Quantum Information Processing (2023) 22:362
https://doi.org/10.1007/s11128-023-04124-9

Hardware design and implementation of high-speed multidimensional reconciliation sender module in continuous-variable quantum key distribution

Shenshen Yang[1] · Zhilei Yan[1] · Qing Lu[2,3] · Hongzhao Yang[2,3] · Zhenguo Lu[2,3] · Xiangyang Miao[1] · Yongmin Li[2,3,4]

Received: 6 June 2023 / Accepted: 15 September 2023

Abstract

In continuous-variable quantum key distribution systems, the multidimensional reconciliation process can correct errors in raw keys at long transmission distances, so that both communicating parties can obtain an identical set of corrected key strings. In this paper, we designed and implemented a hardware acceleration scheme for the complete multidimensional reconciliation sender module using the parallel processing capability of FPGAs. To this end, we simplified the matrix operation based on the characteristic of fewer nonzero elements at the matrix family $\mathcal{A}_8$ and customized an exclusive circuit structure. In addition, we also constructed a multi-edge type (MET) LDPC code with good decoding performance and implemented a high-speed MET-LDPC decoding module based on FPGA. When the signal-to-noise ratio is 0.16, the simulation results show that the reconciliation efficiency is 93.4%, the frame error rate is 19%, and the throughput can reach 9.6 Mbps, which is two orders of magnitude higher compared with CPU. Our work paves the way for the miniaturization and integration of CV-QKD systems.

✉ Xiangyang Miao
sxxymiao@126.com

✉ Yongmin Li
yongmin@sxu.edu.cn

Shenshen Yang
yangssgy@163.com

[1] College of Physics and Information Engineering, Shanxi Normal University, Taiyu Road, Taiyuan 030031, Shanxi, China

[2] State Key Laboratory of Quantum Optics and Quantum Optics Devices, Institute of Opto-Electronics, Shanxi University, Wucheng Road, Taiyuan 030006, Shanxi, China

[3] Collaborative Innovation Center of Extreme Optics, Shanxi University, Wucheng Road, Taiyuan 030006, Shanxi, China

[4] Hefei National Laboratoty, Hefei 230088, China

Published online: 03 October 2023

Keywords Quantum key distribution (QKD) · Continuous-variable · Multidimensional reconciliation · Field programmable gate array (FPGA) · Sender

1 Introduction

Quantum key distribution (QKD) technology relies on the laws of quantum mechanics to provide information-theoretical secure keys for both legitimate communicating parties [1, 2]. Even if the eavesdropper has infinite computing power, she cannot eavesdrop on any information. The basic process of generating secret keys can be outlined as: the sender Alice sends the encoded quantum states to the receiver Bob through a quantum channel, Bob uses a random measurement base to measure the received quantum states, and then, the key can be extracted through the process of key sifting, parameter estimation, information reconciliation, and privacy amplification [3].

Continuous-variable QKD (CV-QKD) [4, 5] is an important branch of QKD with the advantages of good compatibility with existing communication techniques, and high key rates in the metropolitan area. In recent years, a number of progress have been made in various aspects such as protocol design, security analysis, and experimental techniques [6–16]. However, the signal-to-noise ratio (SNR) of CV-QKD systems is usually much lower than classical communication protocols, leading to a relatively complex information reconciliation process, and digital signal processing technologies sometimes need to be introduced to improve the SNR [17, 18].

In the information reconciliation process, the communicating parties use classical error correction codes to correct the errors in the raw keys to obtain a completely identical data string. For Gaussian signals in CV-QKD systems, the commonly used information reconciliation schemes are slice reconciliation [19] and multidimensional reconciliation [20]. Among them, slice reconciliation is suitable for the case of short communication distance and relatively large SNRs; while multidimensional reconciliation can achieve error correction at lower SNRs and can support the system to achieve longer transmission distance, therefore, it has gained wide attention in recent years [21, 22].

Due to the high error rate of raw keys in CV-QKD systems, error correction is difficult and requires very large code length error correction codes, which leads to high computational effort and very low data throughput. One possible way to improve the throughput is to accelerate the algorithm using hardwares, such as the graphics processing unit (GPU) [23–28] or the field programmable gate array (FPGA) [29, 30]. Among them, FPGAs have the advantages of powerful parallel operation, low power consumption, and flexible configuration of parallelism, and their powerful control and programmability make them very attractive for designing prototypes and manufacturing small production-run devices [31–33].

In this paper, we designed a hardware acceleration scheme of a multidimensional reconciliation sender module based on FPGA. The proprietary circuit structure was customized according to the characteristics of the multidimensional reconciliation process with fewer nonzero elements of the orthogonal matrix, which effectively simplifies the matrix operations in hardware. Furthermore, we have constructed multi-edge-type

(MET) LDPC codes with good decoding performance and improved and implemented a high-speed LDPC code iterative decoding module based on FPGA. In the decoding module, we use the methods of shortening the code length and fixed-point number calculations to reduce the consumption of storage resources. The simulation results show that the reconciliation efficiency is 93.4%, the frame error rate is 19%, and the throughput can reach 9.6 Mbps when the SNR is 0.16. The consumption of key hardware resources such as look-up table (LUT), Block RAM (BRAM), and DSP is less than one-third of the Virtex-7XC7VX690T FPGA.

The following paper is organized as follows: In Sect. 2, we introduce the basic principle of multidimensional reconciliation and its impact on secret key rate. In Sect. 3, we present the construction method and decoding algorithm of MET-LDPC codes. In Sect. 4, we present the detailed design scheme of the sender module based on FPGA. In Sect. 5, we present the simulation and implementation results. Finally, the conclusion and outlook are drawn in Sect. 6.

2 Multidimensional reconciliation

The CV-QKD system usually works at a very low SNR, resulting in signals concentrated near zero, which is difficult to discriminate. To increase the discrimination, a multidimensional spatial rotation operation can be performed on the signals to convert the Gaussian signals into binary signals. The schematic of multidimensional reconciliation is shown in Fig. 1.

After the process of preparation, transmission and measurement of quantum states, the communicating parties share a set of interrelated Gaussian variables $\boldsymbol{X}$ and $\boldsymbol{Y}$. Both parties form a vector for each d elements of this Gaussian sequence, labeled $\boldsymbol{X}'$ and $\boldsymbol{Y}'$ and then, normalize each d-dimensional vector as follows:

$$\boldsymbol{x}' = \frac{\boldsymbol{X}'}{\left|\boldsymbol{X}'\right|}, \ \boldsymbol{y}' = \frac{\boldsymbol{Y}'}{\left|\boldsymbol{Y}'\right|}, \tag{1}$$

where $\left|\boldsymbol{X}'\right|$ and $\left|\boldsymbol{Y}'\right|$ denote the modulo of vectors. After normalization, the random vectors are transformed into signal points on an unit sphere and the Gaussian variables are transformed into $\boldsymbol{x}'$ and $\boldsymbol{y}'$. Then, a set of random bit strings $\boldsymbol{u} = \{b_1, b_2, \ldots, b_i\}$ of the same length as the Gaussian variables that obeying a uniform distribution is generated by a true random number generator (TRNG) at the receiver side, and each random bit string is transformed into a d-dimensional spherical vector as follows

$$\boldsymbol{u}' = \left(\frac{(-1)^{b_1}}{\sqrt{d}}, \frac{(-1)^{b_2}}{\sqrt{d}}, \ldots, \frac{(-1)^{b_d}}{\sqrt{d}}\right), \tag{2}$$

The receiver performs a d-dimensional spatial rotation operation to compute α, such that it satisfies $\alpha \cdot \boldsymbol{y}' = \boldsymbol{u}'$. $\boldsymbol{u}$ is encoded to generate the syndrome $\boldsymbol{S}$. Then, α and $\boldsymbol{S}$ are sent to Alice. Alice uses the received information to compute the mapping

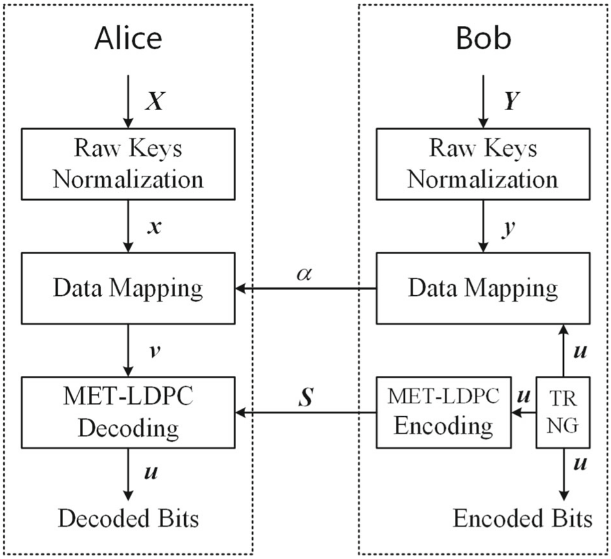

Fig. 1 Schematic of multidimensional reconciliation based on MET-LDPC codes. The inputs $\boldsymbol{X}$ and $\boldsymbol{Y}$ are a set of interrelated but not identical Gaussian variables. α and S denote the mapping function and the syndrome, respectively, which are computed by the receiver and sent to the sender. TRNG (True Random Number Generator) is used to generate a set of random bit strings. The outputs *Decoded Bits* and *Encoded Bits* are an identical set of corrected key strings

function M'. Then, she calculates $\boldsymbol{v}$ using the equation $M' \cdot \boldsymbol{x}' = \boldsymbol{v}$. Finally, Alice uses the computed $\boldsymbol{v}$ as side information to recover the exactly same $\boldsymbol{u}$ as Bob by decoding.

The reconciliation efficiency can be expressed as $\beta = R/C$, where R denotes the code rate of the error-correcting code and C denotes the channel capacity. The secret key rate K of the CV-QKD system can be expressed as

$$K = \gamma \left(1 - R_{\mathrm{FE}}\right) \left(\beta I_{AB} - \chi_{BE}\right), \tag{3}$$

where γ denotes the effect of throughput on the key rate. If the throughput is greater than or equal to the generation rate of raw keys, then we have $\gamma = 1$, otherwise $\gamma < 1$; R_{FE} denotes the frame error rate; I_{AB} denotes the amount of mutual information between the communicating parties; χ_{BE} denotes the amount of information that may be leaked to the eavesdropper in reverse reconciliation. Equation (3) shows that the throughput, frame error rate and reconciliation efficiency of the information reconciliation all have a large impact on the key rate of the system, so it is crucial to improve the performance of information reconciliation.

3 Multi-edge type LDPC codes

MET-LDPC codes were first proposed by Richardson and Urbanke [34], which have good error correction performance even at very low SNRs, and have been applied to multidimensional reconciliation [22, 25, 35–38]. This section briefly describes the construction of the check matrix of our MET-LDPC code check matrix and the decoding algorithm.

3.1 Construction of the check matrix

The degree distribution characterizes the position of nonzero elements in the check matrix. In previous works, researchers have constructed degree distributions with good performance [25, 35, 36, 39], which give good degree distributions for code rates of 0.01, 0.02, 0.025, 0.1, and 0.5, respectively. There are two main types of ways to place nonzero elements in the matrix according to the degree distribution, which are random constructions and structured constructions.

Random construction [40] means randomly placing a 1 at an unpopulated position in the check matrix and then checking whether the required constraints are violated. If the placement is valid, the process is repeated; if the constraint is violated, new location is tried. This operation is repeated until all nonzero elements are placed. This approach provides maximum freedom for nonzero elements and allows for the best error correction performance. However, it takes a long time when constructing check matrices of very large size.

Since MET-LDPC codes are usually applied in channels with very low SNRs, the size of the check matrix is large and the number of iterations is high. Therefore, the decoding delay time is usually high, and hardware acceleration is desired to increase the computation speed and achieve a lower delay time. The quasi-cyclic structure facilitates parallel processing in hardware. In this paper, we combine the two methods, first, constructing the base matrix with a code length of 10,000 by random construction algorithm, and then, extending the base matrix by the quasi-cyclic algorithm to obtain the check matrix with a code length of 160,000 to achieve the trade-off of error correction performance and time complexity.

3.2 Layered sum-product decoding algorithm

The sum-product decoding algorithm has good decoding performance where the layered message passing mechanism converges faster, can reduce the number of iterations, and can reduce the consumption of storage resources when applied to FPGAs [41]. The algorithm consists of the following three main steps:

Step 1: Initialization: calculating the initial log-likelihood ratios (LLRs) for each variable node:

$$\mathrm{LLR}_i^0 = \ln \frac{P_i\ (0)}{P_i\ (1)}. \tag{4}$$

Step 2: Perform row-by-row processing of the nodes in the check matrix:

$$M_{ji} = \mathrm{LLR}_i^{l-1} - E_{ji}^{l-1}, \tag{5}$$

$$E_{ji}^{l} = \left(1 - s_j\right) \times \prod_{i \in N(j)/i'} \operatorname{sgn}\left(M_{ji}\right)$$

$$\times \Psi \left[\Psi \left(M_{ji} \right) - \sum_{i \in N(j)/i'} \Psi \left(M_{ji} \right) \right], \tag{6}$$

$$\mathrm{LLR}_i^l = M_{ji} + E_{ji}^l, \tag{7}$$

where l denotes the number of iterations, and the function Ψ is defined as $\Psi(x) = -\log\left[\tanh\left(|x|/2\right)\right]$. From the node processing equation, the intermediate variables do not need to be cached into the next iteration process, thus effectively reducing the consumption of storage resources when implementing on FPGA.

Step 3: Decision. The quantization $X = [x_1, x_2, \ldots, x_n]$ is performed, if $\mathrm{LLR}(q_i) \geq 0$, then $x_n = 1$, otherwise $x_n = 0$. If $\boldsymbol{H} \cdot X^T = \boldsymbol{S}$, the result $\boldsymbol{X}$ is the decoded output, otherwise return to step 2 for the next processing. Repeat steps 2 and 3 until $\boldsymbol{H} \cdot X^T = \boldsymbol{S}$, or the number of iterations reaches the given maximum value.

4 Implementation of the high-speed multidimensional reconciliation sender module

In designing and implementing the high-speed multidimensional reconciliation sender module based on FPGA, we use a series of methods such as pipelined operation and parallel processing to improve the throughput. In addition, we use fixed-point number calculation and specific matrix storage format to reduce the consumption of hardware resources. Figure 2 gives the top-level design of the overall design scheme, which is divided into four modules, namely, the raw keys normalization module, the data mapping module, the LLR initialization module, and the iterative decoding module. Since the latter module needs to use the computation results of the previous module, the four modules are carried out in a serial structure. The overall throughput depends on the module with the slowest computation speed. The following section describes the FPGA implementation of the four modules in detail and deduces the theoretical throughput of our scheme.

4.1 Raw keys normalization module

This module needs to construct the vector of d elements from $\boldsymbol{X}$ and then, normalize each d-dimensional vector $\boldsymbol{x}' = \boldsymbol{X}'/\left|\boldsymbol{X}'\right|$. This solution uses eight-dimensional reconciliation to process eight data in parallel when executing square and division operations, and performs two-by-two addition in parallel when executing accumulation. The total number of floating-point IP cores required is eight multipliers, seven adders, one square root, and eight dividers. A total of six clocks is required to complete a single calculation. Taking four-dimensional reconciliation as an example, its logical structure of the raw keys initialization module is shown in Fig. 2a.

Since this module and the followed *data mapping* module and *LLR initialization* module have less computation and require multiplication, logarithm, and exponential operations, thus 32-bit floating-point numbers are adopted for these modules. In order

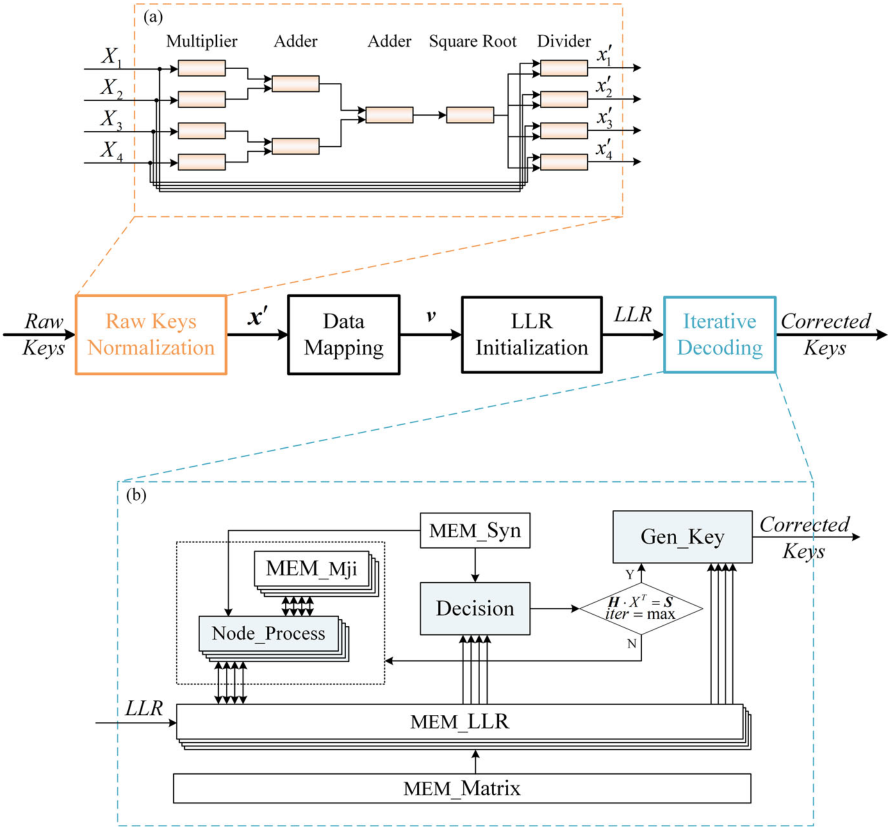

Fig. 2 Schematic diagram of the logic structure of the sender module. **a** The logic structure of raw keys normalization module. Here, we take four-dimensional reconciliation as an example. **b** The logic structure of iterative decoding module, which contains three computation modules and four storage modules. The *Node_Process* module executes Eqs. (5), (6), and (7)

to save LUT resources, the floating-point IP cores used are all implemented using DSP.

4.2 Data mapping module

As mentioned in the previous section, the task to be performed by the data mapping module of the sender is: (1) Calculate the rotation function M' using the α received from Bob; (2) Calculate $\boldsymbol{v}$ using $\boldsymbol{M}' \cdot \boldsymbol{x}' = \boldsymbol{v}$. These calculations require a matrix family [20]. Take the matrix family $\mathcal{A}_4$ as an example, which contains four fourth-order matrices:

$$A_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad A_2 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix}, \tag{8}$$

$$A_3 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \qquad A_4 = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}. \tag{9}$$

From the expression of $\mathcal{A}_4$, we can find that the four matrices have fewer nonzero elements 1 and -1, and their positions do not overlap each other. Therefore, the nonzero elements of these four matrices can be recombined to form a new fourth-order matrix:

$$A_4' = \begin{pmatrix} 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \end{pmatrix} \tag{10}$$

Similarly, in the eight-dimensional reconciliation, the matrix family $\mathcal{A}_8$ consists of eight eighth-order squares with only eight nonzero elements in each matrix, and all nonzero elements can also form a new eighth-order matrix:

$$A_8' = \begin{pmatrix} 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \end{pmatrix} \tag{11}$$

Taking advantage of this feature of the matrix family can greatly simplify operations on FPGAs. When performing matrix calculations, only nonzero elements need to be operated, so the number of IP cores needed is reduced to 1/8. Further, because there are only 1 and -1 in the nonzero elements, we can convert the multiplication operation into an inverse operation. In other words, if the nonzero element is 1, no multiplication is required; if the nonzero element is -1, multiplication can be converted to subtraction. In addition, the accumulation of the eight matrices is equivalent to integrating the nonzero elements into one matrix since the positions of the nonzero elements of the eight matrices do not overlap each other. So, we need not perform floating-point addition, which can save a lot of hardware resources.

4.3 LLR initialization module

The function to be implemented for this module corresponds to Eq. (4), which estimates the initial probability distribution with variable nodes of 0/1 using the data obtained in the previous stage, $|\boldsymbol{X}|$, $|\boldsymbol{Y}|$, and $\boldsymbol{v}$. A total of four multipliers, two dividers, two subtractors, and one exponential operation IP core are required to calculate the probability density function of the Gaussian distribution using IP cores. In the eight-dimensional reconciliation, eight LLR initialization modules are instantiated at the same time, and the LLR is processed in parallel to maximize throughput. After getting the LLR, the 32-bit floating-point number is converted to the fixed-point number and enters into the iterative decoding module.

4.4 Iterative decoding module

The function of this module is to realize the node processing and judgment process of the layered sum-product decoding algorithm, which involves complex calculations and requires several iterations, so that the throughput of this module is bottleneck to restrict the throughput of multidimensional reconciliation process. We have implemented a high-speed decoder for irregular LDPC codes [41] and successfully applied it to slice reconciliation [29]. By optimizing and improving the iterative decoding algorithms, we make it applicable to MET-LDPC codes. The logic structures of the iterative decoding module are shown in Fig. 2b. Firstly, the parallelism is modified according to the needs of the multidimensional reconciliation process. Then, the calculation is deferred to fixed-point numbers, which can save both LUT and storage resource saving compared to 32-bit floating-point numbers. Finally, the storage scheme is optimized according to the characteristics of the check matrix. When caching M_{ji}, only the values are stored sequentially without storing the number of rows. When storing the check matrix, only the number of columns of its nonzero elements are stored row by row, and the rows are swapped according to the change of the number of columns, that is, when the number of the latter column stored is smaller than the previous one.

4.5 Throughput

In the sender module, the four submodules are pipelined, so the total throughput depends on the module with the slowest computational speed. Therefore, the overall throughput is approximately equal to the throughput of the iterative decoding algorithm. It can be approximated as

$$T \approx \frac{f \cdot q}{(1-R) \cdot N_{rn} \cdot N_{\text{iter}}} \tag{12}$$

$$\approx \frac{f \cdot L}{N_{mn} \cdot N_{\text{iter}}} \tag{13}$$

where f denotes the clock frequency of the FPGA chip, q denotes the expansion factor in matrix expansion, N_{iter} denotes the number of iterations in iterative decoding, N_m denotes the average value of nonzero elements in each row of the base matrix, and N_{mn} denotes the total number of nonzero elements in the base matrix.

5 Results of simulation and hardware implementation

The block length of parity check matrix has a great influence on the performance of information reconciliation such as frame error rate and throughput and so on. The

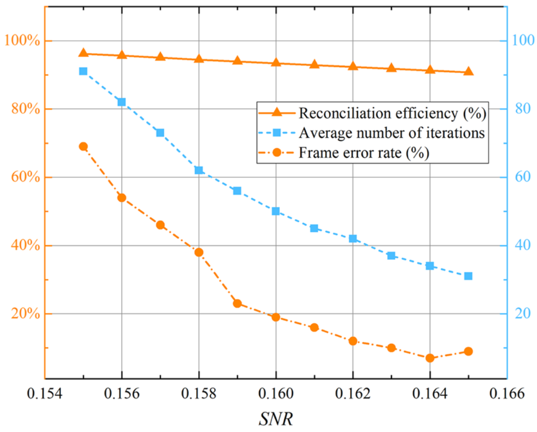

Fig. 3 Decoding performance under different SNRs. Reconciliation efficiency, frame error rate and the average number of iteration all decreases as SNR increases

Table 1 Some parameters in our hardware acceleration scheme

Parameters	Values	Parameters	Values
f	100 MHz	N_{rn}	3.7
R	0.1	N_{mn}	33,375
q	16	β	93.4%
L	160,000	R_{FE}	19%
N_{iter}	50	T	9.6 Mbps

challenge of FPGA-based hardware acceleration is the very limited on-chip storage resources. Shortening the code length is the straightforward way to reduce the amount of required storage resources. To this end, we construct a series of matrices with different code length such as 1×10^5, 1.6×10^5, 5×10^5, and 1×10^6. After simulation tests, the code length of 1.6×10^5 is chosen, and no significant degradation of the decoding performance is observed. By first constructing a base matrix with a code length of 1×10^4 using the random construction algorithm and using the quasi-cyclic expansion with an expansion factor of 16, the check matrix with code length of 1.6×10^5 was constructed. As shown in Fig. 3, the decoding performance of the matrix was tested with the SNR from 0.155 to 0.165. When the SNR is less than 0.165, the reconciliation efficiency can reach above 90%. It is noted that both the frame error rate and the average number of iterations decrease with the increasing SNR.

Compared with the two previous works [26, 27], the code length in our work is effectively shortened. Since the number of nodes in LDPC codes is linearly related to the code length, shortening the code length can play a role in reducing the computational effort. In addition, we use the random construction algorithm to construct the check matrix that has better decoding performance in comparison with Progressive Edge-Growth (PEG) algorithm [42] when the code length is very longer [43].

To test the designed scheme, we use Xilinx VC709 evaluation board for simulation and implementation. The parameters in our scheme are shown in Table 1. The constructed MET-LDPC code has a code rate R of 0.1, which can support CV-QKD systems above 50 km of key distribution. The quasi-cyclic expansion factor q is 16,

Table 2 Hardware resource consumption of the whole sender module ant its four submodules

Module	LUT	BRAM (Kb)	DSP
Raw keys normalization	9676 (2.23%)	0	30 (0.83%)
Data mapping	25,640 (5.92%)	0	256 (7.11%)
LLR initialization	85,960 (19.84%)	0	456 (12.67%)
Iterative decoding	13,817 (3.19%)	15,822 (29.90%)	96 (2.67%)
The whole project	134,852 (31.13%)	15,822 (29.90%)	838 (23.28%)

and this parameter determines the parallelism of iterative decoding, and increasing q can further improve the throughput. The final achieved reconciliation efficiency is 93.4%, and the frame error rate is 19% under SNR of 0.16. The throughput can reach 9.6 Mbps, which is was improved by two orders of magnitude in comparison with 37.4 kbps where the algorithm is executed on an Intel i7-9700K CPU.

The board we chose is populate the Virtex-7XC7VX690T FPGA with 433,200 LUTs, 52,920 Kb BRAMs, and 3600 DSPs. Table 2 shows the quantity and proportion of hardware resources occupied by the whole sender module and its submodules. The first three submodules all call floating-point IP cores to perform calculations, and the instantiated IP cores preferentially use DSPs instead of LUTs, so there is a greater demand for DSPs. In addition, the three submodules do not need to cache data, so no BRAM are consumed. Table 2 also shows that the module with the highest resource consumption is *LLR Initialization*, this is because the table shows the total amount needed for eight *LLR initialization* modules to be processed in parallel, the number of LUTs and DSPs required for a single *LLR initialization* module is 10,745 and 57, respectively. Table 2 also shows that the whole project of the sender module consume less than on-third of the total hardware resources, providing the potential for further performance improvements.

In our scheme, the achieved throughput is sufficient for our CV-QKD prototype with system clock rate of 10 MHz. In principle, we can increase the value of q or instantiate two sender modules to improve the throughput at the cost of hardware resources consumption. For example, when increasing q to 256, the throughput can be improved to be 153.6 Mbps and the hardware resources of our FPGA chip can meet the demand. In contrast, the throughput of GPU-based multidimensional information is 64.11 Mbps with code rate R of 0.1 [27].

6 Conclusion and outlook

We design and implement a high-speed multidimensional reconciliation sender module based on FPGA and achieved a throughput of 9.6 Mbps. Recently, this information reconciliation module has been exploited to realize a real-time data post-processing in our CV-QKD prototype with system clock rate of 10 MHz. Notice that, the hardware resources consumed in this paper are less than 32% of the total resources, leaving adequate space for further performance improvement. In principle, the throughput can be doubled by instantiating two sender modules on the FPGA chip, and the throughput can be further improved by increasing the parallelism of iterative decoding, etc. Currently, integrated photonic technology is progress rapidly [44, 45], and some research teams have tried to implement the sender and receiver of CV-QKD using silicon photonic chips [46, 47]. To realize an overall integration, the information reconciliation module must also be able to be integrated and miniaturized. The advantages of easy integration and low power consumption of FPGAs make it a very competitive candidate. Therefore, our results are useful to the overall integration and miniaturization of CV-QKD systems.

Acknowledgements This work was supported by Innovation Program for Quantum Science and Technology (No. 2021ZD0300703), National Natural Science Foundation of China (NSFC) (No. 62175138, 62205188), Shanxi 1331KSC, Fundamental Research Program in Shanxi Province (No. 202203021222232), Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi Province (No. 2021L258) and Key Research and Development Program of Guangdong Province (No. 2020B0303040002).

Availability of data and materials The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Xu, F., Ma, X., Zhang, Q., Lo, H.-K., Pan, J.-W.: Secure quantum key distribution with realistic devices. Rev. Mod. Phys. **92**(2), 025002 (2020)
2. Portmann, C., Renner, R.: Security in quantum cryptography. Rev. Mod. Phys. **94**(2), 025008 (2022)
3. Li, Y.-M., Wang, X.-Y., Bai, Z.-L., Liu, W.-Y., Yang, S.-S., Peng, K.-C.: Continuous variable quantum key distribution. Chin. Phys. B **26**(4), 040303 (2017)
4. Laudenbach, F., Pacher, C., Fung, C.-H.F., Poppe, A., Peev, M., Schrenk, B., Hentschel, M., Walther, P., Hübel, H.: Continuous-variable quantum key distribution with gaussian modulation—the theory of practical implementations. Adv. Quantum Technol. **1**(1), 1800011 (2018)
5. Pirandola, S., Andersen, U.L., Banchi, L., Berta, M., Bunandar, D., Colbeck, R., Englund, D., Gehring, T., Lupo, C., Ottaviani, C., Pereira, J., Razavi, M., Shaari, J.S., Tomamichel, M., Usenko, V.C., Vallone, G., Villoresi, P., Wallden, P.: Advances in quantum cryptography. Adv. Opt. Photonics **12**(4), 1012–1236 (2020)
6. Zhang, Y., Chen, Z., Pirandola, S., Wang, X., Zhou, C., Chu, B., Zhao, Y., Xu, B., Yu, S., Guo, H.: Long-distance continuous-variable quantum key distribution over 202.81 km fiber. Phys. Rev. Lett. **125**(1), 010502 (2020)
7. Huang, P., Wang, T., Chen, R., Wang, P., Zhou, Y., Zeng, G.: Experimental continuous-variable quantum key distribution using a thermal source. New J. Phys. **23**(11), 113028 (2021)
8. Zhao, W., Shi, R., Ruan, X., Guo, Y., Mao, Y., Feng, Y.: Monte Carlo-based security analysis for multimode continuous-variable quantum key distribution over underwater channel. Quantum Inf. Process. **21**(5), 186 (2022)
9. Aguiar, L.S., Borelli, L.F.M., Roversi, J.A., Vidiella-Barranco, A.: Performance analysis of continuous-variable quantum key distribution using non-gaussian states. Quantum Inf. Process. **21**(8), 304 (2022)
10. Tian, Y., Wang, P., Liu, J., Du, S., Liu, W., Lu, Z., Wang, X., Li, Y.: Experimental demonstration of continuous-variable measurement-device-independent quantum key distribution over optical fiber. Optica **9**(5), 492–500 (2022)
11. Wang, H., Li, Y., Pi, Y., Pan, Y., Shao, Y., Ma, L., Zhang, Y., Yang, J., Zhang, T., Huang, W., Xu, B.: Sub-Gbps key rate four-state continuous-variable quantum key distribution within metropolitan area. Commun. Phys. **5**(1), 162 (2022)

12. Jain, N., Chin, H.M., Mani, H., Lupo, C., Nikolic, D.S., Kordts, A., Pirandola, S., Pedersen, T.B., Kolb, M., Ömer, B., Pacher, C., Gehring, T., Andersen, U.L.: Practical continuous-variable quantum key distribution with composable security. Nat. Commun. **13**(1), 4740 (2022)
13. Li, Z., Wang, X., Chen, Z., Shen, T., Yu, S., Guo, H.: Impact of non-orthogonal measurement in bell detection on continuous-variable measurement-device-independent quantum key distribution. Quantum Inf. Process. **22**(6), 236 (2023)
14. Zhang, M., Huang, P., Wang, P., Wei, S., Zeng, G.: Experimental free-space continuous-variable quantum key distribution with thermal source. Opt. Lett. **48**(5), 1184–1187 (2023)
15. Tian, Y., Zhang, Y., Liu, S., Wang, P., Lu, Z., Wang, X., Li, Y.: High-performance long-distance discrete-modulation continuous-variable quantum key distribution. Opt. Lett. **48**(11), 2953–2956 (2023)
16. Du, S., Wang, P., Liu, J., Tian, Y., Li, Y.: Continuous variable quantum key distribution with a shared partially characterized entangled source. Photonics Res. **11**(3), 463–475 (2023)
17. Chen, Z., Wang, X., Yu, S., Li, Z., Guo, H.: Continuous-mode quantum key distribution with digital signal processing. npj Quantum Inf. **9**(1), 28 (2023)
18. Matsuura, T., Maeda, K., Sasaki, T., Koashi, M.: Finite-size security of continuous-variable quantum key distribution with digital signal processing. Nat. Commun. **12**(1), 252 (2021)
19. Van Assche, G., Cardinal, J., Cerf, N.J.: Reconciliation of a quantum-distributed gaussian key. IEEE Trans. Inf. Theory **50**(2), 394–400 (2004)
20. Leverrier, A., Alléaume, R., Boutros, J., Zémor, G., Grangier, P.: Multidimensional reconciliation for a continuous-variable quantum key distribution. Phys. Rev. A **77**(4), 042325 (2008)
21. Feng, Y., Wang, Y.-J., Qiu, R., Zhang, K., Ge, H., Shan, Z., Jiang, X.-Q.: Virtual channel of multi-dimensional reconciliation in a continuous-variable quantum key distribution. Phys. Rev. A **103**(3), 032603 (2021)
22. Jeong, S., Jung, H., Ha, J.: Rate-compatible multi-edge type low-density parity-check code ensembles for continuous-variable quantum key distribution systems. npj Quantum Inf. **8**(1), 6 (2022)
23. Lodewyck, J., Bloch, M., García-Patrón, R., Fossier, S., Karpov, E., Diamanti, E., Debuisschert, T., Cerf, N.J., Tualle-Brouri, R., McLaughlin, S.W., Grangier, P.: Quantum key distribution over 25 km with an all-fiber continuous-variable system. Phys. Rev. A **76**(4), 042305 (2007)
24. Lin, D., Huang, D., Huang, P., Peng, J.Y., Zeng, G.: High performance reconciliation for continuous-variable quantum key distribution with LDPC code. Int. J. Quantum Inf. **13**(2), 1550010 (2015)
25. Milicevic, M., Feng, C., Zhang, L.M., Gulak, P.G.: Quasi-cyclic multi-edge LDPC codes for long-distance quantum cryptography. npj Quantum Inf. **4**(1), 21 (2018)
26. Wang, X., Zhang, Y., Yu, S., Guo, H.: High speed error correction for continuous-variable quantum key distribution with multi-edge type LDPC code. Sci. Rep. **8**, 10543 (2018)
27. Li, Y., Zhang, X., Li, Y., Xu, B., Ma, L., Yang, J., Huang, W.: High-throughput GPU layered decoder of quasi-cyclic multi-edge type low density parity check codes in continuous-variable quantum key distribution systems. Sci. Rep. **10**, 14561 (2020)
28. Guo, D., He, C., Guo, T., Xue, Z., Feng, Q., Mu, J.: Comprehensive high-speed reconciliation for continuous-variable quantum key distribution. Quantum Inf. Process. **19**(9), 320 (2020)
29. Yang, S.-S., Lu, Z.-G., Li, Y.-M.: High-speed post-processing in continuous-variable quantum key distribution based on FPGA implementation. J. Lightw. Technol. **38**(15), 3935–3941 (2020)
30. Lu, Q., Lu, Z., Yang, H., Yang, S., Li, Y.: FPGA-based implementation of multidimensional reconciliation encoding in quantum key distribution. Entropy **25**(1), 80 (2023)
31. Walenta, N., Burg, A., Caselunghe, D., Constantin, J., Gisin, N., Guinnard, O., Houlmann, R., Junod, P., Korzh, B., Kulesza, N., Legré, M., Lim, C.W., Lunghi, T., Monat, L., Portmann, C., Soucarros, M., Thew, R.T., Trinkler, P., Trolliet, G., Vannel, F., Zbinden, H.: A fast and versatile quantum key distribution system with hardware key distillation and wavelength multiplexing. New J. Phys. **16**(1), 013047 (2014)
32. Paraïso, T.K., Roger, T., Marangon, D.G., De Marco, I., Sanzaro, M., Woodward, R.I., Dynes, J.F., Yuan, Z., Shields, A.J.: A photonic integrated quantum secure communication system. Nat. Photonics **15**(11), 850–856 (2021)
33. Sax, R., Boaron, A., Boso, G., Atzeni, S., Crespi, A., Grünenfelder, F., Rusca, D., Al-Saadi, A., Bronzi, D., Kupijai, S., Rhee, H., Osellame, R., Zbinden, H.: High-speed integrated QKD system. Photonics Res. **11**(6), 1007–1014 (2023)
34. Richardson, T., Urbanke, R.: Multi-edge type LDPC codes. In: Workshop Honoring Prof. Bob McEliece 60th Birthday, California Institute of Technology, Pasadena, CA, USA, pp. 24–25 (2002)

35. Jouguet, P., Kunz-Jacques, S., Leverrier, A.: Long-distance continuous-variable quantum key distribution with a gaussian modulation. Phys. Rev. A **84**(6), 062317 (2011)
36. Wang, X., Zhang, Y., Li, Z., Xu, B., Yu, S., Guo, H.: Efficient rate-adaptive reconciliation for continuous-variable quantum key distribution. Quantum Inf. Comput. **17**(13 & 14), 1123–1134 (2017)
37. Li, Q., Wen, X., Mao, H., Wen, X.: An improved multidimensional reconciliation algorithm for continuous-variable quantum key distribution. Quantum Inf. Process. **18**(1), 25 (2019)
38. Zhang, M., Hai, H., Feng, Y., Jiang, X.-Q.: Rate-adaptive reconciliation with polar coding for continuous-variable quantum key distribution. Quantum Inf. Process. **20**(10), 318 (2021)
39. Mani, H., Gehring, T., Grabenweger, P., Ömer, B., Pacher, C., Andersen, U.L.: Multiedge-type low-density parity-check codes for continuous-variable quantum key distribution. Phys. Rev. A **103**(6), 062419 (2021)
40. MacKay, D.J.C.: Good error-correcting codes based on very sparse matrices. IEEE Trans. Inf. Theory **45**(2), 399–431 (1999)
41. Yang, S.-S., Liu, J.-Q., Lu, Z.-G., Bai, Z.-L., Wang, X.-Y., Li, Y.-M.: An FPGA-based LDPC decoder with ultra-long codes for continuous-variable quantum key distribution. IEEE Access **9**, 47687–47697 (2021)
42. Hu, X.-Y., Eleftheriou, E., Arnold, D.M.: Regular and irregular progressive edge-growth tanner graphs. IEEE Trans. Inf. Theory **51**(1), 386–398 (2005)
43. Bai, Z., Yang, S., Li, Y.: High-efficiency reconciliation for continuous variable quantum key distribution. Jpn. J. Appl. Phys. **56**(4), 044401 (2017)
44. Wang, J., Sciarrino, F., Laing, A., Thompson, M.G.: Integrated photonic quantum technologies. Nat. Photonics **14**(5), 273–284 (2020)
45. Feng, L., Zhang, M., Wang, J., Zhou, X., Qiang, X., Guo, G., Ren, X.: Silicon photonic devices for scalable quantum information applications. Photonics Res. **10**(10), 135–153 (2022)
46. Zhang, G., Haw, J.Y., Cai, H., Xu, F., Assad, S.M., Fitzsimons, J.F., Zhou, X., Zhang, Y., Yu, S., Wu, J., Ser, W., Kwek, L.C., Liu, A.Q.: An integrated silicon photonic chip platform for continuous-variable quantum key distribution. Nat. Photonics **13**(12), 839–842 (2019)
47. Piétri, Y., Vidarte, L.T., Schiavon, M., Grangier, P., Rhouni, A., Diamanti, E.: CV-QKD receiver platform based on a silicon photonic integrated circuit. In: Optical Fiber Communications Conference and Exhibition, San Diego, CA, USA, pp. 1–3 (2023)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.