September 2024

Wells Fargo Technology Case Study

Quantum Key Distribution for Symmetric Key Generation

Wells Fargo Technology | Cybersecurity Data Science
Quantum Security Team (QS)
Optical/Secure Network Services (OSNS)

Contents

Abstract

We live in a time when massive amounts of personal information and financial data are transmitted over public networks. Consequently, the significance of secure communications cannot be overstated. Symmetric cryptography, including data encryption and message authentication, is widely used to protect confidential information. Today, these symmetric keys are managed using either classical symmetric or modern asymmetric key management methods. However, the coming quantum computer threat puts modern asymmetric cryptography, and to a lesser degree, classical symmetric cryptography, at risk. Postmodern solutions, such as the NIST Post-Quantum Cryptography (PQC) asymmetric algorithms, and other quantum-resistant technologies such as Quantum Key Distribution (QKD) provide a cryptographic transition path.

Challenge Statement

Information theory shows that traditional secret-key cryptosystems cannot be totally secure unless the key, used only once, is at least as long as the cleartext. A cryptosystem is considered information-theoretically[1] secure if its security derives from information theory, i.e., it is secure even when the adversary has unbounded computing power and memory. Quantum Key Distribution (QKD) [2] provides information theoretically secure key exchange in the era of quantum computing. Security technologies such as QKD, together with PQC, will extend the application prospects of quantum-resistant security.

Introduction

The Cybersecurity Data Science (CSDS) Quantum Security (QS) team is performing experiments that research quantum-resistant technologies to keep Wells Fargo customer data and operations data safe in the anticipation of cryptographically relevant quantum computers (CRQCs) that threaten classical cryptography.

Quantum Key Distribution, which is based on quantum mechanical physics, provides quantum-safe secure delivery of symmetric keys between entities. The proof-of-technology (PoT) experiments demonstrate operability and provide greater understand of the necessary hardware systems to bring up a QKD system and integrate it into the Enterprise cryptographic ecosystem.

The experiment and accompanying research for this case study demonstrated the understanding and ability to operate a European Telecommunications Standards Institute (ETSI) compliant QKD system. The test bed included the presence of two endpoints set apart over a physical distance, connected by a local fiberoptic infrastructure, where each endpoint represented an enterprise network or data center. The experiment demonstrated the ability to successfully distribute photonic data bits to each endpoint to securely generate cryptographic keys at both locations.

The research and the accompanying experiments were initiated and planned by the Wells Fargo QS team and the Optical/Secure Network Services (OSNS) team. The experiments were conducted in collaboration among the two Wells Fargo teams in partnership with Toshiba and Ciena technical teams.

[1] Liang et al "Information theoretic security." Foundations and Trends® in Communications and Information Theory 5.4–5 (2009): 355-580.

[2] Bennett, Charles H., and Gilles Brassard. "Quantum cryptography: Public key distribution and coin tossing." arXiv preprint arXiv:2003.06557 (2020).

Technical Description

In contrast to traditional methods for distributing symmetric keys, QKD offers secure transmission of key information with measurable security and the capability to detect eavesdropping. To achieve broader adoption and scalability, QKD must seamlessly integrate with existing fiberoptic infrastructures employed in classical data communications. This integration is accomplished by combining quantum and classical data signals onto a single fiber connection using Wavelength Division Multiplexing (WDM).

QKD systems typically consist of four physical components as shown in Figure 1 and detailed in the table below.

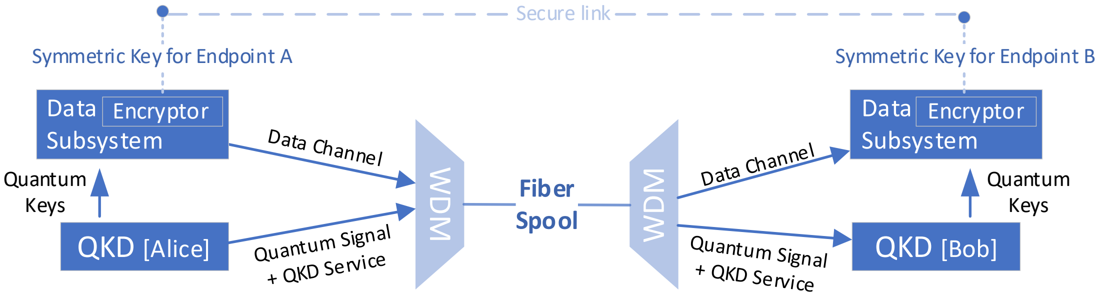

Figure 1 : Generic QKD System

Four typical QKD physical components:

QKD Appliance	The QKD appliance is the key provider for employing techniques required to encode/decode key information using photon states based on the BB84 protocol. QKD systems incorporate a mechanism for generating ‘quantum keys’ i.e., keys formed out of a stream of single photons or employ an integrated QRNG (Quantum Random Number Generator) for key generation.
Data Subsystem	The data subsystem (including high-speed encryptors) supports the implementation of symmetric key cryptography such as the Advanced Encryption Standard (AES). AES security is enhanced by key refresh from QKD systems.
Interconnect WDM	Wavelength Division Multiplexing interconnects use multiple light wavelengths to transmit classical data signals and quantum signals together on the same fiber.
Fiber Cabling	Fiberoptic cables are used to carry coherent light beams (and quantum states) from optical transmission appliances.

Setup and Approach

The experiments were conducted within the Wells Fargo Quantum Security Laboratory using a Toshiba's QKD system, consisting of QKD 4.2-LD and a QKD Controller 4.2 SO3, as well as Ciena's Waveserver 5 high-speed encryptor and optical transceiver. The experiments included two endpoints set approximately 140 meters apart using a fiberoptic cable spool within the lab environment.

The experiments were accomplished collaboratively among technical teams from Wells Fargo, Toshiba, and Ciena in two phases. Phase A consisted of only Toshiba QKD systems in the setup, Phase B included Ciena high-speed encryption devices in the experiment.

Phase A – Only Toshiba QKDs

Figure 2 shows Phase A: Toshiba QKD System (designated "Alice") to Toshiba QKD System (designated "Bob") where connectivity was demonstrated.

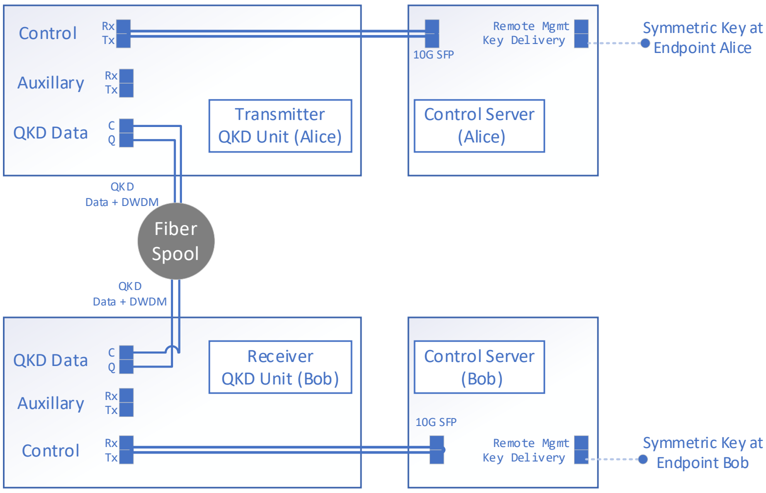

Figure 2 : Phase A Using Toshiba QKD and Control Server

Device label in the experiment	Device Model	Function
Transmitter QKD Unit (Alice)	Toshiba QKD4.2A-LD	Long distance QKD transmitter with forward/backward dual fiber
Receiver QKD Unit (Bob)	Toshiba QKD4.2B-LD	Long distance QKD receiver with forward/backward dual fiber
Control Server (Alice)	Toshiba QKD4.2A-SO3	For user interaction with the QKD Unit
Control Server (Bob)	Toshiba QKD4.2B-SO3	For user interaction with the QKD Unit

Toshiba QKD 4.2 A/B–LD (Long Distance) appliance[3] enables QKD data traffic and additional data traffic to operate over the same fiber as the quantum signals. Toshiba QKD4.2 A/B-SO3 appliance is provided with each QKD unit, allowing the user to interact with the corresponding QKD unit.

[3] Toshiba Europe, 2023, User Manual: QKD 4.2 Transmitter/Receiver - Multiplexed LU/LD/XLD

Phase B - Toshiba QKDs and Ciena Waveserver Devices

Figure 3 shows Phase B: where Toshiba QKD System (Alice) communicates with Toshiba QKD System (Bob), and they enable the Waveserver 5_A and Waveserver 5_B to setup keys for the interconnect platform's Datapath encryption. (Datapath encryption details are beyond the scope of this document.)

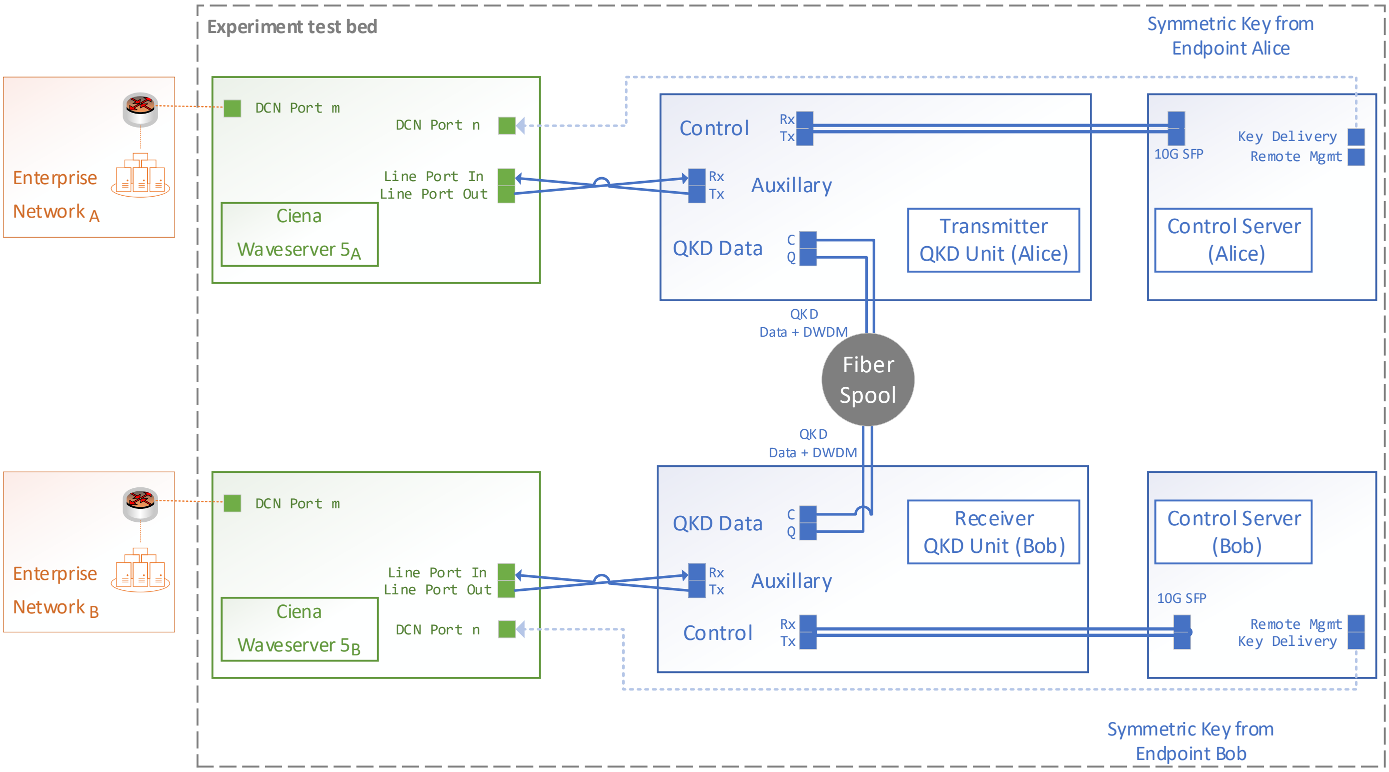

Figure 3 : Phase B Toshiba QKD with Ciena Waveserver 5

For this testbed, the Waveserver 5 had a Control Processor Subsystem (CPS)[4]. (The Waveserver device is also available with a FIPS 140-3 Level 2 compliance-capable Control Processor Subsystem.) The following table lists the Ciena devices added to the setup in Phase A.

Device label	Device Model	Function
Ciena Waveserver 5_A	Ciena Waveserver 5 OS 2.3.11	high-speed, point-to-point connectivity for Data Center Interconnect (DCI) applications
Ciena Waveserver 5_B	Ciena Waveserver 5 OS 2.3.11	high-speed, point-to-point connectivity for Data Center Interconnect (DCI) applications.

[4] Ciena, 2022, User Guide: Waveserver 5, 323-3001-100 - Standard Issue 1

Key Exchange in the Experiments

The key exchange process is the backbone for establishing a secure point-to-point connection, facilitating the availability of the keys. Following is a summary of the key exchange process between Ciena Waveserver (WS) and Toshiba QKD/Control Server units.

1. The Waveserver on Alice's side, using a ETSI based HTTPS request[5] to get a key from QKD Alice.
2. If TLS1.2 authentication is successful, Alice returns a key and its associated Key-ID (KID).
3. The KID is then sent over to the far-end WS (note the key itself is never transferred over the line only the ID), where it is used by the far-end WS to request the same key from Bob.
4. Once a key is successfully received, the new pre-shared key is applied on both ends.

With the current implementation, the Pre-Shared Key (PSK) received from the QKD systems is used in two separate instances. First, it is used as authentication material between the two Waveservers as part of Elliptic-curve Diffie–Hellman (ECDH) key exchange process. Second, it is used in combination with the shared secret from the ECDH process to generate the Advanced Encryption Standard with Galois Counter Mode (AES-256-GCM) ciphers to encrypt the data at the Optical Data Unit-Cn[6] (ODUCn) level.

The Waveserver 5 goes through the ECDH key exchange process every 1 second, with the shared secret refreshed every 1 second. In the experiment, we used the QKD European Telecommunications Standards Institute (ETSI) API, to request a new pre-shared key every 20 seconds (note that the 20s frequency is a configurable parameter).

Figure 4 illustrates how the Waveserver and QKD System work together to enable secure end-to-end data flow in the enterprise.

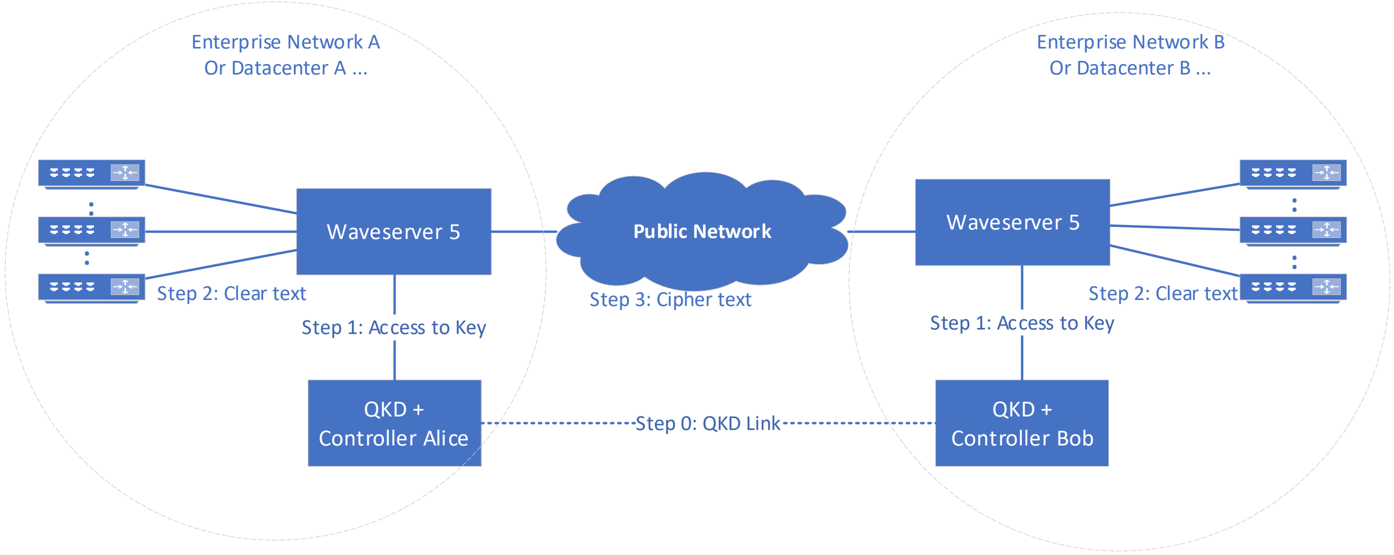

Figure 4 : End-to-End Secure Data Transfer

[5] ETSI GS QKD 014 V1.1.1 "QKD; Protocol and Data Format of REST-Based Key Delivery API" (2019)

[6] Cn indicates the bitrate of approximately n*100 Gbit/s

Key delivery using ETSI API

The European Telecommunications Standards Institute (ETSI) QKD Group specifications specify the communication protocol and data formation for a QKD network to make the cryptographic keys available to the applications. It is this group's specifications that enable the interoperability of the QKD networks and the entities. The ETSI 014 standard[7] defines the key delivery interface as using HTTPS. The ETSI QKD 014 specifications require mutual TLS (Transport Layer Security) authentication, i.e., the server and the client authenticate to each other. Below is the ETSI explanation for Secure Application Entity (SAE) and Key Management Entity (KME).

SAE (Secure Application Entity)	Applications such as encryptors are defined as SAE and are referred to as "client" because it issues the requests.
KME (Key Management Entity)	KMEs are defined as QKD transmitters / receivers.

In practice, the SAE client authenticates the KME server i.e., server authentication of the QKD Control Server. To enable key delivery mechanism, the certificates must be installed into the QKD system. Self-signed certificates were used for this experiment. Figure 5 illustrates a typical key delivery use case.

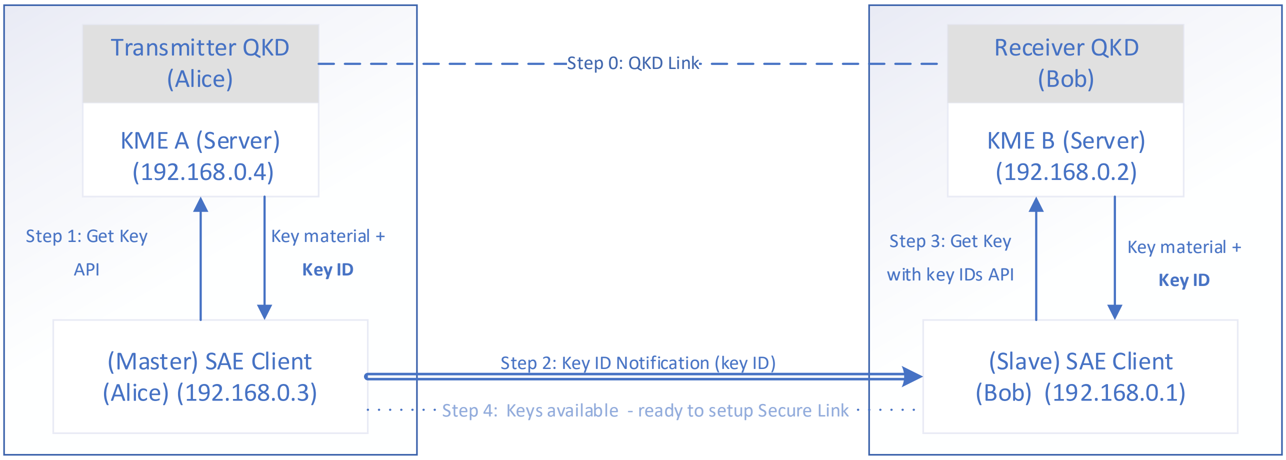
Transmitter QKD (Alice)
KME A (Server) (192.168.0.4)
Step 1: Get Key API
Key material + Key ID
(Master) SAE Client (Alice) (192.168.0.3)
Step 0: QKD Link
Step 2: Key ID Notification (key ID)
Step 4: Keys available - ready to setup Secure Link
Receiver QKD (Bob)
KME B (Server) (192.168.0.2)
Step 3: Get Key with key IDs API
Key material + Key ID
(Slave) SAE Client (Bob) (192.168.0.1)

Figure 5 : Key Delivery Use Case

Observations

Control and configuration of the QKD appliances was accomplished through the graphical user interface (GUI) remotely accessed by a host device via port-forwarding over SSH. The "qkduser" account was used to access and make changes to the network settings, view QKD activity log, and perform essential QKD functions.

Startup and Configuration

The GUI on the QKD devices was used to initiate the QKD start up process.

- The startup, alignment, and initial processes is kicked off by entering "localhost:8081" in the browser on the control server.
- After user confirmation to begin startup process, (status message shows "Starting"), the QKD system cycles through hardware initialization, auto-alignment, and QKD data processing.
- When alignment is complete, the QKD status changes to "Generating Key".
- Once the system is initiated and running, two metrics are displayed and updated in real time:
  - Secure Bit Rate (SBR)
  - Quantum Bit Error Rate (QBER)

[7] ETSI GS QKD 014 V1.1.1 "QKD; Protocol and Data Format of REST-Based Key Delivery API" (2019)

Results

While the Toshiba user manual indicated a typical secure key rate (aka SBR) of 300 kbps at 10dB loss, the observed secure key rate was well above that with a secure key rate of 726 kbps. The team attributed the increased secure key rate to the short distance between the endpoints and the resulting reduced channel transmission loss.

The quantum bit error rate (QBER) is a ratio of the error rate to the secure key rate i.e., it is the measure of the percentage of bit error across the quantum channel during the key distribution. The QBER is used to detect the occurrence of eavesdropping. For example, when an eavesdropper (designed “Eve”) is using the intercept/resend attack on an average the QBER is 25%. During the experiments, QBER was observed to be 3.22% ± 0.2%.

Key delivery between Toshiba devices

To validate the keys are generated, the REST method ‘enc_keys’ was issued from endpoint Bob. The cURL application is used to retrieve the key using the ‘Get key’ command with *192.168.0.4* as the KME_hostname and *bobsae* as the slave_SAE_ID. The calling master SAE supplied ‘Key request’ data to retrieve key_ID for the slave_SAE_ID.

qkduser@bob	
192.168.0.2	Parameter: KME_hostname (KME B).
alicesae	Parameter: slave_SAE_ID.
enc_keys	REST method to pull the key (s).
--ca_cert certs/ca_crt.pem --key certs/client_bob_key.pem --cert certs/client_bob_crt.pem	Request data model with KME B as the calling Master SAE over to the slave SAE at Alice. ca_crt.pem is the self-signed CA (Certificate Authority) certificate client_bob_key.pem is the key that KME B uses to prove it is permitted to use client_bob_crt.pem. client_bob_crt.pem is the certificate for the KME B signed by CA that issued ca_cert.pem.

{“keys”: [{
“key_ID”: “54ad..6d17”,
“key”: “….”
}]
}
	Response data model is the ‘key container’ from KME (‘192.168.0.2’) to SAE (‘alicesae’).
	The ‘key container’ is held in the JSON object – it contains the actual key as the value ‘key’.

Successful execution of the ‘enc_keys’ method indicated that the QKD devices at Alice and Bob were in sync. The symmetric key to connect with Alice was now available at the Bob endpoint.

Key delivery to Waveserver 5

The following python script was executed using the “diag” user account on the Ciena Waveserver 5 device to retrieve the shared key at end point Bob.

```
import requests
r = requests.get(“https://192.168.0.2/api/v1/keys/alicesae/enc_keys?number=1&size=256”,
verify='ca_cert.pem',cert=('client_bob_key.pem', 'client_bob_crt.pem'))
import json
key_repsonse = r.json()
key_response
```

Retrieving a Key ID and the relevant key indicated that the Waveserver 5 had received the key being generated by the QKD system Bob. The same steps were executed at the Waveserver 5 connected to QKD system Alice.

As at endpoint Bob, retrieving a Key ID and the relevant key indicated that the Waveserver 5 at Alice had received the key generated by the QKD system Alice.

At this stage, the key material at Alice and Bob did not match because the Key IDs are different. This demonstrated that the actual key was never sent over the wire between Waveserver devices. Rather, it is the Key ID retrieved by the Waveserver 5 at Bob which is sent to the Waveserver at Alice. The Waveserver at Alice then sends the received Key ID to the Alice KME server to retrieve the corresponding key. The ETSI REST API method “dec_keys” was used to retrieve the matching key by specifying the same Key ID at the two end points.

Conclusion

The experiment successfully demonstrated the ability to integrate and interconnect products and technologies from multiple vendors to achieve a fully operational QKD implementation. This validated the secure generation, sharing and distribution of the cryptographic keys between endpoints, facilitating quantum-resistant secure delivery of the symmetric keys and subsequent secure communications.

It's important to acknowledge that, while QKD provides crucial elements of a quantum-resistant data protection ecosystem, factors such as distance limitations, specialized hardware requirements, authentication requirements, and vulnerability to Denial-of-Service attacks need to be considered when designing and deploying a fully integrated, quantum-resistant architecture. In the broader scope of a fully integrated, quantum-resistant architecture, QKD serves as just one layer in the PQC stack.

This experiment is one of a series of PoTs designed to explore, research, develop, and asses quantum-resistant security technologies needed to protect Wells Fargo and the financial ecosystem in the quantum age.