A Pipeline Optimization Model for QKD Post-processing System

Jianyi Zhou, Bo Liu, Baokang Zhao*, and Bo Liu

School of Computer
National University of Defense Technology
Changsha, Hunan, China
zjy1024kb@gmail.com, {boliu,bkzhao}@nudt.edu.cn,
liuboyayu@163.com

Abstract. Quantum key distribution (QKD) technology can create unconditional security keys between communication parties, but its key generation rate can't satisfy high-speed network applications. As an essential part of QKD system, the design of post-processing system has a huge influence on its key generation rate. For the challenges of real-time and high-speed processing requirements, we propose a pipeline optimization model for QKD post-processing system. With the variable granularity division policies in our model, a high-speed pipeline QKD post-processing system can be designed with the constraints of limited computing and storage resources and security requirements. Simulation results show that for GHz BB84 protocol based QKD system, the security key generation rate of our post-processing system can reach to 600kbps with 25km distance. We believe that our work can provide useful guidance for the design of future QKD system.

Keywords: QKD, Post-processing, pipeline, performance optimization.

1 Introduction

Quantum key distribution [1] technology is currently the most feasible practical application of quantum information. Based on the laws of physics rather than computational complexity of mathematical problems, quantum key distribution can create information-theoretical security (ITS) keys between communication parties. The keys generated by QKD systems can be used for cryptographic applications with one-time-pad [2], AES or other security protection schemes.

QKD system involves two phases, quantum communication phase and classical post-processing phase. Due to its high complexity, the design and implementation of post-processing has a huge influence on the speed of security key generation. For high-speed QKD systems, most works utilize hardware for real-time post-processing [3-8]. However, hardware based methodology suffers from long design cycle, high complexity in realization and troublesome debugging. Therefore some researchers turn to

* Corresponding author.

Linawati et al. (Eds.): ICT-EurAsia 2014, LNCS 8407, pp. 472–481, 2014.

software based post-processing [9], in which the main challenge is how to design the software architecture to speed-up the processing procedures. Currently parallel computing technologies such as pipeline and multithreading have been adopted to improve processing efficiency [10-11].

As a consequence, how to divide the pipeline and design efficient multithread post-processing system has become a hotspot of research. In this paper, we propose a pipeline partition optimization model. To test the performance of our model, a multi-core parallel computing system was built on the basis of our previous works in literature [12-15]. Experimental results show that our pipeline design can reach a speed over 600kbps for BB84 protocol based GHz QKD system, which may prove useful for the future design and optimization of QKD post-processing system.

The rest of the paper is organized as follows. Section 2 provides the background of quantum key distribution and pipeline. Our pipeline optimization model for QKD post-processing system is in section 3. The experimental results and analysis are presented in section 4 while section 5 concludes the paper.

2 Preliminaries

2.1 Quantum Key Distribution

QKD protocols can be divided into preparation-measurement protocol and entanglements based protocol. The most mature in research and feasible protocol is BB84, which consists quantum communication phase and classical post-processing phase.

Security key rate is the most important performance indictor of QKD system. Secure rate R (bits/pulse) which is defined as the possibility that the security key drawing from the photon pulse sent by Alice, as expressed in equation (1).

$$R = I(A:B) - I(B:E) \tag{1}$$

$I(A:B)$ is the mutual information between Alice and Bob while $I(B:E)$ is the mutual information between eavesdropper and Bob. $I(A:B)$ can be further expressed as equation (2), in which q is determined by specific QKD protocol, Q_μ is the photon counting rate, E_μ is quantum bit error rate and $H_2(E_\mu)$ is Shannon entropy.

$$I(A:B) = qQ_\mu[1 - H_2(E_\mu)] \tag{2}$$

$I(B:E)$ can be represented as equation (3), in which Q_0 is count rate of vacuum quantum signals, Q_1 is single photon count rate, Q_{multi} is multi-photon count rate and e_1 is the quantum bit error rate caused by single photons.

$$I(B:E) = qQ_0 \cdot 0 + qQ_1 H_2(e_1) + qQ_{multi} \times 1 \tag{3}$$

Combining (1), (2) and (3), the final security key rate can be represented as equation (4), in which Δ_1 is the ratio of single photon signal in the signals detected by Bob and f is the frequency of QKD system.

$$B_{secure} = qQ_{\mu}\{-H_2(E_{\mu}) + \Delta_1[1 - H_2(e_1)]\} \cdot f \tag{4}$$

2.2 QKD Post-processing Procedures

The post-processing procedures of QKD system in which conducts BB84 protocol mainly include the following steps:

- ***Key Sifting.*** Through sifting, Alice and Bob can drop the signal bits which do not carry quantum state information and obtain sifted key.
- ***Basis Sifting.*** By comparing the basis, the different bits in the basis of Bob and Alice are deleted and the initial key is obtained.
- ***Error Correction.*** Alice calculates and transmits the error correction information to Bob via the public classical channel and Bob corrects the errors in sifted keys.
- ***Key Confirmation.*** Bob and Alice send information to each other to confirm their keys are identical after error correction procedure.
- ***Privacy Amplification.*** Alice randomly chooses a Hash function and sends the generation information of the function to Bob. Bob generates the Hash matrix, and then they can gain the security key after the privacy amplification.

2.3 Concurrency Analysis

The main procedures of post-processing have the characteristics of strict order requirement, rich interaction and huge data amount, but they are uncorrelated for difference data unit, which is suitable for pipeline structure. Performance, resource and security requirement have conflicted influence on the design of QKD post-processing system. For example, performance can be improved by increasing the number of stages in pipeline, but this will result in increased consumption of storage. Moreover, in order to guarantee the security of QKD system, the security key length must be long enough to eliminate the finite size effect [16]. Thus resource requirement will grow, which may consequently limit the number of stages when available resource is constrained. Therefore we need to maximize system performance under the constraints of security demand and resource, which is challenging yet meaningful problem.

3 A Pipeline Optimization Model for QKD Post-processing System

In this section we firstly provide the hypotheses of our model. Then the variable granularity division policies of pipeline are conducted. Finally, an optimization model for QKD post-processing system is formulated to solve the pipeline design problem.

3.1 Hypotheses

We assume that the tasks can be divided evenly which means the sub-tasks after partition has uniform time consumption. In each macro stage, the change of data length is relatively small. Thus, we assume that the data length remains the same in one macro stage. Moreover, only CPU core number and memory needs to be considered as the constraint of system resource.

3.2 VGDP Method for Stage Division

The performance of pipeline can be optimized if all operations in all stages are identical in processing time. Otherwise, the slowest stage will become the bottleneck. We proposed a method called variable granularity division of pipeline (VGDP). VGDP's idea is to conduct coarse division at first so that calculation and communication can be conducted in a parallel way and then conduct fine division for load balancing.

A Coarse Granularity Division

For the scenario depicted in Fig. 1, the entire process can be divided into five steps which consist of three calculation and two communication procedures.

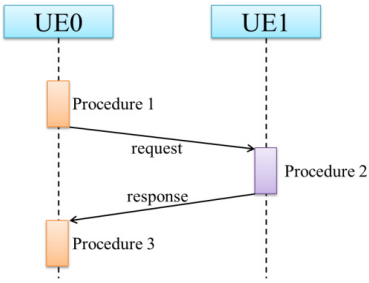

Fig. 1. Cooperative Computation

Our principle of division is to overlay calculation and communication procedures as much as possible with the purpose of utilizing delay hiding to the fullest. We denote the time consumption of the each macro stage on Alice and Bob as Ta_i and Tb_i correspondingly. Two situations may occur for Alice and Bob, one is that both of them are working, and the other is that one of them is working and one is idle. We define step function as:

$$\varphi(x)=\begin{cases}1 & x\neq 0\\ 0 & x=0\end{cases}, \tag{5}$$

then the number of working macro stages on Alice and Bob are $\sum_{i=1}^{j}\varphi(Ta_i)$ and $\sum_{i=1}^{j}\varphi(Tb_i)$ respectively, and in which j is the number of macro stages.

The amount of error corrected key has to be accumulated for a few rounds before privacy amplification and we denote the number of rounds required as n. Key sifting is the first stage of post-processing and we denote the length of the data processed as l_{Raw}. We define ra_i and rb_i as the ratio of the length of data processed in the each macro stage compared with l_{Raw}.The initial macro stages need to be merged to make the number of macro stages not exceed CPU core number. The two stages which produce least time costing resulting stage should be chosen.

B Fine Granularity Division

After the coarse division, the time consumption of each stage still varies greatly. To further improve the performance of pipeline, we need to divide the macro stages into smaller sub-stages. The fine granularity division is only conduct for calculation procedures. T he data interaction among subtasks will introduce additional delay. Let εa_i and εb_i be the data interaction delay, then we have:

$$\begin{cases} \varepsilon a_i = ra_i \cdot l_{Raw} \cdot n \cdot v \\ \varepsilon b_i = rb_i \cdot l_{Raw} \cdot n \cdot v \end{cases}, \tag{6}$$

in which v is the unit data transmit time between successive stages.

Assume that the number of sub-stages in each macro stage of Alice and Bob be Xa_i and Xb_i respectively. The delay of each macro stage can be expressed as equation (7).

$$t_i = \begin{cases} Ta_i / Xa_i + \varepsilon a_i & (Ta_i \neq 0 \ and \ Tb_i = 0) \\ Tb_i / Xb_i + \varepsilon b_i & (Ta_i = 0 \ and \ Tb_i \neq 0) \\ Max\left\{ Ta_i / Xa_i + \varepsilon a_i \ , \ Tb_i / Xb_i + \varepsilon b_i \right\} & (Ta_i \neq 0 \ and \ Tb_i \neq 0) \end{cases} \tag{7}$$

3.3 Model Formulation

There are some constraints in pipeline design. Firstly the number of stages should not exceed the number of computing cores (e.g. CPU). If the number of cores on Alice and Bob is Ca and Cb correspondingly, then the cores constraint can be expressed as (8).

$$\begin{cases} \sum_{i=1}^{j} Xa_i \le Ca \\ \sum_{i=1}^{j} Xb_i \le Cb \end{cases} \tag{8}$$

The next is the usage of memory. The memory needed for data storage is determined by the data length of each stage and the length of shared queen length m while it is determined by the specific algorithms adopted and the final key length l_{key} for computation. If we denote the available storage of Alice and Bob as Ma and Mb correspondingly, and λ as the amount of storage needed to compute key of unit length, then the memory constraints can be expressed as (9).

$$\begin{cases} m(\sum_{i=1}^{j-1} Xa_i \cdot ra_i \cdot l_{Raw} + Xa_j \cdot l_{key}) + \lambda \cdot l_{key} \le Ma \\ m(\sum_{i=1}^{j-1} Xb_i \cdot rb_i \cdot l_{Raw} + Xb_j \cdot l_{key}) + \lambda \cdot l_{key} \le Mb \end{cases} \tag{9}$$

As for security, Procedures such as key confirmation and authentication consume security keys, expressed as R_{cost}. System performance can be improved by increasing the number of stages in pipeline. However, this will cause degradation in the security level. Thus, a threshold P is needed to fulfill the minimal security requirement.

$$\frac{l_{key} - R_{cost}}{l_{key}} \ge P \tag{10}$$

The system performance is usually measured by throughput, which can be defined as the amount of data processed in unit time. The throughput of the system is (11).

$$Q_{parallel} = \frac{N}{\sum_{i=1}^{j} Max\{Ta_i + Xa_i \cdot \varepsilon a_i,\ Tb_i + Xb_i \cdot \varepsilon b_i\} + (N\text{-}1) Max\{t_1, t_2 \cdots t_j\}} \tag{11}$$

When the data length N is large enough, the time of filling the pipeline can be neglected. The final throughout is

$$Q_{parallel} = \frac{1}{Max\{t_1, t_2, \cdots, t_j\}}. \tag{12}$$

With the goal of optimizing system performance and constraints, the pipeline design problem can be expressed as (13) to (15).

$$Max \quad \frac{1}{Max\{t_1, t_2, \cdots, t_j\}} \tag{13}$$

$$t_i = \begin{cases} Ta_i / Xa_i + \varepsilon a_i & (Ta_i \ne 0 \ and \ Tb_i = 0) \\ Tb_i / Xb_i + \varepsilon b_i & (Ta_i = 0 \ and \ Tb_i \ne 0) \\ Max\left\{Ta_i / Xa_i + \varepsilon a_i, Tb_i / Xb_i + \varepsilon b_i\right\} & (Ta_i \ne 0 \ and \ Tb_i \ne 0) \end{cases} \tag{14}$$

$$s.t. \begin{cases} \sum_{i=1}^{j} Xa_i \le Ca \\ \sum_{i=1}^{j} Xb_i \le Cb \\ m(\sum_{i=1}^{j-1} Xa_i \cdot ra_i \cdot l_{Raw} + Xa_j \cdot l_{key}) + \lambda \cdot l_{key} \le Ma \\ m(\sum_{i=1}^{j-1} Xb_i \cdot rb_i \cdot l_{Raw} + Xb_j \cdot l_{key}) + \lambda \cdot l_{key} \le Mb \\ l_{key} = ra_j \cdot l_{Raw} \cdot n \\ \dfrac{l_{key} - R_{cost}}{l_{key}} \ge P \\ \forall i \in \{1, \cdots, j\}, Xa_i \in N, Xb_i \in N \end{cases} \tag{15}$$

3.4 Model Solution Procedure

Algorithm 1. Feasible Direction Algorithm

Input: j, $Ta[1..j]$, $Tb[1..j]$, $ra[1..j]$, $rb[1..j]$, m, l_{Raw}, v, l_{key}, κ, Ca, Cb, Ma, Mb

Output: $t[1..j]$, $Xa[1..j]$, $Xb[1..j]$

1. $Xa[1..j]$, $Xb[1..j]$ Initialization
2. **while** 1
3. **do for** $i \leftarrow 1$ to j
4. **do** calculate $t[i]$
5. **if** $t[i] = \max\ t[1..j]$
6. **then** if $t[i] = Ta[i] / Xa[i] + ra[i] \cdot l_{Raw} \cdot n \cdot v$
7. **then** $Xa[i] \leftarrow Xa[i] + 1$
8. **if** $\sum_{i=1}^{j} Xa[i] > Ca$ or $m(\sum_{i=1}^{j-1} Xa[i] \cdot ra[i] \cdot l_{Raw} + Xa[j] \cdot l_{key}) + \lambda \cdot l_{key} \le Ma$
9. **then** $Xa[i] \leftarrow Xa[i] - 1$
10. **break**
11. **else** $Xb[i] \leftarrow Xb[i] + 1$
12. **if** $\sum_{i=1}^{j} Xb[i] > Cb$ or $m(\sum_{i=1}^{j-1} Xb[i] \cdot rb[i] \cdot l_{Raw} + Xb[j] \cdot l_{key}) + \lambda \cdot l_{key} \le Mb$
13. **then** $Xb[i] \leftarrow Xb[i] - 1$
14. **Break**

Our model is a nonlinear programming problem with variable dimensions. If we use exhaustive search to solve our model with dimension j, the time complexity will be $O(n^j)$. We can adopt feasible direction algorithm to solve the problem, whose idea is choosing a direction that best optimizes the objective function locally. A near optimal solution can be obtained after sufficient iterations. As for our problem, we only need to increase the number of subtask of the most time consuming macro stage by one. Time complexity can be reduced to $O(n)$ if Algorithm 1 is utilized.

Algorithm 2. Solution to Variable Dimension Nonlinear Programming Model

Input: J, $Ta[1..J]$, $Tb[1..J]$, $ra[1..J]$, $rb[1..J]$, m, l_{Raw}, v, l_{key}, κ, Ca, Cb, Ma, Mb

Output: Tmin, XA, XB

1. Tmin $\leftarrow \infty$
2. **while** $j > 1$
3. **do if** $\sum_{i=1}^{j} \varphi(Ta[i]) \le Ca$ *and* $\sum_{i=1}^{j} \varphi(Tb[i]) \le Cb$
4. **then** use Algorithm 1 to solve problem with dimension j
5. **if** max $t[1..j]$ < Tmin
6. **then** Tmin $\leftarrow$ max $t[1..j]$
7. Jmin $\leftarrow$ j
8. XA $\leftarrow$ $Xa[1..j]$
9. XB $\leftarrow$ $Xb[1..j]$
10. Merge the macro stages
11. $j \leftarrow j - 1$

For each value of j, a local optimal solution can be obtained using algorithm 1. Then a global optimal can be chosen from the local optimal. If the system is divided into J macro stages, the overall solution of the model can be expressed as Algorithm 2.

4 Evaluation

4.1 Experiment Setting

We designed and constructed our multi-thread post-processing software. We also designed a quantum communication simulator based on our previous work in [12-15].

Our QKD system conducts BB84 protocol. The hardware platform of our system is Intel(R) Core(TM) i7@ 3.40 GHz, 1G. The system can be divided into 9 macro stages and the parameters such as ra_i, rb_i, Ta_i and Tb_i in our platform was measured. The default values of other experiment parameters are listed in Table 1.

Table 1. Default Parameter Values for QKD Post-processing System

Ca	4	Cb	4	m	20	l_{Raw}	1Mbits
Ma	1G	Mb	1G	v	3123MB/s	l_{key}	85kbits
Q_μ	1%	E_μ	2%	λ	576		

4.2 Quantum Communication Distance

If the quantum communication distance becomes very long, the count rate will decreased rapidly due to quantum channel loss. In this section, we focus on the influence of distance on the performance of post-processing system. We set the Q_μ as 0.5%, 1%, 2% and 5% respectively and the security key rates for both parallel and pipeline post-processing system are depicted in Fig. 3.(a).

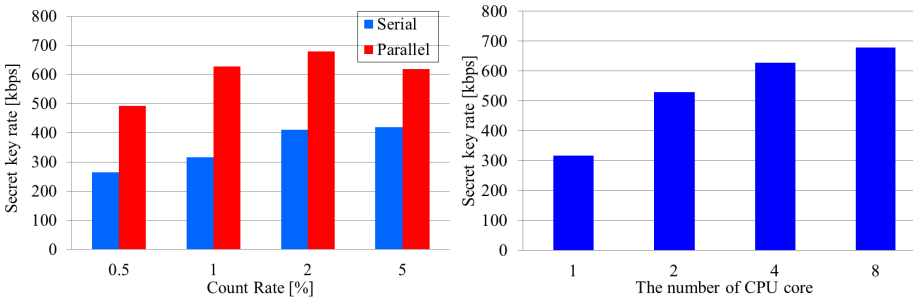

Fig. 2. (a) Security Key Rate under Different Count Rate; (b) Security Key Rate with Different CPU Core Number

Fig. 3.(a) shows that regardless of the quantum communication distance, our pipeline post-processing system can increase security key rate significantly compared with serial system. For example, our pipeline system can reach a security key rate above 600kbps which is 98% higher than serial system, proving the effectiveness of the model. Moreover, the raise in security key rate decreases with count rate, which can be explained by the fact that privacy amplification is the most time-consuming task. Thus, the increase in count rate merely shortens the time of the procedures before privacy amplification while privacy amplification becomes the bottleneck of the system. In this condition, the macro stage division may need to be redone according to pipeline design model and privacy amplification should be divided into different macro stages.

4.3 Post-processing System Resource

CPU core number is an important factor on pipeline performance. When tasks in different stages are allocated on different CPU core, the advantage of pipeline can be maximized. In this section we investigate the influence of CPU core number on system performance. The security key rates for post-processing systems with 1, 2, 4 and 8 cores are obtained and plotted in Fig. 3.(b).

As Fig. 3.(b) indicates security key rate increases with CPU core number. For example, the key rate for two-core platform reaches 500kbps which is 68% more than single core platform. This is because the task can be pipelined to more CPUs. But the increase in key rate gradually slows down with more CPU cores, which could be explained by the growth of synchronization cost with more stages. We can improve the system performance by stage division and using more cores. However, this methodology may reach its limit because of the increase in synchronization cost.

5 Conclusion

In this paper, we proposed an optimization model to maximize the thought of the pipeline under resource and security constraints. And flexible granularity division policies are conducted. Our model proves the ability to improve the performance of QKD system significantly compared to serial system. We also analyzed the influence of parameters such as CPU core number on system performance, which may offer useful guidance for the design of future QKD system.

Acknowledgment. The work described in this paper is partially supported by the grants of the National Basic Research Program of China (973 project) under Grant No.2009CB320503, 2012CB315906; the project of National Science Foundation of China under grant No. 61070199, 61103189, 61103194, 61103182, 61202488, 61272482.

References

1. Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. In: Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, pp. 175–179. IEEE Press, Bangalore (1984)
2. Vernam, G.S.: Secret signaling system. U.S. Patent 1310719 (1919)
3. Xu-Yang, W., Zeng-Liang, B.A.I., Shao-Feng, W., et al.: Four-State Modulation Continuous Variable Quantum Key Distribution over a 30-km Fiber and Analysis of Excess Noise. Chinese Physics Letters 30(1) (2013)
4. Zhang, H.F., Wang, J., Cui, K., Luo, C.L., Lin, S.Z., Zhou, L., ... Pan, J.W.: A real-time QKD system based on FPGA. Journal of Lightwave Technology 30(20), 3226–3234 (2012)
5. Tanaka, A., Fujiwara, M., Yoshino, K.I., Takahashi, S., Nambu, Y., Tomita, A., ... Tajima, A.: High-speed quantum key distribution system for 1-Mbps real-time key generation. IEEE Journal of Quantum Electronics 48(4), 542–550 (2012)
6. Wang, J., Luo, C.L., Lin, S.Z., et al.: Research of hash-based secure key expansion algorithm for practical QKD. Optik-International Journal for Light and Electron Optics (2012)
7. Cui, K., Wang, J., Zhang, H.F., et al.: A real-time design based on FPGA for Expeditious Error Reconciliation in QKD system. IEEE Information Forensics and Security (2011)
8. Walenta, N., Burg, A., Caselunghe, D., et al.: A fast and versatile QKD system with hardware key distillation and wavelength multiplexing. arXiv preprint arXiv: 1309. 2583 (2013)
9. Wijesekera, S., Palit, S., Balachandran, B.: Software Development for B92 Quantum Key Distribution Communication Protocol. In: ICIS 2007: 6th IEEE/ACIS International Conference on Computer and Information Science, pp. 274–278. IEEE Press, New York (2007)
10. Lin, X., Peng, X., Yan, H., Jiang, W., Liu, T., et al.: An Implementation of Post-Processing Software in Quantum Key Distribution. In: 2009 WRI World Congress on Computer Science and Information Engineering, pp. 243–247. IEEE Press, New York (2009)
11. Li, Q., Le, D., Rao, M.: A Design and Implementation of Multi-thread Quantum Key Distribution Post-processing Software. In: Second International Conference on Instrumentation, Measurement, Computer, Communication and Control, pp. 272–275. IEEE Press, New York (2012)
12. Liu, B., Zhao, B., Wei, Z., Wu, C., Su, J., Yu, W., ... Sun, S.: Qphone: A quantum security VoIP phone. In: Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM, pp. 477–478. ACM Press, New York (2013)
13. Liu, B., Liu, B., Zhao, B., Zou, D., Wu, C., Yu, W., You, I.: A real-time privacy amplification scheme in quantum key distribution. In: Mustofa, K., Neuhold, E.J., Tjoa, A.M., Weippl, E., You, I. (eds.) ICT-EurAsia 2013. LNCS, vol. 7804, pp. 453–458. Springer, Heidelberg (2013)
14. Zou, D., Zhao, B., Wu, C., Liu, B., Yu, W., et al.: CLIP: A Distributed Emulation Platform for Research on Information Reconciliation. In: 2012 15th International Conference on Network-Based Information Systems (NBiS), pp. 721–726. IEEE Press, New York (2012)
15. Sun, S.H., Ma, H.Q., Han, J.J., et al.: Quantum key distribution based on phase encoding in long-distance communication fiber. Optics Letters 35(8), 1203–1205 (2010)
16. Experimental Decoy State Quantum Key Distribution with Unconditional Security Incorporating Finite Statistics, `http://arxiv.org/pdf/0705.3081.pdf`