Article **Journal of Computational Technologies**
December 2021, Vol.5 No.16 12-22

Post-quantum software for distillation of non-orthogonal quantum states through binary frames

Software post cuántico para destilación de estados cuánticos no ortogonales a través de frames binarios

SAMPERIO-GUZMAN, Emmanuel H. †*´, LIZAMA-PÉREZ, Luis A.´ and LÓPEZ-ROMERO, J. Mauricio´´

´Sección de Posgrado de la Universidad Politécnica de Pachuca, Ex-Hacienda de Santa Bárbara, 43830, México.

´´CINVESTAV Querétaro, Libramiento Norponiente 2000, Real de Juriquilla, 76230, Santiago de Querétaro, Querétaro, México.

ID 1st Author: *Emmanuel H., Samperio-Guzman* / **ORC ID:** 0000-0002-4531-9598, **CVU CONACYT ID:** 987790

ID 1st Co-author: *Luis A., Lizama-Pérez* / **ORC ID:** 0000−0001−5109−2927, **CVU CONACYT ID:** 204133

ID 2nd Co-author: *J. Mauricio, López-Romero* / **ORC ID:** 0000-0003-3435-9241

DOI: 10.35429/JOCT.2021.16.5.12.22 Received: July 15, 2021; Accepted: December 30, 2021

Abstract

Quantum cryptography is a paradigm for the establishment of secret keys and data confidentiality, which represents an alternative in the quantum era because its security properties are based on the principles of quantum physics. Unfortunately, errors that occur during transmission and detection of quantum states have made it difficult to implement this technology globally. However, a new cryptographic key quantum distribution scheme based on non-orthogonal state pairs has recently been published which considerably outperforms known schemes. This article describes the fundamentals of this protocol which are represented as an algorithm and the pseudo-code of the most relevant functions of the system is shown; The current development of the software for the distillation of non-orthogonal quantum states by means of binary frames is presented, which demonstrates the transmission control, reconciliation and privacy amplification of the shared secret bits. Likewise, we present the results obtained from the computer system and its interpretation in relation to the efficiency of the protocol, which exceeds 50% channel error rates and a quadratic growth of the length of the secret key as a function of the number of double detection events. Objectives: Demonstrate the effectiveness of the non-orthogonal state distillation protocol through binary frames using the software developed. Methodology: For the development of this project, the following methodology has been carried out (see Figure 1). Contribution: The results of this software guide tests for quantum distillation in an experimental communications environment in order to provide a useful solution in the era of quantum information transmission and communication technologies.

Software, QKD, Non-OrthogonalStates

Resumen

La criptógrafa cuántica es un paradigma para el establecimiento de llaves secretas y confidencialidad de datos, el cual representa una alternativa en la era cuántica debido a que sus propiedades de seguridad se basan en los principios de la física cuántica. Desafortunadamente, los errores que se producen durante la transmisión y la detección de estados cuánticos han dificultado la implementación de esta tecnología a nivel global. No obstante, un nuevo esquema de distribución cuántica de llave criptográfica basado en pares de estados no ortogonales ha sido publicado recientemente el cuál supera considerablemente los esquemas conocidos. En este artículo se describen los fundamentos de este protocolo los cuales se representan como algoritmo y se muestra el pseudocódigo de las funciones más relevantes del sistema; se presenta el desarrollo actual del software de destilación de estados cuánticos no ortogonales por medio de frames binarios con lo que se demuestra el control de transmisión, reconciliación y amplificación de privacidad de los bits secretos compartidos. Así mismo, damos a conocer los resultados obtenidos del sistema informático y su interpretación en relación con la eficiencia del protocolo, la cual supera tasas del 50% de error en canal y un crecimiento cuadrático de la longitud de la llave secreta en función del número de eventos de detección doble. Objetivos: Demostrar la efectividad del protocolo de destilación de estados no ortogonales a través de frames binarios mediante el software desarrollado. Metodología: Para el desarrollo de este proyecto, se ha llevado a cabo la siguiente metodología (Ver Figura 1). Contribución; Los resultados de este software dan pauta a pruebas para la destilación cuántica en un entorno experimental de comunicaciones con la finalidad de brindar una solución útil en la era de las tecnologías de la comunicación y transmisión de información cuántica.

Software, QKD, Estados no ortogonales

Citation: SAMPERIO-GUZMAN, Emmanuel H., LIZAMA-PÉREZ, Luis A. and LÓPEZ-ROMERO, J. Mauricio. Post-quantum software for distillation of non-orthogonal quantum states through binary frames. Journal of Computational Technologies. 2021. 5-16:12-22.

* Correspondence to Author: (E-mail: esamperio593@micorreo.upp.edu.mx)
† Researcher contributed as first author.

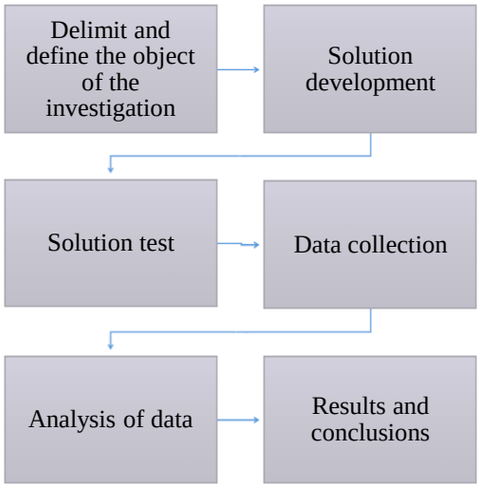


Figure 1 Scheme of the implemented methodology

Introduction

Nowadays, quantum technologies have gained great impact and importance in the field of telecommunications and cybersecurity as exposed by Peter W. Shor in 1999 [1], quantum algorithms are theoretically capable of breaking the security of classical cryptographic systems.

However, the developments carried out so far to transmit quantum pulses secretly through a quantum channel is functional under certain scenarios, obtaining different results in terms of efficiency, accuracy and the security they provide [2-6]. The purpose of quantum cryptography systems is to establish a cryptographic key through a sequence of bits shared between the transmitter and receiver (called Alice and Bob), in order to encrypt a message that in case of being intercepted by an attacker (called Eve), the attacker cannot recover or interpret the original message.

In this scheme, one of the most promising methods for the quantum era consists in sending quantum states to establish a cryptographic key, since thanks to the Heisenberg uncertainty principle [7] and the non-cloning theorem [8], it is presumed that Eve cannot measure these states, copy them and retransmit them without altering their original state. More importantly, if Eve has intercepted these quantum states, its presence in the quantum channel can be detected and the protocol would be disrupted [9].

The operation of this protocol is based on the transmission of pairs of non-orthogonal quantum states, with which binary frames are formed and one of the most interesting properties is that it can operate with channel error rates beyond the limits theorized by Claude Shannon in 1948 [10], managing to generate a secret key in channels with a noise level higher than 50% Unlike its counterpart, BB84, which has been estimated to be able to distill a secret key for quantum bit error rates (QBER) lower than 11% [11].

Quantum key distillation using 2×2 Binary Frames

This is a distillation method that performs the processes of tuning, reconciliation (error correction) and security amplification in a single process by using binary frames. This method is able to increase the percentage of the secret key, since the spy has no control over Bob's detection events, making it secure against the Intercept and Forward (IR) attack and the Photon Numer Split (PNS) attack. Another advantage of this method is that since it is a generalization of the Bennett-Brassard protocol of 1984 (BB84), it can be implemented with the usual optical equipment that produces strong quantum pulses.

Key exchange using quantum streams

In this process, Alice randomly sends pairs of non-orthogonal quantum states to Bob (Figure 2). In turn, Bob measures Alice's quantum states by randomly using a measurement basis, X or Z. If after Bob has measured the pair of quantum states he obtains the same result, a Double Matching (DM) event occurs, successfully transmitting a bit from Alice to Bob. So it is implied that two quantum states are used to encode a single bit.

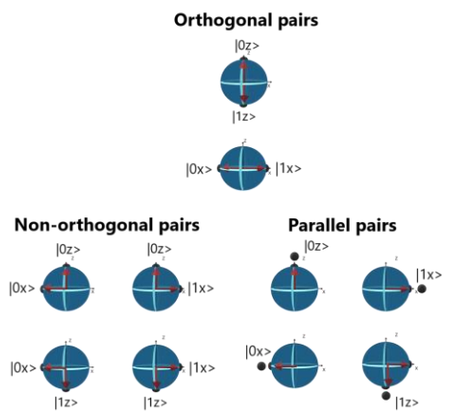


Figure 2 Representation of quantum state pairs

For this scheme, Bob only announces to Alice the indices of events that have produced double coincidence (DM) events, as shown in the schematic in Figure 3.

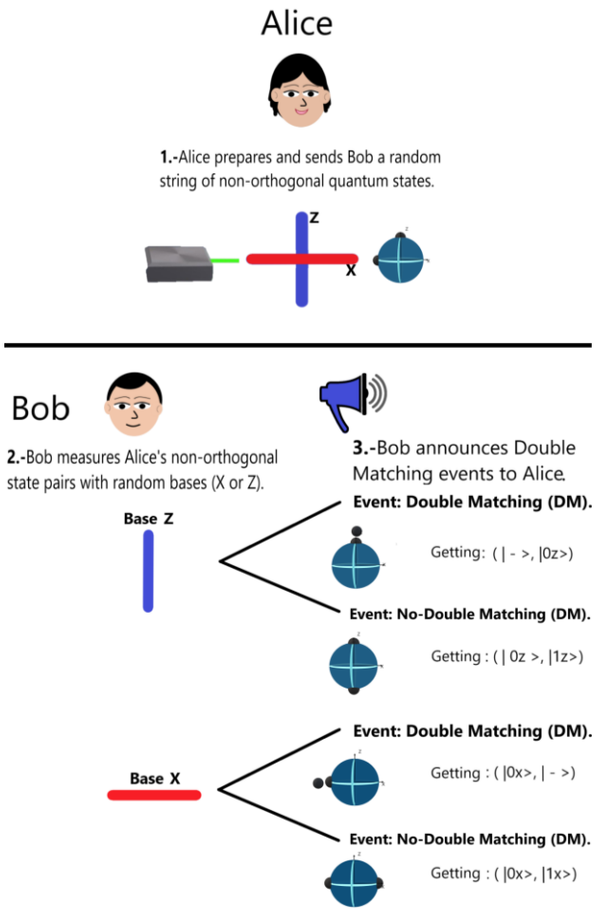


Figure 3 Representation of the detection events produced by Bob's readings

Building binary frames from quantum states

At the conclusion of the detection events, Bob announces to Alice the indices of the DM events with which he proceeds to form the necessary frames for the reconciliation process. Being a 2×2 frame we have 2^4 possible frames, which are shown in Table 1 and are numbered from 1 to 16, these frames are classified into two categories: regular frames and general frames. Each row of a frame contains the cubits of the non-orthogonal pair sent by Alice with left-side X-base and right-side Z-base. After Bob measures a pair of non-orthogonal quantum states and achieves a double coincidence event, he gets one bit per row within a frame, thus two bits per frame in total.

Regular frames	
$f_1 = \begin{pmatrix} \|0x\rangle & \|1z\rangle \\ \|1x\rangle & \|0z\rangle \end{pmatrix}$	$f_2 = \begin{pmatrix} \|1x\rangle & \|0z\rangle \\ \|1x\rangle & \|1z\rangle \end{pmatrix}$
$f_3 = \begin{pmatrix} \|0x\rangle & \|1z\rangle \\ \|1x\rangle & \|1z\rangle \end{pmatrix}$	$f_4 = \begin{pmatrix} \|1x\rangle & \|1z\rangle \\ \|0x\rangle & \|1z\rangle \end{pmatrix}$
$f_5 = \begin{pmatrix} \|1x\rangle & \|0z\rangle \\ \|0x\rangle & \|1z\rangle \end{pmatrix}$	$f_6 = \begin{pmatrix} \|1x\rangle & \|1z\rangle \\ \|1x\rangle & \|0z\rangle \end{pmatrix}$
General frames	
$f_7 = \begin{pmatrix} \|0x\rangle & \|0z\rangle \\ \|0x\rangle & \|0z\rangle \end{pmatrix}$	$f_{12} = \begin{pmatrix} \|1x\rangle & \|1z\rangle \\ \|0x\rangle & \|0z\rangle \end{pmatrix}$
$f_8 = \begin{pmatrix} \|0x\rangle & \|0z\rangle \\ \|1x\rangle & \|1z\rangle \end{pmatrix}$	$f_{13} = \begin{pmatrix} \|0x\rangle & \|0z\rangle \\ \|0x\rangle & \|1z\rangle \end{pmatrix}$
$f_9 = \begin{pmatrix} \|0x\rangle & \|1z\rangle \\ \|0x\rangle & \|0z\rangle \end{pmatrix}$	$f_{14} = \begin{pmatrix} \|0x\rangle & \|0z\rangle \\ \|1x\rangle & \|0z\rangle \end{pmatrix}$
$f_{10} = \begin{pmatrix} \|1x\rangle & \|0z\rangle \\ \|0x\rangle & \|0z\rangle \end{pmatrix}$	$f_{15} = \begin{pmatrix} \|0x\rangle & \|1z\rangle \\ \|0x\rangle & \|1z\rangle \end{pmatrix}$
$f_{11} = \begin{pmatrix} \|1x\rangle & \|1z\rangle \\ \|1x\rangle & \|1z\rangle \end{pmatrix}$	$f_{16} = \begin{pmatrix} \|1x\rangle & \|0z\rangle \\ \|1x\rangle & \|0z\rangle \end{pmatrix}$

Table 1 Usable and general 2×2 frames

Matching results

Each frame sent by Alice has a setting for Bob called Matching Result or MR, which is composed of two bits. Table 2 lists the four possible MRs for Bob and each MR contains two bits which encode its base. This matching process is intended for Alice to be able to identify Bob's RMs.

$MR = 00 \begin{pmatrix} \|\bullet x\rangle & — \\ \|\bullet x\rangle & — \end{pmatrix}$
$MR = 01 \begin{pmatrix} — & \|\bullet z\rangle \\ — & \|\bullet z\rangle \end{pmatrix}$
$MR = 10 \begin{pmatrix} \|\bullet x\rangle & — \\ — & \|\bullet z\rangle \end{pmatrix}$
$MR = 11 \begin{pmatrix} — & \|\bullet z\rangle \\ \|\bullet x\rangle & — \end{pmatrix}$

Table 2 Matching results (MR)

Sifting method using XOR function

To calculate the adjustment bits or Sifting bits (Sb), the XOR function must be applied to the vertical bits within each column of the frame, taking the empty states as a null bit. Sifting bits (Sb) are written at the bottom of each RM as shown in Table 3.

ISSN 2523-6814

The most important property of the set bits is that they must not be redundant, otherwise they will become ambiguous. Therefore, the Sb's of a given frame must not be derived from different MR bases. This condition can be verified in Table 3 and Table 3', where it is shown that the Sb's define a complete set (without repetitions) over the XOR function applied to each frame. At this point, it is shown that not all frames can be used during the filtering process. We can now list the protocol steps to obtain the Sb's:

1. Alice prepares and sends to Bob pairs of non-orthogonal cubits and sends them to Bob over the quantum channel, selecting a random string from among $(|0x\rangle, |0z\rangle)$; $(|0x\rangle, |1z\rangle)$; $(|1x\rangle, |0z\rangle)$ y $(|1x\rangle, |1z\rangle)$.

2. Bob measures Alice's non-orthogonal pairs with random bases (X or Z).

3. At the end of the measurements, Bob announces to Alice the events with double matching (DM).

4. Alice prepares usable frames (Alice knows which cubit pairs are contained in a frame).

As a result, the bits shared by Alice and Bob are the bits that encode each RM, according to Table 2.

Alice	Bob	
$f_1 = \begin{pmatrix} \|0x\rangle & \|1z\rangle \\ \|1x\rangle & \|0z\rangle \end{pmatrix}$	$\begin{pmatrix} \|0x\rangle & - \\ \|1x\rangle & - \end{pmatrix}$ $\begin{matrix} 1 & 0 \end{matrix}$	$\begin{pmatrix} - & \|1z\rangle \\ - & \|0z\rangle \end{pmatrix}$ $\begin{matrix} 0 & 1 \end{matrix}$
$f_2 = \begin{pmatrix} \|1x\rangle & \|0z\rangle \\ \|1x\rangle & \|1z\rangle \end{pmatrix}$	$\begin{pmatrix} \|1x\rangle & - \\ \|1x\rangle & - \end{pmatrix}$ $\begin{matrix} 0 & 0 \end{matrix}$	$\begin{pmatrix} - & \|0z\rangle \\ - & \|0z\rangle \end{pmatrix}$ $\begin{matrix} 0 & 1 \end{matrix}$
$f_3 = \begin{pmatrix} \|0x\rangle & \|1z\rangle \\ \|1x\rangle & \|1z\rangle \end{pmatrix}$	$\begin{pmatrix} \|0x\rangle & - \\ \|1x\rangle & - \end{pmatrix}$ $\begin{matrix} 1 & 0 \end{matrix}$	$\begin{pmatrix} - & \|1z\rangle \\ - & \|1z\rangle \end{pmatrix}$ $\begin{matrix} 0 & 0 \end{matrix}$
$f_4 = \begin{pmatrix} \|1x\rangle & \|1z\rangle \\ \|0x\rangle & \|1z\rangle \end{pmatrix}$	$\begin{pmatrix} \|1x\rangle & - \\ \|0x\rangle & - \end{pmatrix}$ $\begin{matrix} 1 & 0 \end{matrix}$	$\begin{pmatrix} - & \|1z\rangle \\ - & \|1z\rangle \end{pmatrix}$ $\begin{matrix} 0 & 0 \end{matrix}$
$f_5 = \begin{pmatrix} \|1x\rangle & \|0z\rangle \\ \|0x\rangle & \|1z\rangle \end{pmatrix}$	$\begin{pmatrix} \|1x\rangle & - \\ \|0x\rangle & - \end{pmatrix}$ $\begin{matrix} 1 & 0 \end{matrix}$	$\begin{pmatrix} - & \|0z\rangle \\ - & \|1z\rangle \end{pmatrix}$ $\begin{matrix} 0 & 1 \end{matrix}$
$f_6 = \begin{pmatrix} \|1x\rangle & \|1z\rangle \\ \|1x\rangle & \|0z\rangle \end{pmatrix}$	$\begin{pmatrix} \|1x\rangle & - \\ \|1x\rangle & - \end{pmatrix}$ $\begin{matrix} 0 & 0 \end{matrix}$	$\begin{pmatrix} - & \|1z\rangle \\ - & \|0z\rangle \end{pmatrix}$ $\begin{matrix} 0 & 1 \end{matrix}$
$f_1 = \begin{pmatrix} \|0x\rangle & \|1z\rangle \\ \|1x\rangle & \|0z\rangle \end{pmatrix}$	$\begin{pmatrix} - & \|1z\rangle \\ \|1x\rangle & - \end{pmatrix}$ $\begin{matrix} 1 & 1 \end{matrix}$	$\begin{pmatrix} \|0x\rangle & - \\ - & \|0z\rangle \end{pmatrix}$ $\begin{matrix} 0 & 0 \end{matrix}$
$f_2 = \begin{pmatrix} \|1x\rangle & \|0z\rangle \\ \|1x\rangle & \|1z\rangle \end{pmatrix}$	$\begin{pmatrix} - & \|0z\rangle \\ \|1x\rangle & - \end{pmatrix}$ $\begin{matrix} 1 & 0 \end{matrix}$	$\begin{pmatrix} \|1x\rangle & - \\ - & \|1z\rangle \end{pmatrix}$ $\begin{matrix} 1 & 1 \end{matrix}$
$f_3 = \begin{pmatrix} \|0x\rangle & \|1z\rangle \\ \|1x\rangle & \|1z\rangle \end{pmatrix}$	$\begin{pmatrix} - & \|1z\rangle \\ \|1x\rangle & - \end{pmatrix}$ $\begin{matrix} 1 & 1 \end{matrix}$	$\begin{pmatrix} \|0x\rangle & - \\ - & \|1z\rangle \end{pmatrix}$ $\begin{matrix} 0 & 1 \end{matrix}$
$f_4 = \begin{pmatrix} \|1x\rangle & \|1z\rangle \\ \|0x\rangle & \|1z\rangle \end{pmatrix}$	$\begin{pmatrix} - & \|1z\rangle \\ \|0x\rangle & - \end{pmatrix}$ $\begin{matrix} 0 & 1 \end{matrix}$	$\begin{pmatrix} \|1x\rangle & - \\ - & \|1z\rangle \end{pmatrix}$ $\begin{matrix} 1 & 1 \end{matrix}$
$f_5 = \begin{pmatrix} \|1x\rangle & \|0z\rangle \\ \|0x\rangle & \|1z\rangle \end{pmatrix}$	$\begin{pmatrix} - & \|0z\rangle \\ \|0x\rangle & - \end{pmatrix}$ $\begin{matrix} 0 & 0 \end{matrix}$	$\begin{pmatrix} \|1x\rangle & - \\ - & \|1z\rangle \end{pmatrix}$ $\begin{matrix} 1 & 1 \end{matrix}$
$f_6 = \begin{pmatrix} \|1x\rangle & \|1z\rangle \\ \|1x\rangle & \|0z\rangle \end{pmatrix}$	$\begin{pmatrix} - & \|1z\rangle \\ \|1x\rangle & - \end{pmatrix}$ $\begin{matrix} 1 & 1 \end{matrix}$	$\begin{pmatrix} \|1x\rangle & - \\ - & \|0z\rangle \end{pmatrix}$ $\begin{matrix} 1 & 0 \end{matrix}$

Table 3 Sb calculation

Error correction method

We define the Sifting String (SS) as a binary string (Table 4) composed of the set bits (Sb) and the measured bits or Mesure bit (Mb). For each frame a SS is constructed as explained below: The Sb are written from left to right separated by a comma and then the measured bits taken from top to bottom are written.

Mesure bits	0	—
	1	—
Adjustment bits (Sb)	1	0

$$SS = (1^{er}\ Sb \,||\, 2^{do}\ Sb, 1^{er}\ Mb \,||\, 2^{do}\ Mb)$$
$$SS = 10{,}01$$

Table 4 Representation for a string of fit (SS).

The set string (SS) is sent to Alice and with this it helps to recognize if some of the measured bits are wrong since she knows the possible values of the SS that Bob can get. In Table 5 and Table 5' we can observe the valid SS without error and the possible cases with error that Bob can obtain. For the cases where the SS can change because of an error or an intrusion, it can happen that the SS formed can go unnoticed as a valid SS, as well as there are also cases where an erroneous SS is immediately identified.

Frame	SS Valid	MR	1st bit	error
f_1	$ss_{11} = 00{,}00$	10	10,10	Yes
	$ss_{12} = 01{,}10$	01	00,00	No
	$ss_{13} = 10{,}01$	00	00,11	Yes
	$ss_{14} = 11{,}11$	11	10,01	No
f_2	$ss_{21} = 00{,}11$	00	10,01	No
	$ss_{22} = 01{,}01$	01	00,11	No
	$ss_{23} = 10{,}01$	11	11,11	No
	$ss_{24} = 11{,}11$	10	01,01	No
f_3	$ss_{31} = 00{,}11$	01	01,01	No
	$ss_{32} = 01{,}01$	10	11,11	No
	$ss_{33} = 10{,}01$	00	00,11	No
	$ss_{34} = 11{,}11$	11	10,01	No
f_4	$ss_{41} = 00{,}11$	01	01,10	Yes
	$ss_{42} = 01{,}10$	11	11,11	Yes
	$ss_{43} = 10{,}10$	00	00,11	Yes
	$ss_{44} = 11{,}11$	10	10,10	Yes
f_5	$ss_{51} = 00{,}00$	11	01,10	Yes
	$ss_{52} = 01{,}01$	01	00,11	Yes
	$ss_{53} = 10{,}10$	00	00,00	No
	$ss_{54} = 11{,}11$	10	01,01	No

Frame	2nd bit	Error	1st and 2nd bit	Error
f_1	01,01	Yes	11,11	No
	00,11	Yes	01,01	Yes
	00,00	No	1,10	Yes
	01,10	No	00,00	No
f_2	10,10	Yes	00,00	Yes
	00,00	Yes	01,10	Yes
	00,00	Yes	01,10	Yes
	10,10	Yes	00,00	Yes

f_3	01,10 00,00 00,00 01,10	Yes Yes Yes Yes	00,00 10,10 ,10,10 00,00	Yes Yes Yes Yes
f_4	01,10 11,11 00,11 10,10	No No No No	00,00 10,01 10,01 00,00	Yes Yes Yes Yes
f_5	10,01 00,00 00,11 10,10	Yes No Yes No	11,11 01,10 10,01 00,00	No Yes Yes No
f_6	10,10 00,11 11,11 01,10	No No No No	00,00 01,01 01,01 00,00	Yes Yes Yes Yes

Table 5 Error detection

Privacy pre-amplification

To obtain secret bits in case of dissonances in the channel, it is necessary to perform a method to identify the errors that may arise.

Errors in the channel may cause the bits obtained by Bob to be different from those sent by Alice. These errors can be identified by special frames called auxiliary frames and null frames.

Considering the following example in which Alice sends f_3 to Bob, who receives it with MR=10 and responds with SS=11,11. In Table 6 Alice's frame is represented as f_3a. On Bob's side, if the frame is error free we identify it as f_3b and in case of error we can identify it as f(3b^').

Case	MR	Frame	Sifting String
f_{3a}		$\begin{pmatrix} \|0x\rangle & \|1z\rangle \\ \|1x\rangle & \|1z\rangle \end{pmatrix}$	
f_{3b}	11	$\begin{pmatrix} — & \|1z\rangle \\ \|1x\rangle & — \end{pmatrix}$ 1 1	11,11
$f_{3b'}$	10	$\begin{pmatrix} \|1x\rangle & — \\ — & \|1z\rangle \end{pmatrix}$ 1 1	11,11

Table 6 Cases with errors

We can observe that for f_(3b^') the SS=11,11 has an error in the first double matching (DM) event when ├ |0z⟩ is detected as ├ |1z⟩ for MR= 10, which produces an ambiguity for Alice since up to this point she is not able to know if this SS is from a valid case or comes from a case with error. Given this ambiguity it is necessary to validate these cases and ensure that both Alice and Bob are able to generate an identical secret key for both.

Auxiliary frames and null frames

To rule out ambiguities a validation method has been designed which consists of using frame f_7 with states ├ |0x⟩ and ├ |0z⟩ which produce a SS= 00,00 otherwise the error is detected in these states. Frames f_9 and f_10 are used as auxiliaries (table 7 and table 7'), with which we can identify whether the DM events measured by Bob correspond correctly with the quantum states sent by Alice. In the case of f_9 it is used to identify errors in ├ |0x⟩ states that are measured as ├ |1x⟩ and f_10 is used to detect errors in ├ |0z⟩ states measured as ├ |1z⟩.

To validate the auxiliary frames, it is necessary to validate the states ├ |0x⟩ and ├ |0z⟩ using f_7. In cases where Bob measures a ├ |0x⟩ state instead of a ├ |1x⟩. error can be identified using f_9. On the other hand if Bob measures a ├ |0z⟩ state instead of a ├ |1z⟩. This error can be identified using f_10. These cases are demonstrated in the third and fifth rows. The error is represented by a slash over the states: $|\overline{0x}\rangle$ en $f_{9'}$ y $|\overline{1z}\rangle$ en $f_{10'}$.

Alice	Bob	
$f_7 = \begin{pmatrix} \|0x\rangle & \|0z\rangle \\ \|0x\rangle & \|0z\rangle \end{pmatrix}$	$\begin{pmatrix} \|0x\rangle & — \\ \|0x\rangle & — \end{pmatrix}$ 0 0 $SS = 00{,}00$	$\begin{pmatrix} — & \|0z\rangle \\ — & \|0z\rangle \end{pmatrix}$ 0 0 $SS = 00{,}00$
$f_9 = \begin{pmatrix} \|0x\rangle & \|1z\rangle \\ \|0x\rangle & \|0z\rangle \end{pmatrix}$	$\begin{pmatrix} \|0x\rangle & — \\ \|0x\rangle & — \end{pmatrix}$ 0 0 $SS = 00{,}00$	$\begin{pmatrix} — & \|1z\rangle \\ — & \|0z\rangle \end{pmatrix}$ 0 1 $SS = 01{,}10$
$f_{9a} = \begin{pmatrix} \|\overline{0x}\rangle & \|1z\rangle \\ \|0x\rangle & \|0z\rangle \end{pmatrix}$	$\begin{pmatrix} \|\overline{1x}\rangle & — \\ \|0x\rangle & — \end{pmatrix}$ 1 0 $SS = 10{,}10$	$\begin{pmatrix} — & \|1z\rangle \\ — & \|0z\rangle \end{pmatrix}$ 0 1 $SS = 01{,}10$
$f_{10} = \begin{pmatrix} \|1x\rangle & \|0z\rangle \\ \|0x\rangle & \|0z\rangle \end{pmatrix}$	$\begin{pmatrix} \|1x\rangle & — \\ \|0x\rangle & — \end{pmatrix}$ 1 0 $SS = 10{,}10$	$\begin{pmatrix} — & \|0z\rangle \\ — & \|0z\rangle \end{pmatrix}$ 0 0 $SS = 00{,}00$
$f_{10a} = \begin{pmatrix} \|1x\rangle & \|\overline{0z}\rangle \\ \|0x\rangle & \|0z\rangle \end{pmatrix}$	$\begin{pmatrix} \|1x\rangle & — \\ \|0x\rangle & — \end{pmatrix}$ 1 0 $SS = 10{,}10$	$\begin{pmatrix} — & \|\overline{1z}\rangle \\ — & \|0z\rangle \end{pmatrix}$ 0 1 $SS = 01{,}10$
$f_7 = \begin{pmatrix} \|0x\rangle & \|0z\rangle \\ \|0x\rangle & \|0z\rangle \end{pmatrix}$	$\begin{pmatrix} — & \|0z\rangle \\ \|0x\rangle & — \end{pmatrix}$ 0 0 $SS = 00{,}00$	$\begin{pmatrix} \|0x\rangle & — \\ — & \|0z\rangle \end{pmatrix}$ 0 0 $SS = 00{,}00$
$f_9 = \begin{pmatrix} \|0x\rangle & \|1z\rangle \\ \|0x\rangle & \|0z\rangle \end{pmatrix}$	$\begin{pmatrix} — & \|1z\rangle \\ \|0x\rangle & — \end{pmatrix}$ 0 1 $SS = 01{,}10$	$\begin{pmatrix} \|0x\rangle & — \\ — & \|0z\rangle \end{pmatrix}$ 0 0 $SS = 00{,}00$
$f_{9a} = \begin{pmatrix} \|\overline{0x}\rangle & \|1z\rangle \\ \|0x\rangle & \|0z\rangle \end{pmatrix}$	$\begin{pmatrix} — & \|1z\rangle \\ \|0x\rangle & — \end{pmatrix}$ 0 1 $SS = 01{,}10$	$\begin{pmatrix} \|\overline{1x}\rangle & — \\ — & \|0z\rangle \end{pmatrix}$ 1 0 $SS = 10{,}10$
$f_{10} = \begin{pmatrix} \|1x\rangle & \|0z\rangle \\ \|0x\rangle & \|0z\rangle \end{pmatrix}$	$\begin{pmatrix} — & \|0z\rangle \\ \|0x\rangle & — \end{pmatrix}$ 0 0 $SS = 00{,}00$	$\begin{pmatrix} \|1x\rangle & — \\ — & \|0z\rangle \end{pmatrix}$ 1 0 $SS = 10{,}00$
$f_{10a} = \begin{pmatrix} \|1x\rangle & \|\overline{0z}\rangle \\ \|0x\rangle & \|0z\rangle \end{pmatrix}$	$\begin{pmatrix} — & \|\overline{1z}\rangle \\ \|0x\rangle & — \end{pmatrix}$ 0 1 $SS = 01{,}10$	$\begin{pmatrix} \|1x\rangle & — \\ — & \|0z\rangle \end{pmatrix}$ 1 0 $SS = 10{,}00$

Table 7 Auxiliary Frames

Continuing the explanation of Table 6, an ambiguity is observed in SS=11.11 of frame f_3b' when, Alice sends the frame to Bob who measures it using MR=10. However, when applying the measurement basis ┼ |z⟩, the photodetector produces ┼ |0z⟩ instead of ┼ |1z⟩; then we have (Table 8):

Frames correct	Wrong frame	SS
f_{3a}	$\begin{pmatrix} \|0x\rangle & \|1z\rangle \\ \|1x\rangle & \|1z\rangle \end{pmatrix}$	
$f_{3b'}$	$\begin{pmatrix} \|1x\rangle & — \\ — & \|1z\rangle \\ 1 & 1 \end{pmatrix}$	11,11

Table 8 Case of erroneous frames

When Alice receives the string SS=11,11 belonging to f_3, she knows that it implies two possibilities: SS comes from the error-free string SS_34= 11,11 under MR=11 or an error occurs in the first measured bit corresponding to the string SS_32= 01,01 under MR=10. To remove the ambiguity, Alice uses the auxiliary frame f_9. Therefore, she looks at the SS of the possible auxiliary frames formed by f_9 where the ambiguous index (-, ┼ |1z⟩) is assigned, forming a frame, with an index (┼ |0x⟩,-) that has been verified by a frame f_7. Then, Alice finds the following case (Table 9):

Frame auxiliar	Frame Bob	SS
f_9	$\begin{pmatrix} \|1x\rangle & — \\ — & \|0z\rangle \end{pmatrix}$	10,10

Table 9 Error identification using auxiliary frames

The adjustment string 10,10 reveals that there is an error for the index in question, therefore, Alice decides to use the secret bit that is generated from f_3 with SS_32. It should be noted that the adjustment strings of auxiliary frames are indistinguishable from other SS coming from general frames, thus ensuring privacy in this protocol.

Error correction model

In table 5 we find that there are errors that Alice is able to detect easily, however, there are others where it can find ambiguity. In Table 10 and Table 10' each frame is analyzed in order to clarify the ambiguity thanks to the use of auxiliary frames. However, there are some frames in which it is not possible to break the ambiguity, so it is necessary to remove the frame in question.

Frame	Non-orthogonal pair	Sifting String	Auxiliary frame
f_1	$(\|0x\rangle, \|\overline{1z}\rangle)$	$00, \bar{0}0$ $10, \bar{0}1$	— —
	$(\|\overline{1x}\rangle, \|0z\rangle)$	$00, 0\bar{0}$ $01, 1\bar{0}$	
f_2	$(\|\overline{1x}\rangle, \|0z\rangle)$	$10, \bar{0}1$ $01, \bar{0}1$	— f_{10}
	$(1x\rangle, \|\overline{0z}\rangle)$	$00, \bar{1}1$ $11, \bar{1}1$	
f_3	$(\|0x\rangle, \|\overline{1z}\rangle)$	$01, \bar{0}1$ $10, \bar{0}1$	— f_9
	$(\|\overline{0x}\rangle, \|1z\rangle)$	$11, \bar{1}1$ $00, \bar{1}1$	
f_4	$(\|1x\rangle, \|\overline{1z}\rangle)$	$01, 1\bar{0}$ $10, 1\bar{0}$	— f_9
	$(\|\overline{0x}\rangle, \|1z\rangle)$	$11, 1\bar{1}$ $00, 1\bar{1}$	
f_5	$(\|\overline{1x}\rangle, \|0z\rangle)$	$00, \bar{0}0$ $01, \bar{0}1$	— —
	$(0x\rangle, \|\overline{1z}\rangle)$	$00, 0\bar{0}$ $10, 1\bar{0}$	
f_6	$(\|1x\rangle, \|\overline{0z}\rangle)$	$00, 1\bar{1}$ $11, 1\bar{1}$	f_{10} —
	$(\|\overline{1x}\rangle, \|0z\rangle)$	$10, 1\bar{0}$ $01, 1\bar{0}$	
frame	Sifting String	bit erroneous	action
f_1	—	1st	Remove
	—	2nd	Remove
f_2	—	1st	Remove SS_{22}
	01,10	1st	SS_{23}
f_3	—	1st	Remove
	10,10	1st	SS_{32} SS_{33}
f_4	—	2nd	Remove
	10,10	2nd	SS_{42} SS_{43}
f_5	—	1st	Remove
	—	2nd	Remove
f_6	01,10	2nd	SS_{62} SS_{63}
	—	2nd	Remove

Table 10 Detection in ambiguity cases

Secret bit assignment

To preserve security, each SS must correlate with at least two MRs. At the end of the correction model, both Alice and Bob can obtain a string of secret bits, in Alice's case it is generated from the identification of Bob's SS and the frame used (Table 11).

For his part Bob relates his SS to his MR to obtain the secret key. For example, consider that Bob announces SS= 01,01, then there are two possible MR for this case: 10 and 11, In the case of Alice and Bob we have the secret bit equal to 0 for f_3 (Alice), MR =10 (Bob) and a secret bit equal to 1 for f_2 (Alice), MR=11 (Bob).

The security in this protocol is demonstrated since the frames are known only to Alice who can deduce the MR of Bob. With this model the SS can be shared through a public channel in a reliable way since it is correlated with either a secret bit 0 or 1 so that the attacker does not know from which SS each secret bit comes from.

Sifting String	MR	frame	sb
$SS21 = SS31 = SS41 = SS61 = 00,11$	00	f_2, f_6	0
$SS24 = SS34 = SS44 = SS64 = 11,11$	11	f_3, f_6	0
$SS42 = SS62 = 01,10$	01	f_6	0
$SS22 = SS32 = 01,01$	10	f_3	0
$SS23 = SS33 = 10,01$	00	f_3	0
$SS43 = SS63 = 10,10$	00	f_4	0
$SS21 = SS31 = SS41 = SS61 = 00,11$	01	f_3, f_4	1
$SS24 = SS34 = SS44 = SS64 = 11,11$	10	f_2, f_4	1
$SS42 = SS62 = 01,10$	11	f_4	1
$SS22 = SS32 = 01,01$	01	f_2	1
$SS23 = SS33 = 10,01$	11	f_2	1
$SS43 = SS63 = 10,10$	10	f_6	1

Table 11 Secret bit allocation

Algorithm

In this section we will detail the algorithm for the development of the test software for the 2×2 binary frame distillation protocol.

- Create and send list of pseudo-random pairs of Non-Orthogonal (NOT) states over the channel((├ |0x⟩, ├ |0z⟩); (├ |0x⟩, ├ |1z⟩), (├ |1x⟩, ├ |0z⟩) and (├ |1x⟩, ├ |1z⟩)).
- Reading of channel NO state pairs and list with enumeration of Double Coincidence Detection Events (EDDC).
- Reading channel EDDC enumeration list and frame construction.
    - Writing Frame Information (FI) to the channel.
- Receiving IF and building frames.
    - Obtaining and sending SS through classical channel.
- Receiving and reading SS:
    - Identification of removable ambiguity cases.
    - Identification and sending of non-removable ambiguity cases (output).
    - Obtaining key (output)
- Reading of non-removable ambiguity cases.
    - Elimination of cases
    - Obtaining key

Figure 4 briefly depicts the process carried out for this algorithm. Starting with Alice on the left who uses a laser photonic source to send pulses of non-orthogonal states to Bob through a quantum channel. In the middle we observe Eve, who intercepts Alice's states without knowing how they are polarized or the basis Bob will use. After Eve measures these states, she has to send a copy of the pulses to Bob, however, if after measuring the intercepted states they do not match, they become unusable. The case in which Eve's bases match Alice's bases and Bob's bases occurs with a probability of 1/4 for each pair of states that are intercepted.

On the left side we find Bob, who uses random x or z bases to measure the pairs of states sent by Alice. Once the agreed number of double matching (DM) events has been reached, Bob tells Alice the indices of these events to generate the 2×2 binary frames. In the included table we can see the continuation of the process from the measurements obtained by Bob up to obtaining a secret key for Alice and Bob.

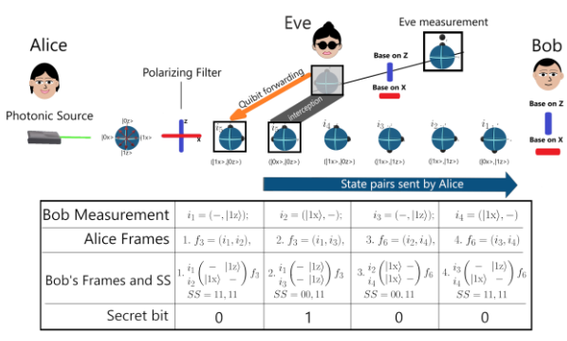


Figure 4 Quantum key distribution using 2×2 binary frames

It is worth mentioning that the fitting and reconciliation procedures are achieved by the computed fitting chains, while the privacy amplification is realized by the combination of all double detection events. Therefore, the distillation process is performed as an integral method. This property designed for this protocol is particular among current quantum key distribution technologies.

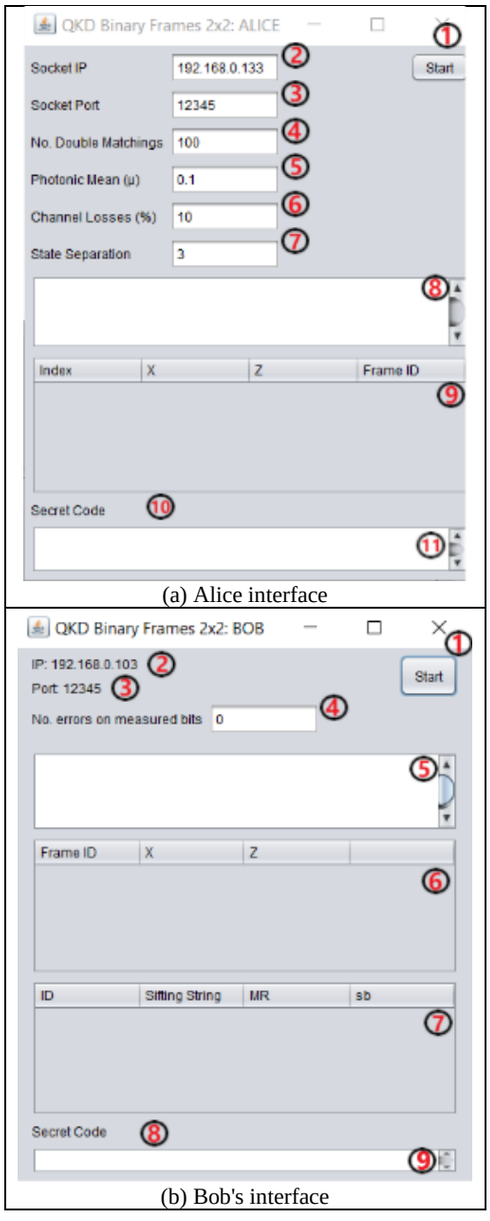
Software for distillation using 2×2 binary frames

This section will show the interface of the software developed to test the distillation protocol using 2×2 binary frames. We will start by describing the graphical user interface (GUI) of the Alice station, which is depicted in Figure 5a. In this interface we find ten important elements that are listed below:

1. Button to connect and start protocol.
2. Section for entering Bob's IP address.
3. Socket port to connect to Bob.
4. Selection of the number of double coincidence events.
5. Photon average value for the simulated laser source.
6. Simulation for noise loss in the quantum channel.
7. Separation of the pairs of non-orthogonal quantum states.
8. Panel of quantum state pairs sent, and statistics obtained.
9. Indices of bob's dm events and list of frames created by alice.
10. Number of secret bits generated.
11. String of generated secret bits.

The following list describes the components of Bob's interface shown in Figure 5b.

1. Connect and start protocol button.
2. Confirmation of IP connection to Alice.
3. Socket port to connect to Alice.
4. Panel to enter the simulated error rate on the communication channel.
5. Display panel for DM event rates.
6. Display of information about the frames created by Bob, as well as display of MR and SS.
7. Display panel for SS display and calculation of secret bits.
8. Number of secret bits generated.
9. String of secret bits generated.



(a) Alice interface

(b) Bob's interface

Figure 5 User interface for distillation software using 2×2 binary frames

Implementation of the error correction process

The structure of the pseudocode implemented in the error correction process for the 2×2 binary frame protocol software is shown below. This process corresponds to the reception and reading of the SS from Bob to Alice (Table 12).

Bob calculates its MR, from the frames indicated by Alice, returning the Sifting String (SS) of all the frames including the SS of the auxiliary frames.

5.- Correction of errors. (A)

Objectives:

(a) Create List of Removable Ambiguity Cases (LCAR).

- a. Identify Null Frames Without Errors
- i. List of Null Frames Without Error (LFNSE)
- b. Identification by Auxiliary Frames (FA)
- i. List of No-Fault Auxiliary Frames (LFAxSE)
- c. List of Cases Ambiguity Not Removable (LCANR)
- d. List of SS with Multiple Instance Error (LSSEMI)
- i. Identify SS Cases with Error in Multiple Instances (CSSEMI)

b) Correct Removable Ambiguity (RA) cases.

i. Table of cases

c) Generate key.

d) List of frames to be removed (LFR).

Input: SS list

Output: List of AR cases, list of frames to be removed and Key.

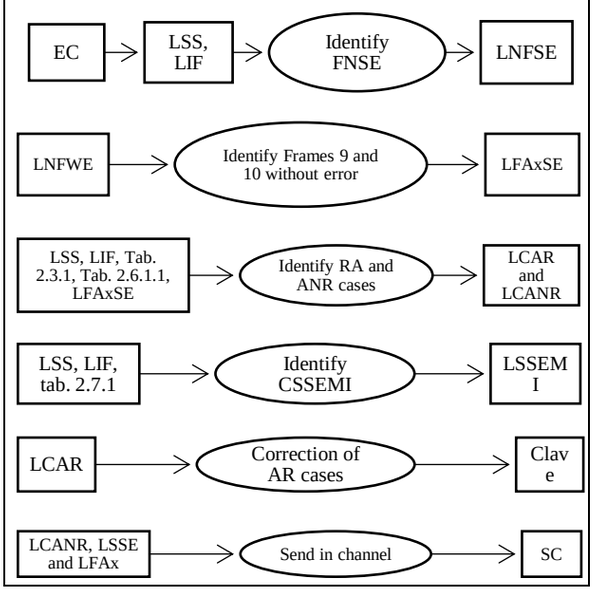
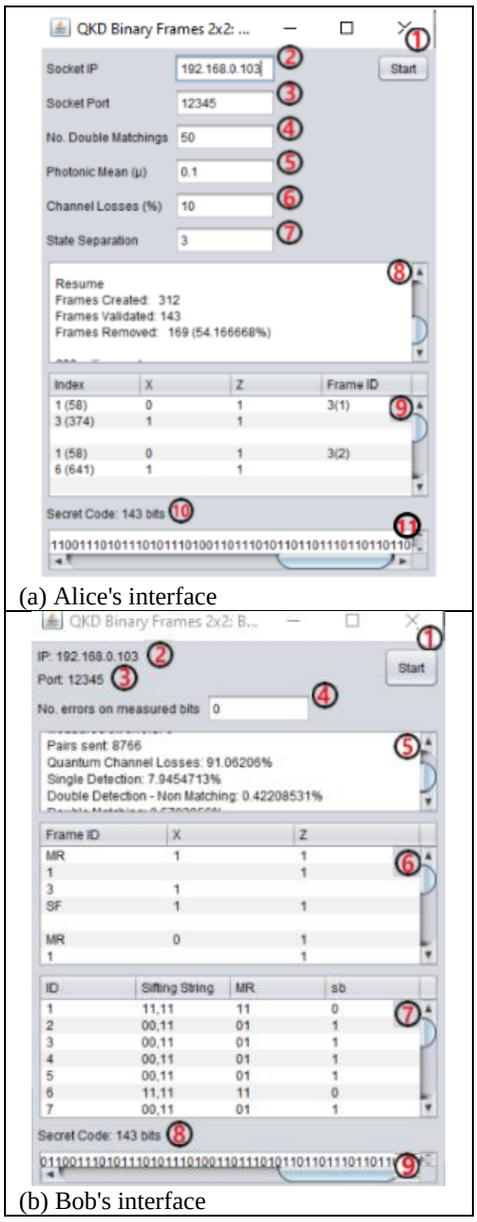


Table 12 Pseudocode implemented for error correction process

Results

The previous sections have presented the methodology and algorithm of this protocol, as well as the pseudocode for the error correction process developed for the non-orthogonal quantum state distillation software using 2×2 binary frames. Figure 6a shows the calculated results after a simulation test using 50 DM and a 10% error rate in the quantum channel. Section 8 shows a summary of the frames created, frames validated and frames removed and the execution time used for this test.



(a) Alice's interface

(b) Bob's interface

Figure 6 Execution of the 2×2 protocol software

Figure 6b shows the software interface with the results calculated for Bob, where point 5 shows a summary of the quantum states received as well as those that generated a DM event and the percentages of these events. Table 13 show the results of 10 tests performed with this distillation software. Figure 7 presents information corresponding to the total number of frames, frames removed and the generated number of secret bits when DM = 100. The total of secret bits is the difference between total frames and frames removed. The tests were carried out on a laptop with a 2.2 GHz Intel Core i7-8750H processor and 12 GB of RAM.

#	Frames created	Frames removed	Secret bits
1	1060	580	480
2	1377	756	621
3	1272	552	720
4	1121	532	589
5	1456	598	858
6	1392	725	667
7	1372	588	784
8	1496	782	714
9	1334	696	638
12	1325	500	825
Prom.	1320.5	630.9	689.6
#	Percentage gain	Time (ms)	Troughput (kbps)
1	45%	752	0. 63829786
2	45%	604	1.0281457
3	57%	592	1.2162162
4	53%	468	1.2585471
5	59%	388	2.2113402
6	48%	457	1.5864333
7	57%	490	1.2000000
8	48%	485	1.6123711
9	48%	479	1.4530271
12	62%	848	0.5896226
Prom.	52%	556.3	1.279400116

Table 13 Results of tests performed

Figure 7 Graph of results obtained

Since the number of frames after combinations is (DM¦2)=(DM(DM-1))/2, it implies that the information shared from detection events grows quadratically with respect to the number of DM events obtained. However, not all frames produce secret bits, only 1/8 of the total frames remain functional [12] to distill secret bits. The results produced by the software agree well with this prediction.

Acknowledgments

I thank the Consejo de Ciencia y Tecnología (CONACYT) and the Universidad Politécnica de Pachuca for their support and sponsorship to make possible the development of this project. Likewise, I am infinitely grateful to Dr. Luis Adrián Lizama Pérez for his guidance and help during this research, to my colleagues Daniel Mejía and Ivan Caballero for their invaluable collaboration and support.

Conclusions

The arrival of the quantum era is imminent, because quantum computers are a potential threat to the security of current data protection methods, QKD technology represents a valuable opportunity to establish secret keys and protect the confidentiality of communications for this new quantum era.

In this work, we have introduced a distillation software based on non-orthogonal state pairs using binary frames that has allowed us to evaluate the efficiency of this QKD protocol when using 2×2 frames. It has been shown for the first time, at least theoretically, that error correction is possible even when the error rate in the quantum channel exceeds 50%. Moreover, the rate of the secret key varies quadratically with respect to the number of DM events. The throughput of the QKD post-processing system reaches about 1 Kbps when DM = 100 and about 1 Mbps when DM = 1000 which represents optimal times for key generation.

References

1. Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, *41*(2), 303-332.

2. Mehic, M., Niemiec, M., Rass, S., Ma, J., Peev, M., Aguado, A., ... & Voznak, M. (2020). Quantum key distribution: a networking perspective. *ACM Computing Surveys (CSUR)*, *53*(5), 1-41.

3. Lovic, V. (2020). Quantum key distribution: Advantages, challenges and policy.

4. Pirandola, S., Andersen, U. L., Banchi, L., Berta, M., Bunandar, D., Colbeck, R., ... & Wallden, P. (2020). Advances in quantum cryptography. *Advances in Optics and Photonics*, *12*(4), 1012-1236.

5. Razavi, M., Leverrier, A., Ma, X., Qi, B., & Yuan, Z. (2019). Quantum key distribution and beyond: introduction. *JOSA B*, *36*(3), QKD1-QKD2.

6. Xu, F., Ma, X., Zhang, Q., Lo, H. K., & Pan, J. W. (2020). Secure quantum key distribution with realistic devices. *Reviews of Modern Physics*, *92*(2), 025002.

7. Heisenberg, W. (1985). Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik. In *Original Scientific Papers Wissenschaftliche Originalarbeiten* (pp. 478-504). Springer, Berlin, Heidelberg.

8. Wootters, W. K., & Zurek, W. H. (2009). The no-cloning theorem. *Physics Today*, *62*(2), 76-77.

9. Brassard, G., & Salvail, L. (1993, May). Secret-key reconciliation by public discussion. In *Workshop on the Theory and Application of of Cryptographic Techniques* (pp. 410-423). Springer, Berlin, Heidelberg.

10. Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, *27*(3), 379-423.

11. Lütkenhaus, N. (1999). Estimates for practical quantum cryptography. *Physical Review A*, *59*(5), 3301.

12. Lizama-Perez, L. A., & López, J. M. (2020). Quantum key distillation using binary frames. *Symmetry*, *12*(6), 1053.

ISSN 2523-6814