PAPER • OPEN ACCESS

Detecting quantum attacks: a machine learning based defense strategy for practical continuous-variable quantum key distribution

New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG
IOP Institute of Physics

Published in partnership with: Deutsche Physikalische Gesellschaft and the Institute of Physics

OPEN ACCESS

RECEIVED
31 March 2020

REVISED
20 July 2020

ACCEPTED FOR PUBLICATION
23 July 2020

PUBLISHED
25 August 2020

PAPER

Detecting quantum attacks: a machine learning based defense strategy for practical continuous-variable quantum key distribution

Yiyu Mao[1], Wenti Huang[2], Hai Zhong[2], Yijun Wang[1], Hao Qin[3], Ying Guo[1] and Duan Huang[2,4]

[1] School of Automation, Central South University, Changsha 410083, People's Republic of China
[2] School of Computer Science and Engineering, Central South University, Changsha 410083, People's Republic of China
[3] CAS Quantum Network Co., Ltd No. 99 Xiupu Rd, Pudong New District, Shanghai 201315, People's Republic of China
[4] Author to whom any correspondence should be addressed.

E-mail: duanhuang@csu.edu.cn

Keywords: attack detection, continuous-variable quantum key distribution, artificial neural network

Abstract

The practical security of a continuous-variable quantum key distribution (CVQKD) system is compromised by various attack strategies. The existing countermeasures against these attacks are to exploit different real-time monitoring modules to prevent different types of attacks, which significantly depend on the accuracy of the estimated excess noise and lack a universal defense method. In this paper, we propose a defense strategy for CVQKD systems to address these disadvantages and resist most of the known attack types. We investigate several features of the pulses that would be affected by different types of attacks, derive a feature vector based on these features as the input of an artificial neural network (ANN) model, and show the training and testing process of the ANN model for attack detection and classification. Simulation results show that the proposed scheme can effectively detect most of the known attacks at the cost of reducing a small part of secret keys and transmission distance. It establishes a universal attack detection model by simply monitoring several features of the pulses without knowing the exact type of attack in advance.

1. Introduction

Quantum key distribution (QKD) [1] is one of the most important application of quantum technologies, which enables two distant parties, Alice and Bob, to exchange secret keys in an untrusted environment without being eavesdropped by an eavesdropper, Eve its theoretical unconditional security is guaranteed by the fundamental laws of quantum mechanics [2, 3], which based on some assumptions that Alice and Bob's device are supposed to behave according to a perfect model. However, there are some deviations between the theoretical perfect assumptions and practical QKD implementations, such deviations may bring loopholes and enable Eve to break the security by stealing information from the legitimate parties [4–6].

According to different implementation methods, QKD can be divided into two types: discrete-variable (DV) QKD [7, 8] and continuous-variable (CV) QKD [9–11]. Compared with DVQKD, CVQKD has higher secret key rate and better compatibility with the current optical networks [12]. Gaussian modulated coherent state (GMCS) protocol is the most popular CVQKD scheme [13, 14], which has been proven theoretically secure against collective attacks [15–17]. However, the security of the practical GMCS CVQKD can be broken by some practical attack strategies, such as Trojan-horse attacks [18, 19], wavelength attacks [20, 21], calibration attacks [22], local oscillator (LO) intensity attacks [23], saturation attacks [24], and homodyne-detector-blinding attacks [25]. The main idea of these attacks is to exploit the imperfections of optical devices to bias the excess noise estimation, and the essence of the corresponding countermeasures is to add suitable real-time monitoring modules on the system, which significantly depend on the accuracy of

the estimated excess noise and the calculated precision of a low bound of optical features disturbance for Eve successfully concealing herself [26]. However, in practice there are some natural fluctuations in the legitimate light as well as real detectors and electronics, Alice and Bob have to implement multiple iterative calculations to obtain an accurate estimation. In addition, the estimation procedure is usually implemented after the key transmission process is completed, once an attack is found the whole key data should be discarded, wasting a lot of time and resources. Moreover, in actual systems we do not know in advance which kind of attack Eve will launch, so we need a universal defense solution which can resist as many attack types as possible.

In this paper, we propose a defense strategy for CVQKD systems to address the disadvantages mentioned above. We investigate several typical features of the pulses that would be effected by the attacks, and the deviations of these features between normal unattacked pulses and abnormal attacked pulses. A set of feature vectors labeled by different attack types is constructed to train an artificial neural network (ANN). The trained ANN model can automatically detect abnormal feature vectors and classify them into different attack types. Consequently, a universal attack detection model is established, which can recognize most of the known attack types by using only one forward propagation calculating process. The secret keys received by Bob can be sequentially input into the model, and the transmission process will be aborted immediately once abnormal data is found. In this way Bob does not need to wait until the key transmission process is complete to check if the system is attacked. In our work, we mainly consider three typical attack strategies against GMCS CVQKD systems with homodyne detection, including the calibration attack, the LO intensity attack, and the saturation attack. In addition, two types of hybrid attack strategies [25, 27] is also investigated. Individual wavelength attacks [20, 21] are not considered here because they are only effective for heterodyne detection CVQKD systems. For one-way GMCS CVQKD systems, isolators and wavelength filters are the most suitable countermeasures against Trojan-horse attacks, thus the Trojan-horse attack is also not contained in our work.

2. Learning for automatic attack classification

2.1. Feature extraction of optical pulses

In a GMCS CVQKD protocol, Alice prepares a train of coherent states $|X_A + iP_A\rangle$ where the quadrature values X_A and P_A subject to a bivariate Gaussian distribution with variance $V_A N_0$. Here N_0 represents the shot noise variance which corresponds to the variance of the homodyne detector output when the input signals are vacuum states. Then Alice sends the prepared states to Bob with a strong LO of intensity I_{LO} by using polarization multiplexing technique. The receiver Bob measures one of the quadratures of the signal states by performing a homodyne detection, with the help of the LO as a phase reference. After this process, Alice and Bob obtain two strings of correlated data $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$ and $\mathbf{y} = \{y_1, y_2, \ldots, y_N\}$, where x represents the quadrature value modulated by Alice (X_A or P_A) and y represents the quadrature value measured by Bob (X_B or P_B). We note that

$$\bar{x} = 0, \quad V_x = V_A N_0, \tag{1}$$

$$\bar{y} = 0, \quad V_y = \eta T V_A N_0 + N_0 + \eta T \xi + V_{el}, \tag{2}$$

where T and η are the quantum channel transmittance and the efficiency of the homodyne detector, respectively. $V_{el} = v_{el} N_0$ is the detector's electronic noise and $\xi = \varepsilon N_0$ is the technical excess noise of the system. In a practical CVQKD system, there are several features could be affected by different attack strategies, such as the intensity I_{LO} of the LO, the shot noise variance N_0, the mean value $\bar{y}$ and variance V_y of Bob's measurement. Table 1 shows the impacts of different attack strategies on the measurable features. We find that the first four types of attacks affect different features. Although the last attack strategy and the saturation attack act on the same features, they have different degree of impact (more details can be found in the appendix A). Therefore, learning the variation of these features can help to detect and classify different attacks.

Figure 1 shows the schematic diagram of Bob's detection setup that is used for simultaneously measuring the features mentioned above. Firstly, the signal and LO pulses are demultiplexed by using a PBS. Then, an AM is applied on the signal path to randomly set a maximum attenuation with a probability of 10% for real-time shot-noise estimation, and the remaining signal pulses are not attenuated. Meanwhile, the LO pulses are split by a 90 : 10 beam splitter, part of which are used for homodyne detection and part of which are used for power monitoring and clock generation. After that, the analog measurement results are fed in the DPC for sampling and attack detection. We assume that Bob receives N pulses in a communication process and all these pulses can be divided into M blocks. For each block, we can calculate the mean and variance, the LO average power, and the shot noise variance. By this way, a feature vector

Table 1. Impacts of different attack strategies on measurable features. The symbol '√' under the features indicates that the corresponding feature will be changed by the corresponding attack.

Features	$\bar{y}$	V_y	I_{LO}	N_0
LO intensity attack [23]	—	√	√	√
Calibration attack [22]	—	√	—	√
Saturation attack [24]	√	√	—	—
Hybrid attack 1 [27]	—	√	√	—
Hybrid attack 2 [25]	√	√	—	—

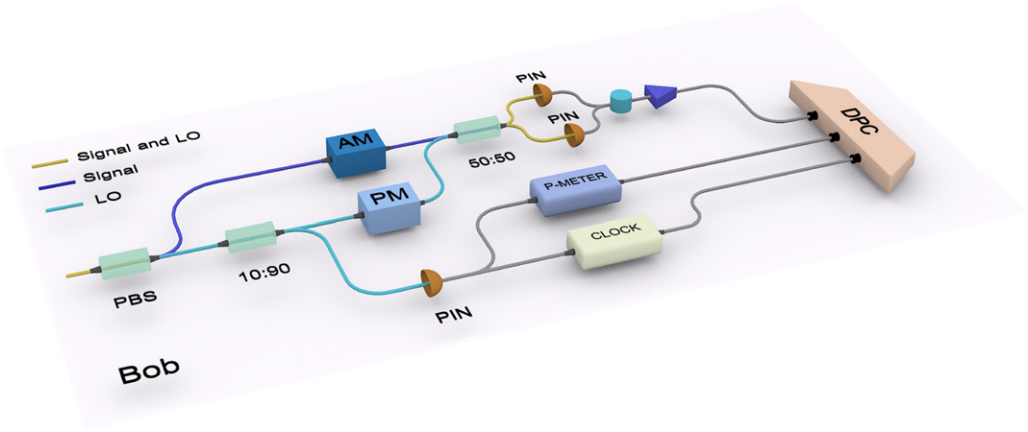

Figure 1. Schematic diagram of Bob's detection setup for protecting a CVQKD system against attacks. PBS: polarization beam splitter. AM: amplitude modulator. PM: phase modulator. PIN: PIN photodiode. P-METER: power meter. CLOCK: clock circuit used to generate clock signal for measurement. DPC: data processing center used to sample analog signal, attack detection and raw key distillation.

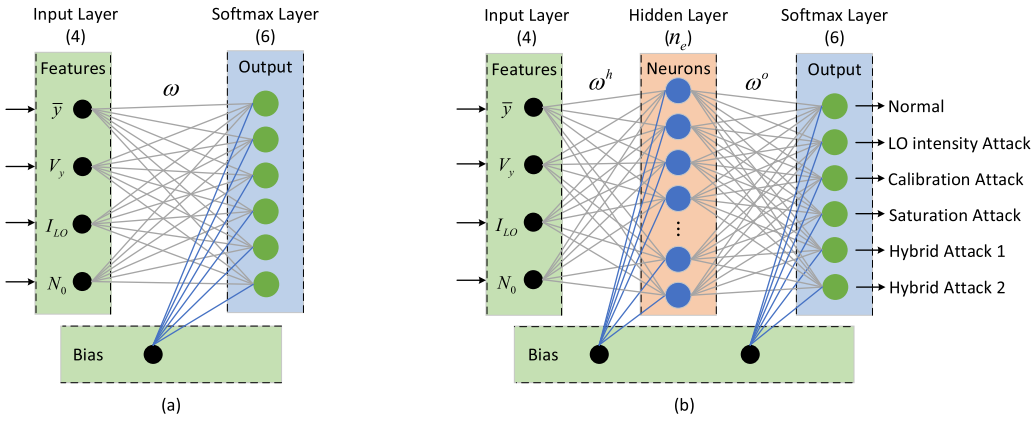

Figure 2. ANN-based quantum attack detection model. The circles represent the artificial neutrons in each layer and the lines represent the weights of the neutrons output to the next layer. (a) A linear ANN model without the hidden layer which can only solve linear separable problems. (b) A nonlinear ANN model with a hidden layer to classify different types of quantum attacks.

$\vec{u} = \{\bar{y}, V_y, I_{\mathrm{LO}}, N_0\}$ is constructed to represent the corresponding block. M feature vectors $\{\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_M\}$ of the M blocks form the input of the ANN model, as shown in figure 2. The values of the feature vector are various under different types of attacks, since different attacks act on different features and change the values of them in different ways. According to the approximation theorem of the neural networks, it is possible to infinitely approximate to any given bounded continuous function on a given domain with a neural network [28]. It suggests that the neural network can fully learning the behaviours of the attacks based on the established feature vectors. It is worth noting that although there may be errors between the feature values of each block and these values of the whole data, the neural network can still use them to distinguish attacks because the errors under different attacks is also different.

2.2. Artificial neural network establishment for attack classification

In this section, we introduce how to establish the ANN attack detection model based on feature vectors. ANN is a popular machine learning technique inspired by the biological neural network in the human brain [29]. As shown in figure 2, an ANN consists of several layers and each layer contains many neurons, ANN sends the weight values of each neuron as output to the next layer after processing with inputs from neurons in the previous layer. Our target is to derive an output vector $\vec{v}$ according to the input vector $\vec{u}$ by constructing a classifier, which is represented by a function $f : \vec{u} \rightarrow \vec{v}$. The construction of the classifier is based on multiple training iterations on a training set $S_{\text{train}} = \{(\vec{u_1}, \vec{v_1}), (\vec{u_2}, \vec{v_2}), (\vec{u_3}, \vec{v_3}), \ldots\}$. In our scheme, the input vector $\vec{u}$ consists of the features listed in table 1, the output vector $\vec{v}$ consists of a set of probability values, which represent the probability that the current input data belongs to each attack type. Figure 2(a) is a linear ANN model without hidden layers which can only solve linear separable problems. In order to applicable to distinguish different types of attacks, we join a hidden layer between the input layer and the output layer, and further construct a nonlinear ANN multi-classifier by using a softmax function. The number of neurons in the hidden layer can be adjusted for optimal performance. Figure 2(b) shows the nonlinear ANN multi-classifier that contains three layers: input layer, hidden layer and softmax layer (output layer). Each neuron in the current layer is a linear combination of neurons in the previous layer with weight ω and bias b. For example, the relationship between the input layer and the hidden layer is expressed as

$$v_j^{\text{h}} = \sigma_{\text{tanH}}\left(\sum_i u_i \omega_{ij}^{\text{h}} + b_j^{\text{h}}\right), \tag{3}$$

where v_j^{h} is the jth output of the hidden layer, u_i is the ith element of the input vector $\vec{u}$, b_j^{h} is the jth bias unit input into the hidden layer, ω_{ij}^{h} is the weight between the ith element of the input layer and the jth element of the hidden layer which will be iterative optimized in the training process. σ_{tanH} is the activation function which is defined as [30, 31]

$$\sigma_{\text{tanH}}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{4}$$

In a similar manner the relationship between the hidden layer and the output layer is obtained by

$$v_j^{\text{o}} = \sigma_{\text{S}}\left(\sum_i v_i^{\text{h}} \omega_{ij}^{\text{o}} + b_j^{\text{o}}\right), \tag{5}$$

where σ_{S} is the softmax function which is given by

$$\sigma_{\text{S}}(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{6} \exp(x_j)}. \tag{6}$$

ω_{ij}^{o} is the weight between the ith element of the hidden layer and the jth element of the output layer, b_j^{o} is the jth bias unit input into the output layer, v_j^{o} is the jth element of the output layer, and the sum of the output $\sum_{j=1}^{6} v_j^{\text{o}} = 1$. The final output $\vec{v}$ of the ANN model consists of six probability values, which represent the probability that the vector $\vec{u}$ belongs to each class. In the training process, the back-propagation algorithm is used to quickly solve the partial derivatives of the objective function on the internal weights in the network [32], and the weights is accordingly adjusted by using the stochastic gradient descent optimization algorithm [33]. Finally, an ANN model that matches the target output is learned by minimizing the objective function $-\log\ v_j^{\text{o}}$ when the target class is j.

2.3. Training and testing process

According to the data preparation process described in the appendix A, we generate six sets of data as training data $Y_{\text{train}} = \{\mathbf{y}_{\text{normal}}, \mathbf{y}_{\text{LOIA}}, \mathbf{y}_{\text{calib}}, \mathbf{y}_{\text{sat}}, \mathbf{y}_{\text{hyb1}}, \mathbf{y}_{\text{hyb2}}\}$ and preprocess them by division and feature vector extraction, as shown in figure 3. Subsequently, the collected feature vectors labeled by the category of data set are fed into the ANN trainer to learn the characteristics of different attack strategies. In a similar way, we also generate another six sets of data as testing data $Y_{\text{test}} = \{\mathbf{y}'_{\text{normal}}, \mathbf{y}'_{\text{LOIA}}, \mathbf{y}'_{\text{calib}}, \mathbf{y}'_{\text{sat}}, \mathbf{y}'_{\text{hyb1}}, \mathbf{y}'_{\text{hyb2}}\}$ and preprocess them. The resulting feature vectors are directly input into the trained ANN classifier to check the performance of attack classification. In our experiments, precision, recall, false positive rate (FPR) and false negative rate (FNR) are selected as the evaluation metrics to evaluate the performance of our scheme, which can be expressed as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{7}$$

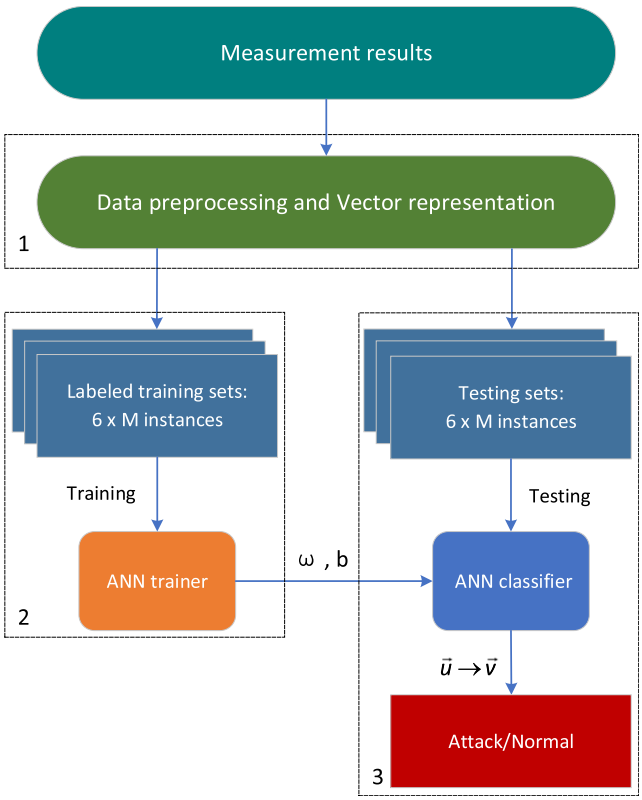

Figure 3. Training and testing procedure of the ANN model.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{8}$$

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}}, \tag{9}$$

$$\text{FNR} = \frac{\text{FN}}{\text{TP} + \text{FN}}, \tag{10}$$

where TP (true positive) denotes the number of the feature vectors that belong to an certain attack type are identified as such attack, FP (false positive) denotes the number of the feature vectors that do not belong to an certain attack type are identified as such attack. FN (false negative) denotes the number of the feature vectors that belong to an certain attack type but are not identified as such attack. TN (true negative) denotes the number of the feature vectors that do not belong to an certain attack type and are not identified as such attack. In general, a fine ANN classifier can achieve high values of precision and recall, and low values of FPR and FNR. In the testing stage, 'one vs others' method is employed to evaluate the performance of the classifier. For example, when calculating the precision of detecting LO intensity attack, the LO intensity attack-related feature vectors are considered as positive instances, while the other five types of vectors are considered as negative instances, which simplifies the multi-class problem to a binary-class problem.

3. Performance analysis

3.1. Implementation details

We implement ANN training and testing on Matlab R2019b, with the help of neural network toolbox. The memory and processor of our computer are 16 GB and Intel Core 4.0 GHz CPU, respectively, and the operating system is Windows 10 Professional. In the experiments the learning rate and error goal of ANN are set as 0.01, and the maximum iterations is 500. The data set size of each attack type is $N = 1 \times 10^7$ and the number of pulses in each block is $Q = 1 \times 10^4$, therefore, the data set of each attack type can be divided into $M = 1000$ feature vectors, 6 types of data constitute 6000 feature vectors. It is worth noting that too small M value will make the ANN model unable to learn the characteristics of each attack type well, and too large M value will bring a large statistical error to the feature values of each block. In practical

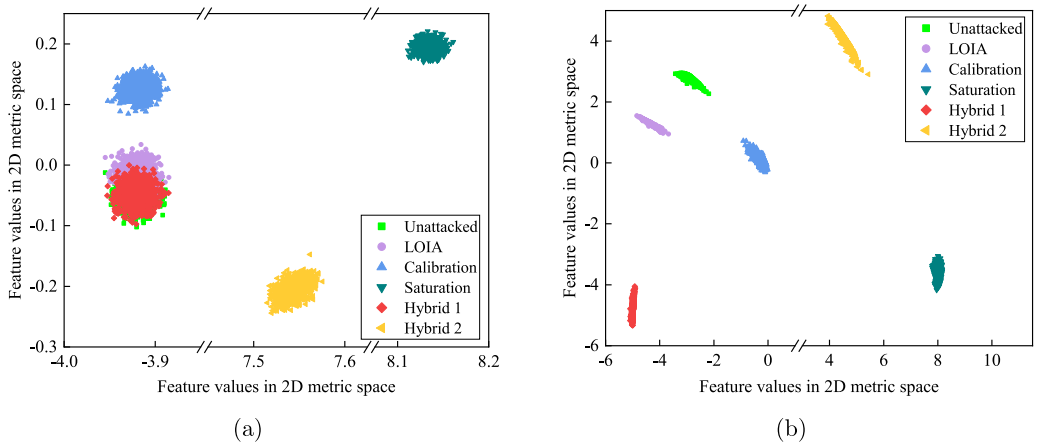

Figure 4. (a) The distribution of 6000 feature vector instances consist of 6 types of attack data before ANN classification. (b) The distribution of 6000 feature vector instances consist of 6 types of attack data after ANN classification.

implementation, the value of M can be optimized by using the grid search algorithm, which is the most widely used strategies for hyper-parameter optimization [34]

3.2. Performance of attack classification for CVQKD system

In this section we analyze the performance of the ANN model for attack detection and classification. Firstly, we introduce principal component analysis [35] to map the collected 6000 feature vectors of six types of data into a 2D metric space, as shown in figure 4(a). We can find that the feature vectors of the calibration attack, the saturation attack and the hybrid attack 2 are very different from the normal unattacked vectors, whereas the feature vectors of the LO intensity attack and the hybrid attack 1 are close to the normal vectors and hard to be separated by statistical analysis. Figure 4(b) shows the mapped instances after ANN classification, we can see that different types of data are significantly separated by the ANN model. In order to determine the optimal number of neurons n_e in the hidden layer, we calculate the values of precision, recall, FPR and FNR of the ANN model for attack classification, all of the results are the average of 20 iterations for fear of overfitting and underfitting. As illustrated in figure 5, the precision and recall of the calibration attack, the saturation attack, the hybrid attack 1 and the hybrid attack 2 reach the maximum 1 when the value of $n_e = 15$. For the LO intensity attack under the same condition, the performance of the ANN is the worst with precision and recall of 0.9969 and 0.9961, respectively. This is because the feature vectors of the LO intensity attack is closest to the normal data compared to other attacks. Similarly, the FPR and FNR of the calibration attack, the saturation attack, the hybrid attack 1 and the hybrid attack 2 achieve the minimum value of 0 at $n_e = 15$, but these two values of the LO intensity attack are 6.2×10^{-4} and 3.9×10^{-3}, respectively. The performance of ANN classification is relatively stable when the value of n_e between 5 and 20, while the precision and recall are low when $n_e = 1$ because the ANN model does not have enough learning ability when the number of neurons in hidden layer is small. In addition, the results of precision, recall, FPR and FNR fluctuate apparently in the condition of $n_e > 20$, because too many neurons in hidden layer greatly increase the complexity of the ANN, thereby neurons in the hidden layer will lose their sensitivity to input signals, and the propagation of information is blocked severely, under this situation the network is easily trapped into a local minimum point and fail to converged to a global minimum within a reasonable number of iterations [36].

3.3. Secret key rate of ANN-based attack defense strategy

In this section, we compare the secret key rates for a CVQKD system that employs the ANN-based attack detection model and for a system that does not employ any countermeasures against attacks. The most commonly used method is the asymptotic secret key rate which is given by [13]

$$K_{\text{asym}} = \beta I_{\text{AB}} - \chi_{\text{BE}}, \tag{11}$$

where β is the reverse reconciliation efficiency, I_{AB} is the Shannon mutual information between Alice and Bob, and χ_{BE} is the Holevo quantity for Eve's maximum accessible information. The detailed calculation about I_{AB} and χ_{BE} can be found in appendix B. In addition to asymptotic security, the finite-size effect [37] is also taken into consideration, since the signals exchanged by Alice and Bob are impossible unlimited in practice. In the finite-size scenario, the characteristics of the quantum channel cannot be known in advance.

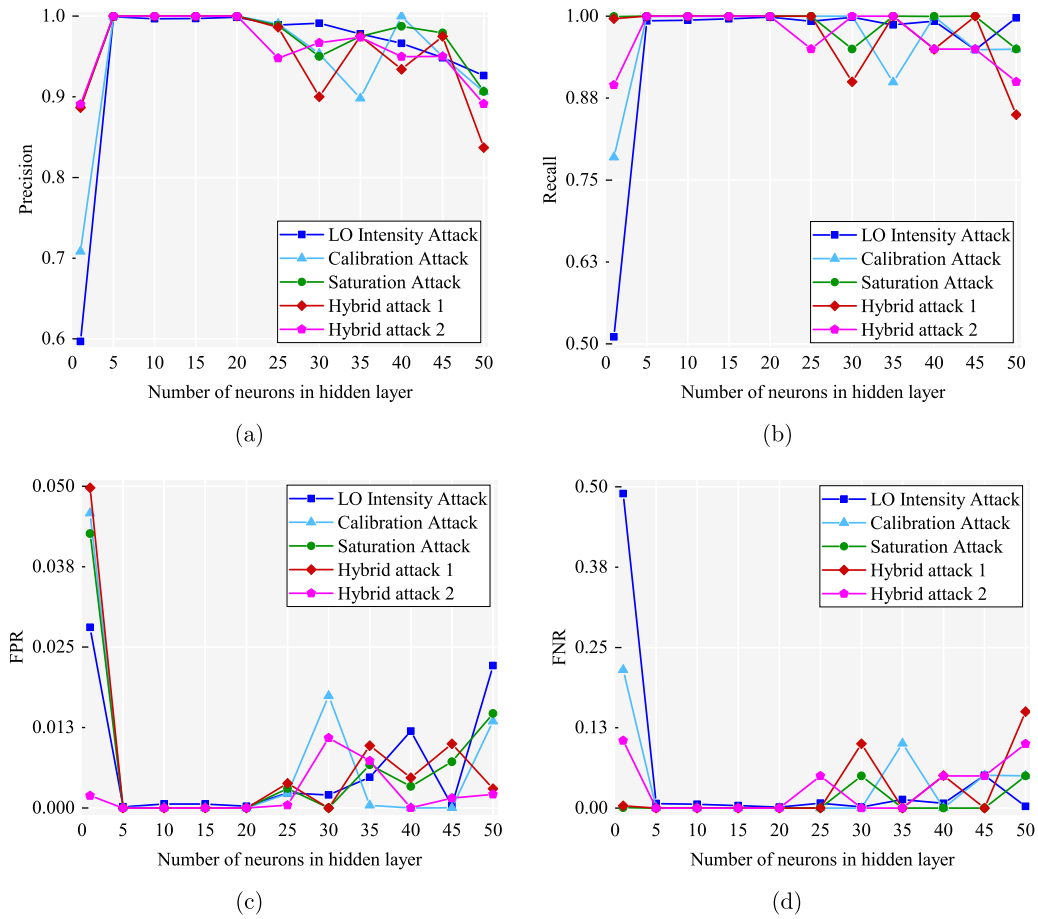

Figure 5. (a) Precision of the ANN model for attack classification versus different values of n_e. (b) Recall of the ANN model for attack classification versus different values of n_e. (c) FPR of the ANN model for attack classification versus different values of n_e. (d) FNR of the ANN model for attack classification versus different values of n_e.

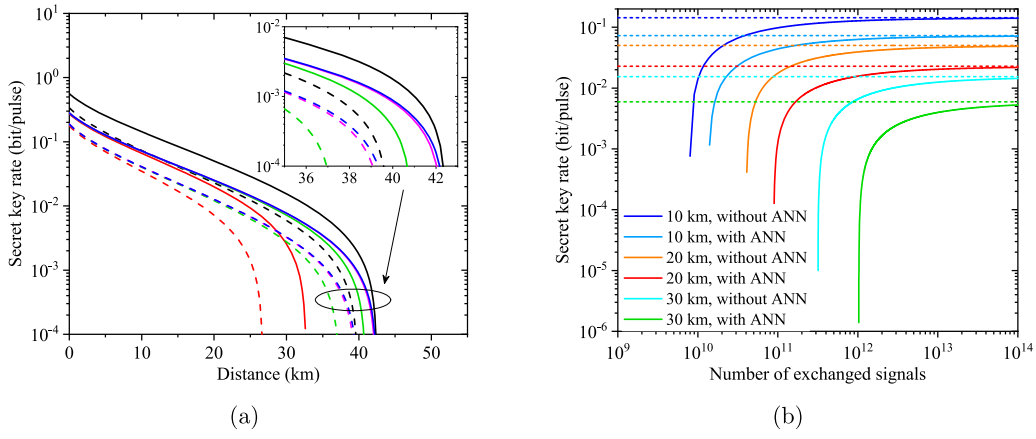

Figure 6. (a) Secret key rate against collective attacks in the asymptotic and finite-size regime. The solid lines correspond to the secret key rates of the system without any countermeasures, and the dashed lines correspond to the secret key rates of the system employ the ANN-based attack detection model. From left to right, the curves correspond, respectively, to the number of exchanged signals $N = 10^8$, $N = 10^{10}$, $N = 10^{12}$, $N = 10^{14}$, and the asymptotic case. (b) Composable secret key rates of a CVQKD system with and without using the ANN-based attack detection model. The solid lines from left to right correspond to the composable secret key rates with and without ANN-based attack detection at transmission distances of 10 km, 20 km, and 30 km, respectively. The dashed lines with the same color as the solid lines are their corresponding asymptotic secret key rates under the same conditions. In all the simulations, the insertion loss of the AM on Bob's signal path is set to a typical value of 2.7 dB.

Even after quantum signals are exchanged, the quantum channel is only partially known. The results of the secret key rates for asymptotic and finite-size scenario are plotted in figure 6(a). We can find that in both asymptotic and finite-size cases, the secret key rate and transmission distance of our scheme are diminished comparing with the system without countermeasures, which is due to 10% of pulses are chosen to estimate the shot noise variance and the AM in Bob's signal path introduces extra insertion loss into the system. But

it is deserving of sacrifice a part of secret keys and transmission distance to enhance the overall defense capability of the system. The detailed calculation about the secret key rate in the finite-size regime can be found in appendix C. Finally, we demonstrate the composable secret key rates of a CVQKD system with and without using the ANN-based attack detection model, and the results are plotted in figure 6(b). The composable security is based on the uncertainty of the finite-size effect, which carefully considers the failure probabilities of every step in CVQKD systems and can obtain the tightest secure bound of a protocol [38]. In figure 6(b), the solid lines from left to right correspond to the composable secret key rates with and without ANN-based attack detection at transmission distances of 10 km, 20 km, and 30 km, respectively. The dashed lines with the same color as the solid lines are their corresponding asymptotic secret key rates under the same conditions. We can see that the results are more pessimistic than that obtained in the finite-size and asymptotic regime, but as the number of exchanged signals increases, the composable secret key rates gradually approach the asymptotic values. The detailed calculation about the composable secret key rate can be found in appendix D.

4. Conclusion

In this work, we introduced and experimentally addressed a quantum attack defense strategy for CVQKD systems by using ANN. We considered the impacts of existing attack strategies on the measurable features of signal and LO pulses, and established a set of feature vectors label by different attack types as input of an ANN model. According to the realistic assumption of the attacks, the training and testing data is prepared for performance evaluation. Simulation results show that the trained ANN can automatically identify and classify attacks with precision and recall values above 99%. Interestingly, we find that the performance of the ANN model is sensitive to the number of neurons n_e in the hidden layer, therefore how to select an appropriate values of n_e is important in practical implementation. Comparing with a system that does not adopt any anti-attack countermeasures, our scheme slightly diminished the secret key rate and transmission distance, but it constructed an overall defense model to anti most of the known attack strategies, significantly improves the security of the system.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) (61972418, 61977062, 61872390, 61871407 and 61801522) and the National Natural Science Foundation of Hunan Province, China (2019JJ40352).

Appendix A. Data preparation and realistic assumption of attacks

In order to investigate the performance of the ANN model for attack classification, we need to establish several valid data sets based on a realistic assumption of Alice and Bob's implementation setup, as well as Eve's capability. Firstly, we assume the fixed parameters mentioned above as: $V_A = 10$, $\eta = 0.6$, $\xi = 0.1N_0$, $V_{el} = 0.01N_0$, $T = 10^{-\alpha L/10}$, where L is the transmission distance which is set as a typical value of 30 km and $\alpha = 0.2$ dB km^{-1} is the loss coefficient of the optical fiber. The attenuation values set by Bob are $r_1 = 1$ (no attenuation) and $r_2 = 0.001$ (maximum attenuation). All of these values are selected according to the standard realistic assumption for CVQKD implementations [22, 39]. In a normal condition without any attacks, the mean and variance of the measurement results are given by

$$\bar{y} = 0, \quad V_i = r_i \eta T(V_A N_0 + \xi) + N_0 + V_{el}, \tag{A.1}$$

where $V_i = \{V_1, V_2\}$ corresponds to the values of r_i, the LO power I_{LO} at Bob side is set as 10^7 photons per pulse with 1% fluctuation [26, 40]. Accordingly, the shot noise variance N_0 under normal condition is set as 0.4 based on the calibrated linear relationship in [22].

Secondly, we briefly recall the principles of the above-mentioned attack strategies, including the LO intensity attack, the calibration attack, the saturation attack, the hybrid attack 1 and the hybrid attack 2.

(a) In the LO intensity attack, Eve attacks the signal beam with a general Gaussian collective attack [15, 41] and attacks the LO beam by using a non-changing phase intensity attenuator with attenuation coefficient $k(0 < k < 1)$. By this way, Eve can arbitrarily reduce the excess noise ε estimated by Alice and Bob to zero and hide her attack. For computational simplicity, we assume the variable attenuation

coefficient k of each LO pulse is the same. Therefore, the variance of Bob's measurement results under this attack can be expressed as

$$V_i^{\mathrm{LOIA}} = k[r_i\eta T(V_{\mathrm{A}}N_0 + \xi + \xi_{\mathrm{Gau}}) + N_0 + V_{\mathrm{el}}], \tag{A.2}$$

where

$$\xi_{\mathrm{Gau}} = \frac{(1-\eta T)(N-1)}{\eta T}N_0, \tag{A.3}$$

represents the noise introduced by Eve's Gaussian collective attack, $N = (1 - k\eta T)/k(1 - \eta T)$ represents the variance of Eve's EPR states. Similarly, the shot noise N_0^{LOIA} is also deviated from the initial level as $N_0^{\mathrm{LOIA}} = kN_0$.

(b) In the calibration attack, Eve intercepts a fraction μ of the signal pulses by implementing a partial intercept-resend (PIR) attack and modifies the shape of LO pulses to control the shot noise estimated by legitimate parties. According to the description in [22], the excess noise introduced by calibration attack is expressed as

$$\frac{\xi_{\mathrm{calib}}}{N_0} = \frac{N_0^{\mathrm{calib}}}{N_0}\left[\frac{\xi_{\mathrm{PIR}}}{N_0^{\mathrm{calib}}} + \frac{1}{\eta T}\left(1 - \frac{N_0}{N_0^{\mathrm{calib}}}\right)\right], \tag{A.4}$$

where $\xi_{\mathrm{PIR}} = \xi + 2\mu N_0$ is the excess noise introduced by Eve's PIR attack, N_0^{calib} is the shot noise after calibration attack and N_0 is the shot noise before attack. In order to make the excess noise estimated by Alice and Bob close to zero, the ratio N_0/N_0^{calib} must satisfy

$$\frac{N_0}{N_0^{\mathrm{calib}}} = 1 + 2.1\eta T, \tag{A.5}$$

with $\mu = 1$ and a typical value of $\xi/N_0^{\mathrm{calib}} = 0.1$. (A.5) indicates that the original shot noise N_0 is reduced into N_0^{calib} by a factor of $\delta = 1/(1 + 2.1\eta T)$. Therefore, the variance of the measurement results under this attack can be expressed as

$$V_i^{\mathrm{calib}} = r_i\eta T(V_{\mathrm{A}}N_0^{\mathrm{calib}} + \varepsilon N_0^{\mathrm{calib}} + 2N_0^{\mathrm{calib}}) + N_0^{\mathrm{calib}} + v_{\mathrm{el}}N_0^{\mathrm{calib}}. \tag{A.6}$$

(c) In the saturation attack, Eve exploits the finite linearity domain of the homodyne detection response. In order to saturate Bob's detector, She intercepts all the pulses send by Alice and measures them with heterodyne detection, then displaces the quadratures of the resent coherent states with a value Δ. As shown in [24], the mean and variance of Bob under saturation attack are expressed as

$$\bar{y}^{\mathrm{sat}} = r_i(\alpha + C), \tag{A.7}$$

$$V_i^{\mathrm{sat}} = V_i'\left(\frac{1+A}{2} - \frac{B^2}{2\pi}\right) - (\alpha - \Delta)\sqrt{\frac{V_i'}{2\pi}}A*B + \frac{(\alpha-\Delta)^2}{4}(1 - A^2), \tag{A.8}$$

where

$$V_i' = r_i\eta T(V_{\mathrm{A}}N_0 + \xi + 2N_0) + N_0 + V_{\mathrm{el}}, \tag{A.9}$$

$$A = \mathrm{erf}\left(\frac{\alpha - \Delta}{\sqrt{2\,\mathrm{V}_i'}}\right), \tag{A.10}$$

$$B = \mathrm{e}^{-(\alpha-\Delta)^2/2\,\mathrm{V}_i'}, \tag{A.11}$$

$$C = -\left[\sqrt{\frac{V_i'}{2\pi}}B + \frac{(\alpha-\Delta)}{2} + \frac{(\alpha-\Delta)}{2}A\right], \tag{A.12}$$

in which α is the boundary of the linear range of the homodyne detector, and the function $\mathrm{erf}(x)$ is the error function defined as

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \mathrm{e}^{-t^2}\mathrm{d}t. \tag{A.13}$$

(d) In the hybrid attack 1, we consider the strategy A that consists of two attack parts. The first part is similar with the LO intensity attack, Eve performs intercept-resend attack to obtain the information sent by Alice and prepares new signal and LO pulses with amplitude $\sqrt{\lambda T}(X_{\mathrm{E}} + \mathrm{i}P_{\mathrm{E}})/2$ and $\alpha_{\mathrm{LO}}/\sqrt{\lambda}$, respectively, where X_{E} and P_{E} are the quadrature values measured by Eve, α_{LO} is the amplitude of the original LO and λ is a real number. In the second attack part Eve prepares and resends two extra

Table 2. Parameters used to generate the data sets of the normal unattacked data and the five attack strategies.

Data sets	Parameters for data generation
$\mathbf{y}_{\text{normal}}$	$\bar{y}$, V_i, I_{LO}
$\mathbf{y}_{\text{LOIA}}$	$\bar{y}$, V_i^{LOIA}, kI_{LO}
$\mathbf{y}_{\text{calib}}$	$\bar{y}$, V_i^{calib}, I_{LO}
$\mathbf{y}_{\text{sat}}$	$\bar{y}^{\text{sat}}$, V_i^{sat}, I_{LO}
$\mathbf{y}_{\text{hyb1}}$	$\bar{y}$, V_i^{hyb1}, I_{LO}/λ
$\mathbf{y}_{\text{hyb2}}$	$\bar{y}$, V_i, ξ_{ext}, D_{ext}, α, I_{LO}

coherent pulses with wavelengths different from the typical communication wavelength of 1550 nm, so that makes the shot noise measurement results seem normal. The variance of Bob's measurement results is given by

$$V_i^{\text{hyb1}} = r_i\eta T(V_{\text{A}}N_0 + 2N_0 + \xi) + \frac{N_0}{\lambda} + V_{\text{el}} + (1-r_i)^2D^2 + (35.81 + 35.47r_i^2)D, \tag{A.14}$$

where D depends on the intensities I^{s}, I^{lo} and wavelengths λ^{s}, λ^{lo} of the extra two pulses. The shot noise level and excess noise estimated by legitimate parties are expressed as

$$N_0^{\text{hyb1}} = \frac{N_0}{\lambda} + (1-r_1r_2)D^2 + (35.81 - 35.47r_1r_2)D, \tag{A.15}$$

$$\frac{\xi^{\text{hyb1}}}{N_0^{\text{hyb1}}} = \left[\frac{(2+\xi)N_0 + (r_1+r_2-2)D^2}{\eta T} + 35.47(r_1+r_2)D\right]. \tag{A.16}$$

(e) In the hybrid attack 2, Eve performs a full intercept-resend attack, and inserts external pluses into the signal port of Bob's homodyne detector along with the re-prepared signals. The pulse width and repetition rate of the external pulses are the same as the pulses sent by Alice. But the wavelength of them is slightly different with Alice's signals, in order to saturate Bob's homodyne detector output. In this way, the external light causes a non-negligible offset on the measurement results of Bob, which is given by

$$D_{\text{ext}} = \sqrt{\eta/I_{\text{LO}}}(1-2T_{\text{ext}})I_{\text{ext}}, \tag{A.17}$$

where T_{ext} is the overall transmission of Bob's homodyne detector regarding the external pulses and is related to the wavelength of the pulse, I_{ext} is the number of photons per pulse of the external light, and D_{ext} is normalized in $\sqrt{N_0}$. The excess noise of the system under this attack becomes

$$\xi_{\text{hyb2}} = \xi + \xi_{\text{IR}} + \xi_{\text{ext}}, \tag{A.18}$$

where $\xi_{\text{IR}} = 2N_0$ is the noise caused by the intercept-resend attack, and ξ_{ext} is the noise caused by the external light, which is related to the value of I_{ext}.

Thirdly, we define the values of the parameters employed in different attack types. For the LO intensity attack, we set the LO fluctuation rate $1-k$ as 0.05 since the analysis in [23] shows that Eve can obtain the full secret keys with an LO fluctuation rate of 0.05 at a transmission distance of 30 km. For the calibration attack, the value of δ is set according to the specific values of η and T based on the equation $\delta = 1/(1+2.1\eta T)$. For the saturation attack, the value of α is set to $20\sqrt{N_0}$ and the value of Δ is set to $19.5\sqrt{N_0}$ since the analysis in [24] shows that the value of Δ should close to α for better attack effect. For the hybrid attack 1, the values of D and λ are selected according to the equations (A.15) and (A.16) to make $N_0^{\text{hyb1}} = N_0$ and $\xi^{\text{hyb1}}/N_0^{\text{hyb1}}$ arbitrarily cloze to zero. For the hybrid attack 2, the value of T_{ext} is set as 0.49, and the value of I_{ext} is selected according to the specific parameter values to make the estimated excess noise smaller than the null key threshold.

Finally, in order to explain the data preparation process more clearly, we summarize the parameters used to generate the data sets for the normal unattacked situation and five attacks strategies, as shown in table 2. The size of each type of data set is $1 \times N$, where 90% of the values in each data set are generated based on $r_i = r_1$, and 10% of the values are generated based on $r_i = r_2$. For example, we generate two groups of normal data, the first group is $\mathbf{y}_1 = \{y_1, y_2, \ldots, y_{N-0.1N}\}$ which follows a Gaussian distribution with zero mean and variance $V_1 = r_1\eta T(V_{\text{A}}N_0 + \xi) + N_0 + V_{\text{el}}$, the second group is $\mathbf{y}_2 = \{y_1, y_2, \ldots, y_{0.1N}\}$ which follows a Gaussian distribution with zero mean and variance $V_2 = r_2\eta T(V_{\text{A}}N_0 + \xi) + N_0 + V_{\text{el}}$. Combining

the two groups of data evenly and obtaining $\mathbf{y}_{\text{normal}} = \{y_1, y_2, \ldots, y_N\}$, which means that 10% of the data in $\mathbf{y}_{\text{normal}}$ is generated for shot noise estimation. In order to establish feature vectors, we divide $\mathbf{y}_{\text{normal}}$ into M blocks $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_M\}$. In each block $\mathbf{b}_m$, the values from $\mathbf{y}_1$ are used for calculating the mean $\bar{y}_m$ and variance V_y^m of this block, the values from $\mathbf{y}_2$ are used for estimating the shot noise variance N_0^m of this block. The LO power of this block is obtained by calculating the average power of the pulses in the current block. Among all of the data sets, $\mathbf{y}_{\text{hyb2}}$ is generated a little differently from the others. Firstly, we generate two groups of data $\mathbf{y}_1$ and $\mathbf{y}_2$. Then, add a value of $D_{\text{ext}}\sqrt{N_0}$ on them, respectively. For each value y_i in these two groups, perform the following calculation, as

$$y_i = \begin{cases} \alpha, & y_i \geqslant \alpha, \\ y_i, & y_i < \alpha. \end{cases} \tag{A.19}$$

Finally, combine the resulting two groups of values evenly and obtain $\mathbf{y}_{\text{hyb2}}$. It is worth noting that we did not describe how to set the value of shot noise N_0 in table 2 because N_0 can be calculated based on the specific data in each block.

Appendix B. Calculation of asymptotic secret key rate

The asymptotic secret key rate under collective attacks with reverse reconciliation is given by equation (11), where the mutual information I_{AB} between Alice and Bob is derived from Bob's measured values $V_{\text{B}} = \eta T(V + \chi_{\text{tol}})$ and the conditional variance $V_{\text{B|A}} = \eta T(1 + \chi_{\text{tol}})$ by using Shannon's equation,

$$I_{\text{AB}} = \frac{1}{2}\log_2\frac{V_{\text{B}}}{V_{\text{B|A}}} = \frac{1}{2}\log_2\frac{V + \chi_{\text{tol}}}{1 + \chi_{\text{tol}}}, \tag{B.1}$$

where $\chi_{\text{tol}} = \chi_{\text{line}} + \chi_{\text{hom}}/T$ represents the total noise referred to the channel input. $\chi_{\text{line}} = T^{-1} + \varepsilon - 1$ is the channel-added noise referred to the channel input and $\chi_{\text{hom}} = [(1-\eta) + v_{\text{el}}]/\eta$ is the detection-added noise referred to Bob's input. χ_{BE} denotes the maximum information available to Eve on Bob's key, which is given by

$$\chi_{\text{BE}} = S(\rho_{\text{E}}) - \int \mathrm{d}_{m_{\text{B}}} p(m_{\text{B}}) S(\rho_{\text{E}}^{m_{\text{B}}}), \tag{B.2}$$

where m_{B} denotes the measurement of Bob, $p(m_{\text{B}})$ denotes the probability density of the measurement, $\rho_{\text{E}}^{m_{\text{B}}}$ denotes Eve's state conditional on Bob's measurement, and S denotes the Von Neumann entropy of the quantum state ρ. In the case of Gaussian attack, equation (B.2) can be simplified to

$$\chi_{\text{BE}} = \sum_{i=1}^{2} G\left(\frac{\lambda_i - 1}{2}\right) - \sum_{i=3}^{5} G\left(\frac{\lambda_i - 1}{2}\right), \tag{B.3}$$

where $G(x) = (x+1)\log_2(x+1) - x\log_2(x)$. $\lambda_{1,2}$ are the symplectic eigenvalues given by

$$\lambda_{1,2}^2 = \frac{1}{2}\left(A \pm \sqrt{A^2 - 4B}\right), \tag{B.4}$$

with

$$A = V^2 + T^2(V + \chi_{\text{line}})^2 + 2T(1 - V^2), \tag{B.5}$$

$$B = T^2(1 + V\chi_{\text{line}})^2. \tag{B.6}$$

$\lambda_{3,4}$ are the symplectic eigenvalues given by

$$\lambda_{3,4}^2 = \frac{1}{2}\left(C \pm \sqrt{C^2 - 4D}\right), \tag{B.7}$$

with

$$C = \frac{A\chi_{\text{hom}} + V\sqrt{B} + T(V + \chi_{\text{line}})}{T(V + \chi_{\text{tot}})}, \tag{B.8}$$

$$D = \frac{\sqrt{B}V + B\chi_{\text{hom}}}{T(V + \chi_{\text{tot}})}. \tag{B.9}$$

The last symplectic eigenvalue $\lambda_5 = 1$. Based on the above equations, we can obtain the secret key rate of the CVQKD system without taking any countermeasures against attacks. When calculating the secret key rate of our scheme, the insertion loss of the *AM* on Bob's signal path should be taken into consideration, as well as the 10% pulses used for real-time shot-noise measurement.

Appendix C. Secret key rate in finite-size scenario

The secret key rate of a CVQKD system considering finite-size effects is given by [37]

$$K_{\text{finite}} = \frac{n}{N}[\beta I_{\text{AB}} - S_{\text{BE}}^{\epsilon_{\text{PE}}} - \triangle(n)], \tag{C.1}$$

where N denotes the number of the exchanged signals between Alice and Bob, and n denotes the number of the signals used for key establishment. $m = N - n$ indicates the number of the remaining signals used for parameter estimation. ϵ_{PE} indicates the failure probability of parameter estimation. $\triangle(n)$ is related to the security of the privacy amplification, which is given by

$$\triangle(n) = (2\ \dim\ \mathcal{H}_Y + 3)\sqrt{\frac{\log_2(2/\bar{\epsilon})}{n}} + \frac{2}{n}\log_2(1/\epsilon_{\text{PA}}), \tag{C.2}$$

where $\bar{\epsilon}$ is a smoothing parameter, ϵ_{PA} is the failure probability of the privacy amplification procedure, and $\mathcal{H}_Y$ is the Hilbert space corresponding to the variable y used in the raw key. We take $\dim\ \mathcal{H}_Y = 2$ for secret key rate evaluation since the raw key is encoded on bits. $S_{\text{BE}}^{\epsilon_{\text{PE}}}$ represents the mutual information between Bob and Eve, which is determined by the covariance matrix Γ_{AB} of the bipartite state shared by Alice and Bob after the quantum channel, that is

$$\Gamma_{\text{AB}} = \begin{bmatrix} (V_{\text{A}}+1)\mathbb{I}_2 & \sqrt{T_{\min}(V_{\text{A}}^2+2V_{\text{A}})}\sigma_z \\ \sqrt{T_{\min}(V_{\text{A}}^2+2V_{\text{A}})}\sigma_z & [T_{\min}(V_{\text{A}}+\varepsilon_{\max})+1]\mathbb{I}_2 \end{bmatrix}, \tag{C.3}$$

where the matrices $\mathbb{I}_2 = \text{diag}(1,1)$ and $\sigma_z = \text{diag}(1,-1)$. $T_{\min}$ and $\varepsilon_{\max}$ correspond, respectively, to the lower and upper bound of T and ε, which are defined as

$$T_{\min} = \frac{\hat{t}_{\min}^2}{\eta}, \qquad \varepsilon_{\max} = \frac{\hat{\sigma}_{\max}^2 - 1 - v_{\text{el}}}{\eta T}, \tag{C.4}$$

with

$$\hat{t}_{\min} \approx \sqrt{\eta T} - z_{\epsilon_{\text{PE}}/2}\sqrt{\frac{1+\eta T\varepsilon + v_{\text{el}}}{mV_{\text{A}}}}, \tag{C.5}$$

$$\hat{\sigma}_{\max}^2 \approx 1 + \eta T\varepsilon + v_{\text{el}} + z_{\epsilon_{\text{PE}}/2}\sqrt{\frac{(1+\eta T\varepsilon + v_{\text{el}})\sqrt{2}}{\sqrt{m}}}, \tag{C.6}$$

where $z_{\epsilon_{\text{PE}}/2}$ follows $1 - \text{erf}(z_{\epsilon_{\text{PE}}/2}/\sqrt{2})/2 = \epsilon_{\text{PE}}/2$. Substituting $T_{\min}$ and $\varepsilon_{\max}$ for the parameters T and ε used in equation (B.3), we can obtain the secret key rate in finite-size scenario. In the simulations, the above-mentioned error probabilities are set to

$$\bar{\epsilon} = \epsilon_{\text{PE}} = \epsilon_{\text{PA}} = 10^{-10}. \tag{C.7}$$

Appendix D. Secret key rate in composable security

In the composable security framework, the secret key rate of a CVQKD protocol against collective attacks is given by [38]

$$K_{\text{comp}} = (1-\epsilon_{\text{rob}})\left\{\beta I_{\text{AB}} - f(\Sigma_a^{\max}, \Sigma_b^{\max}, \Sigma_c^{\min}) - \frac{1}{N}\left[\triangle_{\text{AEP}} + \triangle_{\text{ent}} + 2\log\frac{1}{2\bar{\epsilon}}\right]\right\}, \tag{D.1}$$

where ϵ_{rob} indicates the robustness of the protocol. f is the function computing the Holevo information between Eve and Bob's measurement results for a Gaussian state with covariance matrix parametrized by $\Sigma_a^{\max}$, $\Sigma_b^{\max}$, and $\Sigma_c^{\min}$, that is

$$f(\Sigma_a^{\max}, \Sigma_b^{\max}, \Sigma_c^{\min}) = G\left(\frac{\nu_1 - 1}{2}\right) + G\left(\frac{\nu_2 - 1}{2}\right) - G\left(\frac{\nu_3 - 1}{2}\right), \tag{D.2}$$

where ν_1 and ν_2 are the symplectic eigenvalues of the covariance matrix $\begin{bmatrix} \Sigma_a^{\max}\mathbb{I}_2 & \Sigma_c^{\min}\sigma_z \\ \Sigma_c^{\min}\sigma_z & \Sigma_b^{\max}\mathbb{I}_2 \end{bmatrix}$, $\nu_3 = \Sigma_a^{\max} - (\Sigma_c^{\min})^2/(1 + \Sigma_b^{\max})$. More explicitly,

$$\nu_1^2 + \nu_2^2 = \Sigma_a^{\max 2} + \Sigma_b^{\max 2} - 2\Sigma_c^{\min 2}, \tag{D.3}$$

$$\nu_1^2\nu_2^2 = (\Sigma_a^{\max}\Sigma_b^{\max} - \Sigma_c^{\min 2})^2. \tag{D.4}$$

Then we define

$$\Sigma_a^{\max} = \frac{1}{N}\left[1 + 2\sqrt{\frac{\log_2(36/\epsilon_{\rm PE})}{N/2}}\right]\|X\|^2 - 1, \tag{D.5}$$

$$\Sigma_B^{\max} = \frac{1}{N}\left[1 + 2\sqrt{\frac{\log_2(36/\epsilon_{\rm PE})}{N/2}}\right]\|Y\|^2 - 1, \tag{D.6}$$

$$\Sigma_c^{\min} = \frac{\langle X, Y\rangle}{N} - 5\sqrt{\frac{\log_2(8/\epsilon_{\rm PE})}{(N/2)^3}}(\|X\|^2 + \|Y\|^2). \tag{D.7}$$

Assuming that the success probability of parameter estimation is at least 0.99, thereby the robustness of the protocol is $\epsilon_{\rm rob} \leqslant 0.01$, and the random variables $\|X\|^2$, $\|Y\|^2$, and $\langle X, Y\rangle$ satisfy the following restrains

$$\|X\|^2 \leqslant N(V + 1) + 3\sqrt{2N(V + 1)}, \tag{D.8}$$

$$\|X\|^2 \leqslant N(TV_{\rm A} + T\varepsilon + 1) + 3\sqrt{2N(TV_{\rm A} + T\varepsilon + 1)}, \tag{D.9}$$

$$\langle X, Y\rangle \leqslant N\sqrt{T(V^2 - 1)} - 3\sqrt{V_{\rm A}(T\varepsilon + 1)N/2}. \tag{D.10}$$

The parameters $\triangle_{\rm AEP}$ and $\triangle_{\rm ent}$ in equation (D.1) can be obtained by

$$\triangle_{\rm AEP} = \sqrt{N}\left[(d + 1)^2 + 4(d + 1)\log_2\frac{2}{\epsilon_{\rm sm}^2} + 2\log_2\frac{2}{\epsilon^2\epsilon_{\rm sm}}\right] + 4\frac{\epsilon_{\rm sm}d}{\epsilon}, \tag{D.11}$$

$$\triangle_{\rm ent} = \log_2\frac{1}{\epsilon} + \sqrt{2N\log_2^2 N\log_2\frac{2}{\epsilon_{\rm sm}}}, \tag{D.12}$$

where d is the discretization parameter. $\epsilon = \sqrt{\epsilon_{\rm PE} + \epsilon_{\rm cor} + \epsilon_{\rm ent}} + 2\epsilon_{\rm sm} + \bar{\epsilon}$ is a possible security parameter. In the simulations, we choose $\epsilon_{\rm sm} = \bar{\epsilon} = 10^{-21}$, $\epsilon_{\rm PE} = \epsilon_{\rm cor} = \epsilon_{\rm ent} = 10^{-41}$, and $d = 5$ for simplicity.

ORCID iDs

Yiyu Mao https://orcid.org/0000-0002-4693-6408
Wenti Huang https://orcid.org/0000-0002-3216-3440
Ying Guo https://orcid.org/0000-0002-9306-2061

References

[1] Scarani V, Bechmann-Pasquinucci H, Cerf N J, Dušek M, Lütkenhaus N and Peev M 2009 *Rev. Mod. Phys.* **81** 1301
[2] Gisin N, Ribordy G, Tittel W and Zbinden H 2002 *Rev. Mod. Phys.* **74** 145
[3] Weedbrook C, Pirandola S, García-Patrón R, Cerf N J, Ralph T C, Shapiro J H and Lloyd S 2012 *Rev. Mod. Phys.* **84** 621
[4] Xu F, Qi B and Lo H K 2010 *New J. Phys.* **12** 113026
[5] Lydersen L, Wiechers C, Wittmann C, Elser D, Skaar J and Makarov V 2010 *Nat. Photon.* **4** 686
[6] Lo H K, Curty M and Tamaki K 2014 *Nat. Photon.* **8** 595
[7] Bennett C H and Brassard G 2014 *Theor. Comput. Sci.* **560** 7–11
[8] Xu F, Curty M, Qi B, Qian L and Lo H K 2015 *Nat. Photon.* **9** 772–3
[9] Grosshans F and Grangier P 2002 *Phys. Rev. Lett.* **88** 057902
[10] Lance A M, Symul T, Sharma V, Weedbrook C, Ralph T C and Lam P K 2005 *Phys. Rev. Lett.* **95** 180503
[11] Gong L H, Song H C, He C S, Liu Y and Zhou N R 2014 *Phys. Scr.* **89** 035101
[12] Wang C, Huang P, Huang D, Lin D and Zeng G 2016 *Phys. Rev.* A **93** 022315
[13] Fossier S, Diamanti E, Debuisschert T, Tualle-Brouri R and Grangier P 2009 *J. Phys. B: At. Mol. Opt. Phys.* **42** 114014
[14] Jouguet P, Kunz-Jacques S, Leverrier A, Grangier P and Diamanti E 2013 *Nat. Photon.* **7** 378–81
[15] Garcia-Patron R and Cerf N J 2006 *Phys. Rev. Lett.* **97** 190503

[16] Furrer F, Franz T, Berta M, Leverrier A, Scholz V B, Tomamichel M and Werner R F 2012 *Phys. Rev. Lett.* **109** 100502
[17] Leverrier A 2017 *Phys. Rev. Lett.* **118** 200501
[18] Gisin N, Fasel S, Kraus B, Zbinden H and Ribordy G 2006 *Phys. Rev.* A **73** 022320
[19] Jain N, Anisimova E, Khan I, Makarov V, Marquardt C and Leuchs G 2014 *New J. Phys.* **16** 123030
[20] Huang J Z, Weedbrook C, Yin Z Q, Wang S, Li H W, Chen W, Guo G C and Han Z F 2013 *Phys. Rev.* A **87** 062329
[21] Ma X C, Sun S H, Jiang M S and Liang L M 2013 *Phys. Rev.* A **87** 052309
[22] Jouguet P, Kunz-Jacques S and Diamanti E 2013 *Phys. Rev.* A **87** 062313
[23] Ma X C, Sun S H, Jiang M S and Liang L M 2013 *Phys. Rev.* A **88** 022339
[24] Qin H, Kumar R and Alléaume R 2016 *Phys. Rev.* A **94** 012325
[25] Qin H, Kumar R, Makarov V and Alléaume R 2018 *Phys. Rev.* A **98** 012312
[26] Liu W, Peng J, Huang P, Huang D and Zeng G 2017 *Opt. Express* **25** 19429–43
[27] Huang J Z, Kunz-Jacques S, Jouguet P, Weedbrook C, Yin Z Q, Wang S, Chen W, Guo G C and Han Z F 2014 *Phys. Rev.* A **89** 032304
[28] Kurt H, Maxwell S and Halbert W 1989 *Neural Netw.* **2** 359–66
[29] Saritas M M and Yasar A 2019 *Int. J. Intell. Syst. Appl. Eng.* **7** 88–91
[30] Lau M M and Lim K H 2017 Investigation of activation functions in deep belief network *2nd International Conference on Control and Robotics Engineering (ICCRE) (IEEE)* pp 201–6
[31] Zhang H, Weng T W, Chen P Y, Hsieh C J and Daniel L 2018 Efficient neural network robustness certification with general activation functions *Advances in Neural Information Processing Systems 31* (Red Hook, NY: Curran Associates) pp 4939–48
[32] Zeiler M D 2012 arXiv:1212.5701
[33] Bottou L 2010 Large-scale machine learning with stochastic gradient descent *Proceedings of COMPSTAT'2010* (Berlin: Springer) pp 177–86
[34] Bergstra J and Bengio Y 2012 *J. Mach. Learn. Res.* **13** 281–305
[35] Jolliffe I T and Cadima J 2016 *Phil. Trans. R. Soc.* A **374** 20150202
[36] Wang X, Tang Z, Tamura H, Ishii M and Sun W 2004 *Neurocomputing* **56** 455–60
[37] Leverrier A, Grosshans F and Grangier P 2010 *Phys. Rev.* A **81** 062343
[38] Leverrier A 2015 *Phys. Rev. Lett.* **114** 070501
[39] Fossier S, Diamanti E, Debuisschert T, Villing A, Tualle-Brouri R and Grangier P 2009 *New J. Phys.* **11** 045023
[40] Huang D, Lin D, Wang C, Liu W, Fang S, Peng J, Huang P and Zeng G 2015 *Opt. Express* **23** 17511–9
[41] Navascués M, Grosshans F and Acin A 2006 *Phys. Rev. Lett.* **97** 190502