 doi:10.20944/preprints202209.0485.v1

QKD Reconciliation System Using Conjugate Frames in the Presence of Near-Unity Error Rates

Jesús Ignacio Morán Ramírez[1], Luis Adrián Lizama-Perez[2,*[0000−0001−5109−2927]] and José Mauricio López Romero[3]

[1] Sección de Posgrado de la Universidad Politécnica de Pachuca, Ex-Hacienda de Santa Bárbara, Zempoala, Hidalgo 43830, Mexico; moranramirezjesus@micorreo.upp.edu.mx
[2] Universidad Técnica Federico Santa María, Av. Vicuña Mackenna 3939, San Joaquín, Santiago, Chile; luis.lizamap@usm.cl
[3] Cinvestav Querétaro, Libramiento Norponiente 2000, Real de Juriquilla, Santiago de Querétaro, Querétaro 76230, Mexico; jm.lopez@cinvestav.mx

Abstract. Previously, using the conjugate frame-based reconciliation approach, we defined a method to correct errors produced in pairs of non-orthogonal quantum states that are transmitted through a quantum key distribution (QKD) link. The security of the frame-based reconciliation was discussed in order to deal with Photon Number Division (PNS) attack and Intercept and Forward (IR) attack, among others.
However, until the time of publication we did not have the distillation software to test our method. In this article, following the conjugate frame distillation method, we present the implementation of the post-processing system that demonstrates that it is capable of correcting errors in the presence of error rates close to unity. The system shows that when the number of double-sensing events at Bob's station is as low as 100, the number of secret bits stays above 4500 bits in about 12 seconds, giving a secret rate of 375 bits per second while that the channel error rate reaches 90%.

Keywords: QKD · distillation · reconciliation · conjugate frame

1 Introduction

Quantum cryptography represents one of the most promising schemes for data protection in the quantum era [1,2,3]. Together with post-quantum cryptography techniques which have been selected by NIST [4], Quantum Key Distribution (QKD) is emerging as one of the most secure schemes to deal with quantum computers that are capable of executing the cryptanalytic algorithm for factoring large integer numbers [5].

Unfortunately, commercially available QKD operates through quantum channels that exhibit low noise levels, limiting, in most cases, the total length of the quantum channel [6,7]. The reason why QKD is not executed on highly noisy channels is that, currently, QKD technology does not execute any method that allows error correction beyond 25% rate [8,9]. Traditionally, the error correction process has been made possible by Cascade [10,11], a scheme developed by the pioneers of QKD. Disadvantageously, the protocol is highly interactive and does not guarantee the complete error elimination. In addition to Cascade, other reconciliation techniques have been used, mainly LDPC [12,13], whose computational complexity is greater and requires redundant information to be transmitted.

This situation could change drastically if the currently available QKD technology implements the frame-based error reconciliation algorithm, which allows operating on quantum channels that exhibit high error rates [14,15]. The first version of the frame-based reconciliation method exhibits

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 September 2022 doi:10.20944/preprints202209.0485.v1

a decrease in efficiency as the channel error rate increases, but the method is still functional when the rate is greater than 50% [16].

In our previous work, we have presented the reconciliation method by means of conjugate frames, which has no dependence on the channel error rate [17]. At least theoretically, it is capable of correcting all the errors generated by the quantum channel. It is only necessary to have, at the beginning of the process, some auxiliary frames that serve to start the error correction. In the next section, we'll cover frame-based reconciliation in more detail.

2 Sifting and Reconciliation based on Frames

The frame-based reconciliation is a new approach for error correction of QKD systems [15]. Due to the properties of the quantum states that are sent through the quantum channel, two logical communication channels are established, one for each quantum basis (**X**,**Z**). To do this, Alice prepares the information through logical units called frames, which are binary matrices that group two or more pairs of non-orthogonal states. Each row of a frame is equivalent to a pair of non-orthogonal states. A frame has two columns that identify the quantum bases **X**,**Z**. A bit is transferred through a pair of non-orthogonal states as long as they produce the same result in the receiving station's optical detector. On Bob's side, the frame exhibits the results obtained once the measurements are made. The Figure 1 represents an example using 3×2 frames, where the symbol + denotes the absence of the state. Alice's first pair of non-orthogonal states is formed by the quantum states ($\mathbf{0}_{\mathbf{x}}^{\mathbf{1}}$, $\mathbf{1}_{\mathbf{z}}^{\mathbf{1}}$) where the superscript denotes the number of the pair. The order of the states can be controlled logically since the states are independent of each other, allowing the states of different pairs to be interleaved. The first pair constitutes the first line of Alice's frame as seen in Eq. 1. Pairs two and three are ($\mathbf{1}_{\mathbf{x}}^{\mathbf{2}}$, $\mathbf{1}_{\mathbf{z}}^{\mathbf{2}}$),($\mathbf{1}_{\mathbf{x}}^{\mathbf{3}}$, $\mathbf{0}_{\mathbf{z}}^{\mathbf{3}}$) which correspond to the second and third rows of the frame, respectively. On Bob's side, the first double detection event requires the measurement of the first pair which returns $\mathbf{0}_{\mathbf{x}}^{\mathbf{1}}$, the measurement of the second pair produces $\mathbf{1}_{\mathbf{z}}^{\mathbf{2}}$ while the third pair is detected as $\mathbf{1}_{\mathbf{x}}^{\mathbf{3}}$.

Fig. 1: Frame-based protocol operation: Alice prepares a 3×2 frame, while Bob, after measuring the states, obtains a version of the read frame. The symbol + denotes an empty state.

As can be seen, the information of the pairs of non-orthogonal quantum states are grouped in matrices called frames, which contain the information of two or more pairs of non-orthogonal

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 September 2022 doi:10.20944/preprints202209.0485.v1

states. A 2×2 frame groups two pairs of non-orthogonal states together so it produces two bits on Bob's side. Similarly, a 3×2 frame groups three pairs of states together and produces three bits at Bob's station. The secret key is derived from the geometry of the frame that remains after the detection events, not from the bits that are measured in Bob's detectors. Thus, the fit information between Alice and Bob is extracted from the geometry of the frame. As mentioned before, a useful pair of non-orthogonal states must produce the same result, when measuring with the **X** basis it can be: 0_x,1_x or when measuring with **Z** basis: $\mathbf{0_z}$,$\mathbf{1_z}$. The Figure 2 shows, the different geometries of the possible results using 2×2 frames. In Figure 2, the symbol • denotes a coincident double detection event in which a pair of non-orthogonal states activates the same detector, while the symbol + denotes an empty pulse, which is logically interpreted as a 0 bit. Table 1 presents an overview of the framing approach based on 2×2 frames.

$$MR_1 = \begin{pmatrix} \bullet & + \\ \bullet & + \end{pmatrix}, MR_2 = \begin{pmatrix} + & \bullet \\ + & \bullet \end{pmatrix}, MR_3 = \begin{pmatrix} \bullet & + \\ + & \bullet \end{pmatrix}, MR_4 = \begin{pmatrix} + & \bullet \\ \bullet & + \end{pmatrix}$$

Fig. 2: The secret information is derived from the geometry of the measured frames.

Table 1: This example assumes an error free channel: In a) it is represented the behaviour of a no framing protocol thus exists just one key of 8 bits, b) shows the frames prepared by Alice and c) After Bob's measurement it is exhibited one of 2^8 possible keys of 4 bits each one (due to the sifting process). Using Bob's sifting bits, Alice should be able to identify the pattern that remains after Bob performs the measurements of the quantum states.

Alice ⟶ Bob

a) No framing protocol: one key of 8 bits.

$[\mathbf{0}_{\mathbf{Z}}^{\mathbf{8}}]$	$[\mathbf{1}_{\mathbf{X}}^{\mathbf{7}}]$	$[\mathbf{1}_{\mathbf{Z}}^{\mathbf{6}}]$	$[\mathbf{1}_{\mathbf{X}}^{\mathbf{5}}]$	$[\mathbf{0}_{\mathbf{Z}}^{\mathbf{4}}]$	$[\mathbf{1}_{\mathbf{Z}}^{\mathbf{3}}]$	$[\mathbf{0}_{\mathbf{X}}^{\mathbf{2}}]$	$[\mathbf{0}_{\mathbf{X}}^{\mathbf{1}}]$

b) Alice prepares frames to be transmitted.

$\lceil\mathbf{0}_{\mathbf{X}}^{\mathbf{8}}\rceil$	$\lceil\mathbf{1}_{\mathbf{X}}^{\mathbf{7}}\rceil$	$\lceil\mathbf{0}_{\mathbf{X}}^{\mathbf{6}}\rceil$	$\lceil\mathbf{1}_{\mathbf{X}}^{\mathbf{5}}\rceil$	$\lceil\mathbf{1}_{\mathbf{X}}^{\mathbf{4}}\rceil$	$\lceil\mathbf{1}_{\mathbf{X}}^{\mathbf{3}}\rceil$	$\lceil\mathbf{0}_{\mathbf{X}}^{\mathbf{2}}\rceil$	$\lceil\mathbf{0}_{\mathbf{X}}^{\mathbf{1}}\rceil$
$\lfloor\mathbf{0}_{\mathbf{Z}}^{\mathbf{8}}\rfloor$	$\lfloor\mathbf{0}_{\mathbf{Z}}^{\mathbf{7}}\rfloor$	$\lfloor\mathbf{1}_{\mathbf{Z}}^{\mathbf{6}}\rfloor$	$\lfloor\mathbf{1}_{\mathbf{Z}}^{\mathbf{5}}\rfloor$	$\lfloor\mathbf{0}_{\mathbf{Z}}^{\mathbf{4}}\rfloor$	$\lfloor\mathbf{1}_{\mathbf{Z}}^{\mathbf{3}}\rfloor$	$\lfloor\mathbf{1}_{\mathbf{Z}}^{\mathbf{2}}\rfloor$	$\lfloor\mathbf{0}_{\mathbf{Z}}^{\mathbf{1}}\rfloor$

c) It is shown one of 2^8 possible keys of 4 bits each.

+	$\mathbf{1}_{\mathbf{X}}^{\mathbf{7}}$	+	$\mathbf{1}_{\mathbf{X}}^{\mathbf{5}}$	+	+	$\mathbf{0}_{\mathbf{X}}^{\mathbf{2}}$	$\mathbf{0}_{\mathbf{X}}^{\mathbf{1}}$
$\mathbf{0}_{\mathbf{Z}}^{\mathbf{8}}$	+	$\mathbf{1}_{\mathbf{Z}}^{\mathbf{6}}$	+	$\mathbf{0}_{\mathbf{Z}}^{\mathbf{4}}$	$\mathbf{1}_{\mathbf{Z}}^{\mathbf{3}}$	+	+

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 September 2022 doi:10.20944/preprints202209.0485.v1

To get the sifting bits and proceed to derive the secret key, Bob applies the logical function XOR to each column of the frame and sends the results to Alice. Alice's task is to recognize the frame, or more specifically, the geometry of the frame obtained by Bob, which we denote as MR_i, using the sifting bits that she receives from Bob. The sifting bits make up the Sifting String SS_i. If Alice and Bob use 2×2 frames, considering the absence of errors in the quantum channel, as well as in the optical detection system, Alice can recognize MR_i when SS_i contains only the XOR bits. For 3×2 frames, Bob must add a parity bit to SS_i. However, in the presence of errors, other bits must be added to SS_i. In the case of non-conjugated protocols, the bits that are added are the bits obtained from Bob's measurements, since the secret information is generated from the geometry of the measured frames, not from the measurement results. Therefore, 2 bits are added using 2×2 frames and three bits must be appended for 3×2 frames.

The 2×2 conjugate frame protocol requires Bob to first obtain the conjugate frame, which is achieved by inverting the bits of Bob's frame. To generate SS_i (in the case of conjugate protocols we denote it as CSS_i), Bob concatenates the XOR bits of the columns of both frames (before and after conjugation). The frame conjugation method allows correcting errors when the channel error rate is close to unity and keeping the transfer of secret bits constant. It is only required to initially have the SS_i of two support frames called null and unitary frames: $f_7 = \begin{pmatrix} \mathbf{0^1_X} & \mathbf{0^1_Z} \\ \mathbf{0^2_X} & \mathbf{0^2_Z} \end{pmatrix}$, $f_{11} = \begin{pmatrix} \mathbf{1^1_X} & \mathbf{1^1_Z} \\ \mathbf{1^2_X} & \mathbf{1^2_Z} \end{pmatrix}$, which are the basis of the error correction process.

2.1 Security Condition

For the security of the frame-based sifting process, the sifting bits must not be able to be mapped to a single MR matching result. This property must be achieved to prevent an attacker from obtaining the secret bits. An example can be seen in the Figure 3.

$$MR_1 = \begin{pmatrix} \mathbf{1^1_X} & + \\ \mathbf{1^2_X} & + \end{pmatrix}, MR_2 = \begin{pmatrix} + & \mathbf{1^1_Z} \\ + & \mathbf{1^2_Z} \end{pmatrix}, MR_3 = \begin{pmatrix} \mathbf{0^1_X} & + \\ + & \mathbf{0^2_Z} \end{pmatrix}, MR_4 = \begin{pmatrix} + & \mathbf{0^1_Z} \\ \mathbf{0^2_X} & + \end{pmatrix}$$

Fig. 3: The security requirement establishes that a SS must be correlated to at least two MR. In this example, all MR_i produce the same $SS = 00$.

3 Comparison of frame reconciliation methods

We classify the frame-based error correction methods into conjugated and unconjugated. We have now succeeded in specifying the 2×2 and 3×2 unconjugated methods, and the conjugated method by means of 2×2 frames. Unconjugated reconciliation uses the bits that result directly from Bob's basis measurements. Reconciliation with conjugate frames uses the conjugate frame, which is formed by inverting the bits of Bob's frame.

 doi:10.20944/preprints202209.0485.v1

Reconciliation methods using 2×2 frames (conjugated and unconjugated) exhibit a quadratic order growth of the number of secret bits with respect to the number of double matching detection events. On the other hand, in the (non-conjugated) reconciliation by means of 3×2 frames, the number of secret bits has a cubic variation. So far, we have developed the 2×2 conjugate reconciliation method. Table 1 shows a comparison of frame-based reconciliation methods, where r is the channel error rate, n is the number of coincident double detection events and N is the number of total pulses sent by Alice.

Table 2: Comparison of frame-based reconciliation methods. Here, UC stands for unconjugated while C denotes a conjugated protocol.

Protocol		**Order** (n)	**Throughput**	**Key Rate**	**QBER**
UC	3×2	n^3	$\frac{3}{8}\binom{n}{3}\left(\frac{1}{3}-\frac{2}{7}r\right)$	$\sim \frac{3}{8}\left(1-e^{-\mu}\right)^6 N^6, r=0$	>0.5
	2×2	n^2	$\frac{1}{4}\binom{n}{2}\left(\frac{1}{2}-\frac{1}{3}r\right)$	$\sim \frac{1}{8}\left(1-e^{-\mu}\right)^4 N^4, r=0$	
C			$\frac{1}{2}\binom{n}{2}$	$\sim \frac{1}{4}\left(1-e^{-\mu}\right)^4 N^4$	~ 1

3.1 Scope of the current publication

As we said before, we have developed the 2×2 conjugate reconciliation method. However, in the previous work [17], an analysis of the performance of the protocol under an experimental environment was not presented since we did not have the software to simulate it. Now, in this article, we will introduce the conjugate distillation software specifications.

In subsection 4.1, we will describe the reconciliation algorithm and then in 4.2 we will present the implementation of the reconciliation software. Although it is a simulation software, the distillation method only contemplates the post-processing of the QKD protocol. This implies that what really comes under simulation is the behavior of the quantum channel. The two functions that the reconciliation software simulates are: the channel losses and the quantum measurement. From the results obtained from the simulation, the distillation process is started, which carries out the tasks of adjustment, correction and security amplification. It is worth mentioning that one of the advantages of frame-based reconciliation is that it performs the aforementioned tasks in a single process. The security amplification comes from a singular characteristic of frames: the rows are combined with each other to form all possible frames. This property generates an increase in the global security of the system since in the case that the attacker has information from some rows, he will inevitably lose the bits that result from the frames that are formed, in part, with these rows, and other rows that are not in possession of the attacker.

4 Reconciliation System

We depict in Figure 4 the process diagram of the reconciliation system. As a fundamental part of this system, the error correction algorithm is executed, which we will detail in the next subsection. For detailed information about the reconciliation process as well as the definitions of the frames, please refer to our previous publication [17]. In this article, we will mainly address the features of the software and the results obtained with it.

 doi:10.20944/preprints202209.0485.v1

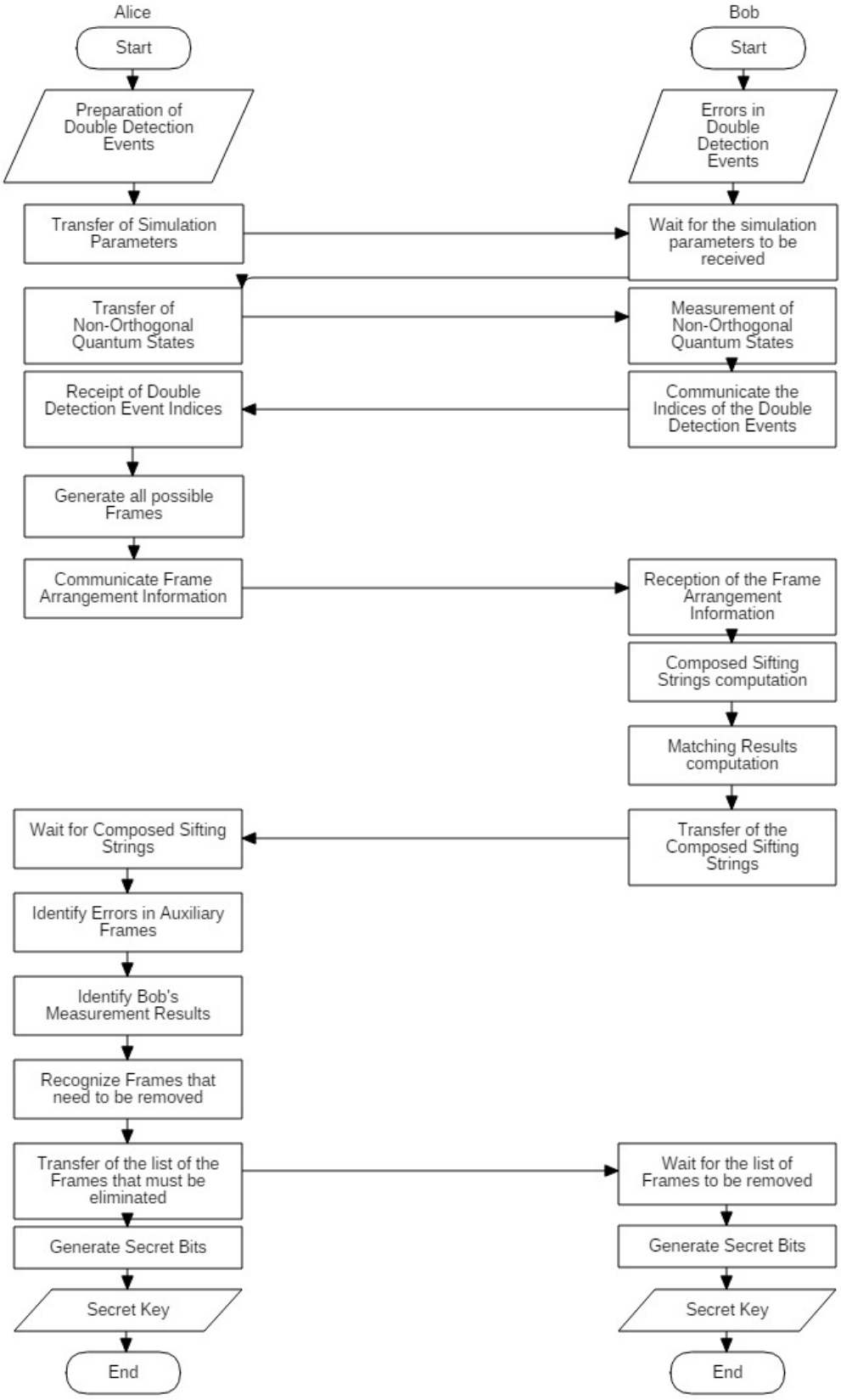

Fig. 4: Process diagram of the reconciliation system.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 September 2022 doi:10.20944/preprints202209.0485.v1

4.1 General Reconciliation Algorithm

For the specification of the reconciliation algorithm we use the overbracket symbol $\overline{\ }$ to represent the error that is produced in a transmitted non-orthogonal quantum pair (NO-QP).

1. Identify $(\mathbf{0_X}, \mathbf{0_Z})$ and $\left(\overline{\mathbf{0_X}}, \mathbf{0_Z}\right)$, $\left(\mathbf{0_X}, \overline{\mathbf{0_Z}}\right)$ errors in the set of f_7. Identify single and parallel errors using DPPE algorithm.
2. Identify $(\mathbf{1_X}, \mathbf{1_Z})$ and $\left(\overline{\mathbf{1_X}}, \mathbf{1_Z}\right)$, $\left(\mathbf{1_X}, \overline{\mathbf{1_Z}}\right)$ errors in the set of f_{11}. Identify single and parallel errors using DPPE algorithm.
3. Identify MR using $(\mathbf{0_X}, \mathbf{0_Z})$ $\left(\overline{\mathbf{0_X}}, \mathbf{0_Z}\right)$, $\left(\mathbf{0_X}, \overline{\mathbf{0_Z}}\right)$ and $(\mathbf{1_X}, \mathbf{1_Z})$ $\left(\overline{\mathbf{1_X}}, \mathbf{1_Z}\right)$, $\left(\mathbf{1_X}, \overline{\mathbf{1_Z}}\right)$ in f_8, f_{12}.
4. Identify $(\mathbf{0_X}, \mathbf{1_Z})$, $(\mathbf{1_X}, \mathbf{0_Z})$ and $\left(\overline{\mathbf{0_X}}, \mathbf{1_Z}\right)$, $\left(\mathbf{1_X}, \overline{\mathbf{0_Z}}\right)$ errors in f_9, f_{10}, f_{13}, f_{14} using $(\mathbf{0_X}, \mathbf{0_Z})$, $\left(\overline{\mathbf{0_X}}, \mathbf{0_Z}\right)$, $\left(\mathbf{0_X}, \overline{\mathbf{0_Z}}\right)$. Identify MR in f_9, f_{10}, f_{13}, f_{14}.
5. Identify $(\mathbf{0_X}, \mathbf{1_Z})$, $(\mathbf{1_X}, \mathbf{0_Z})$ and $\left(\mathbf{0_X}, \overline{\mathbf{1_Z}}\right)$, $\left(\overline{\mathbf{1_X}}, \mathbf{0_Z}\right)$ errors in f_2, f_6, f_3, f_4 using $(\mathbf{1_X}, \mathbf{1_Z})$, $\left(\overline{\mathbf{1_X}}, \mathbf{1_Z}\right)$, $\left(\mathbf{1_X}, \overline{\mathbf{1_Z}}\right)$, $(\mathbf{0_X}, \mathbf{1_Z})$, $(\mathbf{1_X}, \mathbf{0_Z})$, $\left(\overline{\mathbf{0_X}}, \mathbf{1_Z}\right)$, $\left(\mathbf{1_X}, \overline{\mathbf{0_Z}}\right)$. Identify MR in f_2, f_6, f_3, f_4.
6. Identify MR in f_1, f_5 using $(\mathbf{0_X}, \mathbf{1_Z})$, $(\mathbf{1_X}, \mathbf{0_Z})$, $\left(\mathbf{0_X}, \overline{\mathbf{1_Z}}\right)$, $\left(\overline{\mathbf{0_X}}, \mathbf{1_Z}\right)$, $\left(\mathbf{1_X}, \overline{\mathbf{0_Z}}\right)$, $\left(\overline{\mathbf{1_X}}, \mathbf{0_Z}\right)$.

DPPE Algorithm. We provide below the specification of the DPPE algorithm.

1. Given the Composed Sifting String List (CSSL), Alice separates from null frames into the error-detected-null-frames and the error-free-null-frames just checking that $CSS \neq 0000$. The last list contain, however, frames with hidden (parallel) errors.
2. Alice keeps searching into the list to identify cases in the error-detected list that reveals hidden errors.
3. Alice uses an error-free NO-QP of the list of error-free null-frames to identify the position of the error between the non-orthogonal states.

5 Results

We describe in Appendix A the characteristics of the transmitting and receiving nodes, providing a brief description of the behavior of these processes. Each test was performed for an error rate, which was repeated 10 times, obtaining the average key size and execution time. This procedure was carried out at the error percentages 0%, 25%, 50%, 75%, 90% and 95% for 100 double matching detection events. The results were recorded to subsequently conform the performance statistics discussed in the following section. Next, we show the Alice and Bob interfaces for the 75% and 90% error rates. The interfaces of Alice and Bob are shown in the figure Figure 7 and 8 for 100 double matching detection events and Figure 9 and 10.

The Figure 5 shows a comparison in terms of the number of secret bits of the conjugated and unconjugated protocols. On the other side, the Figure 6 shows a comparison in terms of the execution time of the conjugated and non-conjugated protocols. In the Figure 5 we can see the behavior of the unconjugated 2×2 protocol, which rapidly loses the ability to maintain the secret bit rate as the channel error rate increases. The 3×2 protocol shows the best performance if we consider an error rate of less than 10%, thanks to its ability to cubically increase the secret bits.

However, as the error rate on the channel increases, the 3×2 protocol rapidly loses its ability to keep the bit rate secret. In contrast, the 2×2 conjugate protocol keeps the secret bit rate above 50% error. When exceeding 90% error in the channel, its ability to maintain bit rate decreases considerably, as shown in the Figure 5.

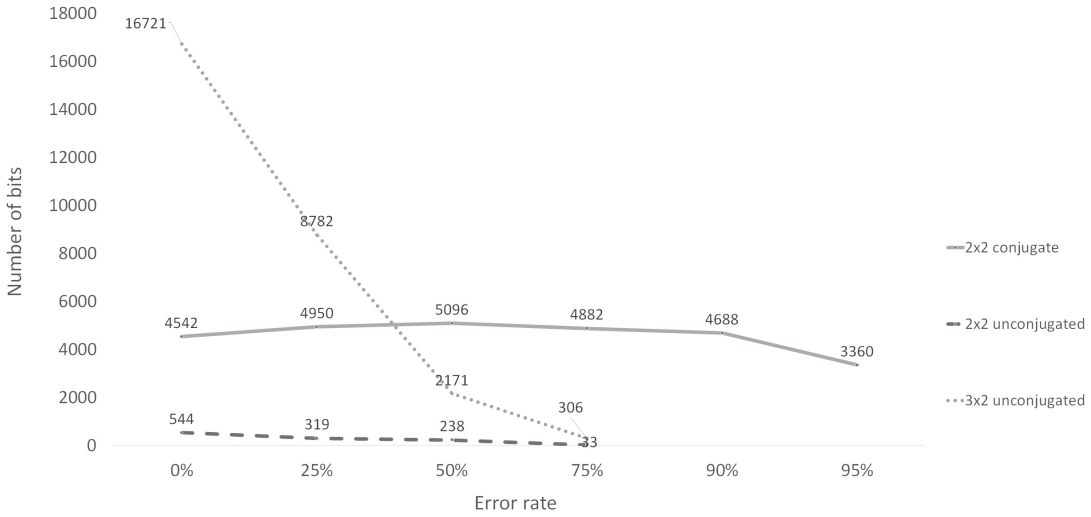

Fig. 5: Number of secret bits of conjugated and unconjugated protocols.

The Figure 6 allows us to visualize the processing time of the protocols, we observe a high-speed processing that the 2×2 unconjugated protocol presents, however, its ability to derive secret bits with an error rate greater than 30% is almost zero, while the 3×2 protocol shows a longer processing time for all error percentage. For its part, the conjugated 2×2 protocol exhibits a higher execution time than its non-conjugated counterpart, possibly due to the computation of the conjugated frames, but always less than the non-conjugated 3×2 protocol. When exceeding 90% error, the protocol loses the ability to maintain the bit rate, increasing its execution times.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 September 2022 doi:10.20944/preprints202209.0485.v1

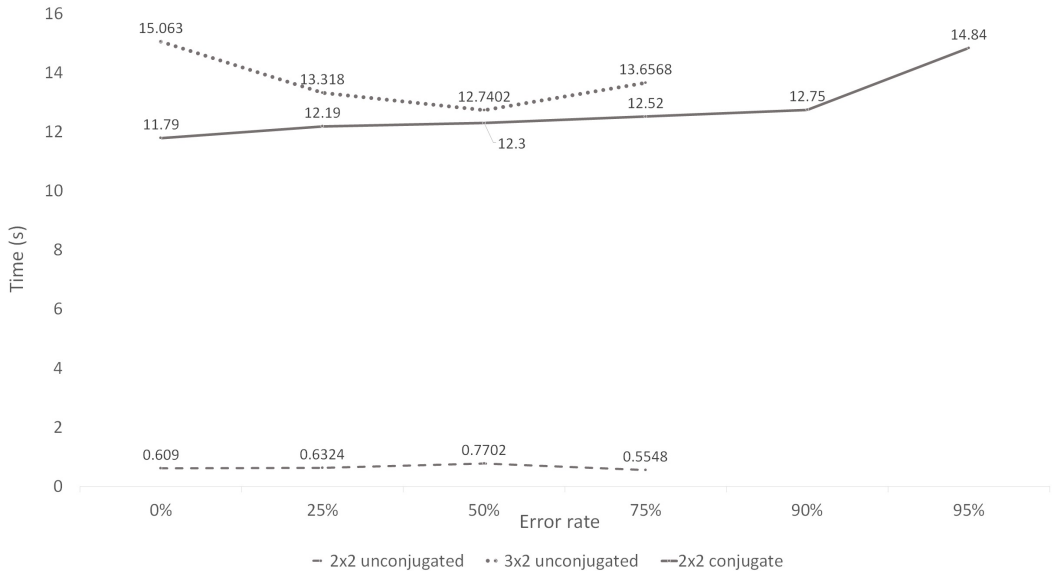

Fig. 6: Execution time of conjugated and non-conjugated protocols.

6 Conclusions

The results obtained with the conjugate frame reconciliation system show that the number of secret bits remains above 4500 bits in about 12 seconds, which is equivalent to a secret rate of 375 bits per second when the error rate of the channel is up to 90% and the number of double detection events is only 100. If the error percentage reaches 95%, the number of secret bits is still above 3500 bits in 15 seconds.

To the best of our knowledge, in QKD never before has it been possible to distill secret bits to 95% error. Therefore, these numbers constitute an important achievement in QKD technology. Although the results were obtained using simulation software, we have emphasized that what we have really simulated is the behavior of the quantum channel, since the information processing occurs after the quantum transfer stage. Therefore, our distillation software can be implemented on commercial equipment that can prepare/measure non-orthogonal quantum states.

A Appendix

In this appendix we will explain some details of the interfaces of the reconciliation system based on conjugate frames.

Transmitter Node (Alice). The transmitter interface contains some elements for the manipulation of the system. Within this interface, we will locate the following components:

— IP Address: Static IP assignment in the sending node for communication between two different devices.
— Port: Port number specification for data exchange.
— Number of double detection events: This field specifies the number of double detection events required at the receiving station.

— Process Monitor: This field displays detailed information about the system processes.
— Table of Frames: It allows visualizing the combination of indexes and the frames built by them.
— CSS table: Shows the construction of the CSS strings of each constructed frame.
— Key: Displays the key resulting from the process, as well as its size..

Alice sends pairs of quantum states until the user-specified parameter in the number of double detections field is met. Once the process of transmitting quantum states is finished, Bob tells Alice his double detection events, with which Alice builds her frames and communicates to Bob the indices that identify the frames. Bob builds the frames indicated by Alice and sends back the CSS strings to her. Then Alice separates the CSS from the auxiliary frames and proceeds to identify the generated errors.

With this information, Alice begins to identify errors, performing this action on frames f_8, f_{12}, f_9, f_{10}, f_{13}, f_{14}, f_2, f_6, f_3, f_4, f_1, f_5. At the conclusion of this process, Alice has identified all the strings in Bob's frames, however, some strings that do not satisfy the security property: each CSS string must be mapped to at least two MR codes, otherwise they must be removed. Once Alice locates these strings, she tells Bob which frames need to be removed.

Now Alice is able to create the key, if the protocol does not present a symmetric key between both nodes, the process monitor will show an error message. At the end of the protocol run, statistics are calculated, which will be displayed in the graphical interface process monitor on Alice's station.

Receiver Node (Bob). The receiver interface shares similar elements with the sending node, such as the IP configuration, data transfer Port, Process Monitor and Key, however, the monitor of the receiving node registers limited activity in comparison of the sending node. In the receiver interface, we identify an important element.

— Noise percentage: Determines the percentage of error with which the quantum channel is established.

Bob's first process includes receiving, measuring, and recording the received quantum states. In this step, Bob generates error in the measurements of the quantum states following the percentage of error predefined by the user. Upon completion of this process and sharing the results of the double detection events, Bob receives the indices from Alice and builds the frames that she indicates. The next step is to generate the CSS strings, which are sent to the transmitting node (Alice). Once Alice sends the list of frames to be discarded, Bob proceeds to obtain the secret key. Finally, statistics are calculated and displayed in Bob's graphical interface process monitor.

References

1. R. Alléaume, C. Branciard, J. Bouda, T. Debuisschert, M. Dianati, N. Gisin, M. Godfrey, P. Grangier, T. Länger, N. Lütkenhaus, *et al.*, "Using quantum key distribution for cryptographic purposes: a survey," *Theoretical Computer Science*, vol. 560, pp. 62–81, 2014.
2. K. W. Hong, O.-M. Foong, and T. J. Low, "Challenges in quantum key distribution: a review," in *Proceedings of the 4th International Conference on Information and Network Security*, pp. 29–33, 2016.
3. D. Bacco, B. Da Lio, D. Cozzolino, F. Da Ros, X. Guo, Y. Ding, Y. Sasaki, K. Aikawa, S. Miki, H. Terai, *et al.*, "Boosting the secret key rate in a shared quantum and classical fibre communication system," *Communications Physics*, vol. 2, no. 1, pp. 1–8, 2019.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 September 2022 doi:10.20944/preprints202209.0485.v1

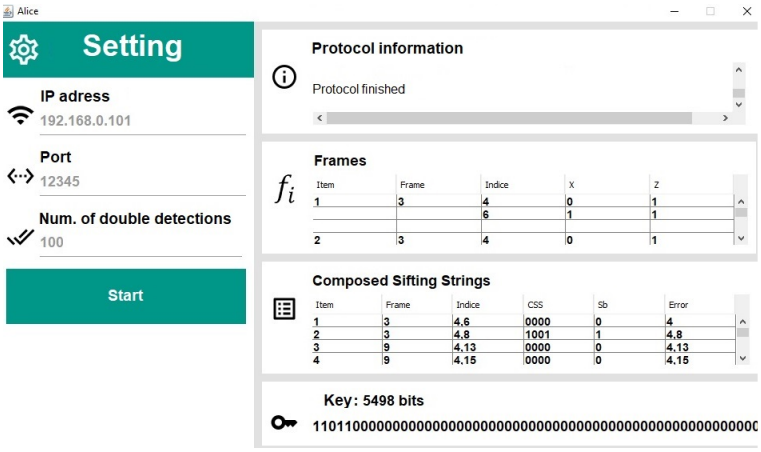

Fig. 7: Overview of Alice's interface when the error rate is 75%.

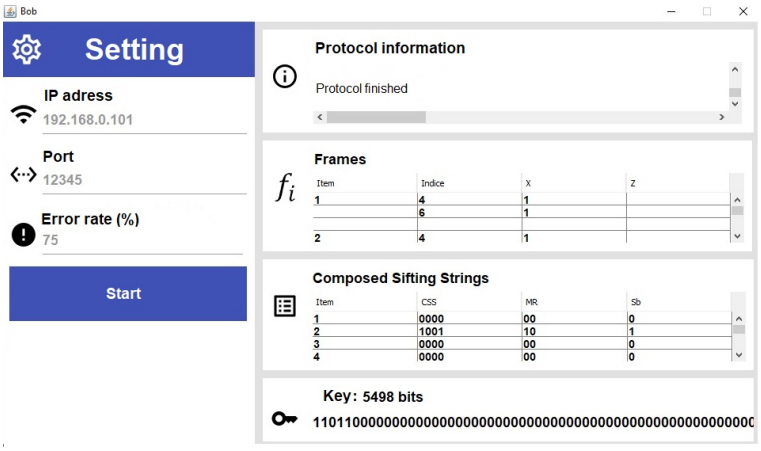

Fig. 8: Overview of Bob's interface when the error rate is 75%.

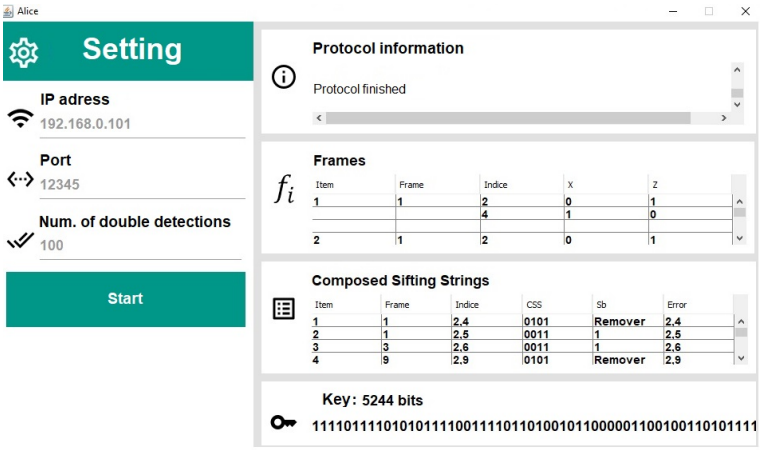

Fig. 9: Results showing the Alice interface with an error rate of 90%.

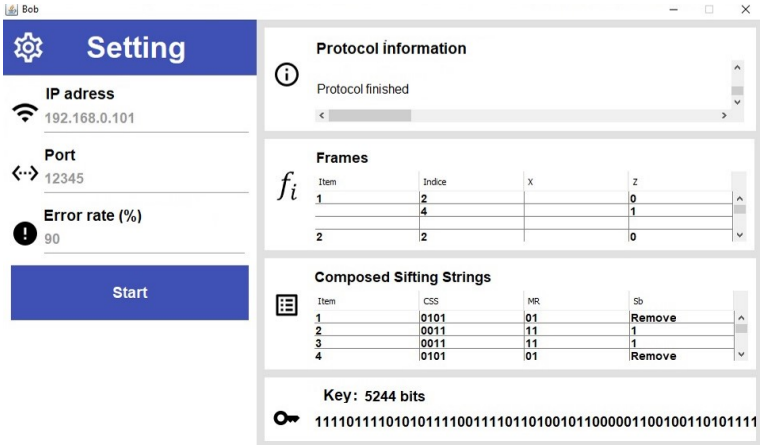

Fig. 10: Results showing the Bob interface with an error rate of 90%.

4. G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, C. Miller, D. Moody, R. Peralta, *et al.*, "Status report on the third round of the nist post-quantum cryptography standardization process," *US Department of Commerce, NIST*, 2022.
5. P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*, pp. 124–134, Ieee, 1994.
6. E. Diamanti, H.-K. Lo, B. Qi, and Z. Yuan, "Practical challenges in quantum key distribution," *npj Quantum Information*, vol. 2, no. 1, pp. 1–12, 2016.
7. T. Sasaki, Y. Yamamoto, and M. Koashi, "Practical quantum key distribution protocol without monitoring signal disturbance," *Nature*, vol. 509, no. 7501, pp. 475–478, 2014.
8. A. Mink and A. Nakassis, "Ldpc error correction for gbit/s qkd," in *Quantum Information and Computation XII*, vol. 9123, pp. 19–31, SPIE, 2014.
9. H. Yan, T. Ren, X. Peng, X. Lin, W. Jiang, T. Liu, and H. Guo, "Information reconciliation protocol in quantum key distribution system," in *2008 Fourth International Conference on Natural Computation*, vol. 3, pp. 637–641, IEEE, 2008.
10. G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion," in *Workshop on the Theory and Application of of Cryptographic Techniques*, pp. 410–423, Springer, 1993.
11. T. B. Pedersen and M. Toyran, "High performance information reconciliation for qkd with cascade," *arXiv preprint arXiv:1307.7829*, 2013.
12. R. G. Gallager, "Low-density parity-check codes," *Information Theory, IRE Transactions on*, vol. 8, no. 1, pp. 21–28, 1962.
13. A. Mink and A. Nakassis, "Ldpc for qkd reconciliation," *arXiv preprint arXiv:1205.4977*, 2012.
14. L. A. Lizama-Pérez, J. M. López, and E. De Carlos López, "Quantum key distribution in the presence of the intercept-resend with faked states attack," *Entropy*, vol. 19, no. 1, p. 4, 2016.
15. L. A. Lizama-Perez and J. M. López, "Quantum key distillation using binary frames," *Symmetry*, vol. 12, no. 6, p. 1053, 2020.
16. L. A. Lizama-Pérez, E. H. Samperio, *et al.*, "Beyond the limits of shannon's information in quantum key distribution," *Entropy*, vol. 23, no. 2, p. 229, 2021.
17. L. A. Lizama-Pérez and J. M. López-Romero, "Perfect reconciliation in quantum key distribution with order-two frames," *Symmetry*, vol. 13, no. 9, p. 1672, 2021.