High Entropy Quantum Communication Framework for Secure Key Distribution and Secure Messaging

Rohit De
Del Norte High School
San Diego, California 92127, USA
Email: de.rohit01@gmail.com

***Abstract*—This work explores quantum computing and quantum communication with a focus on cybersecurity. A high entropy quantum communication framework is set up for secure Quantum Key Distribution (QKD) and secure short messaging, based on the Deutsch-Jozsa (DJ) algorithm. QKD allows Alice and Bob to securely share a secret key and improves over the Public Key Infrastructure (PKI), which can become vulnerable as quantum computing matures. However, QKD itself can be compromised by sophisticated Man In The Middle (MITM) quantum attacks, such as intercept-resend and quantum cloning. Recent research on QKD improved the entropy by reordering the qubits within a DJ-packet and by hopping to a different size for each run of the DJ-algorithm. This paper further increases the entropy by additionally using multiple orthogonal bases for the different qubits in a DJ-packet, called the HRB (Hopping Reorder Basis) scheme. Furthermore, the HRB scheme does not require any pre-sharing to establish the protocol. Functionality of the HRB scheme is tested on Google's Cirq quantum simulator. Simulations show that an attacker's interception success drops 200-times in the HRB scheme when using two orthogonal bases vs. 12-times in the previous work. When three orthogonal bases are used, the attacker's interception success drops more than 1000-times, improving the secrecy of the communication.**

***Keywords*—PKI; QKD; Deutsch-Jozsa; MITM; Qubit.**

I. INTRODUCTION

Quantum technology has a great potential to advance computing and communication. While it can help strengthen internet security through means like Quantum Key Distribution (QKD) [6] [8], its computation power can also be exploited to break classical security schemes such as Public Key Infrastructure (PKI) [4]. In future, resourceful quantum computers utilizing Shor's algorithm can make asymmetric encryption algorithms like RSA [11], which is used in PKI, vulnerable to attacks. This puts sensitive information such as bank transactions, login credentials, and any encrypted communications at risk. To overcome this threat, QKD supports next generation key distribution when quantum networks and quantum computers become prevalent [4]. QKD is used for generating and sharing a secret key between two parties, Alice and Bob, using quantum mechanical properties of qubits. The secret key is then used to set up an encrypted data communication channel between them, as shown in Figure 1.

In quantum technology, information is encoded in elements called qubits. Qubits can exist as superposition of two states but can collapse to either $zero$ or one (i.e., $|0\rangle$ or $|1\rangle$) states when measured or copied. This is called the 'no-cloning' property [12] and is utilized in most of the QKD methods. The 'no-cloning' property lets the receiver, Alice, detect

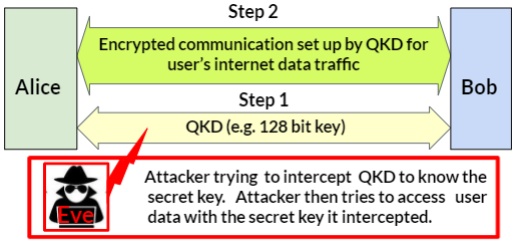


Figure 1. Attack on the QKD step to intercept the shared secret key.

an eavesdropper or a Man-In-The-Middle (MITM) attacker, Eve. This is unlike classical communication system where an eavesdropper can stealthily read, copy and store the bits transmitted, and then do offline brute force analysis. While the no-cloning property benefits QKD, it may still be possible for a very resourceful attacker to timely replace the collapsed qubits with fresh initialized qubits [13], e.g., initialization to $|0\rangle$ or $|1\rangle$ followed by superposition to replace the collapsed qubits. This threat and other attacks, like intercept/resend (faked-state) and quantum cloning [14] [15], can compromise QKD. This paper provides a high entropy quantum communication framework based on the Deustch-Jozsa (DJ) algorithm [1] by leveraging the original work by Nagata and Nakamura [2], and recent work by De et al. [3]. The DJ-algorithm allowed easy addition of new methods to increase entropy. The specific way the DJ-algorithm is leveraged illustrates an unique integration of quantum computing and quantum communication. This work also serves as a case-study on employing quantum technology for security and privacy.

This paper is organized as follows. Section II gives a brief survey of some of the QKD approaches and the previous research using the DJ-algorithm. Section III describes the new HRB mechanism and its entropy improvements, which decreases Eve's chance of successful interception versus previous research. Section IV shows the simulation results of the HRB scheme. Section V describes the end-to-end communication framework based on the HRB scheme, and how it can be used not only for secure QKD but also for secure short messages directly on a quantum communication channel. Finally, Section VI concludes the paper.

II. REVIEW OF LITERATURE

In BB84 [6] [7] QKD protocol, the qubits use two conjugate pairs of states, where the two states within each pair use orthogonal basis (the states of 0° and 90° form the rectilinear basis, while the states of 45° and 135° form the diagonal

 ISBN: 978-1-61208-961-4

basis). Alice creates a random bit (0 or 1) and randomly selects one of the two bases, rectilinear or diagonal, to transmit photons to Bob. Bob does not know the specific basis Alice picked; he also randomly selects either the rectilinear or the diagonal basis, and uses it to measure the photons he receives. When Alice and Bob share the bases they each used for each of the photons, they discard the photons for which they used mismatched bases. An interception by an eavesdropper would introduce errors due to the no-cloning property of qubits. The Six-State protocol [9] is the version of BB84 using a six-state polarization scheme on three orthogonal bases. The decoy-state technique [10] uses multiple intensity levels that are randomly chosen at the transmitter's source. Only one of the intensity levels is the signal state, while the others are the decoy states. Alice at the end publicly announces the intensity level that was used in the transmission of each qubit. The E91 [8] scheme uses perfectly correlated entangled photon pairs. Alice and Bob would get the same result if they measured the polarization of their photons. Any attempt by Eve to eavesdrop destroys these correlations in a way that Alice and Bob can detect. For some of these approaches, particularly those using few qubits with a few states for a secret bit, sophisticated QKD attackers [15] may be able to successfully intercept and replace the collapsed qubits with fresh initialized qubits and stay undetected.

A. Earlier work on QKD using the DJ-algorithm

The quantum DJ-algorithm [1] categorizes an n-qubit function U_f (called an oracle) in a single iteration, making it exponentially faster than the classical counterpart. The oracle U_f is determined to be *balanced* if for half of the inputs the output is $|0\rangle$ and for the other half the output is $|1\rangle$. The oracle U_f is *constant* if for all possible inputs the output is either always $|0\rangle$ or always $|1\rangle$. Each run of the DJ-algorithm requires a set of input qubits and a helper target qubit, which together form the DJ-packet.

The work by Nagata and Nakamura [2] for QKD using the DJ-algorithm is shown in Figure 2. Alice sends a sequence of DJ-packets, e.g., DJ-Packet1, and DJ-Packet2 as requests to Bob with the input qubits in them set to $|0\rangle$, the helper target qubit set to $|1\rangle$, and superposed by the *Hadamard* transform. Bob applies a balanced or a constant oracle U_f for each DJ-packet request and sends them back to Alice. Alice measures the qubits in the received DJ-packets and computes to determine if the oracle U_f Bob applied on each of them was balanced or constant. Bob's choices of { *constant*, *balanced* } map to { 0, 1 } bits of a key which now becomes a secret shared binary information (bit) between Bob and Alice. Figure 2 shows two DJ-packets, each carrying one binary bit information. To share a 128 bit secret key at least 128 DJ-packet communication is needed. The DJ-packets in Figure 2 are of fixed size, each with four qubits. The solid box marked (T) is the helper target qubit, the others are input qubits. With the throughput $= 1/4$ secret bit per qubit, the secrecy is very low due to predictable qubit positions. Similar to some existing QKD approaches, Nagata and Nakamura's [2]

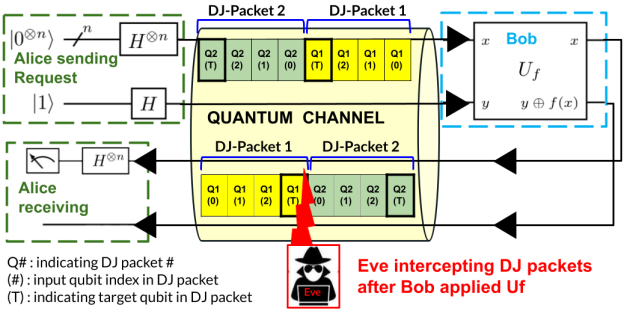


Figure 2. The DJ-algorithm for QKD with fixed-sized DJ-packets [2]

approach is also prone to MITM and eavesdropping attacks. The attacker (Eve) can predictably intercept the fixed size DJ-packets, seen as high as 25% in our simulations. A resourceful attacker can even replace the collapsed qubits with fresh qubits all initialized to $|0\rangle$ state [13] if the oracle U_f is constant, enabling Eve to stay undetected. Resourceful attackers can also do intercept/resend (faked-state) and quantum cloning attacks [14] [15]. An improvement in secrecy is achieved by De et al. [3], by changing the sizes of the consecutive DJ-packets based on a hopping (H) pattern and reordering (R) the position of the qubits within the DJ-packet, called the HR scheme. Hopping and Reordering make it hard for the attacker to identify all the required qubits and their type (input or target (T)), thereby increasing the difficulty of determining if the oracle U_f is constant or balanced. Figure 3 shows a sequence of DJ-packets with size hopping from 3 qubits to 2 qubits, and then to 4 qubits. It also shows the helper target qubit (solid box with (T) in the figure) can be at any position in the DJ-packet. The throughput $= 3/(3+2+4) = 1/3$ secret bit per qubit.

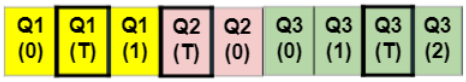


Figure 3. Hopping and reordering (HR) scheme for DJ-packet communication [3]. In the scheme "HR:N1(M1),N2(M2),N3(M3)", 'N1','N2' and 'N3' denote the number of qubits; 'M1', 'M2' and 'M3' denote the target qubit indices, for three consecutive DJ-packets.

However, the secrecy increase in the HR scheme is still not enough and there is a noticeable opportunity of successful interception, seen as much as 2% in our simulations. Furthermore, the HR method needs the specific HR scheme to be pre-shared between Alice and Bob. Hence, there is a need to develop a mechanism with much higher secrecy and that does not require pre-sharing. The next section provides the new HRB scheme that satisfies these requirements.

III. THE NEW HRB SCHEME

The HRB scheme provides a very high entropy quantum communication framework by using multiple orthogonal bases for the qubits in the DJ-packets. This increases the secrecy when compared to the HR mechanism [3] and the BB84-based schemes. Computations in DJ-algorithm still operate with qubits in the standard Z-basis, but during transmission certain selected qubits are transformed into a distinct value such that they are in a different set of orthogonal basis, e.g., the

 ISBN: 978-1-61208-961-4

X-basis or the Y-basis. Alternatively, the selected qubits can be rotated by distinct angle values that are orthogonal to each other. The HRB scheme thus harnesses both the computation and communication benefits of quantum technology.

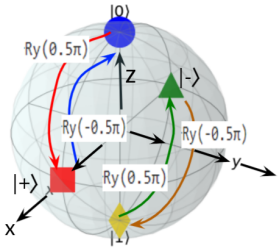


Figure 4. Qubit rotation about the Y-axis to change the values between the Z-basis and the X-basis.

The bloch-sphere in Figure 4 shows that $|0\rangle$ and $|1\rangle$ in Z-basis upon rotation about the Y-axis by 0.5π radians (90°) become the $|+\rangle$ and the $|-\rangle$ in the X-basis, respectively. To recover the qubits into Z-basis values, a rotation of the qubits by -0.5π radians (−90°) about the Y-axis is needed. Alternatively, different qubits can be rotated by different orthogonal angular values (e.g., $\theta 1$ and $\theta 2$, where $\theta 1$ and $\theta 2$ are orthogonal to each other) by the sender and rotated in the reverse direction (by $-\theta 1$ and $-\theta 2$) by the receiver before processing. The values for $\theta 1$ and $\theta 2$ are selected such that practical quantum hardware implementation with error correction and fault-tolerance are feasible. It is possible using a combination of Hadamard (H) and T gates [16]. The T-gate is a rotation around the z-axis by $\pi/4$ radians. With a sequence of H and T gates in specific orders as shown in Figure 5, a single-qubit gate rotation of various angle values can be set-up around an arbitrary axis in the Bloch sphere [16]. However, cost and decoherence problems are a potential limiting factor for expansive use of the T gates. The HRB scheme is described using basis, however, qubit rotation by a specific angle (e.g., $\theta 1$ and $\theta 2$) can be alternatively used instead of basis.

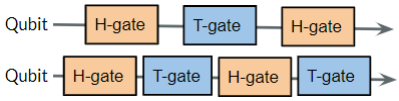


Figure 5. Various qubit rotations using H and T gates.

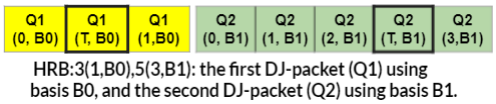
Figure 6 shows two approaches for applying multiple orthogonal bases (or qubit rotations by orthogonal angles) as:

(i) All qubits in a DJ-packet use the same basis, but different DJ-packets in a hopping sequence can use different bases. Example HRB: 3(1, B0), 5(3, B1)

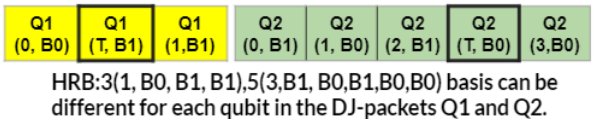
(ii) Qubits within a DJ-packet use different basis. Example HRB: 3(1, B0, B1, B1), 5(3, B1, B0, B1, B0, B0)

For illustration, two orthogonal bases (B0, B1) are used, e.g., B0=Z-basis, and B1=X-basis. The hopping sequence shows DJ-packets of two sizes with target qubit reordering. The scheme description is extended to include the orthogonal basis (or $\theta 1$, $\theta 2$) information for each qubit after the qubit index field. When orthogonal angle rotations are used, B0 and B1 represent two angles $\theta 1$ and $\theta 2$ orthogonal to each other.

Figure 7 shows the Cirq circuits for the HRB scheme that is shown in Figure 6 with two DJ-packets of sizes 3 and 5



(i) Different orthogonal basis only across DJ-packets.



(ii) Different orthogonal basis within each DJ-packet.

Figure 6. The two options, (i) and (ii), for the HRB scheme.

qubits, and using Z-basis(=B0) and the X-basis(=B1). Note the Cirq 'Ry' operator for rotation about the Y-axis by 0.5π or -0.5π radians, which is required for the value changes of the qubits to be in the different orthogonal basis (B0/B1). Alternatively, instead of changing to different orthogonal basis, different orthogonal angular rotations $\theta 1$ and $\theta 2$ can be used.

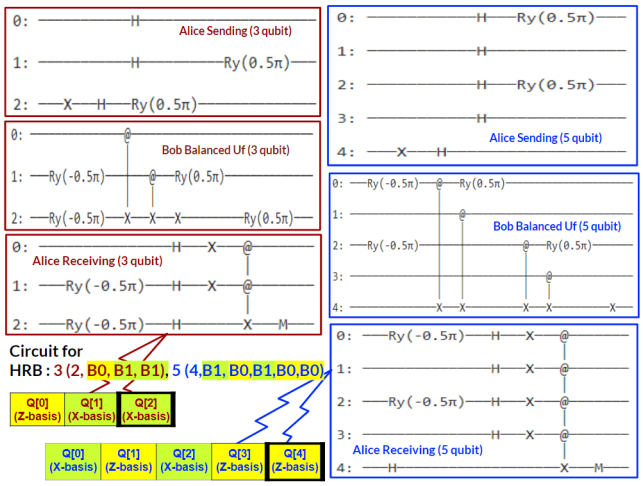


Figure 7. HRB Cirq circuits for 3-qubit and 5-qubit DJ-packets

Unlike BB84 [6], the HRB scheme does not suffer from the problem of basis mismatch between Alice and Bob. The bases (or, rotation angles) are predefined in the HRB scheme and are communicated using the mechanism described in Section V.A. The use of multiple orthogonal bases together with size-hopping and reordering makes the HRB scheme a more secure QKD mechanism than some of the BB84 based approaches [6] [7], and will be discussed more in Section IV B.

A. DJ-packet buffering and transmission

The qubits in the DJ-packets are momentarily buffered before sending to convert from the parallel order as in the DJ-algorithm, into a linear sequence for transmission on the quantum channel. The change from the Z-basis is done just before buffering, while target qubit reordering is done when the buffered qubits are serialized. Figure 8 shows qubits Q[0] and Q[2] are received in X-basis by Bob, then changed to Z-basis to apply the oracle, and then sent out again in X-basis. The reordered qubits are ordered back during buffering as they were in the original parallel form, and then changed back to Z-basis, as illustrated in Figure 8 using qubits Q[0] and Q[2]. Then, the next part of the DJ-algorithm is applied.

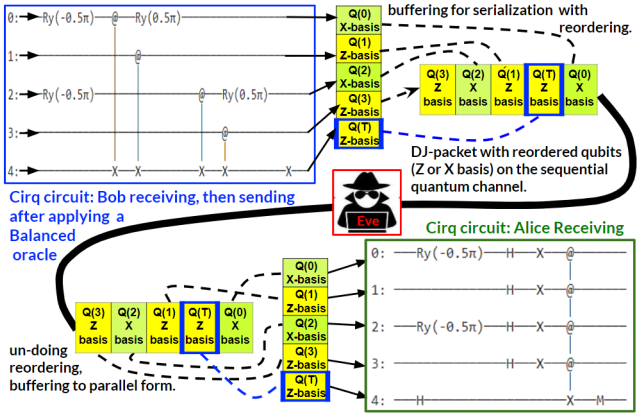


Figure 8. Reordering of DJ-packet qubits with buffering, and Ry rotation for using Z-basis or X-basis, on the quantum channel.

The DJ-algorithm computation always occurs with the qubits in Z-basis and in their natural order, while the quantum communication channel transmits the qubits in different orthogonal bases and positions reordered, leading to a large increase in entropy. The buffering time is determined by the time taken to transmit all the DJ-packet qubits in the quantum channel.

IV. SIMULATION AND TEST RESULTS

The HRB scheme is implemented in Python using Cirq [5] quantum operations and run on the Cirq quantum simulator. Cirq simulates the behavior of quantum hardware using stochastic models while running on classical computers. The Python code using Cirq operations sets up a quantum circuit for the HRB scheme and takes the DJ-packets as input. The Cirq implementation for Eve intercepts the DJ-packet qubits in transit according to the attack models. Being independent entities, Alice, Bob and Eve use three different Cirq simulator instances. The stream of DJ-packets is set up with a specific HRB scheme and sent between the simulator instances of Alice and Bob.

A. Experiments and the results

The attack models use a fixed-size scan window M (e.g., $M = 4$ qubits) with the target qubit (T) at the last index (i.e., $M - 1 = 3$). The models scan the continuous stream of DJ-packets between Bob and Alice with starting qubit offsets between 0 to 'scan window size -1' (i.e., $M - 1$). Attacker Eve expects that one of the offsets will match a DJ-packet boundary with size M qubits. Eve also assumes that up to three orthogonal bases can be randomly selected by Alice and Bob.

Figure 9 shows the simulation results for the attacker's successful interception rate. Figure 10 shows the secrecy, which is defined as "(100% - the successful interception%)", assuming there are no other compromises. Eve's interception is successful only if all the qubits of a DJ-packet are correctly identified. Partial interception of DJ-packet fails to determine the type of oracle U_f Bob applied, and leads to the attacker replacing incorrect collapsed qubits with fresh qubits that get detected by Alice, thereby exposing the attacker. The first bar (F:4) in Figure 9 represents the work by Nagata and Nakamura [2] with fixed size DJ-packets where the attacker's interception success is as high as 25%. The second and the third bars show the HR schemes, which are representative of the prior work by De et al. [3]. HR:2,4,3 uses qubit reordering and has three consecutive DJ packets of 2, 4, and 3 qubits, where the attacker's interception dropped to 2.77%. HR:2,6,4,5,3 has more variable sized DJ-packets and hence more entropy, where the attacker's successful interception dropped to 2.0%, which is a 12.5-times drop compared to F:4. The fourth, fifth

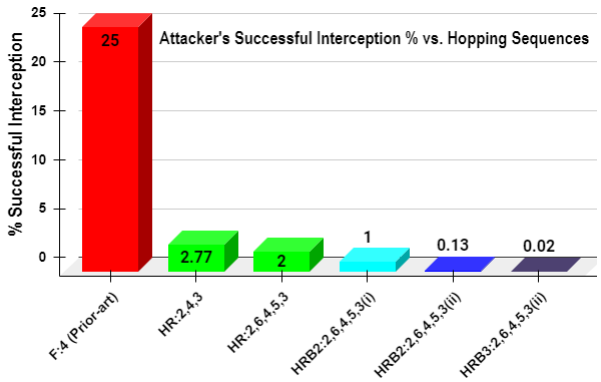


Figure 9. Bar chart comparing attacker's successful interception rates for Fixed, HR, and HRB schemes. Lower is better.

and the sixth bars show the new HRB scheme that combines size hopping, reordering, and multi-basis. The fourth bar uses two orthogonal bases and option-1 (HRB2:2,6,4,5,3(i)) where Eve's successful interception is 1%. The fifth bar also uses two orthogonal bases but is with option-2 (HRB2:2,6,4,5,3(ii)), where Eve's interception success drops further to 0.13%, which is 200-times lower than F:4. Hence, option-2 for multi-basis is much more effective than option-1. The sixth bar uses three orthogonal bases with option-2 (HRB3:2,6,4,5,3(ii)) and has the highest entropy. Eve's successful interception rate drastically drops to 0.02%, which is more than 1000-times lower than F:4.

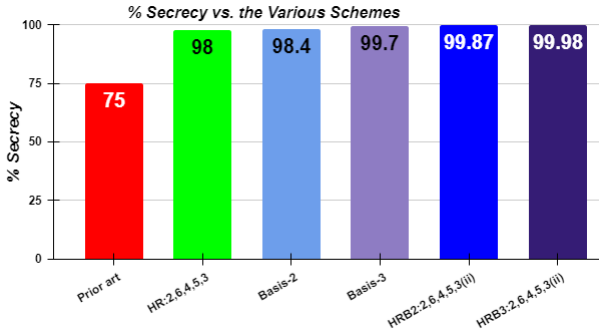


Figure 10. Bar chart comparing secrecy for Fixed (prior art), HR, Fixed-Basis, and HRB schemes. Higher is better.

Figure 10 shows that secrecy increases as the entropy through DJ-packet size-diversity, reordering and multi-basis increase. Basis-2 (secrecy=98.4%) and Basis-3 (secrecy=99.7%) are with fixed sized packets without reordering, but they use two and three different orthogonal bases, respectively. They show secrecy can be better from just using multiple orthogonal bases, rather than hopping/reordering as in HR scheme (98%) [3]. The best secrecy is achieved for

 ISBN: 978-1-61208-961-4

HRB3:2,6,4,5,3(ii) (99.98%) that uses three different bases, the next best is HRB2:2,6,4,5,3(ii) (99.87%) that uses two different orthogonal bases. These tests compared results among the use of one, two and three bases, where the secrecy improves from 75% for one basis, to 98.4% for two bases, and to 99.7% for three bases. This shows a gradually flattening increase in secrecy with further increase in the number of bases. However, the complexity of the hardware implementation increases with the number of bases due to increase in quantum gates needed. Hence, a trade-off should be done for the secrecy needed to thwart the existing interception threat on the quantum communication channel versus the number of bases needed to attain the secrecy requirements.

B. Secrecy and efficiency comparison with BB84

A probabilistic comparison is provided between BB84 and the HRB scheme. Each qubit in BB84 can be in one of the four different states and maps to a binary bit, which results in 25% successful interception probability assuming equally likely presence of the four states. For comparison, we select the HRB scheme HRB2:2,6,4,5,3(ii) that has five DJ-packets of sizes 2, 6, 4, 5, and 3 qubits, with a total of 20 ($H = 20$) qubits in the hopping sequence. There is only one DJ-packet ($P = 1$) that matches the attacker scan window size ($M = 4$ qubits). In HRB2:2,6,4,5,3(ii) each qubit can be using one of the two orthogonal bases ($B = 2$). The probability for the attacker matching the specific orthogonal basis, out of 'B' possible orthogonal bases, for all the M qubits in the DJ-packet is $(1/B)^M$. The probability of matching the reordered qubits is $1/M$. The probability of matching the DJ-packet boundary is (P/H). Hence, the total probability of successful interception is $= (P/H) * (1/M) * (1/B)^M = (1/20) * (1/4) * (1/2)^4 * 100\% = 0.078\%$, and is much lower than the BB84 protocol. If three orthogonal bases (or distinct rotation angles) are used, the probability of the attacker's successful interception is theoretically reduced to 0.015%. Unlike BB84 [6] [7], the HRB scheme does not suffer from the problem of orthogonal basis mismatch between Alice and Bob. This fact, together with the high entropy, makes the HRB scheme more secure than some of the BB84 based QKD.

V. THE COMMUNICATION FRAMEWORK

Alice, Bob and all participants using this technology have the list of all possible HRB schemes as a part of the system software available with their computing system equipped to perform QKD. As may be occasionally needed, the list can be updated as a software update to their computing system.

A. Setting up the HRB scheme before communication

Whenever QKD or secure messaging is needed, the specific HRB scheme to be used is first communicated, shown as 'step 0' in Figure 11. It can be done by a few possible approaches. One such approach is to use the BB84 to randomly select and communicate an N-bit (e.g., N=8,12,16) value that will indicate the index of the HRB scheme within the list (e.g., 8-bits when list size $\leq$ 256, 12-bits when $\leq$ 4,096, 16-bits when $\leq$ 65,536) of all predefined HRB schemes. BB84 is only used for sharing the index of the HRB scheme secretly. Once the HRB scheme for this session is communicated, both Bob and Alice loads the relevant portions of the quantum circuitry (as shown in Figure 7) for the particular HRB scheme into the quantum hardware. The actual 128 bit key is then shared securely by the HRB scheme, as shown in 'step 1' in Figure 11. This strategy is used since the secrecy achieved by HRB DJ-algorithm is much more than BB84, as discussed in Section IV.D. Furthermore, BB84 has a statistical rejection rate of 50% for the shared secret due to Alice and Bob's random mismatch in basis. This problem of BB84 is now limited to only sharing the list index with small number of bits (between 8 to 16) instead of all the 128 bits for the secret key. The combination of BB84 for the initial step (step 0) to communicate the specific HRB scheme index and then using the HRB scheme for the actual QKD or secure messaging ('step 1') as in Figure 11 leads to an overall improved secrecy and effectiveness than using just BB84.

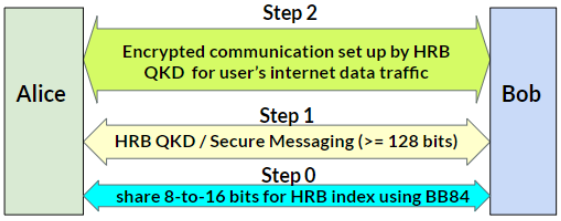


Figure 11. The HRB QKD framework showing 'step 0' for communicating the HRB scheme index. The 'step 1' can do QKD or secure messaging

B. Secure short messages over quantum channel

With a specific HRB scheme set up in 'step 0' as in Figure 11, it is possible to directly send secure short messages in 'step 1' over the quantum channel by encoding the message as a specific sequence of constant or balanced oracles. This is not possible by most other QKD mechanisms as the secret bits are randomly generated with high chances of rejection due to mismatch at the two end points (e.g., basis mismatch between Bob and Alice in BB84). Step 2 is unused in this case.

C. Detecting attacker's interceptions and actions thereafter

Any DJ-packet for which Alice detected interception is discarded. Alice notifies Bob of the sequence numbers of the DJ-packets to discard over the classical communication channel. Bob reconstructs the 128 bit shared secret key by throwing away the discarded DJ-packet sequence numbers sent by Alice. The attacker can still intercept these sequence numbers, but it is irrelevant since those indicate discarded DJ-packets by Alice and Bob. Thus, the actual shared M-bit (e.g., $M = 128$) key stays secret between Alice and Bob. Alice measures the input qubits in the DJ-packet received from Bob, computes to determine if Bob used a constant or balanced U_f, and updates the result in the target qubit, forming the output DJ-packet. The qubit states of the output DJ-packet are then compared with the expected qubit states for the same sized DJ-packets. Alice detects an interception if the states of one or more qubits in the output DJ-packet do not match

 ISBN: 978-1-61208-961-4

the states in any of the expected output DJ-packets. Figure 12 shows the flowchart for detecting interception. Table I shows the Cirq code for some constant and balanced oracles. The

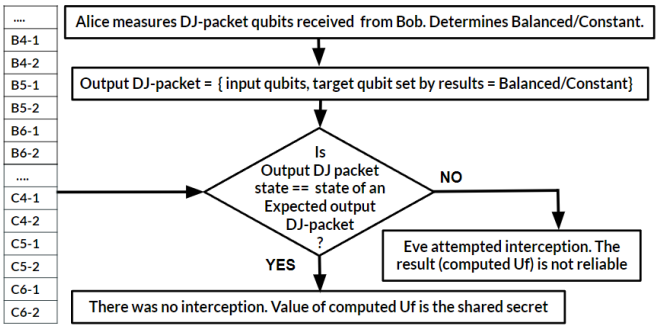


Figure 12. The flowchart for detecting interception by Alice

TABLE I. EXPECTED OUTPUT DJ-PACKETS FOR 4-QUBIT ORACLES

U_f: Example 4-qubit oracle (q0, q1, q2, q3) forms	Expected output DJ-packet
`[ ]`	Constant `C4-1`
`[cirq.X(q3)]`	Constant `C4-2`
`[cirq.CNOT(q0, q3), cirq.CNOT(q1, q3), cirq.CNOT(q2, q3)]`	Balanced `B4-1`
`[cirq.CNOT(q0, q3), cirq.CNOT(q1, q3), cirq.CNOT(q2, q3), cirq.X(q3)]`	Balanced `B4-2`

suffix "-1" or "-2" indicate the different expected output DJ-packets for all possible forms of the constant and the balanced oracles. Alice maintains a repository of expected output DJ-packets for different sized constant and balanced oracle forms. It is shown on the left side of Figure 12 as the list (.., B4-1,.., B6-2,.., C4-1,..,C6-2). As an example, prefixes C4 and B4 are for expected output DJ-packets with 4-qubits when Bob applied the constant and the balanced oracles, respectively. The number of expected output DJ-packets per DJ-packet size is a small finite number after the effects of reordering and multi-basis are removed.

VI. CONCLUSION

A novel way to tremendously increase the entropy of the DJ-packets communicated over the quantum channel is developed by employing different orthogonal basis for the qubits in the DJ-packets. Simulations showed that attacker's successful interception rate drops 200-times when using two orthogonal bases, and more than 1000-times with three orthogonal bases vs. prior work. This framework can be used for QKD and also for secure messages due to the very high secrecy ($\geq 99.98\%$) it provides, and also because it sets up a new HRB scheme for every new session. Hence, this work enhanced communication secrecy and broadened the scope compared to the earlier published works [3] [6]- [10].

Future work needs to evaluate different HRB schemes on real quantum hardware and perform trade-offs on HRB quantum circuit size vs. sustainability to decoherence effects and noisy channels. These studies can also help determine the need for dynamic selection of the HRB schemes of different secrecy levels depending on the existing level of threat on the quantum communication channel from a MITM attacker or an interceptor. If an increased interception rate is detected by Bob or Alice, they can decide to select a HRB scheme with an even higher secrecy, but at the cost of increased quantum circuit size and qubit requirements. Alternatively, if Alice or Bob finds zero interception, then they can decide to use a HRB scheme of reduced secrecy level so as to reduce the quantum circuit size and the number of qubits required. Finally, this research can also provide a foundation for interested readers to learn more about how quantum computing and quantum communications impact cybersecurity.

ACKNOWLEDGMENT

The author would like to immensely thank Mr. Jeremy Juybari, Mr. Colton Beery, and late Dr. Raymond Moberly of Faster Logic LLC; Dr. Kyle Sundqvist of the Physics Department at San Diego State University; for their support on the early research foundations for this project. Many thanks to Ms. Jo Buehler, Rohit's High School Calculus teacher, for her encouragement and support.

REFERENCES

[1] D. Deutsch and R. Jozsa, "Rapid Solutions of Problems by Quantum Computation," Proceedings of the Royal Society of London A.439, pp.553–558, Dec. 1992. doi:10.1098/rspa.1992.0167.

[2] K. Nagata and T. Nakamura, "The Deutsch-Jozsa Algorithm Can Be Used for Quantum Key Distribution," Open Access Library Journal Vol.2, No.8, Aug. 2015. doi:10.4236/oalib.1101798

[3] R. De, R. Moberly, C. Beery, J. Juybari and K. Sundqvist, "Multi-Qubit Size-Hopping Deutsch-Jozsa Algorithm with Qubit Reordering for Secure Quantum Key Distribution," in 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), Broomfield, CO, USA, pp. 473-474, 2021. doi:10.1109/QCE52317.2021.00084

[4] A. Ananthaswamy, "The Quantum Internet Is Emerging, One Experiment at a Time," Scientific American, June 2019.

[5] A. Ho and D. Bacon, "Announcing Cirq: An Open Source Framework for NISQ Algorithms," Google AI Blog, July 2018.

[6] C. H. Bennett, G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," Theoretical Computer Science, Volume 560, Part 1, 2014, pp.7-11, ISSN 0304-3975, doi:10.1016/j.tcs.2014.05.025.

[7] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, M. Peev, "The security of practical quantum key distribution," Rev. Mod. Phys. 81 (3): pp.1301–1350, 2009.

[8] A. Ekert, "Quantum cryptography based on Bell's theorem," Physical Review Letters. American Physical Society. 67 (6): pp.661–663, 1991.

[9] H. Bechmann-Pasquinucci and N. Gisin, "Incoherent and coherent eavesdropping in the six-state protocol of quantum cryptography," Physical Review A 59: pp.4238-4248, 1999.

[10] H. Lo, X. Ma, and K. Chen, "Decoy State Quantum Key Distribution," Physical Review Letters. American Physical Society (APS). 94 (23): 230504, June 2005. doi:10.1103/PhysRevLett.94.230504.

[11] A. P. Bhatt and A. Sharma, "Quantum Cryptography for Internet of Things Security," Journal of Electronic Science and Technology, Volume 17, Issue 3,2019, pp 213-220, ISSN 1674-862X.

[12] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," Nature, vol. 299, no. 5886, pp. 802–803, 1982.

[13] G. L. Long and Y. Sun, "Efficient scheme for initializing a quantum register with an arbitrary superposed state," Physical Review A, vol. 64, no. 1, 2001. doi:10.1103/PhysRevA.64.014303.

[14] Y. Fei, X. Meng, M. Gao, H. Wang, and Z. Ma, "Quantum man-in-the-middle attack on the calibration process of quantum key distribution," Sci Rep 8, 4283, 2018.

[15] D. R. Kuhn, "Vulnerabilities in Quantum Key Distribution Protocols," NISTIR 6977, May 2003, NIST.

[16] Learn Quantum Computation using Qiskit, https://qiskit.org/textbook/ch-gates/more-circuit-identities.html: May, 2022