common matrix. The secret of success is the memory power of the leader, if he able to remember his team members' matrix then it is a simple game. For example, the leader matrix's index n contains value 0. He also knows his team players' matrix index nth value. Suppose one of his player got value 1 in the nth index of the matrix. Leader simply gives a command which is the position or index of the matrix to the specific player. The corresponding team player understands that the particular matrix element position should change to 0. Indeed, it is a matrix with binary elements i.e. 0's and 1's. For the players who got their nth index value 0 then leader has no need to give any command on that position to the team players. Using this same strategy, leader able to convert all his team members' matrix into common matrix under the assumption of leader knows exact values of all matrices.

Let us map this above game to multiparty environment scenario and each party's shared secret key with the sender as player's matrix in the game. Sender can view as a leader and the receivers can regard as team players. Sender established a shared secret key with each user. In the game scenario, leader knows the matrix of the other players and gives voice command to convert. Likewise, sender creates a common secret key and gives commands to the player to convert their respective private shared secret keys into common secret keys.

Let us evaluate difficulty of eavesdropping in the above puzzle game. The only information for converting into common matrix is voice commands given by the leader to his respective team players. This command is just a position of matrix. Eve tries to hear all the commands and guess the matrix. As long the players' matrices are secret, Eve has to try all possibilities for matrix size and sequence of matrix's elements. Additionally, leader can give some false commands like imaginary matrix's index to confuse eavesdropper and easy to differentiate by his team player.

To achieve an efficient MQKD, we apply the concept of the above puzzle game. This can be done by transforming the shared secret keys between sender and receiver into group of matrices and inform the positions or indices for converting their shared keys into common secret key. Sender includes fake index and imaginary matrices in the information to confuse the Eve. As far as shared secret key and matrix information are fully secret, it is subtle for Eve to determine the information about key even she computationally strong. To increase the complexity of mechanism we convert the one dimensional key into square matrix format (two dimensions). Hence, without the knowledge of secret key and matrix size, for the Eve would be difficult to guess the correct key exactly like eavesdropping in the puzzle game.

4. Protocol Description

This section introduces our proposed protocol. Section 4.1 explains the assumptions of the proposed protocol. Section 4.2 illustrates the protocol's components or functions. The detailed sequence of proposed protocol is presented in section 4.3 and a three-party environment protocol setup is demonstrated as an example of proposed protocol in section 4.4.

4.1 Assumptions

Each party has point-to-point quantum channel or private channel to other parties and a common public channel. Quantum channel can be eavesdropped and modified by adversary or Eve while public channel's message can be eavesdropped but cannot be modified by Eve. These assumptions are pointed in Ref. [10]. In our proposed protocol, additional to the above assumptions, we have some more assumptions which are elaborated in the following points.

- Each party established a shared secret key with all parties in the network. We assume that this establishment can be done by any QKD protocols. From this point of writing, we refer shared secret key between sender and receiver as a private key. Let say, a network consist of n parties and then each party has (n-1) keys. The total number of keys in the network is (n * (n-1)). This is a limitation of our proposed protocol, as the party increases in the network then the total number of keys also increases.

- Sender can send a common secret number to the receivers. We assume that transmission of secret number in quantum channel. Until now, there no such quantum protocol to send the short common bits or deterministic data to the receivers. This limitation leads towards the realization of short key quantum deterministic protocol as our future work. Here we consider the secret number or square matrix's dimension value from range 1 - 99.

4.2 Illustrations of Protocol's Components

In this section, we define and elaborate some of the proposed protocol's components.

4.2.1. Definition of key

This key is same as digital cryptography key presentation. Key is a random binary string. In other words, a row matrix with random of 0's and 1's. In our proposed protocol, the position or index of the elements is important. The following diagram illustrates a key structure.

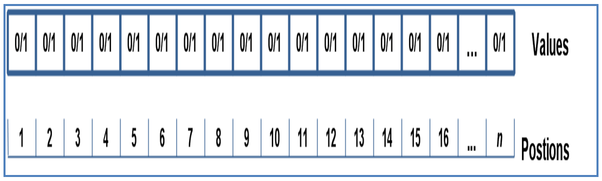


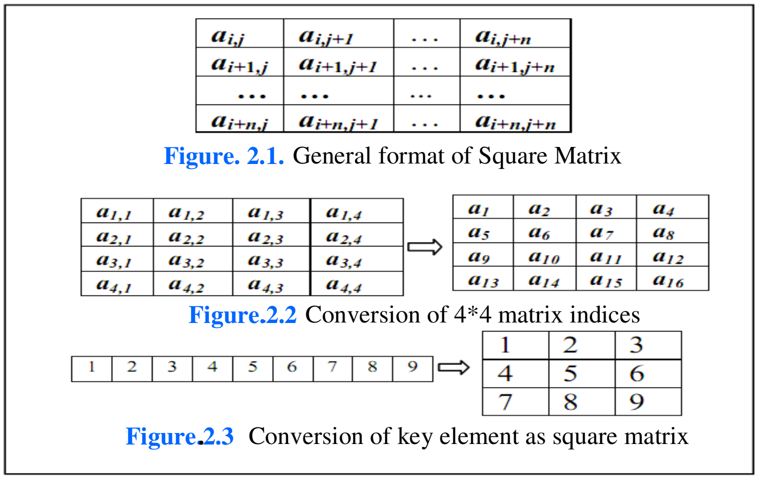
Figure 1. Key with n size

4.2.2. Conversion of the key into square matrices

This conversion mechanism is the heart of our proposed protocol. Private keys are in a structure of row matrix with the random binary numbers. MQKD aims that all parties have common key or common row matrix. Simply say, all parties' private key should have common order or arrangement of 0's and 1's. Converting one dimensional matrix in to two dimensional matrixes needs well defined algorithm. Instead of applying complex algorithm, we use a simple method. Our method still considers the key as a one dimension matrix.

Firstly, we split the key into multiple keys or sub keys by the size of square of secret number or size of the matrix. Secondly, we create few empty square matrices by defined condition. The total number matrices are equal to the total number of sub keys. Then, we match each sub key with a matrix. Lastly, we merely fill square matrix's empty spaces with the sub key's binary elements. Thus, creating and filling the matrix are the important functions of our proposed protocol. These two functions are explained elaborately in this remaining section.

Basically a square matrix has two subscripts represents row and column. Both row and column values are same. Let say mi,j is a square matrix. The following figure 2 (2.1, 2.2, 2.3) explain the general format of square matrix with n * n size, conversion of standard 4*4 matrix indices and an example of conversion of secret key's element into square matrix.



$$\begin{array}{|c|c|c|c|}\hline a_{i,j} & a_{i,j+1} & \cdots & a_{i,j+n} \\ \hline a_{i+1,j} & a_{i+1,j+1} & \cdots & a_{i+1,j+n} \\ \hline \cdots & \cdots & \cdots & \cdots \\ \hline a_{i+n,j} & a_{i+n,j+1} & \cdots & a_{i+n,j+n} \\ \hline \end{array}$$

Figure. 2.1. General format of Square Matrix

$$\begin{array}{|c|c|c|c|}\hline a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ \hline a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ \hline a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ \hline a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \\ \hline \end{array} \Rightarrow \begin{array}{|c|c|c|c|}\hline a_{1} & a_{2} & a_{3} & a_{4} \\ \hline a_{5} & a_{6} & a_{7} & a_{8} \\ \hline a_{9} & a_{10} & a_{11} & a_{12} \\ \hline a_{13} & a_{14} & a_{15} & a_{16} \\ \hline \end{array}$$

Figure.2.2 Conversion of 4*4 matrix indices

$$\begin{array}{|c|c|c|c|c|c|c|c|c|}\hline 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline \end{array} \Rightarrow \begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline 7 & 8 & 9 \\ \hline \end{array}$$

Figure.2.3 Conversion of key element as square matrix

Figure 2. Mechanism of Key Conversion

In our proposed protocol sender sends the column (or row) value of the square matrix as a common secret number Φ to the receivers. Then receivers break their private key into smaller keys or group of blocks by the number of matrix size or square of secret number. Let denote k as size of a key and m as the matrix size or square of secret number (Φ). Let T_m denotes the total number of matrices can be extracted from the key. To find the T_m , we perform a simple calculation called matrix calculator. Matrix calculator has two criteria. Depends on the private key size and secret number, matrix calculator satisfies one of the criteria.

Matrix Calculator

If $k \,\%\, m = 0$ then, → criteria (*i*)

T_m = quotient (k / m)

else $k \,\%\, m \neq 0$ then, → criteria (*ii*)

$T_m = Integer(quotient(k / m) + 1$

$Empty\ spaces = (T_m * m) - k$

From the above calculation, users can attain the total number of empty square matrix. The main idea around this calculation is to create small size square matrices for filling the big size key's elements which is in one dimensional matrix. Suppose the size of the sub key is exactly equals to size of matrix or total number of the matrix's elements. This situation occurs where the size of key is equal to the square of any positive integer number. Then, all the sub key's elements are filled evenly in the matrices. Thus Tm satisfies the criteria (i). On the other hand, the size of sub key is lesser than the size of the matrix, then the matrix's indices that don't have equivalent key's element to fill. Then fill those indices as empty. This situation occurs when Tm falls on the criteria (ii). To find the total number of empty matrix's elements by simply subtracting total size of the key from total number of matrices' element.

The following figure.3 illustrates the conversion of n length key into n matrices with secret number or matrix dimension is 2. So size of matrix or the total number of elements in a matrix is 4. Let say k, m denote key size and matrix size. By the matrix calculator, If the k % m = 0 then all matrix elements are filled with elements of key. On the other hand, if the k % m ≠ 0, then extra matrix indices are filled with empty.

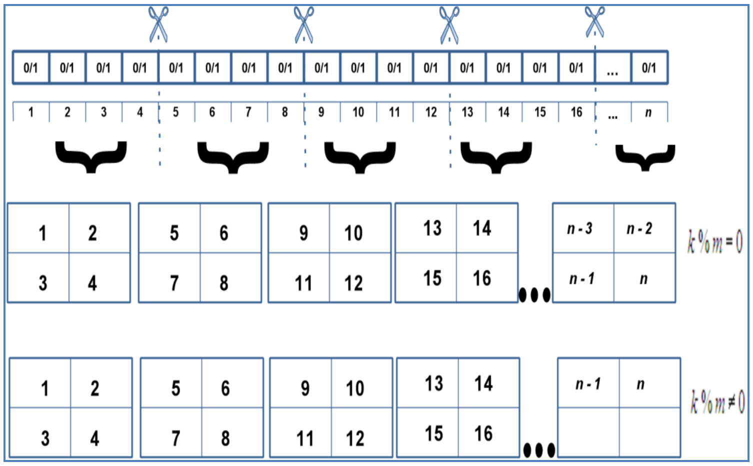


Figure 3. Conversion of key into matrices

4.2.3. Convertor Command

This is the vital component of our proposed scheme which converts the receivers' private key into common secret key. This command is a plaintext consist of some numbers transmit through public channel by the sender to the receivers. It has two segments, namely conversion is a mandatory segment and an optional segment called destruction. Conversion segment contains set of lists. Each list is a collection of positive integers and enclosed in square brackets. The destruction segment contains a positive integer value which is enclosed in parenthesis.

4.2.4. Conversion segment

Each list represents the positions or indices of an actual matrix or imaginary matrix. When receivers obtain this message, they change the element's value of the particular matrix's index or position. Here, changing the value means to convert 0's to 1's and vice versa. To make subtle for the eavesdropping, fake indices are included in the actual matrix. The index of a matrix is range from 1-9800. The maximum number is fixed because of the assumption that maximum value of the secret number is 99. Hence the maximum number of elements in a square matrix is 9800 (992). The total number of matrices or list depends on the sender, because he/she can include many imaginary matrices to obscure the Eve. This command varies for the receivers depending on their private keys.

4.2.5. Destruction segment

Typically, the receivers' private key established by the QKD protocols may not be same in size. Unlikely, our proposed protocol requires same size private keys to achieve the common secret key. Thus, sender's responsibility is to make all the different size private keys into equal size. To accomplish this task, sender chooses the shortest private key's size as the size for the common secret key. Besides, sender knows all the private keys length. Sender sends key reducing size as a plain text to the receivers whose private keys are greater than common secret key. Upon receiving this value by the receivers, they simply delete the rightmost elements of their private key. Like conversion segment, this value also varies according to the receivers. This segment is an optional.

The format of the convertor command as follows

{
n List [Actual matrices' real+ fake positions]
n List [Imaginary matrices' element positions]
(Positive integer value)// optional
}
where Matrix index ∈ *{1, 2, 3…9800}*

4.2.6. Key scheduler command

This is the final function of our proposed protocol. This function transmits the plaintext or command in the public channel to the corresponding receiver by the sender. The purpose of this command is to extract multiple common secret key from the common secret which is established by the convertor command. To derive the multiple keys from the key scheduler command, receivers need same private shared keys with the sender and same or different secret number (matrix size). Then, receivers have to convert their respective private keys into matrices. This process is same as conversion of key into square matrices. Thus, our proposed protocol is more efficient than the previous protocols in terms of repeat usage of shared keys. This command can be act as privacy amplification for the common secret key too. Therefore, we propose few simple techniques to extract or retrieve a new key from the old key. These techniques are publicly defined. So user can create his/her own methods and publish in the public. Sender issues single or combination of extraction techniques to the receivers, only the legitimate receiver can retrieve a new key. The following figure. 4 explains our techniques.

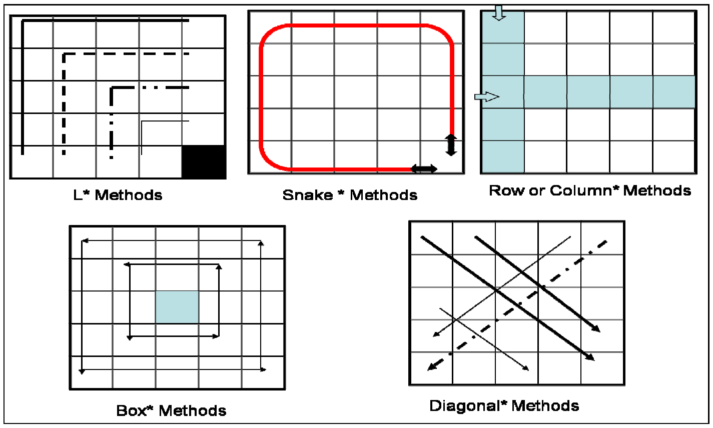


Figure 4. Key Scheduler Methods

The rationale behind these techniques is once the parties attain common secret key, they can manipulate it to derive more secret keys. Our main idea, sender can dictate some fetching order from the matrices and combine all these order to attain a new key. Square matrix is a combination of rows and columns. Therefore, sender can dictate the row or column values in a defined order. To get a new key just combine all outputs of the methods. The figure. 4 explains few methods for creating a new key from old key. For example, L* is a method of retrieving the combination of row and column value in an L shape. The * refers to the attributes of the method i.e. refers to different angles of L like inverted L, flipped L and so on. Row * or Column* method is used to fetch the sender determined row or column elements. The attributes are ones operation or two's operation on row or column binary elements. Snake* method retrieves by cyclic combination of rows and column elements. It contains attributes like full cover or half cover. Box* method retrieves like a box style and Diagonal* read the diagonal element of a matrix. Users can develop their own method and apply it.

4.3 Higher key rate MQKD protocol

1. This section explains the proposed protocol with all functionalities together to achieve the higher key rate in MQKD.

2. Sender creates the common secret key (Ω). Sender can generate the key randomly or use any other techniques from digital cryptography to derive from all the private keys. Let say, Kxy_i be the private keys between sender and receivers. Here, K represents the private key and x refers to a sender and y refers to receivers (where i = {1…n}). The length of the common secret key is lesser or equal to the shortest private key.

$$\text{Length } (\Omega) <= \text{length (shortest ((Kxy))}$$

3. Sender sends the common secret number or matrix's size (Φ) to all the receivers in the quantum channel.
4. Sender converts Ω into set of matrices and receivers transform their respective private keys into group of matrices by the value of matrix size.
5. Sender creates and sends the convertor commands to the receivers for converting their respective shared secret key into common secret key.
6. Sender sends same key scheduler command to the receivers to derive multiple secret key or privacy amplified common secret key.

4.4. Three-party MQKD protocol setup

In this section, three-party system namely Alice, Bob and Charlie are establishing the common secret key by using proposed protocol. Here, we assume that Alice is the sender, Bob and Charlie as receivers. Receivers already established a private key with sender. From this assumption, Alice knows both private secret keys of Bob and Charlie. Let denote K_{AB} as a private shared secret key between Alice and Bob and K_{AC} is between Alice and Charlie. Alice creates the common secret key (Ω) randomly with the length as following condition

$$\text{Length}(\Omega) < \min(\text{length}(K_{AB}), \text{length}(K_{AC}))$$

For instance, K_{AB} is 850 bits long and K_{AC} is 860. Alice chooses the length (Ω) < 850. For this example, Alice chooses 850 as the size of Ω. Next she has to choose the value for secret number Φ. Here, Alice chooses 8 as a secret number or matrix size and transmits to both Bob and Charlie in the quantum channel.

4.4.1. Preparation of matrices by Alice

After completion of sending the secret number, she has to prepare the matrices from the common secret key. Here k refers to the common secret key. Let m denotes the total number of elements or size of the matrix. By applying the matrix calculator (refer section: 4.2). She calculates total matrices and total empty space as following.

```
Total Size of Key ($k$) = 850
Matrix Size ($m$) = 64 ($\Phi^2 = 8^2$)
Matrix Calculator
  i) $k \% m = 0$ then
       $Tm$ = quotient ($k / m$)
  ii) $k \% m \neq 0$ then
       $T_m$ = Integer (quotient ($k / m$)) + 1
       Empty spaces = = ($T_m * m$) - $k$
Criteria (ii) fits, Total number of matrices ($Tm$) can be calculated as follows,
    850/ 64 = 13.28
    $T_m$ = 13 + 1 = 14
    Empty spaces= (14 * 64) – 850 = 46
```

4.4.2. Preparation of matrices by Bob and Charlie

Bob and Charlie also apply the similar way by Alice to convert their private keys into set of matrices by utilizing received secret number. Both of them fill with blank for the empty matrix indices.

Bob Matrix Calculator

Private Key size $(k) = 850$
Matrix size $(m) = 64$
Matrix Calculator
→850 % 64 $\neq 0$
$\therefore$ criteria (ii)
$T_m = 13 + 1$
$T_m = 14$
Empty matrix spaces
=896 -850
= 46

Charlie Matrix Calculator

Private Key size $(k) = 860$
Matrix size $(m) = 64$
Matrix Calculator
→860 % 64 $\neq 0$
$\therefore$ criteria (ii)
$T_m = 13 + 1$
$T_m = 14$
Empty matrix spaces
= 896 – 860
= 36

4.4.3. Bob's convertor command by Alice

Alice creates a convertor command for Bob. The following diagram represents a convertor command for Bob which exclude destruction segment. This is due to the size of his private key is equal to the common secret key size. Actually, convertor command contains no attributes to the font. This below convertor command is only for illustration purpose. The numbers in bold and single underline are represents the actual value position and numbers in italic and single underline represents fake elements. Three dots with underline represent the combination of actual matrix's real element position and fake position. Three dots with double underline represent the fake matrix's positions.

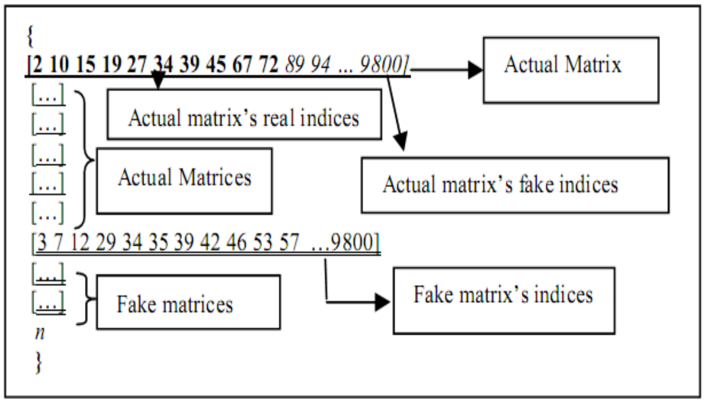


Figure 5. Convertor Command for Bob

4.4.4 The convertor command for Charlie by the Alice

For Charlie, both conversion and destruction segments are included in convertor command. The destruction segment is included because of the size of his private key is greater than common secret key (860 > 850). Number 10 is enclosed in parentheses as discarding bits for Charlie's private key.

```
{
[2 10 15 19 … 9800]
[1…9800] [1…9800]
[1…9800]
[3 7 12 29 34 35 …9800]
[…] … n
(10)
}
```

4.4.5. Deduce the common secret key from convertor command

For legitimate users to derive the key is simple, because they know the matrix size and total number of matrix. Receiver can easily differentiate between real and fake matrix's index. In the case of Eve, she has to use brute force attack. Eve captures all the receivers' convertor commands in order to guess key. As for the matrix dimension and shared key are fully secret, Eve has to try all possible values of matrix dimension, arrangement of binary elements and total number of matrices. Due to the unavailable of verification scheme between sender and receiver in our proposed protocol makes more complication to Eve, even if she determine a key by guessing. Our proposed protocol is a one-way public message transmission scheme. Moreover, to calculate the complexity of guessing the key by Eve is explained in the next section.

5. Security analyses

The only region in our protocol requires security analysis is during the transmission of different convertor commands to the receivers. The convertor command contains some numbers, in order to deduce the key from this numbers Eve has to apply brute force attack. In other words, guessing all possibilities of the value. Hence we analyze the security of our proposed protocol by directly relating it to the min-entropy and guessing entropy [11]. We estimate the complexity of probability to guess the common secret key. To make this paper self-contained, we shall briefly explain the entropy measurement.

5.1 Reviews on Min-Entropy and Guessing Entropy

In this section we give overview of the entropy measurement and mapping the key guessing complexity with guessing entropy. Massey [12] initiated the problem of guessing the value of a random variable X by asking only questions of the form "is X equal to x?" until the correct value is found and the answer is "yes". This type of situation happens in the cryptanalysis of computationally secure ciphers. Massey's assumption is that a cryptosystem is secure in the way intended by its designers, the only attack for finding the secret key is trying all the possible keys in sequence for a given plaintext-ciphertext pair. The probability that the correct value is guesses in the first trial is directly linked to the min-entropy of X and is equal to $2^{-H\infty(X)} = \max_{x\in X} P_X(x)$ under an optimal strategy. An upper bound on this probability in terms of

Shannon entropy is provided by the well-known Fano inequality, which gives a lower bound on the error probability of guessing X from knowledge of a correlated random variable Y. The optimal strategy for successive guessing until the value of X is found is obviously to try the possible values in order of decreasing probability. Denote the elements of the probability distribution P_X by $p_{1,\ldots,}p_n$ such that $p_1 > p_2 > \ldots > p_n$ with $n = |\chi|$. For a fixed optimal guessing strategy, let a guessing function for X is a function $G: \chi -> N$ such that G(x) denotes the number of guesses needed when X = x. The average number of guesses needed to determine X can be called the guessing entropy of X [11].

$$\mathrm{E}\,[G(X)] = \sum_{i=1}^{n} ip_{i,} \tag{1}$$

In the case of guessing X with knowledge of a correlated random variable Y, let G(X | Y) be a guessing function for X given Y when G(X | y) is a guessing function for the probability distribution $P_{X\,|\,Y\,=\,y}$ for any fixed $y \in \gamma$. Thus can be called the conditional guessing entropy of X given Y.

$$\mathrm{E}[G(X|Y)] = \Sigma_{y\in\gamma} P_Y(y)\ \mathrm{E}[G(X \mid y)] \tag{2}$$

5.2. Guessing Complexity of the Common Secret Key

To guess the key by Eve needs all matrices' element arrangements. Eve has to guess the size of the matrix and the total number of matrices. Further, she has to predict a probability of arrangement or sequence of the elements for each matrix. Therefore, matrix size, total number of matrices and sequence of matrices' elements can be considered as random variables to Eve. Eve has the only information from the convertor command which contains combination of actual and fake matrices' index. Therefore, we relate this guesses by Eve with min-entropy and guessing entropy to generalize the complexity of determining key. Min-entropy can be applied if Eve chooses all the random variable guesses on a first trial. Otherwise, guessing entropy is more suitable determining the key's complexity. Thus, we applied guessing entropy for finding common secret key complexity.

5.2.1. To determine the dimension of the matrix or secret number by the Eve

The secret number or matrix dimension is an important value to extract the common secret key from the private keys. The value is publicly known as in range from 1 – 99. Therefore, Eve tries all the option and chooses one for applying to other guesses. We are applying the guessing entropy to guess the dimension of matrix. Let denote D_m as the dimension of matrix, then expectation value by guessing entropy using eqn.1.

$$\mathrm{E}\,[G(D_m)] = \sum_{i=1}^{99} ip_i \tag{3}$$

5.2.2. Guessing the sequence of matrix's elements

After the guessing of matrix dimension, now Eve tries to guess the sequence of a matrix's elements. To find the sequence of matrix's elements, the total number of matrix elements is important. Finding the total number of elements is simple because of the square matrix. Thus, total number of elements in a square matrix is equal to [matrix dimension] ^{2} = (E [$G\ (D_m)$])^2 . Since the matrix elements are binary, then total number of possibilities in a matrix (P_m) is equal to $2^{\text{Total number of elements in matrix}}$, the following equation expresses the total possibilities.

$$P_m = 2^{(D_m)^2} \tag{4}$$

5.2.3. Guessing the total number of matrices

Now Eve attempts for guessing the total number of matrices. By using the guessing entropy, we can estimate the total possibilities for the matrix. Let T_m be the total number of matrices, then expectation value of T_m as follows

$$E[G(T_m)] = \sum_{i=1}^{n} iP_i \tag{5}$$

5.2.4. Guessing the common secret key

Ultimately, Eve tries to guess the common secret key. Eve should guess all possibilities of matrix's elements sequence. Then all possibilities should apply to all matrices. Thus complexity increases as big *$O(2^n)$*. In order to generalize the complexity, we calculate product of all matrices' element with all possible sequences. By applying guessing entropy, the expectation value for common secret key Ω,

$$E(\Omega) = \prod_{x} 2^{\left(\sum_{i=1}^{99} iPi\right)^2} \quad (1 \le x \le n) \tag{7}$$

Here x = number of matrices and substitute the equation (4) in equation (7) then,

$$E(\Omega) = \prod_{x} P_m \tag{8}$$

Finally, substitute the equation (8)'s variable x eqn.5 then,

$$E(\Omega) = \prod_{E[G(T_m)]} P_m \quad \text{where } E[G(Tm)] = \left(\sum_{i=1}^{n} iP_i\right) \tag{9}$$

The equation (9) generalizes the complexity of guessing the common secret key. As the number of matrices increases then the complexity also increases. In our proposed protocol addition of fake matrices method increases the hardness of guessing. Further, each matrix consists of fake indices which also increase the possibilities. In conclusion, Eve's information about the key is negligible.

5.3 Discussion

From the above section, to achieve multi-party key distribution is simple for legitimate users and hard for illegitimate users are proved by the complexity calculation. The increase of the fake indices and imaginary matrices will increase the complexity of guessing the key. Our proposed protocol achieves this by a simple mechanism.

For instance, Eve guesses correctly the matrix dimension. She cannot retrieve any information about the key because the convertor command only contains the matrix position not its value. This situation is same as public discussion of BB84 [1] protocol which reveals the polarizer not the outcome. Additionally, this protocol is one-way post processing in which sender only communicate with the receivers. Thus, it reduces the usage of extensive public discussion among the parties. Furthermore, key scheduler command contains only publicly known methods name and Eve computes nothing from this information unless the private keys and matrix size are not fully secret.

6. Conclusions

We have proposed a scheme that allows *n* receivers to convert their private secret key with the sender into common secret key. The advantage of the scheme is sender can sends *n* public message to the *n* receivers for conversion. The amount of extraction of common secret key is almost same size as the private key. Additionally, same key can be used to derive multiple secret keys without any compromise in security. Thus high efficiency in usage of established key results in cost-effective protocol. Furthermore, it requires one-way public communication for transmitting and unavailable of verification scheme among the legitimate users makes harder for Eve to determine the key. The brute force method is only way to guess the key and probability of guessing the key involves so much complex. In discussion section is showed the Eve's knowledge about key is negligible. However, our scheme requires pre-shared secret key and a common secret number among the parties. The future research agenda is practical realization of sending the secret number using quantum channel to all the parties.

REFERENCES

[1] C.H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing* Bangalore, India, 1984.

[2] A. Cabello, "Multiparty key distribution and secret sharing based on entanglement swapping," *Arxiv preprint quant-ph/0009025*, 2000.

[3] C.H. Hong, et al., "N quantum channels are sufficient for Multi-user Quantum Key Distribution protocol between n users," *Optics Communications*.

[4] T. Nishioka, et al., "Circular type'quantum key distribution," *Arxiv preprint quant-ph/0106083*, 2001.

[5] D. Zheng, et al., "Multiparty authentication services and key agreement protocols with semi-trusted third party," *Journal of Computer Science and Technology*, vol. 17, no. 6, 2002, pp. 749-756.

[6] S.K. Singh and R. Srikanth, "Unconditionally Secure Multipartite Quantum Key Distribution," *Arxiv preprint quant-ph/0306118*, 2003.

[7] M. Ramzan and M.K. Khan, "Multiparty quantum cryptographic protocol," *Chinese Physics Letters*, vol. 25, 2008, pp. 3543-3546.

[8] H. Nihira and C.R. Stroud Jr, "Robust multipartite multilevel quantum protocols," *Physical Review A*, vol. 72, no. 2, 2005, pp. 22337.

[9] K. Chen and H.K. Lo, "Conference key agreement and quantum sharing of classical secrets with noisy GHZ states," 2005, pp. 1607-1611.

[10] R. Matsumoto, "Multiparty quantum-key-distribution protocol without use of entanglement," *Physical Review A*, vol. 76, no. 6, 2007, pp. 62316.

[11] C. Cachin, *Entropy measures and unconditional security in cryptography*, Zürich, 1997.

[12] J.L. Massey, "Guessing and entropy," *Proc. IEEE International Symposium on Information Theory, 1994. Proceedings*