BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **20**, No 1
Sofia • 2020 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2020-0002

EnQuad: A Publicly-Available Simulator for Quantum Key Distribution Protocols

Mohamed S. Abdelgawad[1], *Botrous A. Shenouda*[1], *Sameh O. Abdullatif*[2]

[1]*Electrical Engineering department, The British University in Egypt (BUE), Cairo, Egypt*
[2]*Electrical Engineering Department, and FabLab, Centre for Emerging Learning Technologies (CELT), The British University in Egypt (BUE), Cairo, Egypt.*
E-mails: mohamed127811@bue.edu.eg botrous127265@bue.edu.eg sameh.osama@bue.edu.eg

Abstract**:** *In this paper, we present EnQuad version 1.0: a high-speed and expandable simulator for Quantum Key Distribution (QKD) protocols. Surpassing available simulators, EnQuad does not only simulate a QKD stack, but also does security testing and guides the researcher on which reconciliation protocol should be used in his experimental setup. On the top of that, it recommends changes for the researcher to satisfy security or a given target key-rate if any of them is not already fulfilled. Although EnQuad V1.0 is concerned with depolarizing channels and Individual Intercept-and-Resend attacks, EnQuad is featured with 24 parameters and 9 modular functions so that it could be expanded to a wide range of QKD protocols. In addition, we validated EnQuad outcomes against a comparable simulator and against theory. Furthermore, a set of 11 experiments showed that EnQuad runs 6.12× to 12.2× faster than a comparable simulator. EnQuad was implemented in MATLAB and the code is available online.*

Keywords**:** *QKD simulators, Information Security, Quantum Cryptography, Scientific Computations, Software Technologies.*

1. Introduction

Classical cryptographic protocols that are being deployed at this moment for secure data communication are highly threatened to get broken by quantum computers, once built on a large scale. Quantum cryptography came as a promising substitute to provide unconditional security based on ever-sustainable laws of quantum physics such as Heisenberg's uncertainty principle and no-cloning theorem [1].

Quantum cryptographic protocols employ Quantum Key Distribution (QKD) schemes based on photon polarization or electron spin encoding [2]. A secret random key is shared between two parties, usually known as Alice and Bob, with immunity against Eavesdropper's (Eve attacks). However, both performances, impersonated in the secret-key rates, and security of QKD protocols heavily depend on numerous critical system parameters such as input number of photons, channel noise type/quantity, Eve attack method/level and source/detector imperfections [3]. As such, a researcher might go into arduous experimental work before knowing which combinations/lower-or-higher bounds of those parameters could achieve his target key-bitrate in real life [4, 5]. Consequently, QKD simulators stand as urgent tools to increase the pace of QKD deployment and to guide the researcher on the parameters values that work best in his conditions, and on the choice of sub-protocols in order to satisfy security and target performance.

A few works have developed QKD simulators [6-8]. However, the simulator in [6] was built as a part of quantum mechanics visualization project for the undergraduate levels. Its target was to visualize basic principles rather than performance analysis or security requirements reporting. In [8], the simulator does not currently include the quantum channel noise, which is a serious drawback since it heavily reflects on the estimated secret-key rates and plays a profound role in the protocol security. Furthermore, the number of input pulses is limited to only 1000, leading to finite-size overheads [9]. Work in [7] is not publicly-available. To add, it does neither tell the researcher how to reach his target secret-key rates nor give information about the question of security violation/satisfaction at the researcher's input conditions.

In this paper, we present EnQuad V1.0: a publicly-accessible simulator for QKD protocols, where the code is given in [10]. The contributions of this work are as follows: first, we considered depolarizing channel effect, Individual Eve Intercept-and-Resend attacks and number of input photons as controllable variables that developers can change according to their prospective setups. Second, EnQuad does not only output figure-of-merit parameters such as Quantum Bit Error Rate (QBER) and secret-key rates, it also guides the researcher through which system parameters he should use to satisfy security and to reach his target key-rate. This saves arduous experimental work. Third, we compressed full behaviour of the depolarizing channel and the polarization modulator in the receiver from intensive matrices operations to only 4 statements. That reflected on speeding up the simulation time 6.12× to 12.2× compared to work in [8]. This speedup is of high importance, especially when the number of input photons are scaled to hundreds of thousands. Finally, EnQuad, unlike all other simulators, is featured with modularity, in the sense that all parts were implemented separately/independently with clear interfaces (sets of well-defined variables). This greatly helps EnQuad to seamlessly embrace a wide range of prospective QKD setups/protocols.

The rest of this paper is organized as follows: first, we briefly describe "BB84": the most widely known QKD protocol, developed by H. Bennett and G. Brassard in 1984 (see [3]), second, we explain the settings of EnQuad V1.0, where it lies in the much larger picture of Quantum Cryptography and its expansion opportunities; fourth, we elaborate on our simulation model and how we compressed the description of complex parts into a few statements which heavily contributed to EnQuad speed; fifth, we validate EnQuad QBER outcome against previous work and against theoretical formulas in variant conditions, also, we demonstrate how EnQuad guides the researcher to achieve security and target-key rates; finally, we run multiple experiments to test the speed of EnQuad against another accessible simulator and show EnQuad results in a set of scenarios. We further guide the reader through how to use EnQuad in Appendix.

2. BB84

As any communication system, BB84 is mainly composed of a transmitter, a channel and a receiver. At the transmitter side, Alice sends a stream of pulses, one at a time. The pulse should be generated by a Single-Photon Source (SPS). Each pulse gets individually polarized by one of two mutually non-orthogonal bases: rectilinear (+) or diagonal (×). If (+) basis is selected, key bit 0 is encoded as horizontally-polarized pulse with state vector of $|0^{\circ}\rangle$ and key bit 1 is encoded as vertically-polarized $|90^{\circ}\rangle$. If (×) basis is selected, key bit 0 is encoded as circular-right polarized pulse with state vector of $|45^{\circ}\rangle$ and key bit 1 is encoded as circular-left $|-45^{\circ}\rangle$. This is practically done by a polarization modulator controlled by bases selector and key bits generator as depicted in Fig. 1. The pulses stream goes through the quantum channel that could be fiber or free-space. The channel might be an amplitude-damping channel, a phase-damping channel or a depolarizing channel.

At the receiver side, Bob decodes the received pulses by measuring their polarization states. Alice and Bob then reveal their bases choices and discard the bits that were measured with unmatched based. The measurement setup is practically achieved by a polarization modulator followed by a Polarizing Beam Splitter (PBS) and two single photon detectors (SPD0 and SPD1) [11], as shown in Fig. 1. The remaining key after discarding the bits corresponding to unmatched bases is called the sifted key. Alice and Bob then undergo classical post-processing schemes: reconciliation followed by privacy amplification. Reconciliation is to correct the erroneous key bits as a result of channel noise or Eve attacks. The error rate in the sifted key is called Quantum Bit Error Rate (QBER). Privacy amplification is then applied where further key bits are discarded to maximize security against the information that Eve has gained from the bits that have been revealed during data reconciliation. The remaining key at the disposal of Alice/Bob is called *the secret key*. The ratio between length of the sifted key and that of the secret key is defined as *secret-key bit rate k*, which is of a great interest to the QKD development community.

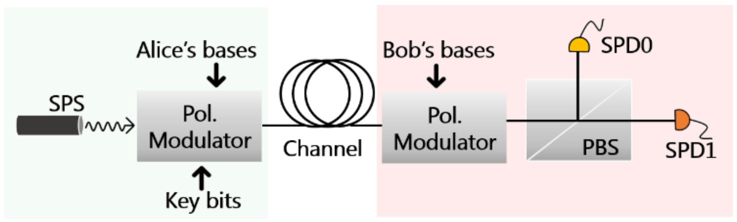

Fig. 1. Schematic of an experimental BB84 setup: Polarizing Beam Splitter (PBS), Single Photon Source/ Single Photon Detector (SPS/SPD) and polarization modulator

3. EnQuad settings

3.1. Why BB84?

In QKD literature, there is a myriad of QKD protocols as variations around BB84, including but not limited to, SARG04 [12] and Decoy-state Protocol [13]. Since BB84 and its variants are the most proven to be secure and the most targeted forms of quantum cryptography to be experimentally implemented, it is useful to offer a simulator with its core coded as per BB84. Exploiting EnQuad modularity, explained in *How to use and expand EnQuad*, we strongly believe that it could be extended in future releases to include SARG04 and Decoy-state protocols that are arguably more practical and higher secret-key rate achievers against certain attacks.

3.2. Source and detector

In EnQuad, a single-photon source (SPS) was employed since it is a critical key for the security of BB84. A single-photon source prevents Eve from applying Photon Number Splitting (PNS) attack (see [12] for details), which compromises the unconditional security of BB84. We also employ a single-photon detector with no dark counts. Nonetheless, when expanding EnQuad to SARG04 and Decoy-state protocols in future releases, it would be safe and sensible to include a multi-photon source (faint laser) and the dark-count rate; since such protocols were primarily built to maintain unconditional security against such imperfections.

3.3. Channel noise and eavesdropping

We implemented a depolarizing channel: a channel that introduces a bit-flip error or a phase-flip error or both, at equal probabilities. Depolarization is an important type of quantum noise that arises in free-space due to weather change and in optical fiber due to the phase changes along the channel [14]. The overall noise exclusively depends on the depolarizing parameter p. As for Eve, she has different strategies of attack to listen in on Alice's sent key, including Intercept-and-Resend, PNS, cloning [15], and faked-states attacks [16]. In Version 1.0 of EnQuad, we implement a combination of individual Intercept-and-Resend attacks where Eve measures the on-going pulses, one after another, with the same types of bases available at Alice/Bob side (($\times$) or ($+$)). We also define a parameter called *attack level* ε representing the ratio of pulses to be attacked. In practice, channel noise and Eavesdropping are eminent challenges to secret-key bit rate k and QKD security. EnQuad tells the researcher whether his inputs p and ε satisfy the lower bound of security S_{lb} based on Shannon mutual information between Alice and Bob and Alice and Eve in our settings. If his input conditions violate S_{lb}, EnQuad tells the researcher how much p need to change (since the researcher usually has no control on ε). In this way, we are assisting him to pick the suitable channel for his experimental setup.

3.4. Reconciliation and privacy amplification

Instead of reinventing the wheel and re-implementing available classical reconciliation and privacy amplification schemes that have been relentlessly researched in a plethora of studies [17-21], we took a reverse approach. We apply our settings to the Shannon mutual information between Alice, Bob, and Eve, from which it is possible to tell the researcher the theoretical secret-key bit rate k_{th} at its input conditions. Based on which, EnQuad tells the researcher the lower bound of the efficiency f_{lb} of the post-processing schemes required to reach his target secret-key bit rate k_{tr}. Afterwards, the researcher could make use of the results in [17] to select the scheme suitable for his physical experiment based on our reported f_{lb}. If there is yet no scheme to satisfy the f_{lb} at the researcher's input conditions, then we are basically setting the standards for the yet-to-be-developed post-processing schemes required to realize QKD protocols in real-life at the conditions of interest.

4. EnQuad simulation model

At the transmitter side, the bases selector and the key bits generator are simulated as two uniformly-distributed pseudorandom generators. The built-in MATLAB function rand, with the Mersenne twister set as the default generator, was used to generate uniformly-distributed pseudorandom numbers in the interval $(0, 1)$. Outcomes are then rounded using round, another built-in MATLAB function, that rounds to the nearest integer. This how uniformly-distributed random binary bits for the two generators are produced. The first randomly selects bases by setting 0 for $(+)$ basis and 1 for $(\times)$ basis, and the second generates the bits of the key to be sent. Basis selection and key bits are passed to four if-else cases with output of 1, 2, 3 and 4 representing $|0^\circ\rangle$, $|90^\circ\rangle$, $|45^\circ\rangle$ and $|{-45^\circ}\rangle$, respectively. Since the inputs to the four if-else cases are uniformly distributed, the output states are also uniformly distributed.

As for the channel, the depolarizing channel model is given by [22] (see the next equations (1)-(6)): in (1), where ρ denotes the density matrix of the original polarization state, i.e., before the channel, and ρ' is the output density matrix after being depolarized by the channel; σ_1 is the matrix responsible for bit-flip-error; σ_2 is the one responsible for phase-flip error and σ_3 is doing both errors, all with equal probabilities $\frac{p}{3}$ and are defined in (2). To follow, the density matrix ρ could be computed as in (3), where θ is the angle from the reference state, which in our case is the $|0^\circ\rangle$. Thus, ρ of a given polarization state before the channel can be computed as in (4). When substituting $\theta = 0^\circ$, 90°, 45° and -45° in (4), ρ of the four polarization states are found as in (5). When substituted in (1), we get $\rho'(p)$ for each state as in (6):

$$\rho'(p) = (1-p)\rho + \frac{p}{3}\sigma_1\rho\sigma_1 + \frac{p}{3}\sigma_2\rho\sigma_2 + \frac{p}{3}\sigma_3\rho\sigma_3, \tag{1}$$

$$\sigma_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad \sigma_2 = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \qquad \sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{2}$$

(3) $$\rho = |\psi\rangle\langle\psi|;\, |\psi\rangle = \begin{bmatrix}\cos\theta \\ \sin\theta\end{bmatrix}, \langle\psi| = [\,\cos\theta \quad \sin\theta\,],$$

(4) $$\rho = |\psi\rangle\langle\psi| = \begin{bmatrix}\cos^2\theta & \cos\theta\sin\theta \\ \sin\theta\cos\theta & \sin^2\theta\end{bmatrix},$$

(5) $$\rho_0 = \begin{bmatrix}1 & 0 \\ 0 & 0\end{bmatrix},\ \rho_{90} = \begin{bmatrix}0 & 0 \\ 0 & 1\end{bmatrix},\ \rho_{45} = \frac{1}{2}\begin{bmatrix}1 & 1 \\ 1 & 1\end{bmatrix},\ \rho_{-45} = \frac{1}{2}\begin{bmatrix}1 & -1 \\ -1 & 1\end{bmatrix},$$

(6) $$\rho'_0(p) = \begin{pmatrix}1-\frac{2p}{3} & 0 \\ 0 & \frac{2p}{3}\end{pmatrix},\qquad \rho'_{90}(p) = \begin{pmatrix}\frac{2p}{3} & 0 \\ 0 & 1-\frac{2p}{3}\end{pmatrix},$$
$$\rho'_{45}(p) = \frac{1}{2}\begin{pmatrix}1 & 1-\frac{2p}{3} \\ 1-\frac{2p}{3} & 1\end{pmatrix},\qquad \rho'_{-45}(p) = \frac{1}{2}\begin{pmatrix}1 & -1+\frac{2p}{3} \\ -1+\frac{2p}{3} & 1\end{pmatrix}.$$

As for the receiver, a received photon has four measurement outcomes with probabilities depending on p and on the basis selected. To see how probabilities are calculated, a photon with ρ'_0 is considered. In case the $(+)$ basis is selected for measurement, ρ_0 and ρ_{90} are possible outcomes. The probability of the correct outcome $P(\rho_0)$ is calculated in (7) where $Tr(\rho)$ is the trace of matrix ρ. Similarly, the probability of the incorrect outcome is calculated in (8). It is useful to observe that in case of an ideal channel, $P(\rho_0) = 1$, and $P(\rho_{90}) = 0$. In case the $(\times)$ basis is selected $P(\rho_{45})$ and $P(\rho_{-45})$ are calculated in (9).

(7) $$P(\rho_0) = \mathrm{Tr}\{\rho'_0\rho_0\} = 1 - \tfrac{2p}{3},$$

(8) $$P(\rho_{90}) = \mathrm{Tr}\{\rho'_0\rho_{90}\} = \tfrac{2p}{3},$$

(9) $$P(\rho_{45}) = \mathrm{Tr}\{\rho'_0\rho_{45}\} = \tfrac{1}{2}\ , P(\rho_{-45}) = \mathrm{Tr}\{\rho'_0\,\rho_{-45}\} = \tfrac{1}{2}.$$

After calculating the measurement outcomes probabilities for the other three states (ρ'_{90}, ρ'_{45} and ρ'_{-45}), we found they are completely analogous with same probabilities, yet with different sequence as shown in Table 1. That is how we compressed the full behavior of the depolarizing channel and the polarizing modulator in the receiver from computationally-intense matrices operations to only four statements, which positively contributes to the simulator speed. In our simulator, each state is represented by the corresponding four outcomes probabilities in a 1D vector. According to the basis selected for measurement, only two outcomes' probabilities are selected, then passed to a random binary generator with non-uniform distribution that is controlled by those two probabilities to generate bit 0 or 1 representing the decoded bit.

Table 1. Probability of measurement outcomes for the four possible states of the received photon

Received photon state	Measurement with $(+)$ basis		Measurement with $(\times)$ basis	
	Probabilities			
	$P(\rho_0)$	$P(\rho_{90})$	$P(\rho_{45})$	$P(\rho_{-45})$
$\rho_0^{'}$	$1-\frac{2p}{3}$	$\frac{2p}{3}$	0.5	0.5
$\rho_{90}^{'}$	$\frac{2p}{3}$	$1-\frac{2p}{3}$	0.5	0.5
$\rho_{45}^{'}$	0.5	0.5	$1-\frac{2p}{3}$	$\frac{2p}{3}$
$\rho_{-45}^{'}$	0.5	0.5	$\frac{2p}{3}$	$1-\frac{2p}{3}$

5. QKD QBER, security and reconciliation in EnQuad

5.1. QBER

The transition (error) probability in the channel q_{ch} could be seen from Table 1 as $2p/3$, where p is the channel depolarizing parameter. QBER is the error ratio in the sifted key due to channel and Eve. Transition probability at Bob side due to the Eve Intercept-Resend q_{e} is known as $\varepsilon/4$ [1], where ε is the attack level defined as ratio of the number of pulses to be attacked over the number of input pulses. $\varepsilon/4$ comes from the fact that Eve has no idea about the polarization states sent by Alice, thus, Eve inevitably intercept and resend a pulse in an incorrect (unmatched with Alice) measurement basis 50% of the time. When Bob receives a pulse incompatible with his measurement base (turns to be matched with Alice after sifting), then the bit is also decoded incorrectly 50% of the time, introducing a total of $50\% \times 50\% = 25\%$ error in Bob's decoded bits. Though, Eve may not intercept-resend all the input pulses, she may select a number of pulses to attack, which is represented by $\boldsymbol{\varepsilon}$, if, however, she chooses to attack all pulses, $\boldsymbol{\varepsilon}$ turns to be 1 and $1/4$ of the sifted key will be erroneous. QBER could be calculated given q_{ch} and q_{e}. Since the channel is binary symmetric, the overall transition probability of a bit starting from Alice source to Bob detector going through Eve followed by the channel (also regarded as the error probability in the sifted key) is calculated by the next equation:

$$\text{QBER} = q_{\mathrm{e}}(1-q_{\mathrm{ch}}) + (1-q_{\mathrm{e}})q_{\mathrm{ch}} = \frac{\varepsilon}{4}\left[1-\frac{2p}{3}\right] + \left[1-\frac{\varepsilon}{4}\right]\frac{2p}{3} = \quad (10)$$

$$= \frac{\varepsilon}{4} + \frac{2p}{3}(2-\boldsymbol{\varepsilon}).$$

Without the presence of Eve, QBER turns to be equal q_{ch} . In practice, q_{ch} that is estimated beforehand is the threshold revealing the presence of Eve. In the reconciliation process, QBER is estimated over a randomly-permuted part of the sifted key, if the estimated QBER largely exceeds q_{ch}, the QKD protocol may be aborted and re-run. If QBER exceeds q_{ch} with a tolerable amount (still satisfies the lower bound of security to be discussed below), Alice and Bob may proceed to post-processing schemes to correct the errors in the sifted key and minimize Eve information. As such, QBER is a critical parameter in QKD protocol security analysis. EnQuad computes QBER at the researcher's input conditions as the ratio of errors in the sifted key. In QBER verification, EnQuad QBER is compared with QBER resulted from simulator in [8] at the same conditions, and the theoretical QBER in (10) at such conditions.

5.2. Lower bound of security

A QKD protocol is secured as long as the mutual information between Alice and Bob $I(\mathcal{A};\mathcal{B})$ is greater than the mutual information between Alice and Eve $I(\mathcal{A};\mathcal{E})$ as stated in (11). Since we are studying a memoryless symmetric source with alphabet $\mathcal{A} = \{a_0,\ a_1\} = \{0,\ 1\}$ then $p(a_0) = p(a_1) = 1/2$ and the entropy of the source $H(\mathcal{A})$ is at its maximum: $H(\mathcal{A}) = 1$. In addition, since we are dealing with a binary symmetric channel where conditional probabilities $p(a_0|a_1) = p(a_1|a_0)$ and $p(a_0|a_0) = p(a_1|a_1)$ then conditional entropies $H(\mathcal{A}|\mathcal{B})$ and $H(\mathcal{A}|\mathcal{E})$ are equal to $1 - h(\alpha)$ where $h(\alpha)$ is the Shannon binary entropy with transition probability α: $h(\alpha) = -\alpha\log_2\alpha - (1-\alpha)\log_2(1-\alpha)$. In the case of communication between Alice and Bob, α is actually the QBER calculated in (1). In the case of Eve attacking Alice, α is defined as $\left(\frac{1}{2} - q_{\mathrm{e}}\right)$. Accordingly, the lower bound of security in (11) can be reformulated as per our settings as in (12). Note that it is valid to say $\mathrm{QBER} < \left(\frac{1}{2} - q_{\mathrm{e}}\right)$ if $h(\mathrm{QBER}) < h\left(\frac{1}{2} - q_{\mathrm{e}}\right)$ as long as $\mathrm{QBER} \leq 0.5$ and $(\frac{1}{2} - q_{\mathrm{e}}) \leq 0.5$, which is true in our case. The maximum of $\left(\frac{1}{2} - q_{\mathrm{e}}\right)$ is 0.5 that's when there is no Eve's attack; QBER also cannot exceed 0.5 since the depolarization parameter p is limited to 0.25 for usable channels [22].

$$(11)\qquad S_{\mathrm{lb}}:\ \left(I(\mathcal{A};\mathcal{B}) = H(\mathcal{A}) - H(\mathcal{A}|\mathcal{B})\right) > \left(I(\mathcal{A};\mathcal{E}) = H(\mathcal{A}) - H(\mathcal{A}|\mathcal{E})\right),$$

$$(12)\qquad \begin{aligned} &S_{\mathrm{lb}}:\ (1 - h(\mathrm{QBER})) > \left(1 - h\left[\tfrac{1}{2} - q_{\mathrm{e}}\right]\right):\ h(\mathrm{QBER}) < h\left[\tfrac{1}{2} - q_{\mathrm{e}}\right],\\ &:\mathrm{QBER} < \left[\tfrac{1}{2} - q_{\mathrm{e}}\right]\ :\ \left[\tfrac{\varepsilon}{4} + \tfrac{2p}{3}(2-\varepsilon)\right] < \left[\tfrac{1}{2} - \tfrac{\varepsilon}{4}\right]\ :\ p < \tfrac{4(1-\varepsilon)}{3(2-\varepsilon)}.\end{aligned}$$

Based on (12), EnQuad tells the researcher whether his input conditions violate the security condition (i.e., he should abort the protocol). Additionally, if the lower bound of security can be satisfied by decreasing his channel depolarizing parameter p (which practically means to replace his channel by a lower-p one), it tells him so with how much it should be decreased.

5.3. Lower bound of reconciliation efficiency

The theoretical secret-key rate k_{th} is the difference between $I(\mathcal{A};\mathcal{B})$ and $I(\mathcal{A};\mathcal{E})$ [17]. When we apply our settings, k_{th} can be defined as in (13). The quantity $h(\mathrm{QBER})$ is actually seen as the Shannon minimum amount of information that Bob needs from Alice to correct the errors in his decoded bits due to channel depolarization or Eavesdropping. In practice, however, post-processing reconciliation protocols reveal $\frac{1}{f}\big(h(\mathrm{QBER})\big)$ for error-correction, where f is the reconciliation efficiency factor and is less than 1. Expanding on its physical meaning, when $\frac{1}{f}\big(h(\mathrm{QBER})\big)$is multiplied by length of the sifted key, the result denotes the number of bits that are expected to be leaked in the post-processing schemes to obtain an error-free secret key. For an input target secret-key rate k_{tr} lower than k_{th}, lower bound of reconciliation efficiencyf_{lb} is calculated according to (14); if and only if the researcher's input conditions satisfy the lower bound of security formulated in (12) at the first place. The researcher is then able to take the reciprocal of our f_{lb}, and use the results in [17] to see which reconciliation protocol satisfies it and thus could be implemented along with his setup. If the researcher's k_{tr} is greater than k_{th}, EnQuad computes and reports the p that could achieve k_{th} almost greater than his k_{tr}:

(13) $$k_{\mathrm{th}} = h\left[\tfrac{1}{2} - q_{\mathrm{e}}\right] - h(\mathrm{QBER}),$$

(14) $$k_{\mathrm{tr}} = h\left[\tfrac{1}{2} - q_{\mathrm{e}}\right] - \frac{1}{f_{\mathrm{lb}}}\big(h(\mathrm{QBER})\big).$$

6. EnQuad simulation results

6.1. EnQuad verification

EnQuad QBER is calculated as the ratio of the unmatched bits in the sifted keys at the hand of Alice and Bob. EnQuad QBER is validated against the QBER of simulator in [8], where QBER was captured by Getdata Graph Digitizer software from their plot, and against the theoretical QBER at the same conditions: channel depolarization effect is disabled (i.e., p is set to zero) since it was not implemented there; number of sent photons is also set to be 1000, which is their maximum, to reduce QBER fluctuation as much as possible; Eve attack level ε is varied from 0 to 1 with a step of 0.1. Results are depicted in Fig. 2. In order to examine the utility of the proposed simulator with respect to the corresponding in literature, the QBER is calculated versus Eve attack level using various number of photons (Fig. 3). It can be observed that the deviation from the theoretical limit tends to minimum by increasing the number of simulated photons in the system (Fig. 4). Over and above, for the same number of photons, 1K photons, our simulation platform shows better (i.e., lower) deviation with respect to literature.

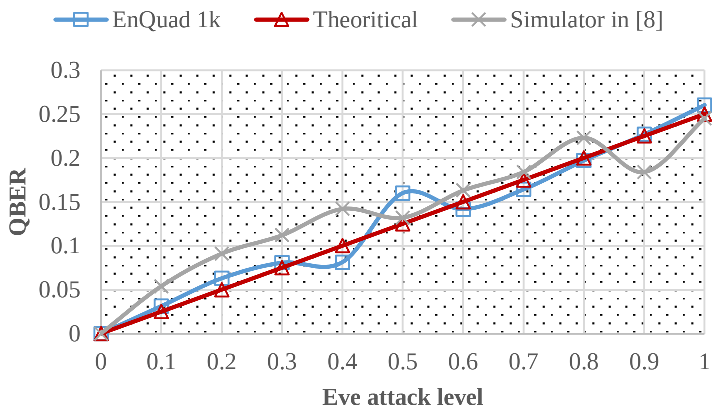

Fig. 2. EnQuad Validation: Plot of QBER with different Eve attack levels

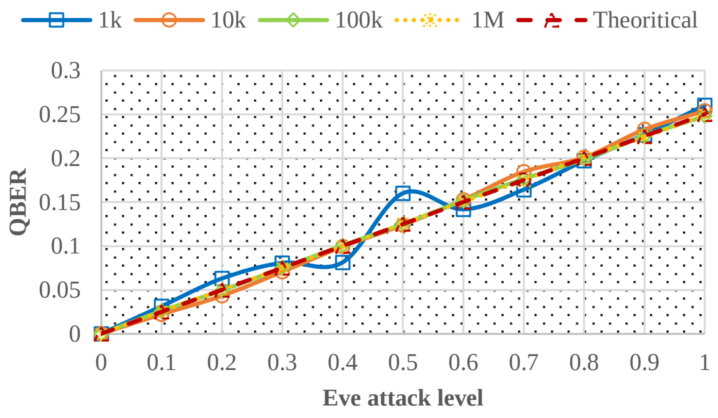

Fig. 3. EnQuad Validation: Plot of QBER with different number of photons

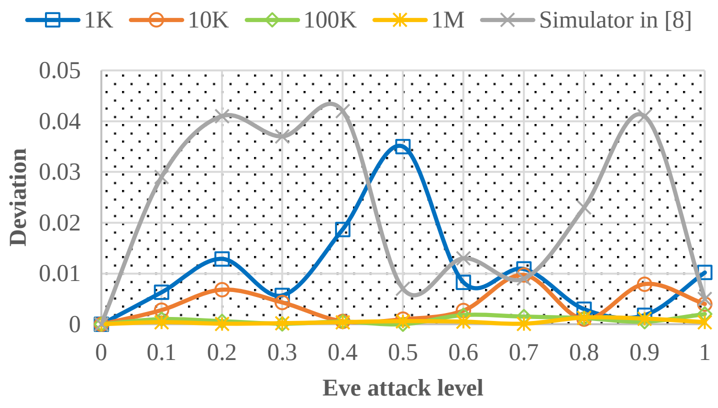

Fig. 4. QBER deviation from theoretical values for various number of photons

6.2. Speed evaluation

Compressing the full behaviour of the depolarizing channel and the polarization modulator in the receiver from computationally-intense matrices operations to only four statements, as demonstrated in Section 4, greatly reflected on EnQuad speed; we choose the name EnQuad as an acronym for **En**cryption in **4** statements. Despite the fact that EnQuad considers channel depolarization effect while work in [8] does not, experiments showed a faster simulation by 6.12× to 12.2×. We could not compare our speed with the simulator in [7] because it is not publicly available and no speed results are reported. Furthermore, we found work in [6] not comparable, since it is a visualization project of basic principles for students with no interest in speed.

Simulation time is defined as the time taken to compute the sifted key from the moment the researcher enters his input conditions and the run button is hit. We chose sifting as the termination phase since it is the last common phase between our work and work in [8]. More clearly, after sifting, EnQuad computes mutual information between Alice, Bob and Eve, then runs a security test, then reports the lower bound of reconciliation efficiency at the researcher's input conditions to reach a target key-rate and recommends changes in the input conditions when the target key-rate is not achievable. However, simulator in [8] directly runs a predetermined reconciliation protocol and output the key rate. Thus, the processes after sifting in the two simulators are uncorrelated.

Table 2. Comparison of simulation time with varying number of input photons and Eve's attack levels

Number of input photons	Eve's attack level	Simulator in [8] elapsed time (s)	EnQuad elapsed time (s)	Speed up
500	0	0.94	0.086	10.9×
600	0	1.20	0.098	**12.2**×
700	0	1.19	0.114	10.4×
800	0	1.27	0.154	8.24×
900	0	1.31	0.145	9.03×
1000	0	1.41	0.176	6.47×
1000	0.2	1.40	0.173	8.10×
1000	0.4	1.49	0.193	7.72×
1000	0.6	1.53	0.201	6.12×
1000	0.8	1.62	0.216	7.50×
1000	1	1.76	0.222	7.92×

As to the method used in recording our simulation time, we used *tic-toc*: a built-in MATLAB function that measures the amount of time for a script included within tic-toc keywords and reports time in seconds. As to recording simulation time for work in [8], the simulator type is set to *sifting* and JMeter™ was used: a desktop Java software designed to measure the performance of web applications. Since simulation time might depend on the CPU usage at the time of simulation in our case and since it might depend on web traffic in the work we are comparing against, we repeated each experiment, of same conditions, 5 times each separated by 30 seconds and took the average. In the first set of experiments, we disabled Eve's attack and incremented the number of input photons by 100. Since their allowable range of number of input photons is from 500 to 1000, we end up with six experiments. In the second set of experiments, we kept the number of input photons constant at 500 and incremented Eve's attack level from 0.2 to 1 by a step of 0.2, thus, we end up with another five experiments. In all experiments, we disabled the channel depolarization effect as it is not currently considered in their simulation. Average simulation time for each experiment at the aforementioned conditions are reported in Table 2.

6.3. Security and reconciliation

Following Subsections, 5.2. Lower bound of security, and 5.3. Lower bound of reconciliation efficiency, we report EnQuad results of a set of scenarios as listed in Table 3. Number of input photons is set to 10,000 in all combinations of input conditions (p and ε). S denotes the security testing. EnQuad reports Yes when S_{lb} is satisfied (i.e., $I(\mathcal{A};\mathcal{B})$ exceeds $I(\mathcal{A};\mathcal{E})$), and No otherwise. R is the recommendation that EnQuad gives to the researcher when S_{lb} is not satisfied or his/her target key-rate k_{tr} is not achievable (i.e., k_{tr} is larger than k_{th}) at his/her input conditions. L denotes the maximum allowable number of bits to be disclosed in the post-processing schemes to obtain k_{tr}, and is calculated as $\frac{1}{f_{\mathrm{lb}}}\left(h(\mathrm{QBER})\right)$ multiplied by the length of the sifted key. f_{lb} or L helps the researcher with the selection of a reconciliation protocol that could achieve his/her k_{tr} at his input conditions.

Table 3. EnQuad security and reconciliation results in different depolarizing parameters and Eve's attack levels

p	ε	S	k_{th}	k_{tr}	f_{lb}	L (bits)	R
0.1	0.5	Yes	0.185	0.173	0.98	3901	—
0.2	0.9	No	—	—	—	—	p should be 0.068 at max to satisfy security
0.15	0.6	Yes	0.065	0.1	—	—	p should be 0.123 at max to achieve k_{tr}
0.01	0.1	Yes	0.767	0.75	0.93	1246	—

7. Conclusion

In this work, we presented a publicly-available simulator to accelerate the demanding process of designing quantum key distribution setups with target secret-key rates. EnQuad outperforms available simulators in both speed, with an improvement of more than an order of magnitude, and built-in features. It offers security tests, recommendation on reconciliation protocols choice and guidance on lower/higher bounds of significant parameters such as the depolarizing channel parameter, number of input photons and Eve's attack level, for a target secret-key rate. EnQuad also reports insightful outputs such as QBER, theoretical secret-key rates, mutual information between Alice and Bob or Alice and Eve and the expected number of bits to be leaked in the post-processing schemes. What's more, EnQuad was validated against a comparable simulator and against theory where clear accordance was shown and closer values to theory were achieved. Since EnQuad source code was built with modularity and well-defined interfaces in mind, we believe that it has the potential to be continually expanded with the aim of realizing a QKD simulator that could act as a universal counsel for Quantum Cryptography development community.

Appendix. How to use and expand enquad

EnQuad was built in nine MATLAB functions, including the main, with clear interfaces. In Table 4, the parameters of all interfaces are numbered and their allowable values are listed. To follow, Table 4 lists which parameter belongs to which function, and whether it is an input to or an output from this function.

Table 4. EnQuad parameters

No	Parameter name	Allowable values
1	Number of photons	Real positive integer
2	Key bits	1D array of zeros/ones
3	Alice basis selection	1D array of zeros/ones
4	Alice polarized photons states	1D array of 1/2/3/4
5	Eve's attack level	Real positive number from 0 to 1
6	Photons states after Eve's attack	1D array of 1/2/3/4
7	Channel depolarization probability	Real positive number between 0 and 0.25
8	Photons states after channel	1D array of 1/2/3/4
9	Bob basis selection	1D array of zeros/ones
10	Photon states after beam splitter	1D array of 1/2/3/4
11	Alice photos states after polarizing beam splitter	1D array of 1/2/3/4
12	Measured bits	1D array of zeros/ones
13	Alice sifted key	1D array of zeros/ones
14	Bob sifted key	1D array of zeros/ones
15	QBER	Real positive number from 0 to 1
16	Simulation time	Real positive number
17	Theoretical key rate	Real positive number from 0 to 1
18	Target key rate	Real positive number from 0 to 1
19	Mutual information between Alice and Bob	Real positive number from 0 to 1
20	Mutual information between Alice and Eve	Real positive number from 0 to 1
21	Security Decision	Text
22	Lower bound of reconciliation efficiency for a target key rate	Real positive number from 0 to 1
23	Recommendation on depolarizing channel parameter change to achieve security (if not already satisfied)	Text
24	Recommendation on depolarizing channel parameter change to achieve target key rate (if not already achievable)	Text

Parameters number 1, 2, 3, 5, 7 and 9 are the inputs to EnQuad. Parameters number 15, 16 and 17 are the outputs of EnQuad. All parameters are accessible and tuneable. Furthermore, as to expanding EnQuad to a wide range of QKD protocols or

BB84 variants, EnQuad was built with keeping modularity in our mind, in the sense that all functions listed in Table 5 were implemented separately and independently. In other words, functions speak with each other only through interfaces. This means that the code of any function could be safely edited/replaced as per the developers' prospective QKD setups/protocols (see Section 3. EnQuad settings), with EnQuad still perfectly working as long as the interfaces are adjusted accordingly. While EnQuad started with the de-facto standards of a QKD protocol, impersonated in BB84, we seek for including more scenarios for each function, to expand EnQuad usability to BB84 variants such as Decoy-state protocol, SARG04 protocol in addition to the quantum entanglement and Bell's inequality-based protocols [23, 24]. With EnQuad parameters and functions listed, a quick live demo is provided in [10].

Table 5. EnQuad functions

No	Function	Interface	
		Input	Output
1	EnQuad Main	1, 5, 7	16, 17,18, 19, 20, 21, 22, 23, 24
2	Polarizer	1, 2, 3	4
3	Eve's Intercept and Resend	1, 4, 5	6
4	Channel	1, 4/6, 7	8
5	Beam Splitter	8, 9	10
6	Polarizing Beam Splitter	1, 10	11
7	SPAD	11	12
8	Sifting	2, 3, 9	13, 14, 15
9	Security Testing and Reconciliation Efficiency Reporting	1, 5, 7, 13	17, 18, 19, 20, 21, 22, 23, 24

References

1. G i s i n, N., et al. Quantum Cryptography. – Reviews of Modern Physics,Vol. **74**, 2002, No 1, p. 145.
2. C h o u, Y.-H., et al. Quantum Entanglement and Non-Locality Based Secure Computation for Future Communication. – IET Information Security, Vol. **5**, 2011, No 1, pp. 69-79.
3. H a n s c h k e, L., et al. Quantum Dot Single-Photon Sources with Ultra-Low Multi-Photon Probability. – NPJ Quantum Information, Vol. **4**, 2018, No 1, p. 43.
4. S i b s o n, P., et al. Chip-Based Quantum Key Distribution. – Nature Communications, Vol. **8**, 2017, p. 13984.
5. L i a o, S.-K., et al. Satellite-Relayed Intercontinental Quantum Network. – Physical Review Letters, Vol. **120**, 2018, No 3, p. 030501.
6. K o h n l e, A., A. R i z z o l i. Interactive Simulations for Quantum Key Distribution. – European Journal of Physics, Vol. **38**, 2017, No 3, p. 035403.
7. N i e m i e c, M., Ł. R o m a ń s k i, M. Ś w i ę t y. Quantum Cryptography Protocol Simulator. – In: International Conference on Multimedia Communications, Services and Security, 2011, Springer.
8. A t a s h p e n d a r, A., P. R y a n. Simulation and Analysis of QKD (BB84). – Interdisciplinary Center for Security, University of Luxembourg, 2014-2019. **https://www.qkdsimulator.com/**
9. S c a r a n i, V., R. R e n n e r. Quantum Cryptography with Finite Resources: Unconditional Security Bound for Discrete-Variable Protocols with One-Way Postprocessing. – Physical Review Letters, Vol. **100**, 2008, No 20, p. 200501.

10. A b d e l g a w a d, M. Github EnQuad Repository. 2019. **https://github.com/Mo-Abdelgawad/EnQuad-a-QKD-Simulator**
11. D i f f i e, W., M. H e l l m a n. New Directions in Cryptography. – IEEE Transactions on Information Theory, Vol. **22**, 1976, No 6, pp. 644-654.
12. S c a r a n i, V., et al. Quantum Cryptography Protocols Robust against Photon Number Splitting Attacks for Weak Laser Pulse Implementations. – Physical Review Letters, Vol. **92**, 2004, No 5, 057901.
13. L o, H.-K., X. M a, K. C h e n. Decoy State Quantum Key Distribution. – Physical Review Letters, Vol. **94**, 2005, No 23, 230504.
14. J e o n g, Y.-C., Y.-S. K i m, Y.-H. K i m. Effects of Depolarizing Quantum Channels on BB84 and SARG04 Quantum Cryptography Protocols. – Laser Physics, Vol. **21**, 2011, No 8, pp. 1438-1442.
15. N i e d e r b e r g e r, A., V. S c a r a n i, N. G i s i n. Photon-Number-Splitting Versus Cloning Attacks in Practical Implementations of the Bennett-Brassard 1984 Protocol for Quantum Cryptography. – Physical Review A, Vol. **71**, 2005, No 4, 042316.
16. L i z a m a-P é r e z, L., J. L ó p e z, E. D e C a r l o s L ó p e z. Quantum Key Distribution in the Presence of the Intercept-Resend with Faked States Attack. – Entropy, Vol. **19**, 2017, No 1, p. 4.
17. E l k o u s s, D., et al. Efficient Reconciliation Protocol for Discrete-Variable Quantum Key Distribution. – In: 2009 IEEE International Symposium on Information Theory, 2009, IEEE.
18. B e n l e t a i e f, N., H. R e z i g, A. B o u a l l e g u e. Toward Efficient Quantum Key Distribution Reconciliation. – Journal of Quantum Information Science, Vol. **4**, 2014, No 2, p. 117.
19. K e r n, O., J. M. R e n e s. Improved One-Way Rates for BB84 and 6-State Protocols. – arXiv preprint arXiv:0712.1494, 2007.
20. M e h i c, M., M. N i e m i e c, M. V o z n a k. Calculation of the Key Length for Quantum Key Distribution. – Elektronika i Elektrotechnika, Vol. **21**, 2015, No 6, pp. 81-85.
21. G o t t e s m a n, D., H.-K. L o. Proof of Security of Quantum Key Distribution with Two-Way Classical Communications. – IEEE Transactions on Information Theory, Vol. **49**, 2003, No 2, pp. 457-475.
22. S m i t h, G., J. A. S m o l i n. Additive Extensions of a Quantum Channel. – In: Information Theory Workshop, 2008 (ITW'08), IEEE, 2008.
23. E k e r t, A. K. Quantum Cryptography Based on Bell's Theorem. – Physical Review Letters, Vol. **67**, 1991, No 6, p. 661.
24. H w a n g, T., K.-C. L e e. EPR Quantum Key Distribution Protocols with Potential 100% Qubit Efficiency. – IET Information Security, Vol. **1**, 2007, No 1, pp. 43-45.

Received: 15.10.2019; Second Version: 01.01.2020; Accepted: 06.01.2020 (fast track)