

Quantum Direct Communication using Quantum Walks

A Thesis

submitted to

Indian Institute of Science Education and Research, Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Srikara S



Indian Institute of Science Education and Research, Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

June, 2020

Supervisor: Dr. C M Chandrashekar

© Srikara S 2020

All rights reserved

Certificate

This is to certify that this dissertation entitled Quantum Direct Communication using Quantum Walks towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Srihara S at The Institute of Mathematical Sciences, Chennai under the supervision of Dr. C M Chandrashekar, Reader, Department of Theoretical Physics, during the academic year 2019-2020.



Dr. C M Chandrashekar

Committee:

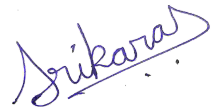
Dr. C M Chandrashekar

Dr. T S Mahesh

To my dearest Amma and Appa...

Declaration

I hereby declare that the matter embodied in the report entitled Quantum Direct Communication using Quantum Walks are the results of the work carried out by me at the Department of Theoretical Physics, the Institute of Mathematical Sciences, Chennai, under the supervision of Dr. C M Chandrashekar and the same has not been submitted elsewhere for any other degree.

A handwritten signature in blue ink, appearing to read 'Srikanth S', written in a cursive style.

Srikanth S

Acknowledgments

I would like to sincerely thank my supervisor, Dr. C M Chandrashekar and his group from IMSc for their guidance throughout the project and also for hosting me at IMSc, and providing me the accomodation facilities. I am also thankful to the whole of the staff and administration of IMSc for providing me with facilities such as hostel, computer account, cluster account, an IMSc webmail ID, access to great mess food, etc. and for treating me, a visitor, like their own student. I would like to thank Dr. T S Mahesh from IISER Pune for being my co-guide, and also for his valuable guidance throughout my tenure as a BS-MS student in IISER Pune. I would also like to thank Prof. Anirban Pathak and Dr. Kishore Thapliyal from IIIT Noida for their invaluable guidance and care since the past two years and without whom I would not have met Dr. Chandrashekar. I would like to thank all the friends I made at IMSc who made my stay at IMSc highly enjoyable and stress free. Finally, I would like to thank my family for being my biggest emotional and moral support throughout my MS thesis and also throughout my life.

Contents

1	Introduction	3
2	Discrete-time quantum walk on a cycle - preliminaries	5
3	The Protocols	7
3.1	Encoding of the message	7
3.2	Quantum Walk based Quantum Secure Direct Communication (QSDC) Protocol	8
3.3	Quantum Walk based Controlled Quantum Dialogue (CQD) Protocol	9
4	Security	10
4.1	Intercept-and-Resend Attack	11
4.1.1	Mutual Information between Alice and Eve	11
4.2	Denial of Service attack	16
4.3	Man-in-the-middle attack	16
4.4	Attack by an untrusted Charlie	16
5	Conclusion	17

List of Figures

1	Quantum Walk based QSDC protocol	8
2	Quantum Walk based QDC protocol	9
3	(a) I_{AE} vs ζ for $N = 3, n(T) = 7, \theta = \frac{\pi}{4}, \xi = \frac{\pi}{4}$ and (b) I_{AE} vs ξ for $N = 3, n(T) = 7, \theta = \frac{\pi}{4}, \zeta = \frac{\pi}{4}$	14
4	(a) I_{AE} vs θ for $N = 3, n(T) = 7, \zeta = \frac{\pi}{4}, \xi = \frac{\pi}{4}$ and (b) I_{AE} vs N for $n(T) = 7, \theta = \frac{\pi}{4}, \zeta = \frac{\pi}{4}, \xi = \frac{\pi}{4}$	14
5	(a) I_{AE} vs $n(T)$ for $N = 4, \theta = \frac{\pi}{4}, \zeta = \frac{\pi}{4}, \xi = \frac{\pi}{4}$ and (b) $n(T)$ for $N = 3, \theta = \frac{\pi}{4}, \zeta = \frac{\pi}{4}, \xi = \frac{\pi}{4}$. Green line represents the I_{AE} for one channel of the LM05 protocol, which is the same as the I_{AE} for the BB84 protocol.	15

Abstract

Quantum computation and quantum information has been one of the most extensive fields of research since the past three decades. Many groundbreaking quantum algorithms have been designed, which can perform the assigned tasks much faster than their classical counterparts. Shor's algorithm, one of the most well known quantum algorithms, can be used to prime factorise large integers in polynomial time, while even the best classical integer factorization algorithms are exponential in time. As the security of most classical cryptosystems rely on the difficulty to prime-factorize large numbers, the Shor's algorithm, if implemented on a scalable quantum computer, can compromise almost all of the classical cryptosystems that are widely used today. Hence, the requirement of quantum cryptography is greatly essential. In this thesis, I attempt to look into the possibility of exploiting quantum walks for the purposes of quantum cryptography. Quantum walks are the quantum analogues of the classical random walks. Unlike classical random walks, Quantum walks have unique properties such as superposition and entanglement of the position and the coin spaces, which can be exploited to design unconditionally secure quantum cryptographic protocols. I propose two new protocols, namely a Quantum Secure Direct Communication (QSDC) protocol and a Controlled Quantum Dialogue (CQD) protocol using discrete time quantum walks on a cycle. The proposed protocols have been shown to be unconditionally secure against various attacks such as the intercept-resend attack, the denial of service attack and the man-in-the-middle attack. Additionally, the proposed CQD protocol is shown to be unconditionally secure against an untrusted service provider. Also, it is shown that the proposed protocols are more secure against the intercept resend attack as compared to the qubit based LM05 protocol.

1 Introduction

Cryptography i.e. the study of reversibly morphing messages into an unreadable form, in order to securely share confidential information and maintain secrecy, has been in use since a few thousand years. Ancient Indians and Greeks used the substitution and the shift ciphers i.e. substituting one letter with another, or shifting the letters by a fixed number positions in the alphabet [11]. Medieval Persians had separate scripts exclusively used for secret communication [12]. From early 8th century to the early 15th century, many Arab mathematicians and scholars developed various innovative cryptosystems mostly based on exploiting the properties of Arabic linguistics [13, 14].

Though cryptography was studied and was in use since a long time, its progress was slow. The progress in research in cryptography increased rapidly in the 20th century, starting during the first and the second world war with the introduction of the enigma cypher in Germany, which was cracked with the turing machine, which is opined by many to be one of the first ever digital computers, and paved way to the birth of a totally new branch of study: Computer Science [15]. Then, with the advent of more powerful computers, more mathematical ciphers and cryptosystems such as the AES, DES, RSA, elliptic curve cryptosystems, El-Gamal, etc. were introduced [16, 17, 18, 19]. The security of these cryptosystems is dependent upon certain invertible functions that are easy to compute, but whose inverse is hard to compute, even by the best and the most powerful computers. This kind of security is called conditional security, as the difficulty to crack the encryption is conditional to the computing power of the machine used for cracking.

The early 20th century also witnessed the birth of Quantum Mechanics. Quantum mechanics could explain various physical phenomena that were impossible to explain via classical mechanics. Although quantum mechanics turned out to be one of the most fundamental and most important branches of physics, the idea of using quantum mechanics for computational purposes wasn't thought about until in the 1980s, when Paul Benioff developed the idea of a Quantum Turing Machine [20]. Later, Richard Feynman and Yuri Manin suggested that quantum computers could perform various tasks which a classical computer could not [21, 22]. This gave birth to two new fields of research, namely Quantum Computation and Quantum Information.

Inspired by the quantum turing model, many attempts were then made

to design algorithms that would work on a quantum computer, called Quantum Algorithms [23]. Quantum Algorithms, in many cases, provided a much higher computational speedup as compared to their classical counterparts. One of the most famous quantum algorithms, the Shor's Prime Factorization algorithm, introduced in 1994, could prime factorize integers in polynomial time [24], whereas even the best classical algorithms were exponential in time. Since the security of most of the widely used classical public-key cryptosystems, such as the RSA, depended upon the difficulty in prime factorizing integers, the Shor's algorithms meant the compromise and the collapse of almost all of today's most widely used cryptosystems. Hence, this boosted the need for unconditionally secure cryptosystems.

The idea of exploiting the properties of quantum mechanics for cryptographic purposes started off with the paper by Stephen Wiesner on conjugate coding and quantum money, which although was written in 1969, didn't get published until 1983 [25]. The first ever quantum cryptographic protocol was proposed in 1984 by Charles Bennett and Gilles Brassard [9]. This was a quantum key distribution protocol i.e. a protocol used to securely generate a secret key between two parties. This key can be later used to encrypt messages via a one time pad. The introduction of BB84 protocol sparked off the research in quantum cryptography in full swing. BB84 was then followed by the E91 protocol [26], B92 [10], Lo-Chau protocol [27], etc. Unlike classical cryptographic schemes, the QKD protocols are unconditionally secure i.e. their security is intrinsic to the protocol and is governed by the laws of quantum mechanics and does not depend upon any conditions such as computational difficulty in solving a problem. Hence, these protocols could be a very feasible replacement to the classical cryptographic schemes, which in general are vulnerable to attacks by a quantum computer.

The research in quantum cryptography was largely limited to QKD, until during 2003-2005, when three novel protocols, namely the Bostrom-Felbinger protocol [28], the LM05 protocol (see appendix), and the quantum dialogue protocol [30]. These protocols belong to the category of Quantum Direct Communication (QDC) protocols. Unlike QKD which are key-generating protocols, the QDC protocols are used to transfer the message directly and securely without the requirement of a key. These QDC protocols have shown that an unconditionally secure quantum communication can be achieved even without a key.

On the other hand, in 1993, the concept of quantum walks was introduced [31]. Quantum walks are the quantum analogues of classical random walks.

Unlike classical random walks where the walker is at just one deterministic position at a given time, in quantum walks, the walker can be at multiple positions, (i.e. a superposition of positions), at the same time. Also, the tossed coin that decides the movement of the walker, can also be at a superposition of head and tails. Adding to that, the coin and the position of the walker can also get entangled. These unique features of quantum walks can help traverse multiple positions faster, a feature which has been exploited in the design of various quantum search algorithms [32]. Quantum walks have also been used for studying and describing various physical phenomena [33, 34] and also in the study and design of quantum networks [35]. But the usage of quantum walks for the purposes of cryptography and secure communication has largely been unexplored, except for a few designs of QKD protocols [36] and public key cryptosystems [37]. In this thesis, we delve into an unexplored cryptographic potential of quantum walks: the quantum direct communication.

The paper is organized as follows: In section 2, we introduce the preliminary concepts of quantum walks required to understand the protocols proposed in section 3. In section 4, we discuss the security of the proposed protocols against various attacks. Finally, in section 5, we provide the concluding remarks. Also, in the Appendix, we provide relevant background details and codes that can be referred to if required.

2 Discrete-time quantum walk on a cycle - preliminaries

Quantum walks are a quantum analogue of the classical random walks. In discrete time quantum walks on an N -cycle, the walker moves along N discrete points on a cycle, which are realised by N dimensional quantum states $|x\rangle$ which are orthogonal to each other belong to the Hilbert space H_p where

$$H_p = span\{|x\rangle, x \in \{0, 1, 2, \dots, N - 1\}\}.$$

At every step of the quantum walk, the walker moves one position either to his left or to his right based on the result ($|0\rangle$ or $|1\rangle$) of the quantum

coin, which is given by a two dimensional quantum state $|c\rangle$ belonging to the Hilbert space H_c where

$$H_c = \text{span}\{|0\rangle, |1\rangle\}.$$

The initial state of the walker starting at position x_{in} and with an initial coin state $|c_{in}\rangle$ which can be in superposition of the two allowed basis states is given by

$$|\Psi_{in}\rangle = |x_{in}\rangle \otimes |c_{in}\rangle = |x_{in}\rangle |c_{in}\rangle \quad ; \quad |x_{in}\rangle \in H_p \quad ; \quad |c_{in}\rangle \in H_c. \quad (1)$$

The dynamics of the walker (i.e., how the walker moves during each step of the quantum walk) is governed by the action of the unitary operator, a composition of a quantum coin operation on the coin space followed by a conditioned shift operation on the complete Hilbert space,

$$U = U(\theta, \xi, \zeta) = S(I_p \otimes R_c). \quad (2)$$

Here I_p is the identity operator on position space and

$$R_c = R_c(\theta, \xi, \zeta) = \begin{bmatrix} e^{i\xi} \cos \theta & e^{i\zeta} \sin \theta \\ e^{-i\zeta} \sin \theta & e^{-i\xi} \cos \theta \end{bmatrix} \quad (3)$$

In simpler cases, when ζ and ξ are fixed, $R_c(\theta, \xi, \zeta) = R_c(\theta)$ is the coin operator on the coin space. The shift operator on $H = H_p \otimes H_c$, which shifts the position of the walker in the direction which is determined by the coin state is given by

$$S = \sum_{x=0}^{N-1} (|x-1 \pmod{N}\rangle \langle x| \otimes |0\rangle \langle 0| + |x+1 \pmod{N}\rangle \langle x| \otimes |1\rangle \langle 1|). \quad (4)$$

The state after t steps of the walk on an N -cycle in general will be in the form,

$$|\Psi_t\rangle = U^t |\Psi_{in}\rangle = \sum_{x=1}^N (\alpha_{x,t} |0\rangle + \beta_{x,t} |1\rangle) \otimes |x\rangle \quad (5)$$

and the probability of finding the walker at any position x after t steps of walk will be $P(x, t) = |\alpha_{x,t}|^2 + |\beta_{x,t}|^2$. In addition to the quantum walk evolution operator we will also define the translation operator and measurement operator which will be needed for QSCD and CQD protocols. Translation operator is defined on the space H_p in the form given by

$$T(y) = \sum_{x=0}^{N-1} |x + y \pmod{N}\rangle\langle x| \quad (6)$$

and the measurement operator M is defined on the entire space H in the form given by

$$M = M_p \otimes M_c$$

where

$$M_p = \sum_{x=0}^{N-1} |x\rangle\langle x| \quad \text{and} \quad M_c = \sum_{c=0}^1 |c\rangle\langle c|. \quad (7)$$

Note that $[T(y), U] = 0$ i.e., $T(y)$ and U commute with each other [37].

3 The Protocols

Here we first present the encoding scheme, and then present the Quantum Secure Direct Communication protocol (Fig. 1) and the Controlled Quantum Dialogue protocol (Fig. 2). In both Fig. 1 and 2, the “random path switcher” is a device that switches the path of the quantum channel so as to move a particular state into encoding the message or into checking eavesdropping, similar to using the lever to change railway tracks.

3.1 Encoding of the message

The message m (or a part m of the total message) is encoded on a quantum walk state $|\phi\rangle = \sum_i |x_i\rangle|c_i\rangle$ by applying the translation operator $T(m)$ on $|\phi\rangle$, resulting in the state $T(m) \otimes I_c |\phi\rangle = \sum_i |x_i + m\rangle|c_i\rangle$

3.2 Quantum Walk based Quantum Secure Direct Communication (QSDC) Protocol

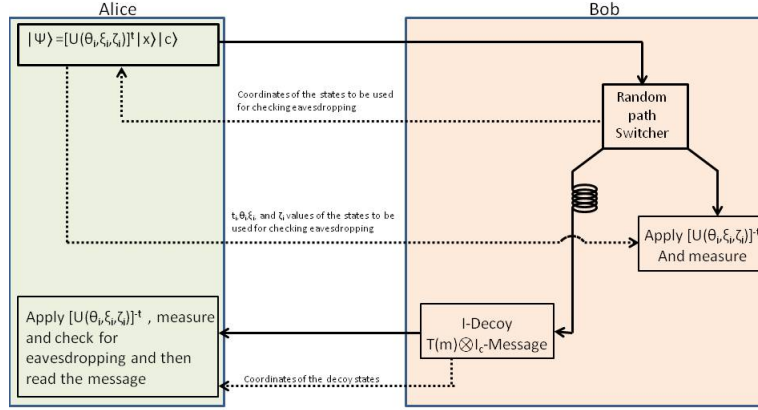


Figure 1: Quantum Walk based QSDC protocol

1. Alice randomly chooses $3n$ integers $\{t_1, t_2, \dots, t_n\}$, $\{x_1, x_2, \dots, x_n\}$ and $\{c_1, c_2, \dots, c_n\}$ such that $x_i \in \{0, 1, 2, \dots, N - 1\}$ and $c_i \in \{0, 1\} \forall i \in \{1, 2, \dots, n\}$ and $3n$ random real numbers $\{\theta_1, \theta_2, \dots, \theta_n\}$, $\{\xi_1, \xi_2, \dots, \xi_n\}$ and $\{\zeta_1, \zeta_2, \dots, \zeta_n\}$ and prepares n walk states $[U(\theta_i, \xi_i, \zeta_i)]^{t_i} |x_i\rangle |c_i\rangle = U^{t_i} |x_i\rangle |c_i\rangle \forall i \in \{1, 2, \dots, n\}$ and sends them to Bob. (In the rest of this and the next protocol, we will refer to $[U(\theta_i, \xi_i, \zeta_i)]$ as U)
2. On receiving the walk states, Bob randomly chooses $n/2$ of them for checking eavesdropping and classically sends their corresponding coordinates i to Alice. Alice classically sends to Bob the corresponding values of $t_i, x_i, c_i, \theta_i, \xi_i$ and ζ_i . Bob applies the corresponding operation U^{-t_i} on those states and measures them and checks the measurement result with the value of x_i and c_i . If the error is within a tolerable limit, he continues to step 3. Else, the protocol is aborted and they start all over again.
3. Out of the remaining $n/2$ walk states, Bob chooses $n/4$ of them for encoding the message. On each of those $n/4$ states, Bob codes a part of his message m_i by applying the translation operator $T(m_i) \otimes I_c$. He

does nothing to the other $n/4$ states (let us call them decoy states). He then sends all the $n/2$ states back to Alice.

4. Once Alice confirms the receiving of the states, Bob classically sends the coordinates of the decoy states to Alice. Alice applies the corresponding operator U^{-t_i} on the decoy states and checks for eavesdropping just like how Bob does it in step 2.
5. Once no eavesdropping is confirmed, Alice then applies U^{-t_i} on the remaining $n/4$ message states and measures them to obtain the message sent by Bob.

3.3 Quantum Walk based Controlled Quantum Dialogue (CQD) Protocol

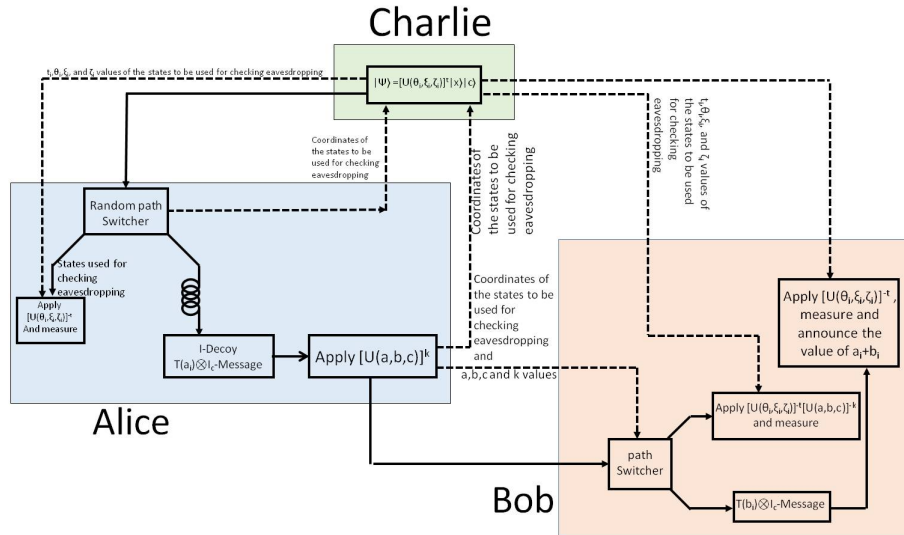


Figure 2: Quantum Walk based QDC protocol

1. Charlie randomly chooses $3n$ integers $\{t_1, t_2, \dots, t_n\}$, $\{x_1, x_2, \dots, x_n\}$ and $\{c_1, c_2, \dots, c_n\}$ such that $x_i \in \{0, 1, 2, \dots, N-1\}$ and $c_i \in \{0, 1\} \forall i \in \{1, 2, \dots, n\}$.

$\{1, 2, \dots, n\}$ and $3n$ random real numbers $\{\theta_1, \theta_2, \dots, \theta_n\}$, $\{\xi_1, \xi_2, \dots, \xi_n\}$ and $\{\zeta_1, \zeta_2, \dots, \zeta_n\}$ and prepares n walk states $[U(\theta_i, \xi_i, \zeta_i)]^{t_i}|x_i\rangle|c_i\rangle = U^{t_i}|x_i\rangle|c_i\rangle \forall i \in \{1, 2, \dots, n\}$ and sends them to Alice.

2. On receiving the walk states, Alice randomly chooses $n/2$ of them for checking eavesdropping and classically sends their corresponding coordinates i to Charlie. Charlie classically sends to Alice the corresponding values of $t_i, x_i, c_i, \theta_i, \xi_i$ and ζ_i . Alice applies the operation U^{-t_i} on those states and measures them and checks the measurement result with the value of x_i and c_i . If the error is within a tolerable limit, she continues to step 3. Else, the protocol is aborted and they start all over again.
3. Out of the remaining $n/2$ walk states, Alice chooses $n/4$ of them for encoding the message. On each of those $n/4$ states, Alice encodes a part of her message a_i by applying the translation operator $T(a_i)$. She does nothing to the other $n/4$ states (let us call them decoy states). She then chooses a random integer k and 3 random real numbers a, b, c and applies $[U(a, b, c)]^k$ on all the $n/2$ states and sends them to Bob.
4. Once Bob confirms the receiving of the states, Alice publicly announces the values of a, b, c and k and the coordinates of the decoy states. Charlie, upon receiving the announcement, sends the t_i, x_i and c_i values of the decoy states to Bob. Bob then applies the corresponding operator $U^{-t_i}[U(a, b, c)]^{-k}$ on the decoy states and checks for eavesdropping just like how Alice does it in step 2.
5. Meanwhile, Bob encodes his message b_i on the remaining message states by applying the translation operator $T(b_i)$. Once he confirms that there is no eavesdropping, Charlie sends the t_i, x_i and c_i values of the message states to Bob. Bob applies the operator U^{-t_i} on the message states, measures them and publicly announces the measurement results $a_i + b_i$. Alice and Bob subtract a_i and b_i respectively from the results to obtain each others' messages.

4 Security

In this section, we analyse the security of our protocol against various attacks, namely the intercept-resend attack, the denial of service attack, man-in-the-middle attack, and the attack by an untrusted Charlie.

4.1 Intercept-and-Resend Attack

In this attack, Eve intercepts the quantum channel and tries to extract information from the incoming state by measuring it. Then, she re-prepares the appropriate state (based on the information she receives) and sends it to the receiver. Our protocols are robust against this attack. This is due to the fact that quantum walk states are usually superposition states and that the position and the coin Hilbert spaces are usually entangled. Hence, Eve can't determine the incoming state by measurement alone. Instead of directly measuring the state, Eve can apply U^{-t_i} and then measure the state. But this attack also cannot be performed by Eve because the value of t_i will be only known to Alice at the time of attack. If Eve attempts to perform this attack, she will raise the error during the eavesdropping checking of the control mode states, and hence will be caught.

4.1.1 Mutual Information between Alice and Eve

In practical scenarios, Alice can choose her parameters $t_i, x_i, c_i, \theta_i, \xi_i$ and ζ_i only from a finite set or a finite range of values. Hence, the amount of mutual information I_{AE} gained between Alice and Eve during the intercept-resend attack is dependent upon the size of these sets and ranges. The higher the mutual information, the more will be known by Eve about the state sent by Alice, thus making the protocol less secure. Let us consider a practical scenario where Alice can choose:

- t_i from the set T containing $n(T)$ integers (from 0 to $n(T) - 1$)
- x_i from the set $X = \{0, 1, 2, \dots, N - 1\}$ (set of N values), N being the dimension of the position space
- c_i from the set $C = \{0, 1\}$ (set of 2 values)
- θ_i from the range $R_\theta = [\theta_{min}, \theta_{max}]$
- ξ_i from the range $R_\xi = [\xi_{min}, \xi_{max}]$
- ζ_i from the range $R_\zeta = [\zeta_{min}, \zeta_{max}]$

Let us say, that for a particular round of transmission, Alice chooses the values $t_A \in T$, $x_A \in X$, $c_A \in C$, $\theta_A \in R_\theta$, $\xi_A \in R_\xi$ and $\zeta_A \in R_\zeta$ and prepares the state $|\psi_A\rangle = [U(\theta_A, \xi_A, \zeta_A)]^{t_A}|x_A\rangle|c_A\rangle$. Now Eve can perform the intercept-resend attack in two ways i.e. Eve can either:

1. directly measure the incoming state to obtain the position and coin values x_E and c_E respectively (Let us call this strategy IR1) , or
2. randomly choose the values $t_E \in T$, $x_E \in X$, $c_E \in C$, $\theta_E \in R_\theta$, $\xi_E \in R_\xi$ and $\zeta_E \in R_\zeta$ and perform the operation $[U(\theta_E, \xi_E, \zeta_E)]^{-t_E}|\psi_A\rangle$ and then measure the position and coin values of the resulting state in order to obtain the values x_E and c_E respectively (let us call this strategy IR2)

Let us now examine IR2. We can consider $t_A, x_A, c_A, t_E, x_E, c_E, \theta_A, \xi_A, \zeta_A, \theta_E, \xi_E, \zeta_E$ as uniformly distributed random variables, where $t_A, x_A, c_A, t_E, x_E, c_E$ are discrete and $\theta_A, \xi_A, \zeta_A, \theta_E, \xi_E, \zeta_E$ are continuous. Now, for IR2, the mutual information I_{AE_2} between Alice and Eve is given by:

$$\begin{aligned}
I_{AE_2} = & \sum_{t_E \in T} \sum_{x_E \in X} \sum_{c_E \in C} \sum_{t_A \in T} \sum_{x_A \in X} \sum_{c_A \in C} \int_{\theta_A = \theta_{min}}^{\theta_{max}} \int_{\theta_E = \theta_{min}}^{\theta_{max}} \int_{\xi_A = \xi_{min}}^{\xi_{max}} \int_{\xi_E = \xi_{min}}^{\xi_{max}} \int_{\zeta_A = \zeta_{min}}^{\zeta_{max}} \int_{\zeta_E = \zeta_{min}}^{\zeta_{max}} \\
& p(t_A, x_A, c_A, t_E, x_E, c_E, \theta_A, \xi_A, \zeta_A, \theta_E, \xi_E, \zeta_E) \\
& \log_2 \frac{p(t_A, x_A, c_A, t_E, x_E, c_E, \theta_A, \xi_A, \zeta_A, \theta_E, \xi_E, \zeta_E)}{p(t_A)p(x_A)p(c_A)p(t_E)p(x_E)p(c_E)p(\theta_A)p(\xi_A)p(\zeta_A)p(\theta_E)p(\xi_E)p(\zeta_E)} \\
& d\theta_A d\xi_A d\zeta_A d\theta_E d\xi_E d\zeta_E \quad (8)
\end{aligned}$$

where $p(a_1, a_2, \dots, a_n)$ is the joint probability distribution-mass function of the random variables a_1, a_2, \dots, a_n where $a_i \in \{t_A, x_A, c_A, t_E, x_E, c_E, \theta_A, \xi_A, \zeta_A, \theta_E, \xi_E, \zeta_E\}$. For IR1, the mutual information I_{AE_1} between Alice and Eve is given by:

$$\begin{aligned}
I_{AE_1} = & \sum_{x_E \in X} \sum_{c_E \in C} \sum_{t_A \in T} \sum_{x_A \in X} \sum_{c_A \in C} \int_{\theta_A = \theta_{min}}^{\theta_{max}} \int_{\xi_A = \xi_{min}}^{\xi_{max}} \int_{\zeta_A = \zeta_{min}}^{\zeta_{max}} \\
& p(t_A, x_A, c_A, x_E, c_E, \theta_A, \xi_A, \zeta_A) \log_2 \frac{p(t_A, x_A, c_A, x_E, c_E, \theta_A, \xi_A, \zeta_A)}{p(t_A)p(x_A)p(c_A)p(x_E)p(c_E)p(\theta_A)p(\xi_A)p(\zeta_A)} \\
& d\theta_A d\xi_A d\zeta_A \quad (9)
\end{aligned}$$

The above formulas of I_{AE_1} and I_{AE_2} contain 3 and 6 integrals respectively. Due to lack of access to good computing power to calculate I_{AE_1} and I_{AE_2} , we modify the protocol for the purpose of analysis of this attack, by keeping the coin parameters θ, ξ and ζ constant and publicly known throughout the protocol, thus reducing the number of secret parameters and avoiding the integrals. Now, the revised formulas for I_{AE_1} and I_{AE_2} will be:

$$I_{AE_2} = \sum_{t_E \in T} \sum_{x_E \in X} \sum_{c_E \in C} \sum_{t_A \in T} \sum_{x_A \in X} \sum_{c_A \in C} p(t_A, x_A, c_A, t_E, x_E, c_E) \log_2 \frac{p(t_A, x_A, c_A, t_E, x_E, c_E)}{p(t_A)p(x_A)p(c_A)p(t_E)p(x_E)p(c_E)} \quad (10)$$

and

$$I_{AE_1} = \sum_{x_E \in X} \sum_{c_E \in C} \sum_{t_A \in T} \sum_{x_A \in X} \sum_{c_A \in C} p(t_A, x_A, c_A, x_E, c_E) \log_2 \frac{p(t_A, x_A, c_A, x_E, c_E)}{p(t_A)p(x_A)p(c_A)p(x_E)p(c_E)} \quad (11)$$

where:

$$p(t_A, x_A, c_A, t_E, x_E, c_E) = \frac{1}{2N[n(T)]^2} (\langle x_E | \langle c_E | U^{-t_E} U^{t_A} | x_A \rangle | c_A \rangle)^2 \quad (12)$$

and

$$p(t_A, x_A, c_A, x_E, c_E) = \frac{1}{2N[n(T)]} (\langle x_E | \langle c_E | U^{t_A} | x_A \rangle | c_A \rangle)^2 \quad (13)$$

and

$$p(a_i) = \sum_{a_1, a_2, \dots, a_{i-1}, a_{i+1}, \dots, a_n} p(a_1, a_2, \dots, a_n) \quad (14)$$

where $a_j \in \{t_A, x_A, c_A, t_E, x_E, c_E\}$ and $U = U(\theta, \xi, \zeta)$ where $\theta, \xi,$ and ζ are the publicly known coin parameters constant throughout the protocol.

We can see that I_{AE_1} and I_{AE_2} are a function of $n(T)$ and N , and also depend on the fixed coin parameters θ, ξ and ζ .

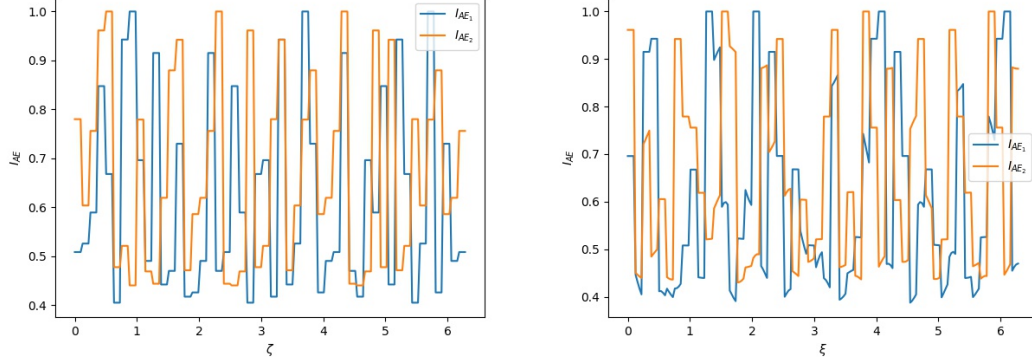


Figure 3: (a) I_{AE} vs ζ for $N = 3, n(T) = 7, \theta = \frac{\pi}{4}, \xi = \frac{\pi}{4}$ and (b) I_{AE} vs ξ for $N = 3, n(T) = 7, \theta = \frac{\pi}{4}, \zeta = \frac{\pi}{4}$

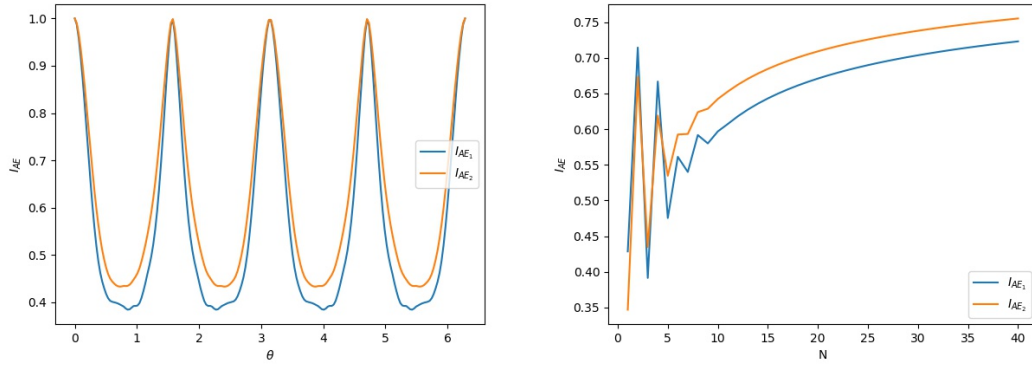


Figure 4: (a) I_{AE} vs θ for $N = 3, n(T) = 7, \zeta = \frac{\pi}{4}, \xi = \frac{\pi}{4}$ and (b) I_{AE} vs N for $n(T) = 7, \theta = \frac{\pi}{4}, \zeta = \frac{\pi}{4}, \xi = \frac{\pi}{4}$

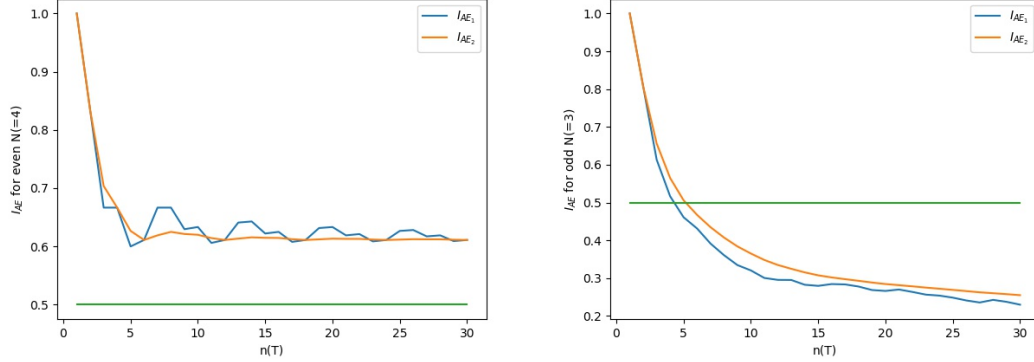


Figure 5: (a) I_{AE} vs $n(T)$ for $N = 4, \theta = \frac{\pi}{4}, \zeta = \frac{\pi}{4}, \xi = \frac{\pi}{4}$ and (b) $n(T)$ for $N = 3, \theta = \frac{\pi}{4}, \zeta = \frac{\pi}{4}, \xi = \frac{\pi}{4}$. Green line represents the I_{AE} for one channel of the LM05 protocol, which is the same as the I_{AE} for the BB84 protocol.

We can see that in Figure 3, I_{AE} (by I_{AE} , we are referring to both I_{AE_1} and I_{AE_2} at the same time) is low for some values of ζ and ξ and high for some values of ζ and ξ . Hence, we must choose and fix the appropriate value of ζ and ξ depending upon the values of $n(T)$, N and θ , such that the mutual information is at its lowest. In Figure 4(a), we can see that I_{AE} is at its lowest when θ is an odd multiple of $\frac{\pi}{4}$ and is at its highest ($I_{AE} = 1$) when θ is an even multiple of $\frac{\pi}{4}$, hence, for θ equal to even multiples of $\frac{\pi}{4}$, the security of the protocol will be compromised. In Figure 4(b), we can see that for odd N , I_{AE} increases with increase in N , whereas for even N , I_{AE} initially decreases with N , but then increases. From Figure 5(b) and Figure 4, we can see that $I_{AE_2} > I_{AE_1}$, implying that that IR2 is a better strategy for Eve than IR1 for odd N . In Figure 5, we see that I_{AE} decreases with $n(T)$ and its value is greater for even N than for odd N . In fact, for odd N , the I_{AE} drops much below 0.5 (which is the I_{AE} for the LM05 protocol (see appendix)) for large $n(T)$, and in fact is less than 0.25 for $n(T) > 25$. This shows that, for an odd, low value of N , and a high value of $n(T)$, and θ being an odd multiple of $\frac{\pi}{4}$ and for appropriate values of ξ and ζ , our quantum walk protocols are more secure against the intercept-resend attack than the LM05 protocol (whose $I_{AE} = 0.5$), even with the modification that the coin parameters remain constant and publicly known throughout the protocol.

4.2 Denial of Service attack

Instead of trying to extract information from the incoming state, Eve can rather perform a denial-of-service attack i.e. she can just stop the incoming state from going forward and can instead prepare and send a random quantum walk state. This attack also cannot be performed by Eve because if she does so, she introduces an added error and noise in the channel and hence the eavesdropping checking performed by the sender and the receiver at each quantum channel will detect Eve.

4.3 Man-in-the-middle attack

Let's consider the QSDC protocol. In this attack, Eve initially puts the incoming state from Alice into her quantum memory. Then, she sends her own walk state to Bob. Bob, assuming that Alice may have sent this state, encodes his message on this state and sends it back to Alice. Eve intercepts that channel also and reads the message. She then encodes the message onto the Alice's state which she had earlier stored in her quantum memory and sends it back to Alice, thus being able to read the message. Eve can perform a similar kind of attack in the CQD protocol to obtain the message of one of the two communicating parties. In both cases of this attack, Eve will be detected by the communicating parties during eavesdropping checking. Hence both our protocols are unconditionally secure against this attack.

4.4 Attack by an untrusted Charlie

Let us consider the QDC protocol. In this attack, Charlie intercepts the channel, applies U^{-t_i} and obtains Alice's message by measuring the state. Then, he re-prepares the state and sends it to Bob. Then when Bob encodes his message b_i and announces the value $a_i + b_i$, Charlie can then get Bob's message as well. But our QDC protocol is robust against this attack because as Alice applies an additional U^k to the states, Charlie will not know the value of a, b, c or k and hence he cannot apply $[U(a, b, c)]^{-k}$ to retrieve the state.

5 Conclusion

Motivated by the unique properties of quantum walks (such as superposition of positions and entanglement between the position and the coin states [38]) and their potential for providing cryptographic security, a one-way two-party Quantum Secure Direct Communication (QSDC) protocol and a two-way three-party Controlled Quantum Dialogue Protocol (CQD) have been designed using quantum walks. It has been shown that the proposed protocols are unconditionally secure against various attacks, such as the intercept-resend attack, the denial of service attack and the man-in-the-middle attack. The CQD protocol, in particular, is shown to be secure against an attack by an untrusted Charlie. Also, for the intercept-resend attack, the mutual information gained between Alice and Eve is shown to be much lower for the proposed protocols as compared to the qubit based protocols such as the LM05 protocol [29], thus making the proposed protocols more secure than LM05 against this attack. Also, unlike the qubit based protocols which transfer just one bit per state, the proposed protocols can transfer multiple bits per state, which can possibly lead to advantages such as faster transmission of messages and a lower requirement of resources (both subject to practical/experimental conditions). These direct communication schemes could potentially lead to secure feasible solutions for many social and economic problems such as the socialist millionaire problem [39], quantum E-commerce [40], quantum voting [41], etc. and the work towards finding these potential solutions will be attempted in the future.

Appendix

LM05 Protocol

This qubit based protocol was introduced in 2005 [29]. In this protocol, the encoding rules for the message sender is as follows:

To encode the bit 0, do nothing to the incoming qubit.

To encode the bit 1, apply the operator $iY = ZX$ on the incoming qubit. The transformations are as follows:

$$iY|0\rangle = -|1\rangle$$

$$iY|1\rangle = |0\rangle$$

$$iY|\pm\rangle = \pm|\mp\rangle$$

The protocol is as follows:

1. Alice chooses n random qubits from the set $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$ and sends them to Bob.
2. Out of these n qubits received from Alice, Bob randomly chooses $n/2$ of them and classically sends their coordinates to Alice.
3. Alice publicly announces the states of the $n/2$ qubits which Bob chose in step 2. Bob measures each of the $n/2$ qubits in their corresponding bases and checks for eavesdropping. If the error is within a tolerable limit, then the protocol continues to step 4. Else, the protocol is discarded and they start all over again.
4. Among the remaining $n/2$ qubits, Bob randomly chooses $n/4$ of them and encodes the message in them according to the encoding rules above and does nothing to the remaining $n/4$ qubits. He sends all these $n/2$ qubits back to Alice.
5. After Alice confirms receiving of the $n/2$ qubits, Bob sends the coordinates of the qubits on which he didn't encode the message. Alice uses these qubits to check for eavesdropping just like how Bob does it in step 3.
6. After confirming no eavesdropping, Alice measures the remaining qubits in their respective bases to obtain the message sent by Bob.

Mutual Information

Let us take two random variables, say x and y . The mutual information I_{XY} between two random variables x and y is the decrease in uncertainty of one random variable when the value of the other random variable is observed, measured or determined. If x and y are discrete, the formula for I_{XY} is given by [42]:

$$I_{XY} = \sum_x \sum_y p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)} \quad (15)$$

Where $p(x, y)$ is the joint probability mass function and $p(x)$ and $p(y)$ are the individual probability mass functions.

If x and y are continuous, then the formula for I_{XY} is given by:

$$I_{XY} = \int_x \int_y p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)} dx dy \quad (16)$$

Where $p(x, y)$ is the joint probability density function and $p(x)$ and $p(y)$ are the individual probability density functions.

There can also be a case where one of the random variables is discrete and the other is continuous. For example, if x is discrete and y is continuous, then the formula for I_{XY} becomes:

$$I_{XY} = \sum_x \int_y p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)} dy \quad (17)$$

where $p(x)$ is the probability mass function of x , $p(y)$ is the probability density function of y and $p(x, y)$ is a function that is a probability density-mass function that is discrete in x and continuous in y .

This concept of mutual information can also be generalized to $r = mn > 2$ random variables $\{x_1, x_2, \dots, x_m\}$ and $\{y_1, y_2, \dots, y_n\}$ where x_i are discrete and y_i are continuous. The generalised mutual information I_{mutual} is given by [42]:

$$I_{mutual} = \sum_{x_1, x_2, \dots, x_m, y_1, \dots, y_n} \int p(x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_n) \log_2 \frac{p(x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_n)}{p(x_1)p(x_2)\dots p(x_m)p(y_1)p(y_2)\dots p(y_n)} dy_1 dy_2 \dots dy_n \quad (18)$$

Mutual Information for the intercept-resend attack for the LM05 protocol:

Let us consider the first transmission from Alice to Bob. In this transmission, Alice first selects either of the four states and prepares them and sends them to Bob. Eve intercepts this channel before the state reaches Bob and randomly chooses a basis for each incoming state and measures the state in that basis. Let $a, e \in \{0, 1, +, -\}$. Let the probability of Alice sending the qubit a and Eve receiving the qubit e be $p(a, e)$. For example, the probability $p(0, 0)$ is:

$$p(0, 0) = \begin{array}{c} \text{probability of Alice choosing } 0 \\ \frac{1}{4} \\ \times \\ \text{probability of Eve choosing the computational } Z \text{ basis} \\ \frac{1}{2} \\ \times \\ \text{probability of Eve getting } 0 \\ 1 \end{array} = \frac{1}{8} \quad (19)$$

Similarly,

$$p(0, 1) = \frac{1}{4} \times \frac{1}{2} \times 0 = 0 \quad (20)$$

$$p(0, +) = \frac{1}{4} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{16} \quad (21)$$

$$p(0, -) = \frac{1}{4} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{16} \quad (22)$$

And similar probabilities for $p(1, e)$, $p(+, e)$, and $p(-, e)$, where $e \in \{0, 1, +, -\}$. Hence, the mutual information I_{AE} for the LM05 protocol is given by:

$$I_{AE} = \sum_a \sum_e p(a, e) \log_2 \frac{p(a, e)}{p(a)p(e)} \quad (23)$$

$$= 4 \left(\frac{1}{8} \log_2 \frac{\frac{1}{8}}{\frac{1}{4} \frac{1}{2}} + \frac{1}{16} \log_2 \frac{\frac{1}{16}}{\frac{1}{4} \frac{1}{4}} + \frac{1}{16} \log_2 \frac{\frac{1}{16}}{\frac{1}{4} \frac{1}{4}} \right) = 0.5$$

(We can see that for all a and e , $p(a) = p(e) = \frac{1}{4}$. Hence, $p(a)p(e) = \frac{1}{16}$)

Codes used to plot the graphs

Calculation of I_{AE_1}

Listing 1: stuff6.py

```
import numpy as np
import scipy as sp
import matplotlib as mpl
import matplotlib.pyplot as plt
import math as m
import copy as cp
from scipy import integrate
import multiprocessing as mp

pie=np.pi
thet=pie/4
Ee=pie/4
Cc=pie/4

def walkstate(N, pos , coin ):
    if coin==0:
        c = [[1] , [0]]
    if coin==1:
        c = [[0] , [1]]
    wsi=np.matrix(np.kron(np.matrix(np.eye(N))[: , pos] , np.matrix(c)))
    return wsi

def Qwalk(N, wsi , theta , E, C, t ):
    wsi=np.matrix(wsi)
    p=np.matrix(np.eye(N))
    c0=np.matrix([[1] , [0]])
    c1=np.matrix([[0] , [1]])
    S=np.kron(np.matrix(np.zeros((N,N))) , np.matrix(np.zeros((2 , 2))))
    for k in range(N):
        forward=k+1
        backward=k-1
        if k==0:
```

```

        backward=N-1
    if k==N-1:
        forward=0
        S=S+np.kron(np.matmul(p[:, forward], p[:, k].getH()), np.matmul(c1, c1.getH()))
+np.kron(np.matmul(p[:, backward], p[:, k].getH()), np.matmul(c0, c0.getH()))
        Ip=np.matrix(np.eye(N))
        Uc=np.matrix([[np.exp(1.j*E)*np.cos(theta), np.exp(1.j*C)*np.sin(theta)],
[-np.exp(-1.j*C)*np.sin(theta), np.exp(-1.j*E)*np.cos(theta)]]])
        U=np.matmul(S, np.kron(Ip, Uc))
        Ut=np.linalg.matrix_power(U, t)
        ws=np.matmul(Ut, wsi)
    return ws

```

```

def Pall(xa, ca, ta, xe, ce, nt, N):
    wsi=walkstate(N, xa, ca)
    ws1=Qwalk(N, wsi, thet, Ee, Cc, ta)
    prob=(abs(np.matmul(walkstate(N, xe, ce).getH(), ws1))**2)/(2*N*nt)
    return prob.tolist()[0][0]

```

```

def Iae(nt, N):
    summ=0
    #summ2=0
    for ta in range(nt):
        for xa in range(N):
            for xe in range(N):
                for ca in range(2):
                    for ce in range(2):
                        pall=Pall(xa, ca, ta, xe, ce, nt, N)
                        if pall==0:
                            inf=0
                        else:
                            inf=pall*np.log2(pall*4*N*N*nt)
                        summ+=inf
                        #summ2+=pall
    return summ/(np.log2(N)+1)#, summ2

```


Calculation of I_{AE_2}

Listing 2: stuff4.py

```
import numpy as np
import scipy as sp
import matplotlib as mpl
import matplotlib.pyplot as plt
import math as m
import copy as cp
from scipy import integrate
import multiprocessing as mp

pie=np.pi
thet=pie/4
Ee=pie/4
Cc=pie/4

def walkstate(N, pos , coin ):
    if coin==0:
        c=[[1],[0]]
    if coin==1:
        c=[[0],[1]]
    wsi=np.matrix(np.kron(np.matrix(np.eye(N))[: , pos ] , np.matrix(c)))
    return wsi

def Qwalk(N, wsi , theta ,E,C, t ):
    wsi=np.matrix(wsi)
    p=np.matrix(np.eye(N))
    c0=np.matrix([[1],[0]])
    c1=np.matrix([[0],[1]])
    S=np.kron(np.matrix(np.zeros((N,N))), np.matrix(np.zeros((2,2))))
    for k in range(N):
        forward=k+1
        backward=k-1
        if k==0:
            backward=N-1
        if k==N-1:
```

```

        forward=0
        S=S+np.kron(np.matmul(p[:, forward], p[:, k].getH()), np.matmul(c1, c1.getH()))
+np.kron(np.matmul(p[:, backward], p[:, k].getH()), np.matmul(c0, c0.getH()))
        Ip=np.matrix(np.eye(N))
        Uc=np.matrix([[np.exp(1.j*E)*np.cos(theta), np.exp(1.j*C)*np.sin(theta)],
[-np.exp(-1.j*C)*np.sin(theta), np.exp(-1.j*E)*np.cos(theta)]])
        U=np.matmul(S, np.kron(Ip, Uc))
        Ut=np.linalg.matrix_power(U, t)
        ws=np.matmul(Ut, wsi)
        return ws

def Pall(ta, xa, ca, te, xe, ce, nt, N):
    wsi=walkstate(N, xa, ca)
    ws1=Qwalk(N, wsi, thet, Ee, Cc, ta)
    ws2=Qwalk(N, ws1, thet, Ee, Cc, -te)
    prob=(abs(np.matmul(walkstate(N, xe, ce).getH(), ws2))**2)/((2*N)*((nt)**2))
    return prob.tolist()[0][0]

def Iae(nt, N):
    summ=0
    #summ2=0
    for ta in range(nt):
        for te in range(nt):
            for xa in range(N):
                for xe in range(N):
                    for ca in range(2):
                        for ce in range(2):
                            pall=Pall(ta, xa, ca, te, xe, ce, nt, N)
                            if pall==0:
                                inf=0
                            else:
                                inf=pall*np.log2(pall*4*N*N*nt*nt)
                            summ+=inf
                            #summ2+=pall
    return summ/(np.log2(N)+1)#, summ2

```

Generating the plots

Listing 3: plots.py

```
import stuff4 as s4
import stuff6 as s6
import matplotlib as mpl
from matplotlib import pyplot as plt
import multiprocessing as mp
import numpy as np
from time import time

t1=time()

pie=np.pi
nTset=np.linspace(1,30,30).astype(int).tolist()
Nset=np.linspace(1,40,40).astype(int).tolist()
thetaset=np.linspace(0,2*np.pi,200).tolist()
Eset=np.linspace(0,2*np.pi,200).tolist()
Cset=np.linspace(0,2*np.pi,200).tolist()

nTf=7
Nodd=3
Neven=4
thetaf=pie/4
Ef=pie/4
Cf=pie/4

def s4iaevsntnodd(nt):
    return s4.Iae(nt,Nodd)

def s6iaevsntnodd(nt):
    return s6.Iae(nt,Nodd)

def s4iaevsntneven(nt):
    return s4.Iae(nt,Neven)

def s6iaevsntneven(nt):
```

```

        return s6.Iae(nt, Neven)

def s4iaevsn(N):
    return s4.Iae(nTf, N)

def s6iaevsn(N):
    return s6.Iae(nTf, N)

def s4iaevsttheta(theta):
    s4.thet=theta
    return s4.Iae(nTf, Nodd)

def s6iaevsttheta(theta):
    s6.thet=theta
    return s6.Iae(nTf, Nodd)

def s4iaevsE(E):
    s4.Ee=E
    return s4.Iae(nTf, Nodd)

def s6iaevsE(E):
    s6.Ee=E
    return s6.Iae(nTf, Nodd)

def s4iaevsC(C):
    s4.Cc=C
    return s4.Iae(nTf, Nodd)

def s6iaevsC(C):
    s6.Cc=C
    return s6.Iae(nTf, Nodd)

pool=mp.Pool(mp.cpu_count())

s4iaevsntnodd_arr=pool.starmap(s4iaevsntnodd, zip(nTset))
s6iaevsntnodd_arr=pool.starmap(s6iaevsntnodd, zip(nTset))
s4iaevsntneven_arr=pool.starmap(s4iaevsntneven, zip(nTset))

```

```

s6iaevsntneven_arr=pool.starmap(s6iaevsntneven , zip(nTset))
s4iaevsn_arr=pool.starmap(s4iaevsn , zip(Nset))
s6iaevsn_arr=pool.starmap(s6iaevsn , zip(Nset))
s4iaevstheta_arr=pool.starmap(s4iaevstheta , zip(thetaset))
s6iaevstheta_arr=pool.starmap(s6iaevstheta , zip(thetaset))
s4iaevsE_arr=pool.starmap(s4iaevsE , zip(Eset))
s6iaevsE_arr=pool.starmap(s6iaevsE , zip(Eset))
s4iaevsC_arr=pool.starmap(s4iaevsC , zip(Cset))
s6iaevsC_arr=pool.starmap(s6iaevsC , zip(Cset))

```

```
pool.close()
```

```

def filewrite(n,l):
    with open(n, 'w') as filehandle:
        for listitem in l:
            filehandle.write('%s\n' % listitem)

```

```

def fileread(n,dtype):
    places = []
    with open(n, 'r') as filehandle:
        for line in filehandle:
            currentPlace = line[:-1]
            if dtype == 'int':
                currentPlace=int(currentPlace)
            elif dtype=='float':
                currentPlace=float(currentPlace)
            places.append(currentPlace)
    return places

```

```

filewrite('s4iaevsntnodd_arr.txt',s4iaevsntnodd_arr)
filewrite('s6iaevsntnodd_arr.txt',s6iaevsntnodd_arr)
filewrite('s4iaevsntneven_arr.txt',s4iaevsntneven_arr)
filewrite('s6iaevsntneven_arr.txt',s6iaevsntneven_arr)
filewrite('s4iaevsn_arr.txt',s4iaevsn_arr)
filewrite('s6iaevsn_arr.txt',s6iaevsn_arr)
filewrite('s4iaevstheta_arr.txt',s4iaevstheta_arr)
filewrite('s6iaevstheta_arr.txt',s6iaevstheta_arr)
filewrite('s4iaevsE_arr.txt',s4iaevsE_arr)

```

```

filewrite('s6iaevsE_arr.txt',s6iaevsE_arr)
filewrite('s4iaevsC_arr.txt',s4iaevsC_arr)
filewrite('s6iaevsC_arr.txt',s6iaevsC_arr)

```

```

plt.plot(nTset,s6iaevsntnodd_arr)
plt.plot(nTset,s4iaevsntnodd_arr)
plt.plot(nTset,[0.5]*len(nTset))
plt.xlabel(r'n(T)')
plt.ylabel(r'$I_{AE}$ for odd N(='+str(Nodd)+'')')
plt.legend([r'$I_{AE_1}$',r'$I_{AE_2}$'])
plt.savefig('iaevsntnodd.jpg')
plt.show()
plt.figure()

```

```

plt.plot(nTset,s6iaevsntneven_arr)
plt.plot(nTset,s4iaevsntneven_arr)
plt.plot(nTset,[0.5]*len(nTset))
plt.xlabel(r'n(T)')
plt.ylabel(r'$I_{AE}$ for even N(='+str(Neven)+'')')
plt.legend([r'$I_{AE_1}$',r'$I_{AE_2}$'])
plt.savefig('iaevsntneven.jpg')
plt.show()
plt.figure()

```

```

plt.plot(Nset,s6iaevsn_arr)
plt.plot(Nset,s4iaevsn_arr)
plt.xlabel(r'N')
plt.ylabel(r'$I_{AE}$')
plt.legend([r'$I_{AE_1}$',r'$I_{AE_2}$'])
plt.savefig('iaevsn.jpg')
plt.show()
plt.figure()

```

```

plt.plot(thetaset,s6iaevstheta_arr)
plt.plot(thetaset,s4iaevstheta_arr)
plt.xlabel(r'$\theta$')
plt.ylabel(r'$I_{AE}$')

```

```

plt.legend([r'$I_{AE_1}$', r'$I_{AE_2}$'])
plt.savefig('iaevsttheta.jpg')
plt.show()
plt.figure()

```

```

plt.plot(Eset, s6iaevsE_arr)
plt.plot(Eset, s4iaevsE_arr)
plt.xlabel(r'$\xi$')
plt.ylabel(r'$I_{AE}$')
plt.legend([r'$I_{AE_1}$', r'$I_{AE_2}$'])
plt.savefig('iaevsE.jpg')
plt.show()
plt.figure()

```

```

plt.plot(Cset, s6iaevsC_arr)
plt.plot(Cset, s4iaevsC_arr)
plt.xlabel(r'$\zeta$')
plt.ylabel(r'$I_{AE}$')
plt.legend([r'$I_{AE_1}$', r'$I_{AE_2}$'])
plt.savefig('iaevsC.jpg')
plt.show()

```

```
t2=time()
```

```
print('Done. Time taken:', t2-t1)
```

References

- [1] Thapliyal, Kishore, and Anirban Pathak. "Quantum e-commerce: a comparative study of possible protocols for online shopping and other tasks related to e-commerce." *Quantum Information Processing* 18.6 (2019): 191.
- [2] Nguyen, Ba An. "Quantum dialogue." *Physics Letters A* 328.1 (2004): 6-10.
- [3] Thapliyal, Kishore, and Anirban Pathak. "Applications of quantum cryptographic switch: various tasks related to controlled quantum communication can be performed using Bell states and permutation of particles." *Quantum Information Processing* 14.7 (2015): 2599-2616.
- [4] Ince, Robin AA, Stefano Panzeri, and Simon R. Schultz. "Summary of information theoretic quantities." *arXiv preprint arXiv:1501.01854* (2015).
- [5] Lucamarini, Marco, and Stefano Mancini. "Secure deterministic communication without entanglement." *Physical review letters* 94.14 (2005): 140501.
- [6] Pathak, Anirban. *Elements of quantum computation and quantum communication*. CRC Press, 2013.
- [7] Vlachou, Chrysoula, et al. "Quantum key distribution with quantum walks." *Quantum Information Processing* 17.11 (2018): 288.
- [8] Vlachou, Chrysoula, et al. "Quantum walk public-key cryptographic system." *International Journal of Quantum Information* 13.07 (2015): 1550050.
- [9] Bennett, Charles H., and Gilles Brassard. "Quantum cryptography: Public key distribution and coin tossing." *arXiv preprint arXiv:2003.06557* (2020).
- [10] Bennett, Charles H. "Quantum cryptography using any two nonorthogonal states." *Physical review letters* 68.21 (1992): 3121.
- [11] I Ashchenko, V.V. (2002). *Cryptography: an introduction*. AMS Bookstore. p. 6. ISBN 978-0-8218-2986-8.

- [12] electricpulp.com. "CODES – Encyclopaedia Iranica". www.iranicaonline.org.
- [13] Al-Jubouri, I. M. N. (19 March 2018). *History of Islamic Philosophy: With View of Greek Philosophy and Early History of Islam*. Authors On Line Ltd. ISBN 9780755210114.
- [14] Singh, Simon (2000). *The Code Book*. New York: Anchor Books. pp. 14–20. ISBN 978-0-385-49532-5.
- [15] Hakim, Joy (1995). *A History of US: War, Peace and all that Jazz*. New York: Oxford University Press. ISBN 978-0-19-509514-2
- [16] "FIPS PUB 197: The official Advanced Encryption Standard" (PDF). Computer Security Resource Center. National Institute of Standards and Technology. Archived from the original (PDF) on 7 April 2015. Retrieved 26 March 2015.
- [17] Rivest, Ronald L.; Shamir, A.; Adleman, L. (1978). "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". *Communications of the ACM*. 21 (2)
- [18] NIST, Special Publication 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, March, 2006.
- [19] Taher ElGamal (1985). "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms" (PDF). *IEEE Transactions on Information Theory*. 31 (4): 469–472.
- [20] Benioff, Paul (1980). "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines". *Journal of Statistical Physics*. 22 (5): 563–591.
- [21] Manin, Yu. I. (1980). Vychislimoe i nevychislimoe [Computable and Noncomputable] (in Russian). *Sov.Radio*. pp. 13–15. Archived from the original on 2013-05-10. Retrieved 2013-03-04.
- [22] Feynman, Richard (June 1982). "Simulating Physics with Computers" (PDF). *International Journal of Theoretical Physics*. 21 (6/7): 467–488. Bibcode:1982IJTP...21..467F. doi:10.1007/BF02650179. Retrieved 28 February 2019.

- [23] Montanaro, Ashley. "Quantum algorithms: an overview." *npj Quantum Information* 2.1 (2016): 1-8.
- [24] Shor, P.W. (1994). "Algorithms for quantum computation: discrete logarithms and factoring". *Proceedings 35th Annual Symposium on Foundations of Computer Science*. IEEE Comput. Soc. Press: 124–134. doi:10.1109/sfcs.1994.365700. ISBN 0818665807.
- [25] Wiesner, Stephen. "Conjugate coding." *ACM Sigact News* 15.1 (1983): 78-88.
- [26] Ekert, Artur K. (5 August 1991). "Quantum cryptography based on Bell's theorem". *Physical Review Letters*. 67 (6): 661–663.
- [27] Lo, Hoi-Kwong, and Hoi Fung Chau. "Is quantum bit commitment really possible?." *Physical Review Letters* 78.17 (1997): 3410.
- [28] Boström, Kim, and Timo Felbinger. "Deterministic secure direct communication using entanglement." *Physical Review Letters* 89.18 (2002): 187902.
- [29] Lucamarini, Marco, and Stefano Mancini. "Secure deterministic communication without entanglement." *Physical review letters* 94.14 (2005): 140501.
- [30] Nguyen, Ba An. "Quantum dialogue." *Physics Letters A* 328.1 (2004): 6-10.
- [31] Aharonov, Y., Davidovich, L., and Zagury, N., 1993, *Phys. Rev. A*, 48, 1687
- [32] Santha, Miklos. "Quantum walk based search algorithms." *International Conference on Theory and Applications of Models of Computation*. Springer, Berlin, Heidelberg, 2008.
- [33] Chawla, Prateek, C. V. Ambarish, and C. M. Chandrashekar. "Quantum percolation in quasicrystals using continuous-time quantum walk." *Journal of Physics Communications* 3.12 (2019): 125004.
- [34] Badhani, Himanshu, and C. M. Chandrashekar. "Gravitationally induced entanglement between two quantum walkers." *arXiv preprint arXiv:1907.06953* (2019).

- [35] Yang, Yuguang, et al. "Quantum network communication: a discrete-time quantum-walk approach." *Science China Information Sciences* 61.4 (2018): 042501.
- [36] Vlachou, Chrysoula, et al. "Quantum key distribution with quantum walks." *Quantum Information Processing* 17.11 (2018): 288.
- [37] Vlachou, Chrysoula, et al. "Quantum walk public-key cryptographic system." *International Journal of Quantum Information* 13.07 (2015): 1550050.
- [38] Kempe, Julia. "Quantum random walks: an introductory overview." *Contemporary Physics* 44.4 (2003): 307-327.
- [39] Saxena, Ashwin, Kishore Thapliyal, and Anirban Pathak. "Continuous variable controlled quantum dialogue and secure multiparty quantum computation." arXiv preprint arXiv:1902.00458 (2019).
- [40] Thapliyal, Kishore, and Anirban Pathak. "Quantum e-commerce: a comparative study of possible protocols for online shopping and other tasks related to e-commerce." *Quantum Information Processing* 18.6 (2019): 191.
- [41] Thapliyal, Kishore, Rishi Dutt Sharma, and Anirban Pathak. "Protocols for quantum binary voting." *International Journal of Quantum Information* 15.01 (2017): 1750007.
- [42] Ince, Robin AA, Stefano Panzeri, and Simon R. Schultz. "Summary of information theoretic quantities." arXiv preprint arXiv:1501.01854 (2015).