Optimization of Hardware Architecture for Quantum Key Distribution

RajaAdhithan Radhakrishnan
r.rajaadhithan@gmail.com
Society For Electronic Transactions and Security(SETS)

***Abstract*—The main objective of this paper is to accelerate the post-processing of Quantum Key Distribution (QKD) using an energy-efficient pipelined architecture implemented on a Field-Programmable Gate Array (FPGA). The proposed architecture aims to improve processing speed while efficiently utilizing hardware resources. In addition, this work compares the proposed approach with existing approaches to demonstrate its performance and resource efficiency.**

***Index Terms*—FPGA - Post processing of QKD - Pipelined Architecture - Energy efficiency.**

I. Introduction

Traditional cryptography can be broken because it depends on hard math problems and secure sharing of secret keys, which may still be intercepted or solved using powerful computers, whereas Quantum Key Distribution (QKD) is safer because it can detect any attempt of eavesdropping, making attacks immediately noticeable. In QKD, key distillation is a critical post-processing step that improves the reliability and security of the quantum key shared over noisy quantum channels by correcting errors caused by channel and device imperfections and reducing any possible information leakage, thereby producing a final secure key suitable for cryptographic applications. In this Paper our main target is how to fast up this post processing of QKD with efficient energy resource.

The rest of this paper is organized as follows. Section II reviews existing FPGA-based architectures for QKD post-processing. Section III presents the motivation for the proposed hardware architecture. Section IV describes the optimized hardware architecture for the post-processing of Quantum Key Distribution (QKD). Finally, Section V concludes the paper.

II. Existing approches and challanges

In this section, the process of Quantum Key Distribution (QKD), and its working principle from an implementation perspective, and the current challenges are described [1], [2], [3], [4].

In Quantum Key Distribution, there are two parties: a sender (Alice) and a receiver (Bob) will authenticate each other. First, random bits are generated using a True Random Number Generator (TRNG). These random bits are encoded into qubits using a suitable QKD protocol, such as BB84 or COW .., and transmitted to the receiver through an optical fiber channel. At the receiver, the qubits are detected using a single-photon detector (SPD), and the corresponding binary bits are extracted using a Time-to-Digital Converter (TDC) and stored in DDR memory. Similarly, at the sender's side, the transmitted binary bits corresponding to the selected protocol are also stored in DDR memory. Based on the exchanged timestamps, the corresponding bits are synchronized, and an equal number of raw key bits are obtained by both the sender and the receiver.

The synchronized data are then fetched from DDR memory for parameter estimation. During this process, a small randomly selected subset of key bits is exchanged, compared, and discarded after estimating the quantum bit error rate (QBER). Based on the estimated QBER, appropriate error correction parameters are selected to perform efficient information reconciliation.In other hand this QBER is used scale QKD Protocal. After successful error correction and verification, the reconciled key data are stored in DDR memory, enabling the processing of a large number of raw key bits while supporting adaptive compression during the privacy amplification stage to attain secure final key as shown in Fig.1.

A. Challenges

During the above process, DDR memory is accessed at every stage because all modules are integrated on an embedded platform and executed using bare-metal software. Each post-processing stage is executed sequentially, resulting in multiple processing iterations and increased overall execution time.

Communication between hardware IPs is performed through the AXI interface. Therefore, reliable data authentication and synchronization are required for every data transfer during board-to-board communication. In addition, the data discarding operation performed during the error verification stage introduces significant computational complexity and increases the overall processing time.

Among all the post-processing modules, the Privacy Amplification (PA) module is the most time-consuming due to its intensive computational requirements. Furthermore, efficient data synchronization and communication between processing stages remain major challenges that must be addressed to achieve high-throughput and low-latency QKD post-processing.

III. Motivation

To address the above challenges, we first focus on optimizing the Authentication, Error Verification, and Privacy Amplification (PA) modules.First to identify a suitable PA technique for QKD systems, we reviewed several existing approaches reported in the literature. Among them, Universal Toeplitz Hashing is one of the most widely adopted methods due to its strong security properties. However, achieving both high throughput and efficient hardware utilization while maintaining the required level of security remains a significant challenge on embedded platforms [6].

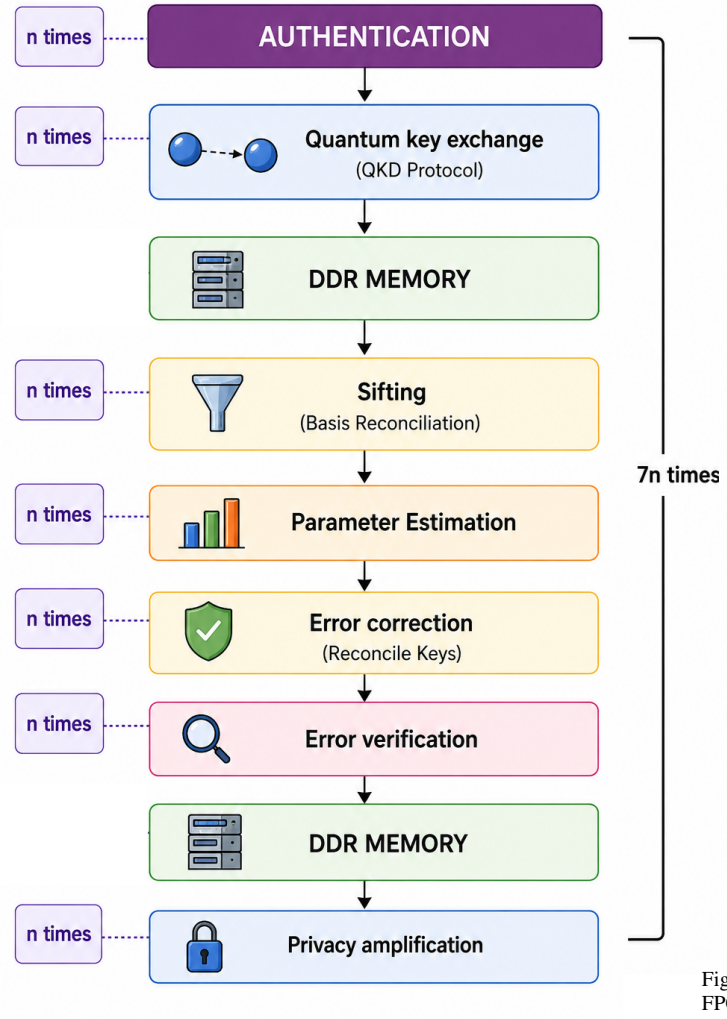

Fig. 1. Existing approach of Post Processing of Quantum key exchange in FPGA.

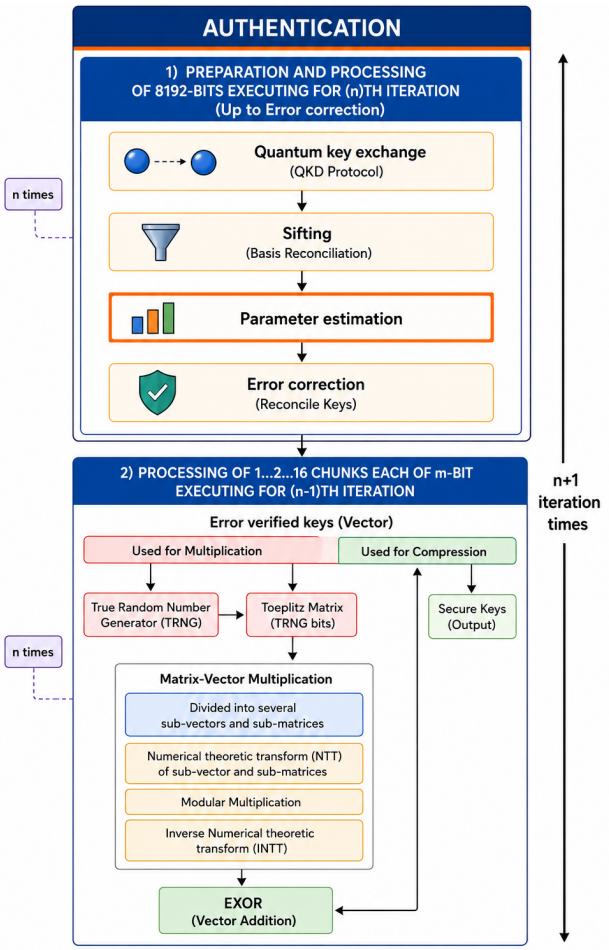

Fig. 2. Proposed approach of Post Processing of Quantum key exchange in FPGA.

To overcome these limitations, we adopt a hardware-oriented implementation using Verilog and implement the parallel and pipelined Toeplitz hashing architecture proposed in [5]. This approach enables efficient hardware resource utilization while significantly improving processing throughput.

The following section presents the proposed hardware implementation of the Authentication, Error Verification, and Toeplitz matrix-based Privacy Amplification modules based on the architecture described in [5].

IV. Optimized Hardware Architecture for Post Processing of Quantumn Key Exchange

To improve the performance and efficiency of FPGA-based Quantum Key Distribution (QKD) post-processing, this section presents the proposed optimizations for the Authentication, Error Verification, and Privacy Amplification (PA) modules.

For the Privacy Amplification module, both communicating parties require an identical random seed. To achieve this, a load-balancing mechanism is proposed in which each party generates half of the random seed locally and securely exchanges the remaining half with the other party. This approach reduces the randomness generation overhead while ensuring that both parties derive the same seed. Furthermore, message integrity is incorporated to authenticate all transmitted data, providing secure communication during both the authentication process and the random seed exchange required for Privacy Amplification.

Instead of storing intermediate data in DDR memory after every processing stage, the proposed architecture buffers the incoming data until a sufficient number of bits are available to initiate processing. The data then flows sequentially through Sifting, Parameter Estimation (8,896 bits), Error Correction (8,192 bits), and Error Verification (8,192 bits). During Error Verification, the data is divided into 16 chunks, enabling flexible error discarding, minimizing data loss, and improving processing efficiency.

The proposed Privacy Amplification architecture employs a dynamically configurable Toeplitz matrix implementation based on the Number Theoretic Transform (NTT) and Inverse Number Theoretic Transform (INTT) for efficient sub-matrix multiplication. Compared with conventional FFT/IFFT-based approaches, the NTT-based implementation reduces hardware complexity while improving computational efficiency. In addition, the architecture integrates the TRNG-based random seed

generation mechanism described above, optimized parallel NTT processing, and efficient BRAM reuse. These optimizations collectively provide high throughput, low latency, efficient hardware utilization, and flexible compression control, making the proposed architecture well suited for high-speed FPGA-based QKD post-processing as shown in Fig.2.

The first step in our approach is to fix the input block size and output block size of the privacy amplification module. To determine these sizes, we define the following parameters: Weak key (W): of length n bits Random seed (RS): of length n bits Final key length (y): calculated as y = n × Compression Ratio The second step is to construct the Toeplitz matrix struc- ture, which is divided into sub-matrices of size m × m for faster multiplication with the weak key, as shown in [7]. The sub-matrix size m also determines the NTT/INTT module size (a 2m-point transform). Using these parameters, we calculate the matrix dimen- sions as follows: Number of sub-matrix rows: R = y / m Number of sub-matrix columns: C = (n × y) / m In this configuration, each row and column of the matrix corresponds to a sub-matrix rather than a single element. The number of required NTT and Inverse NTT (INTT) operations depends on the values of R and C. Our design automatically configures R and C based on the selected compression ratio. This enables real-time adaptation of BRAM usage, NTT/INTT operations, and the overall key compression process. For efficient processing, the weak key ((W)) and random seed ((RS)) are divided into smaller sub-blocks based on the parameters (m), (R), and (C). Furthermore, the proposed design supports variable chunk processing, ranging from 0 to 16 chunks, enabling flexible and scalable hardware execution depending on the available data size and system requirements.This enables the privacy amplification (PA) architecture can operate in parallel with other post-processing steps, including sifting, parameter estimation, error correction, and verification as shown in Fig.2.

A. Existing vs Proposed

When comparing Fig. 1 and Fig. 2, which illustrate the existing and proposed approaches respectively, it is observed that the existing architecture relies heavily on DDR memory access, introducing significant overhead. In addition, the overall processing requires approximately (7n) time units to complete the full post-processing pipeline.

In contrast, the proposed approach minimizes DDR memory dependency and introduces optimized and pipelined designs for the Authentication, Error Verification, and Privacy Amplification modules. From a timing perspective, the proposed architecture reduces the overall processing requirement to approximately (n + 1) iterations to complete the entire pipeline.

Thus, the proposed design significantly improves system efficiency by reducing memory overhead and enabling parallel and pipelined execution. However, further performance improvements can still be achieved by optimizing each individual sub-module in the post-processing stack.

B. Special Features in the Proposed Design

1) Board-to-Board Communication:

- **Load Balancing:** The same random seed is required on both sides of the privacy amplification process. To achieve this, a mutual sharing mechanism is employed: half of the seed is generated on one side and the remaining half on the other, after which both halves are exchanged.
- **Message Integrity:** To ensure secure and authenticated communication between the transmitter and receiver, the architecture incorporates a message integrity verification mechanism using `Poly1305`.

2) Flexible Block Size and Parallelism:

- **Dynamic Error Discarding Mechanism:** The architecture is designed to flexibly process block sizes ranging from minimum to maximum values.
- **Parallelism:** The privacy amplification (PA) architecture can operate in parallel with other post-processing steps, including sifting, parameter estimation, error correction, and verification.

V. Conclusion and Future Work

This work presents an efficient FPGA-based pipelined architecture for QKD post-processing that reduces memory overhead, improves throughput, and lowers processing latency by optimising authentication,error verifiaction and privacy amplification module. The proposed architecture offers a scalable and practical solution for high-speed, real-time QKD systems. Future work will focus on optimizing individual post-processing modules to further improve resource efficiency, reduce power consumption, and enhance secure key generation performance.

ACKNOWLEDGMENT

The author would like to acknowledge and thank Foram P. Shingala, and Karunya Roseline for their invaluable assistance and insightful discussions. The authors also gratefully acknowledge the QSRG (Quantum Security Research Group),for their support and assistance.

References

[1] S. S. Yang, X. Wang, Y. Zhang, and H. Guo, "FPGA-based implementation of size-adaptive privacy amplification in quantum key distribution," *IEEE Photonics Journal*, vol. 9, pp. 1–8, 2017. doi: 10.1109/JPHOT.2017.

[2] Q. Li, Y. Luo, H.-K. Mao, and X. Cheng, "High-speed and adaptive FPGA-based privacy amplification in quantum key distribution," *IEEE Access*, vol. 7, pp. 21482–21490, 2019. doi: 10.1109/ACCESS.2019.2896259.

[3] X. Wang, Y. Zhang, S. Yu, and H. Guo, "High-speed implementation of length-compatible privacy amplification in continuous variable quantum key distribution," *IEEE Photonics Journal*, vol. 10, pp. 1–9, 2018. doi: 10.1109/JPHOT.2018.2824316.

[4] N. Venkatachalam et al., "Scalable QKD Postprocessing System With Reconfigurable Hardware Accelerator," in IEEE Transactions on Quantum Engineering, vol. 4, pp. 1-14, 2023.

[5] B. Y. Tang, B. Liu, Y. P. Zhai, et al., "High-speed and large-scale privacy amplification scheme for quantum key distribution," *Scientific Reports*, vol. 9, no. 15733, 2019. doi: 10.1038/s41598-019-50290-1.

[6] A. Tanaka, T. Matsumoto, K. Aikawa, et al., "High-speed quantum key distribution system for 1-Mbps real-time key generation," *IEEE Journal of Quantum Electronics*, vol. 48, no. 4, pp. 542–550, Apr. 2012. doi: 10.1109/JQE.2012.2187327.

[7] N. Nalla Anandakumar, S. K. Sanadhya, and M. S. Hashmi, "FPGA-based true random number generation using programmable delays in oscillator-rings," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 3, pp. 570–574, Mar. 2020. doi: 10.1109/TCSII.2019.2919891.