FPGA-Based Implementation of an Underwater Quantum Key Distribution System With BB84 Protocol

Burak Kebapci, Vecdi Emre Levent, Sude Ergin, Gorkem Mutlu, Ibrahim Baglica, Anilcan Tosun, Pietro Paglierani, Konstantinos Pelekanakis, Roberto Petroccia, João Alves, and Murat Uysal

***Abstract*—As threats in the maritime domain diversify, securing data transmission becomes critical for underwater wireless networks designed for the surveillance of critical infrastructure and maritime border protection. This has sparked interest in underwater Quantum Key Distribution (QKD). In this paper, we present an FPGA-based real-time implementation of an underwater QKD system based on the BB84 protocol. The QKD unit is built on a hybrid computation system consisting of an FPGA and an on-board computer (OBC) interfaced with optical front-ends. A real-time photon counting module is implemented on FPGA. The transmitter and receiver units are powered with external UPS and all system parameters can be monitored from the connected computers. The system is equipped with a visible laser and an alignment indicator to validate successful manual alignment. Secure key distribution at a rate of 100 qubits per second was successfully tested over a link distance of 7 meters.**

***Index Terms*—Quantum key distribution, underwater communication, BB84 protocol.**

I. Introduction

DESPITE the increasing deployment of underwater sensor networks (USNs) and a growing relevant literature, cyber security aspects have received relatively low attention. Particularly for maritime applications such as the surveillance of critical infrastructure (i.e., harbors, ports, offshore oil platforms, underwater pipelines etc) and border protection, secure communication is the key to ensure the confidentiality, integrity and authentication of the transmitted information. Some countermeasures for cyber attacks have been investigated for USNs [1]. However, all potential solutions offer only computational security based on some mathematical complexity of the encryption. In the quest for quantum advantage, the realization of sufficiently powerful quantum computers is predicted to be possible in the foreseeable future. This would make today's cryptosystems practically useless. USNs are no exception and will be left vulnerable to all types of cyber-attacks bringing a huge threat on maritime security.

The new era of quantum computing brings the necessity of "quantum-secure" cryptography schemes. Based on the firm laws of physics rather than unproven foundations of mathematical complexity, quantum cryptography promises unconditional security for various marine operations [2]. The Proof-of-Concept (PoC) underwater QKD (Quantum Key Distribution) system presented on this article designed to work on relatively short distances under the consideration of several use cases. For example, one specific use case is the pre-mission key exchange. During the initiation phase of a marine mission, various vessels, submarines, and Autonomous Underwater Vehicles (AUVs) can update or refresh their keys. For this purpose, they can maintain a sufficiently close distance to the command node for successful QKD operation. Another use case is the updating of secure keys of underwater sensor nodes. These underwater sensors transmit information on a regular basis through acoustic or optical channels, and the keys used in these systems can be updated via the aid of AUVs.

In the last decade or so, significant advances have been made in the area of QKD and successful experimental demonstrations over fiber optic, atmospheric or satellite links have been performed for various transmission ranges and data rates [3], [4]. The current results are however not directly applicable to underwater environments with unique challenges. Underwater optical transmission suffers from severe attenuation as a result of absorption and scattering due to water molecules and other particles in solution and suspension in water [5]. Unlike free space and fiber optic links [6] which typically operate at infrared wavelengths, visible wavelengths are typically preferred to minimize the underwater attenuation [7]. In particular, the blue-green wavelengths outperform the red-yellow-green wavelengths at open ocean [7], [8] while red-yellow-green wavelengths outperform blue-green wavelengths at coastal turbid waters. Especially coastal and turbid waters have more gelbstoff concentration and it mainly absorbs blue-green wavelengths, while being transparent to red wavelengths. On the other hand open ocean absorption is more like a pure water absorption.

Manuscript received 5 March 2023; revised 14 April 2023; accepted 13 June 2023. Date of publication 19 June 2023; date of current version 2 August 2023. This paper was presented in part at the 10th Underwater Communications and Networking (UComms'22) [DOI: 10.1109/UComms56954.2022.9905688]. *(Corresponding author: Burak Kebapci.)*

Burak Kebapci, Vecdi Emre Levent, Sude Ergin, and Anilcan Tosun are with the Hyperion Technologies, 34794 Istanbul, Turkey (e-mail: burak.kebapci@hyperiontechs.com).

Gorkem Mutlu, Ibrahim Baglica, and Murat Uysal are with the Department of Electrical and Electronics Engineering, Ozyegin University, 34794 Istanbul, Turkey.

Pietro Paglierani, Konstantinos Pelekanakis, Roberto Petroccia, and João Alves are with the NATO STO-CMRE, 19126 La Spezia, Italy.

Digital Object Identifier 10.1109/JPHOT.2023.3287493

The initial works on underwater QKD are theoretical in nature [5], [9], [10], [11], [12]. Based on BB84 protocol, the work in [9] investigated both horizontal and vertical links assuming various transmission distances and depths. The study in [10], investigated feasibility of horizontal submarine-to-submarine QKD links. In [11], performance analysis of BB84 protocol over turbulent underwater channels was presented discussing the effect of different water types, weather conditions and various system parameters. Decoy-state BB84 was further analyzed in [5], [12].

Experimental underwater QKD studies are relatively limited, see e.g., [7], [13], [14], [15], [16], [17], [18]. In [13], BB84 protocol with decoy state is implemented using off-line processing and tested over an air-to-water channel. The Alice and Bob (traditional names for the transmitter and receiver units in the QKD literature) are built on optical benches. A waveform generator is used for generating pulses at Alice and a timestamp instrument is used to record measured pulses at Bob. The underwater part of the link is 30 m long, the achieved QBER(Quantum Bit Error Rate) is %2.48 and key rate of the system is 220 bps. [7] presents another experimental study of decoy state underwater QKD where a waveform generator is used for generating pulses at Alice and an oscilloscope is used to record pulses at Bob. The average QBER of signal state is %0.95 and key rate of the system is 711 kbps. [14] uses a spatial light modulator (SLM) to generate different orders of OAM. As wavelengths, 710 nm and 943 nm are used as idle and signal transmisson signals which are typically not preferred for underwater channels. The work in [15] characterizes underwater channel for quantum communications in the Ottawa River. Their system uses a wavefront error sensor(WFS) and a CCD camera at receiver to analyze effect of turbulence to generated states. At [17] researchers built a 55 m long experimental air to water QKD test setup where 6 polarization states are generated and generated states are successfully received with very low distortion more than 95% fidelity. [18] investigates underwater quantum channels in a 30-meter flume tank and use 532 nm wavelength. The QBER is calculated as 0.91% and key per transmitted photon is measured as 0.84 at 30 m after post processing of the recorded information.

The experimental underwater QKD set-ups in the aforementioned works typically use laboratory equipment and off-line processing. The exceptions are [16], [19] which used FPGAs for the development of the underwater QKD experiments. The QKD set-up in [19] implements the decoy state BB84 protocol where FPGA is used for sending pulses and receiving them. All QKD implementation is implemented on an external user PC including the error checking, error correction and privacy amplification. They have reported a final key rate of 245.6 bps with an average QBER of %1.91 in 2.4 m water channel. [16] also implemented BB84 protocol and achieved %3.5 QBER in 2.37 m water channel where FPGA is used just for sensing incoming pulses and sending the timestamp information to computer. The rest of the BB84 implementations are also done on user computers. While their partial implementation builds upon FPGA, the works in [16], [19] still heavily rely on offline processing and computers to retrieve final key.

In this article, as a first step towards real-time quantum-secure underwater wireless networks, we develop an underwater QKD PoC built on a hybrid computation system. A real-time photon counting module is implemented on FPGA while the rest of the QKD algorithm works on the onboard computer (OBC) unit. Since the OBC will handle heavy computing tasks, this design choice is expected to be instrumental in reducing the execution time of the QKD algorithm. To the best of our knowledge, this work is the first fully integrated underwater QKD terminal prototype that processes all QKD operation without any involvement of user. The design relies on one of the most simple and cost-effective FPGA Chips (Intel Cyclone 10 LP) available on market and no external FPGA memory is used on the implementation.

The rest of the article is organized as follows: In Section II, we present the system architecture. In Section III, we have deep dived in to how realtime QKD is implemented using OBC and FPGA platforms. In Section IV, we have showed the Final PoC and shared experimental results of the system and Section V is the conclusion where we have summarized the current status and shared some of planned potential improvements.

II. System Architecture

The PoC is built on the BB84 protocol [2]. In BB84 protocol, each binary bit is encoded using a pair of mutually unbiased basis. A typical choice in practice is the use of pre-defined polarization states known as bases. For example, to represent the “qubit zero”, either a vertical or a right-diagonal state can be used while a horizontal or a left-diagonal state can be used for “qubit one”. During transmission, Alice (traditional name of transmitter in quantum cryptography terminology) randomly swaps between these polarization states. Bob (receiver) measures the photons in one of the two bases chosen at random and records his choices as well as the outcome of detections referred to as “raw key”. Alice and Bob then compare publicly the two independent random sets of polarization bases that were used, making use for this purpose of a standard communication channel. This channel is not necessarily optical and can take any form based on the communication application. The bit values of those polarization states measured in the compatible bases yield the “sifted key” and the rest of the raw key is discarded. Any adversary (Eve) can intercept both the quantum and the communication channel. The communication channel however leaks no information to third parties due to intrinsic randomness, i.e., each base has equal probability of resulting in a one or a zero. Furthermore, since quantum measurements are destructive, any attempt by the eavesdropper on the quantum channel will introduce noise into the system revealing her presence.

The overall system architecture is presented in Fig. 1. To generate the required four polarization states at Alice (transmitter), we use single-mode pulse laser sources (denoted by LS1, LS2, LS3, and LS4) operating at 405 nm (blue color). The blue color is selected due to its favorable propagation characteristics in the underwater medium. These four laser sources are driven by an FPGA. Each of the laser outputs is followed by a tunable linear neutral-density (ND) filter, denoted by LNDF1, LNDF2,

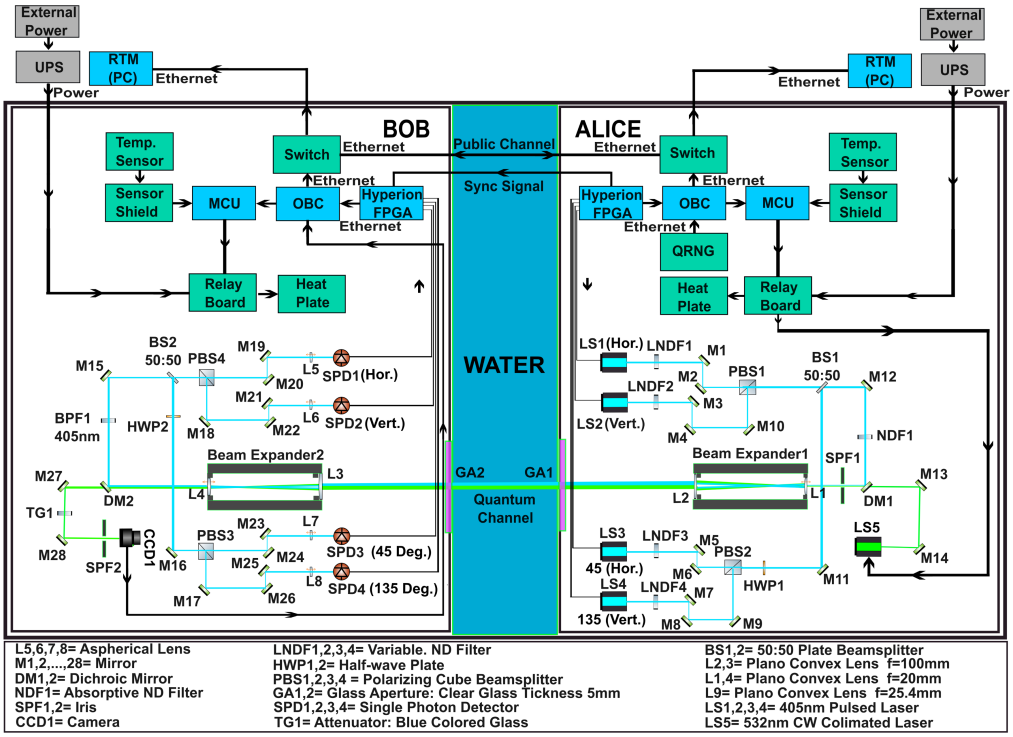


Fig. 1. Underwater QKD system architecture.

LNDF3, LNDF4, to attenuate laser pulses. The mirrors (denoted by M1, M2, M3, M4, M5, M6, M7, and M8) are used as pairs with respect to the laser path to perform the so-called "beam walking" and align lasers to the same spot. The mirror pair (M9 and M10) is used to change the beam position for effective utilization of the space. The polarizing beam splitters (PBS1 and PBS2) are used to combine horizontal and vertical polarizations. To obtain $+45°$ and $-45°$ polarizations, we first combine horizontal polarized (0°) LS3 and vertical polarized (90°) LS4, then use a half-wave plate (HWP) denoted by HWP1 at 22.5°. The resulting signal is then fed to a 50/50 non-polarizing beam splitter denoted by BS1. The combined polarized signals are redirected to a constant ND filter denoted by NDF1. The mirrors M14 and M13 are used for the purpose of beam walking and align blue and green (alignment) laser sources at the dichroic mirror denoted by DM1. The spatial filter SPF1 is used to limit beam size and mitigate back reflections in system. The Beam Expander 1 is used for increasing the beam width to 7.2 mm from 1 mm. The glass apertures (GA1 and GA2) are made with clear glass for minimum loss.

An OEM QRNG(Quantum Random Number Generator) module that provides guaranteed uniform distributed random binary bits at a rate of 4 Mbit/s is used as a random source. It generates the random bits and feeds them to the OBC for the proper selection of the laser modules, i.e., the polarization state. The randomness of data generated keys are verified according to [20]. The OBC provides information on the laser selection to the FPGA. This information is loaded to the block RAM of the FPGA. The FPGA reads this data using a FIFO module and accordingly generates precise short-duration electrical pulses. The timing of laser pulses is achieved by triggering laser diode modules with these FPGA-generated electrical pulses. Moreover, it also communicates with the receiver node over a public channel (Ethernet connection[1] in our case) for sifting. The synchronization of the two FPGA boards is achieved through an SMA cable. At Bob, the received optical signals from the blue colored (405 nm) lasers and the green colored (532 nm) alignment laser are passed through the Beam Expander 2 to reduce beam width back to 1 mm. Then they are demultiplexed using a dichroic mirror DM2. The green beam is passed through the spatial filter (denoted by SPF2) to limit beam size and blue epoxy glass (denoted by TG1) for slight attenuation. It is then redirected to the CCD1 using mirrors M27 and M28. A 405 nm band pass filter (denoted by BPF1) is used to reject unwanted wavelengths from the incoming photons. Using the mirror M15, the blue beam is redirected to the non-polarising 50/50 beam splitter (denoted by NPBS2). The NBPS2 randomizes basis selection by blindly forwarding incoming photons to two paths regardless of their polarizations. One path feeds to a polarizing beam splitter PBS3 to obtain polarization states of 0° and 90° while the other path is fed to a HWP2 for 45° rotation. The rotated polarization bases are redirected to PBS3 using the mirror M16 where polarization states of $-45°$ and $+45°$ are extracted afterwards. The mirror pairs (M19, M20, M21, M22, M23, M24, M25, M26) are used for beam walking to align beams at the center of the aspheric lenses denoted by L5, L6, L7, L8. These aspheric lenses focus laser beams to single photon

[1]In practice, this should be replaced by either an acoustic or optical link. In the case of an optical classical channel, data rates exceeding hundreds of Mbps or even Gbps can be achieved. This is 4-5 orders of magnitude faster than the quantum link. If the classical channel is based on acoustics, for the typical underwater QKD distances on the order of 10 m - 50 m, the acoustic link could provide data rates on the order of 100 kbps which is still well above the quantum link.

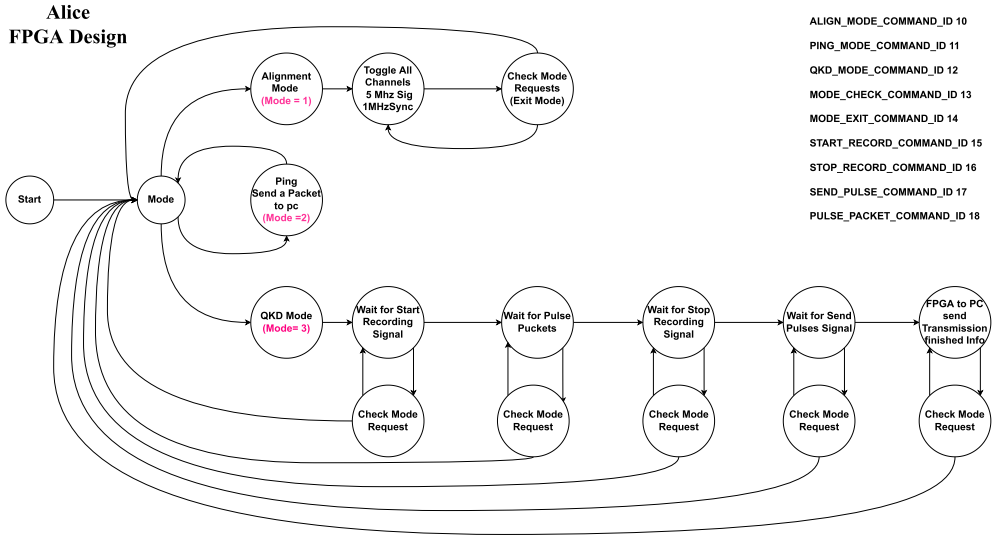


Fig. 2. Alice FPGA architecture.

detectors (SPDs) denoted by SPD1, SPD2, SPD3, SPD4 which generate electrical pulses if they detect any photons. The SPDs has less than 60 Hz noise and dead time is less than 45 ns. The active area of the SPDs is 50 um and efficiency at 405 nm is 18%.

The output of these detectors are connected to the FPGA for high resolution sampling of the received pulses. This part of the FPGA basically works as a timestamp unit. The FPGA sends the measured pulses to the OBC. Bob's OBC shares the measurement basis information with Alice's OBC through the public channel. Alice compares the received measured basis information with the transmitted basis information. Alice picks 128 samples from the matching bases for the sifted key generation. For error correction, it is possible to use various forward error correction techniques including turbo codes, polar codes, LDPC codes [21]. Due to its simplicity we have used Reed Solomon (RS) coding [22] for error correction. The selected basis measurements and the parity bits are then transferred back to Bob. Using the selected basis measurements, Bob generates the sifted key with the measurements recorded before and uses RS parity bits to reduce the effects of possible errors (e.g., due to noise etc). The parameter estimation phase is not yet implemented.

III. Real-Time Software Development

The underwater QKD system is designed to operate in two main modes, namely "Alignment Mode" and "QKD Mode". The alignment mode is useful to assist the system operator during the manual alignment stage. In this mode, Alice sends continuous pulses to all four lasers and Bob records the amount of photons received in the last 100 ms. The operator can make small adjustments in manual alignment by referring to the received photon counts on each detector. In the QKD mode, the system runs a full BB84 cycle and generates 128-bit keys in each successful QKD iteration.

Although the hardware designs of Bob and Alice's units are identical, their digital design differs. Their flow charts are respectively provided in Figs. 2 and 3. As can be seen from Fig. 2, once Alice receives the alignment mode request, it toggles all lasers at 5 MHz and 1 MHz sync channel. Synchronization is made by a cable in our implementation; a similar signal is provided through that channel to check possible cabling issues. The photon generation in QKD is a time-sensitive process and the transmitter should generate photons with a very accurate timing. To prevent any delays while transferring the QRNG laser selection information from OBC to FPGA, the OBC first loads the desired laser selection sequence to the FPGA. When the QKD mode request arrives at the FPGA, it switches to a state where it records the incoming pulse sequence. After the process of loading the pulses to the FPGA is completed, the OBC transfers the "stop record" pulse and the FPGA waits for the "send pulses" packet from the OBC as a final trigger to send all the recorded pulses. When the FPGA receives the "send pulses" command, it starts generating 20 ns pulses with the repetition rate of 10 MHz to lasers according to the information stored in the block RAMs of the FPGA, and at the same time it sends a logic (high) signal to the synchronization cable. After completing the transmission, it returns back to the beginning of the QKD mode state to wait for the next QKD cycle. As can be seen from Fig. 3, during the alignment mode, the FPGA switches to state to counting the pulses from all four detectors and the sync channel. It then sends the recorded pulse counts to the OBC in every 100 ms. In addition, to accomplish timing at Bob's side, it loads the incoming information to the block RAM of the FPGA and sends the measured pulse information to the OBC. When the QKD mode is activated, Bob waits for the sync signal which is sent by Alice at the same time when she sends pulses to the lasers.

A. Details of FPGA Implementation

The custom designed credit card sized FPGA board can be seen at Fig. 4. Same FPGA HW used for both Alice and Bob. There is MXM connector placed on FPGA board where the transmit and recieve signals received via daughterboard. The

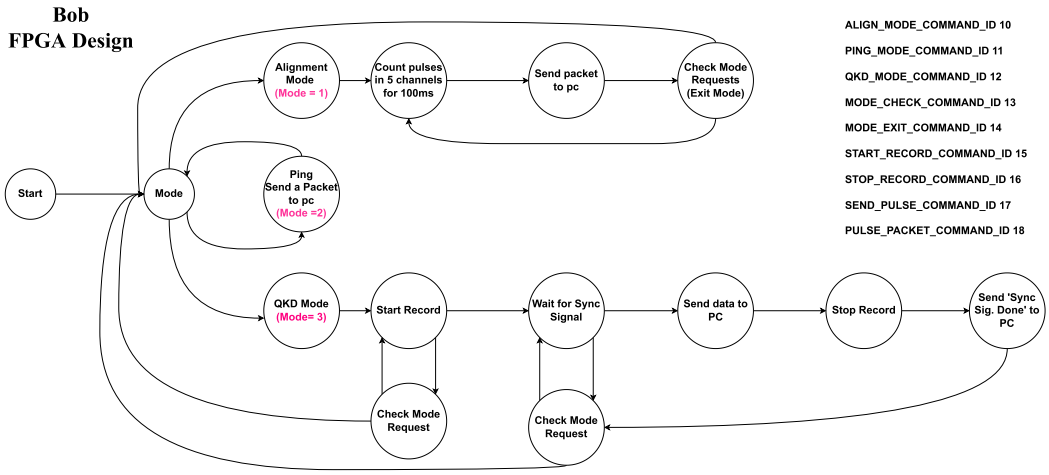


Fig. 3. Bob FPGA architecture.

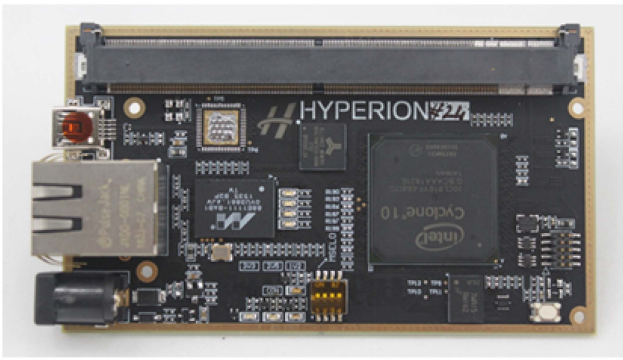


Fig. 4. Custom designed FPGA Board.

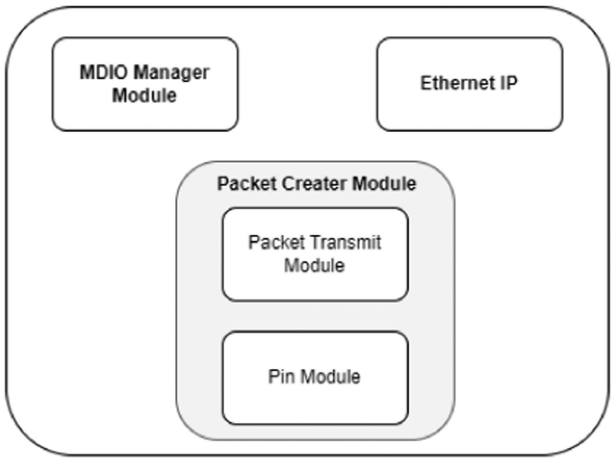


Fig. 5. FPGA TX Top Level Diagram.

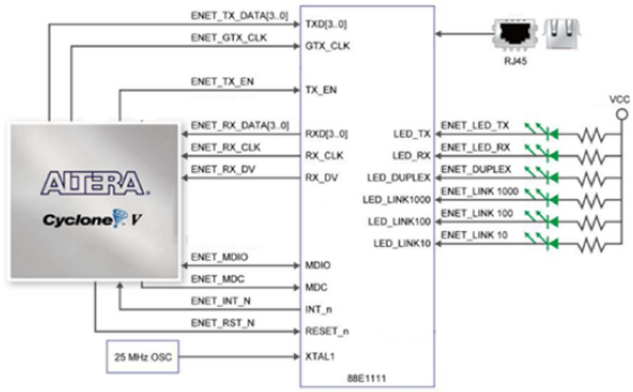


Fig. 6. Ethernet PHY – FPGA Connections.

design done in extendable form to ease implementation of all detectors and lasers on a extension board in future by keeping the most complicated FPGA HW design same. In our study, we developed a transmitter structure that can generate signals for four channels at the desired time and a receiver structure that records the arrival times of signals from four separate channels. In order to transfer the relevant signals to a computer, various auxiliary modules were developed and a receiver and transmitter system has been created. The internal structure of the transmitter system is given in Fig. 5.

The transmitter includes Ethernet IP, MDIO (Management Data Input/Output) Manager and Packet Creator modules. FPGA Ethernet MAC (Media Access Control) IP refers to a pre-designed and tested digital logic circuit that implements the MAC layer of the Ethernet protocol in an FPGA device. This Ethernet MAC IP is responsible for controlling the flow of data on an Ethernet network, including handling the transmission and reception of Ethernet frames and performing error checking.

The MDIO Manager Module was created to write and read address and data information to the registers of Ethernet IP. It waits for request while performing write and read operations. It also allows communication between the physical layer (PHY) and the MAC layer for tasks such as monitoring status, and controlling various PHY functions. MDIO provides a standardized way for the MAC layer to control and configure the PHY devices in the network. This allows for interoperability between different types of PHYs and MACs, making it easier to upgrade or replace components in the network. The use of MDIO also enables advanced features such as energy-saving modes and link-partner auto-negotiation, which are important for optimizing network performance and reducing power consumption. Fig. 6 shows

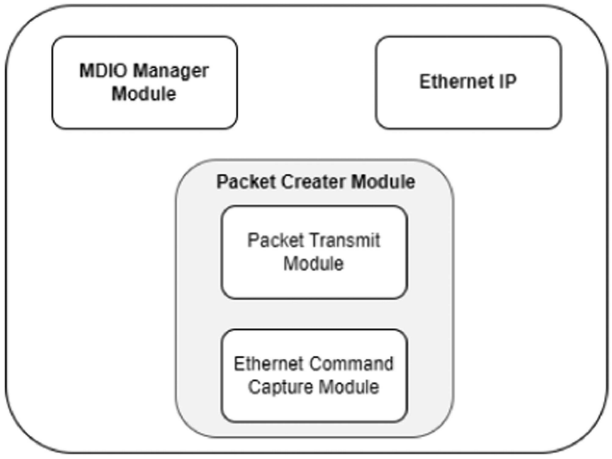


Fig. 7. FPGA RX Top Level Diagram.

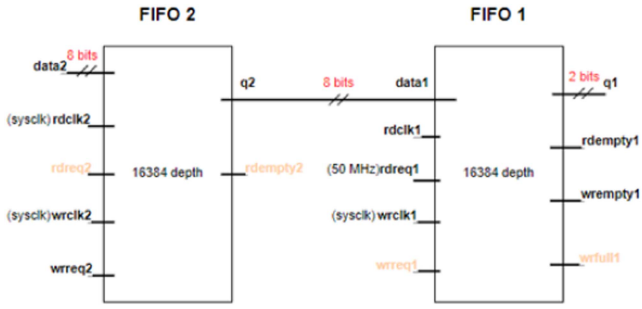


Fig. 8. Back-to-Back Buffering Mechanism.

Fig. 9. Timestamp Packet Structure.

the connection of the PHY controlled by the MDIO manager with the FPGA.

Packet Creator Module consists of 2 submodules. The Pin Submodule handles precise pulse generation operations while the Packet Transmit Submodule is used to enable switching between different modes of operation. In Alignment Mode, the outputs from the Pin Module are 5 MHz signals for data channels and 1 MHz signals for synchronization channels. In QKD mode, the channel information from the Ethernet is written to the FIFO in the Pin Module. Subsequently, a message is sent to the OBC via the Ethernet with the Packet Transmit Module. When a Ping Mode request is received from the Ethernet, the Packet Transmit Module pings the OBC over Ethernet in 200 ms. Subsequently, the Packet Transmit Module generates the data package for transmission.

As illustrated in Fig. 7, the receiver includes Ethernet IP, MDIO manager and Packer Create Module blocks. The MDIO manager and Ethernet IP used in the receiver are the same as the IPs used in the transmitter. The Ethernet Command Capture module receives the mode information from the Ethernet. If Alignment mode is selected, data from 4 channels is counted every 100 ms. The 1 MHz signal coming from the synchronization channel is counted to start recording the data and to ensure synchronization. In QKD mode, incoming data is written to FIFOs. The process of writing to FIFO continues as long as the signal comes from the synchronization channel. Datas written to FIFO are recorded with the time they arrive. The RX design operates at twice the speed of the TX design. This causes some data not to be captured. Synchronization is required when a signal is transferred between circuits in unrelated or asynchronous clock domains. A signal that is asynchronous with the clock is captured by passing it through Flip-Flops and ensuring its synchronization. After the captured data is saved in FIFO, it is packaged in the Packet Transmit Module and sent to the computer. Mode information can be checked during operation. When ping mode comes, Packet Transmit Module sends a packet to a computer in 200 ms.

The Alice transfers channel information to FPGA using UDP protocol. The Alice's FPGA has a state machine that decodes the UDP packet. First it confirms that the received UDP packet is a valid channel information data packet. The data loader state machine takes the payload part of the UDP packet and redirects it to buffering module. The buffering module fills all block RAM of the Cyclone 10LP FPGA. To perform this task, the largest 16 K option supported by Altera FIFO IP was used, but since there was more space, two 16 K FIFOs were connected back-to-back to create a large buffer memory. The final buffer module takes 8-bit inputs and generates 2-bit laser selection output. After Alice receives transfer command from Alice OBC through UDP interface, Alice starts reading loaded information at FIFOs and uses output 2 b laser selection information to generate very short electrical pulses at related channel. The implemented information loading and pulse sending mechanism lets FPGA to have deterministic pulse timing rate and simplifies overall system operation. The Fig. 8 shows the back-to-back buffering mechanism where two FIFOs are used.

At the receiver side, a 28-bit counter is used for high precision time of arrival information of the pulses received from the channels. Next to this counter, pulse channel information is added as 1 b for each of the 4 channels. The 32 b timestamp packet structure is presented in Fig. 9.

The functionalities implemented in FPGA design are verified in the test/development tools prior to flashing bit files to the actual FPGA board. For this purpose, Universal Verification Methodology (UVM) infrastructure was used to verify the design in the simulation environment. Receiver and transmitter design overhead modules include Ethernet and channel signals. Two separate agents have been created in the UVM environment for Ethernet and channel signals. The Ethernet agent handles packets sent and received over Ethernet to the design. The other channel agent is designed to control the pulses to be sent to the line and to examine the pulses coming from the line. Fig. 10 shows the UVM structure created using 2 agents.

B. *Metastability Prevention*

Metastability could pose a problem in digital circuits because it can lead to errors or malfunction. For example, if a circuit is metastable, it may produce an incorrect output value, or it may

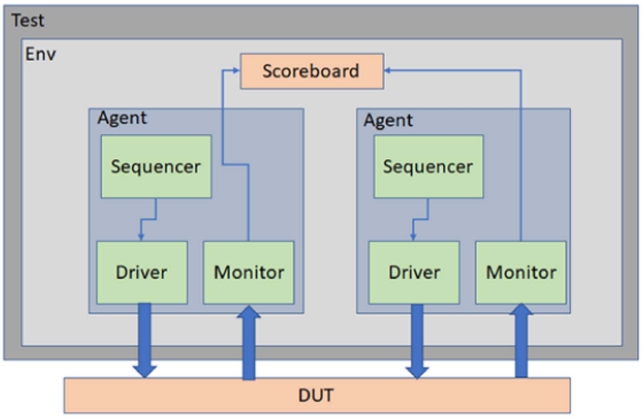


Fig. 10. Verification Environment.

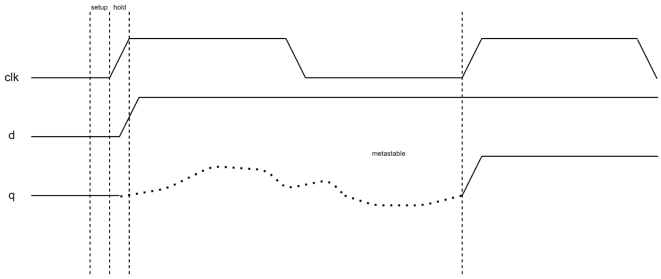


Fig. 11. Sampled Metastable Signal.

oscillate between two or more possible output values. This can cause problems in systems that rely on the accuracy of digital signals, such as computers or other digital devices. There are a number of factors that can contribute to metastability in digital circuits. One common cause is rapid transitions in the input signal, which can cause the circuit to become uncertain about the correct value. Other factors that can contribute to metastability include noise on the input or power supply, or variations in the timing or operation of the circuit. To address the problem of metastability, digital designers can use a variety of techniques, such as glitch filters, debouncing circuits, and specialized circuit designs. In addition, it may be necessary to use synchronization techniques, such as phase-locked loops, to ensure that the circuit is able to latch the correct value. In the Fig. 11, it is seen that the clock in the receiving system is asynchronous with the data in the sending system. Therefore, a metastable signal is sampled when the data changes in the setup-hold time interval of the receiving system. This will cause the remaining logic of the system to work inconsistently.

If the period of the metastable signal leaving the sending system is shorter than the period of the clock of the receiving system, it is also possible that the relevant data is not sampled at all during the sampling period. In the Fig. 12, the clock period of the transmitter is lower than that of the receiver. When a metastable signal is given to the back of the circuit, it may cause different results in each logic. This situation is shown in the NOT Gates in the Fig. 13. Even if the same metastable signal comes to the inputs of NOT Gates, it can cause each NOT Gate to produce a different output value.

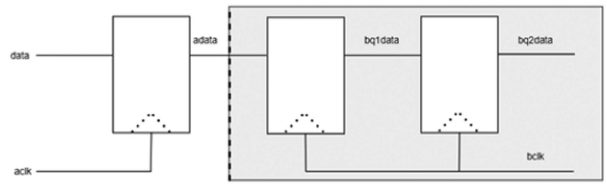


Fig. 12. Inconsistency of Metastable Signal in Circuit.

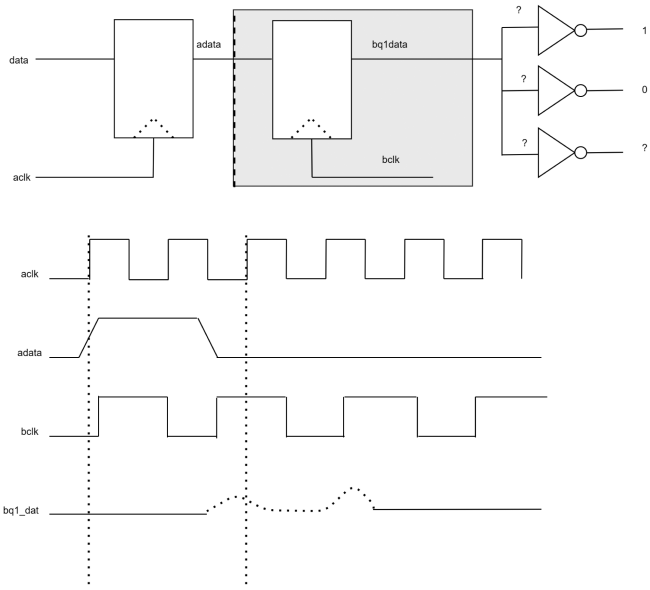


Fig. 13. Metastability Leader-Follower Register Solution.

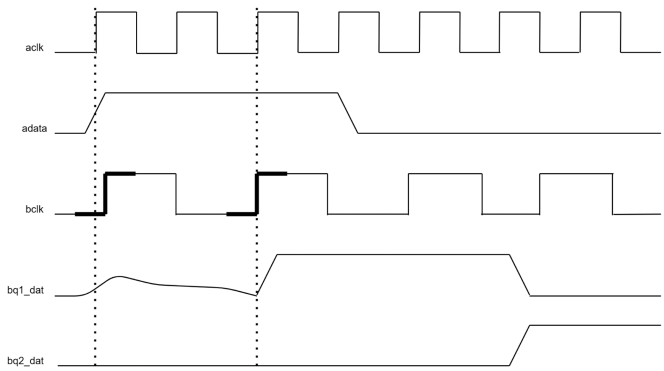
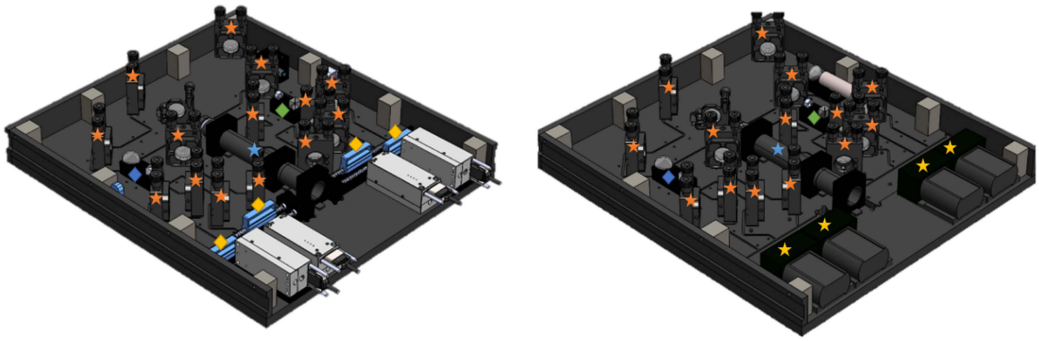


Fig. 14. Stable Signal Capturing.

One way to mitigate the effects of metastability in a digital circuit is to use a “leader-follower” flip-flop. This type of circuit uses two flip-flops connected in series, with the output of the first flip-flop (the “leader”) driving the input of the second flip-flop (the “follower”). The key feature of a leader-follower flip-flop is that the follower flip-flop is only allowed to update its output value on the rising edge of the clock signal. This means that the follower flip-flop will only change its output value when the clock signal is stable, rather than while it is transitioning. As a result, the leader-follower flip-flop is less prone to metastability than a simple flip-flop circuit. To use a leader-follower flip-flop in a circuit, the input value is applied to the leader flip-flop, and the output of the follower flip-flop is used as the circuit’s output.



(a) *Alice* (b) *Bob*

Fig. 15. Alice and Bob's optical benches.

(a) *Alice* (b) *Bob*

Fig. 16. Photo of Alice and Bob's optical benches.

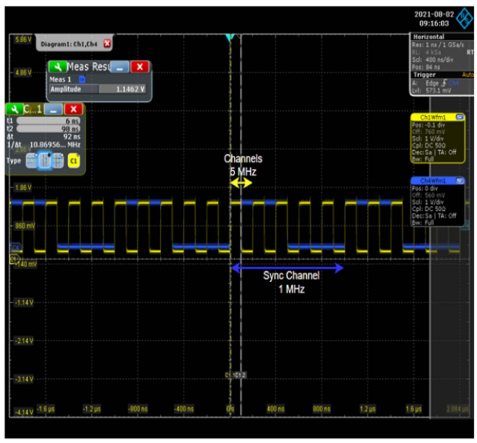


Fig. 17. Output signal of the FPGA when alignment mode is on.

Fig. 18. Underwater performance tests.

The rising edge of the clock signal is used to latch the value from the leader flip-flop into the follower flip-flop. This ensures that the output of the circuit remains stable, even if the input value is changing rapidly or is uncertain. Fig. 13 shows the metastability elimination solution set up using two flip flops.

In a leader-follower flip-flop, it is possible for the leader flip-flop to become metastable, while the follower flip-flop remains stable. This can happen if the input value to the leader flip-flop changes while the clock signal is transitioning. If the leader flip-flop becomes metastable, it will be unable to decide between the two possible output values. However, the follower flip-flop will only update its output value on the rising edge of the clock signal, when the clock signal is stable. As a result, the follower flip-flop will not be affected by the metastability of the leader flip-flop, and will continue to produce a stable output. In this situation, the output of the leader-follower flip-flop may be delayed slightly, as the follower flip-flop will not update its output until the next rising edge of the clock signal. However, the output will still be

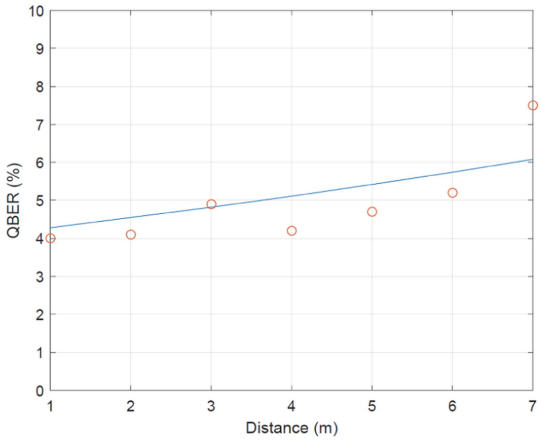


Fig. 19. QBER versus distance.

stable and correct, even if the input value is uncertain or changing rapidly. Fig. 14 illustrates the metastability-eliminated signals are seen using two flip flops. While there is a metastable signal in the first flip flop, the problem in its transfer to the second is eliminated.

IV. Final PoC and Experimental Results

Before the system integration, various optomechanical tests were conducted. The effect of temperature changes on the beam expander and kinetic mirrors were tested in the lab environment. For this purpose, the temperature was changed between 0 and 40 degrees. No deformation or permanent damage was observed on the mechanical system. The validated sub components were assembled on optical benches following the assembly plan in Fig. 15. Different colors are used to denote different components, i.e., orange star - the kinematic mirror mounts, blue star - beam expander, blue diamond - beam splitter, green diamond - HWP, yellow diamond - Linear ND filters, yellow star - SPDs and attached aspheric lenses. Using these benches, laser transmission was successfully tested. Then they were integrated on a rack. The heat plates are assembled to copper sheets to provide homogenous temperature control at each point of the optical benches. The copper plate is then assembled bottom of optical benches with a heat transfer compound applied between them. The assembled final versions of Alice and Bob optical benches are provided in Fig. 16. One of the design challenges is alignment and temperature stabilization of such systems, because underwater temperature levels are much lower than the actual calibrated temperature and the thermal expansion might create misalignment inside of optical benches. To solve this issue, we have implemented a temperature stabilization system for optical benches. Under the consideration of maximum sea temperature is around 30,2 we have stabilised internal temperature of the system to 35 degrees.

The initial tests of the developed underwater QKD system were conducted by directly connecting Alice and Bob's FPGA ports. Fig. 17 shows the output ports of Alice during the alignment mode. The blue channel is the sync channel and it is toggled at 1 MHz. The yellow channel is the signal that goes to all four lasers, which toggles at 5 MHz. The Sync Channel clock and Signal Channels clocks are generated using the same clock source and Signal Channel and Sync Channels are perfectly synchronized. To test Bob's alignment mode, the Wireshark network sniffing tool was used to capture the generated packets with the count rates of all channels in the last 100 ms. The ping modes of Alice and Bob are similar. The unit sends a predefined packet to the OBC until the exit packet is transferred.

After sub-system integration and validation, the PoC was tested for underwater transmission. For this purpose, a PVC pipe filled with tap water was installed between the two terminals (see Fig. 18). QBER measurements versus distance were taken and presented in Fig. 19. The rapid increase in QBER in short distances are mainly based on polarization distortion and low SNR due to imperfections associated with the thick clear glass aperture used in the implementation to withstand high pressure. A linear curve is also included in this figure via data fitting to measurement results. It exhibits a linear behaviour. An average secure key rate of about 100 qubits per second was recorded during the experiments. It is generally accepted that the BB84 protocol is secure against a sophisticated quantum attack if the QBER is less than 0.11 [23]. It is observed that QBER safely remains below 0.11 in our implementation.

V. Conclusions and Future Directions

In this article, we presented the implementation of a BB84-based underwater QKD system with real-time operation capability. The system was built on a hybrid computation system consisting of an FPGA and an OBC interfaced with optical front-ends. A real-time photon counting module and photon generation designs are implemented on FPGA while the rest of the QKD algorithm works on the OBC unit.[2] The system was also equipped with a visible laser and an alignment indicator to validate successful manual alignment. The implementation of real-time QKD systems using FPGAs has several challenges which we have satisfactorily addressed in our design. One of the main challenges is need for low latency operation when generating pulses and reading them from the detector. We have used digital single ended electrical interfaces to send and receive these signals. To have a cost-effective system design, we selected a basic FPGA and only used integrated block RAMs of FPGA. The total time duration of sending/receiving encoded photons is chosen to utilize all memory size of FPGA while keeping the target secure key requirements of 100 qubits per second. In real time QKD systems, the main oscillator that drives that is being used for generating photons and reading them plays critical role to know "Which state is transmitted?" and "When it is transmitted?. If the synchronisation can not be performed preferably much better than transmit pulse rate, system may not perform well and lead to higher QBER. We have observed that

[2]The SoC(System on Chip) FPGAs are powerful alternatives when flexibility of programming on ARM CPU and real-time processing capability of FPGA is needed. Different from the [24], we have aimed to build a system that completely isolates users from all QKD operations. It can be readily checked from that it still requires a PC to make offline processing of the retrieved data. This indicates that adding SoC FPGA did not avoid using powerful external processors in the overall system architecture. In addition, our system is quite flexible; the OBC (Intel i7 CPU) and FPGA parts can be easily replaceable.

using the same FPGA series but different logic elements at transmit and receive side also may lead to long term stability issues due to the way IDE optimizes the implementation. Although IDE claimed that operating frequency is satisfied, shift between clocks observed at long duration tests and lowered resulting QBER in long run tests.

This PoC system can be enhanced in several directions. In the current system, an Ethernet connection between transmitter and receiver channel serves as the public channel and Sync Cable used for syncing FPGA clocks. This can be replaced with an optical link to implement an end-to-end quantum-secure communication system demo. Such an optical link can be simultaneously used for synchronization purposes between transmitter and receiver.

The current PoC system builds upon the BB84 protocol. This protocol is commonly used in QKD systems, owing to its simplicity and effectiveness. Nonetheless, the laser sources sometimes produce pulses containing two or more photons. Thus, an eavesdropper could in principle perform a so-called Photon-Number-Splitting (PNS) attack, and obtain information about the generated key. The most common counter measure to protect QKD systems from such PNS attacks is the combination of the BB84 and the decoy-state method. The decoy-state method requires the variation of intensity during pulse generation, so as to create signal-states and decoy states. With additional upgrades on software and hardware, the developed system can be used to implement decoy-state BB84 protocol. For example, the power of each laser can be adjusted dynamically by a software upgrade and required additional states can be obtained through this. However, the resulting delay of power adjustments using the existing RS232 connection can take up to few seconds and might be problematic for real-time implementation of decoy-state BB84. As an alternative, an electro-optical modulator (EOM) can be included prior or after the beam expander to vary the output photons dynamically in a fast manner. By applying some voltage to EOM, an additional attenuation can be employed on transmitted beam.

The key rate of the current PoC system is limited to around 100 bps which is mainly limited by the FPGA capabilities. In the current implementation, the actual time of the pulse transmission is around 9.8 msec limited by the available logic element size in the deployed FPGA. In each iteration, Alice loads 98304 samples to its FPGA and Alice waits confirmation from Bob side to start transferring the loaded pulses. It is possible to increase the generated bit sequences using a more powerful FPGA with more logic elements or OBC and FPGA can be integrated to a single SoC FPGA. Another alternative is to use a pipelined software implementation. The current version of software controls every step in separate threads and performs each operation step by step. There can be some improvements such that it pipelines multiple QKD operations and tags all of them for further processing. Another possible improvement can come from the adoption of flexible key size. The current version of implementation used a fixed length of 128 b key in each QKD iteration. It simply discards any other measurement if the system has more measurements. It also does not process the QKD iteration if matching basis information is lower than 128 samples. The key rate can be improved by adopting a flexible key size. In such case, error correction lengths and packet sizes are required to be dynamically calculated according to the number of matching bases.

REFERENCES

[1] S. Jiang, "On securing underwater acoustic networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 729–752, Firstquarter 2019.

[2] S. Loepp and W. K. Wootters, *Protecting Information: From Classical Error Correction to Quantum Cryptography*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[3] E. Diamanti, H.-K. Lo, B. Qi, and Z. Yuan, "Practical challenges in quantum key distribution," *npj Quantum Inf.*, vol. 2, Nov. 2016, Art. no. 16025,.

[4] F. Xu, X. Ma, Q. Zhang, H.-K. Lo, and J.-W. Pan, "Secure quantum key distribution with realistic devices," *Rev. Modern Phys.*, vol. 92, no. 2, May 2020, Art. no. 025002.

[5] A. H. F. Raouf, M. Safari, and M. Uysal, "Performance analysis of decoy state quantum key distribution over underwater turbulence channels," *J. Opt. Soc. Amer. B*, vol. 39, no. 6, pp. 1470–1478, 2022.

[6] F. Grünenfelder, A. Boaron, D. Rusca, A. Martin, and H. Zbinden, "Performance and security of 5GHz repetition rate polarization-based quantum key distribution," *Appl. Phys. Lett.*, vol. 117, no. 14, 2020, Art. no. 144003.

[7] Z. Feng, S. Li, and Z. Xu, "Experimental underwater quantum key distribution," *Opt. Exp.*, vol. 29, pp. 8725–8736, 2021.

[8] L. Johnson, R. Green, and M. Leeson, "A survey of channel models for underwater optical wireless communication," in *Proc. 2nd Int. Workshop Opt. Wireless Commun.*, Newcastle Upon Tyne, U.K., 2013, pp. 1–5.

[9] S. C. Zhao, X. H. Han, Y. Xiao, Y. Shen, Y. J. Gu, and W. D. Li, "Performance of underwater quantum key distribution with polarization encoding," *J. Opt. Soc. Amer. A*, vol. 36, pp. 883–892, 2019.

[10] J. Gariano and I. B. Djordjevic, "Theoretical study of a submarine to submarine quantum key distribution systems," *Opt. Exp.*, vol. 27, pp. 3055–3064, 2019.

[11] A. H. F. Raouf, M. Safari, and M. Uysal, "Performance analysis of quantum key distribution in underwater turbulence channels," *J. Opt. Soc. Amer. B*, vol. 37, pp. 564–573, 2020.

[12] M. Lopes and N. Sarwade, "Optimized decoy state QKD for underwater free space communication," *Int. J. Quantum Inf.*, vol. 16, 2018, Art. no. 1850019.

[13] C.-Q. Hu et al., "Decoy-state quantum key distribution over a long-distance high-loss air-water channel," *Phys. Rev. Appl.*, vol. 15, 2021, Art. no. 024060.

[14] F. Bouchard et al., "Quantum cryptography with twisted photons through an outdoor underwater channel," *Opt. Exp.*, vol. 26, 2018, Art. no. 22563.

[15] F. Hufnagel et al., "Characterization of an underwater channel for quantum communications in the Ottawa River," *Optics Express*, vol. 27, pp. 26346–26354, 2019.

[16] S. Zhao et al., "Experimental investigation of quantum key distribution over a water channel," *Appl. Opt.*, vol. 58, pp. 3902–3907, May 2019.

[17] C.-Q. Hu et al., "Transmission of photonic polarization states through 55-m water: Towards air-to-sea quantum communication," *Photon. Res.*, vol. 7, 2019, Art. no. A40.

[18] F. Hufnagel et al., "Investigation of underwater quantum channels in a 30 meter flume tank using structured photons," *New J. Phys.*, vol. 22, 2020, Art. no. 093074.

[19] S. Dong et al., "Practical underwater quantum key distribution based on decoy-state bb84 protocol," *Appl. Opt.*, vol. 61, pp. 4471–4477, May 2022.

[20] L. Bassham et al., "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, 2010. Accessed: Jul. 23, 2023. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=906762

[21] J. J. Boutros and E. Soljanin, "Time-entanglement QKD: Secret key rates and information reconciliation coding," 2023, *arXiv:2301.00486*.

[22] I. S. Reed and S. Solomon, "Polynomial codes over certain finite fields. Journal of the society for industrial and applied mathematics," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 2, pp. 300–304, 1960.

[23] C. Cardoso-Isidoro and F. Delgado, "Shared quantum key distribution based on asymmetric double quantum teleportation," *Symmetry*, vol. 14, no. 4, 2022, Art. no. 713.

[24] A. Stanco et al., "Versatile and concurrent FPGA-based architecture for practical quantum communication systems," *IEEE Trans. Quantum Eng.*, vol. 3, 2022, Art. no. 6000108.

[25] E. Rosenkrantz and S. Arnon, "Optimum LED wavelength for underwater optical wireless communication at turbid water," *Proc. SPIE*, vol. 9224, pp. 349–354, 2015.