Quantum Key Distribution-bootstrapped Authentication for Secure Communication of Distributed Energy Resources

M. Alshowkan*, Member, IEEE, P. G. Evans, Member, IEEE, M. Starke, Senior Member, IEEE, D. Earl, Member, IEEE, and N. A. Peters, Senior Member, IEEE

Abstract—In this work, the first quantum key-bootstrapped authentication for smart grid communications on an electric utility fiber network is demonstrated. The developed method was prototyped in a software package to manage and utilize quantum keys from a quantum key distribution (QKD) system to authenticate machine-to-machine communications used for supervisory control and data acquisition (SCADA). This demonstration showcases the feasibility of using QKD to improve the security of critical infrastructure, including future distributed energy resources (DERs), such as energy storage.

Index Terms—Distributed energy resources, SCADA, Cybersecurity, authentication, electrical substation, networking, quantum key distribution, smart grid.

I. INTRODUCTION

The electric grid is evolving from an electrical network composed primarily of large centralized fossil fuel plants to more distributed renewable and energy storage type plants. Wind, photovoltaic (PV), and energy storage systems (ES) technologies have observed significant cost reductions as the technologies have continued to mature and reach mass production [1], [2], [3]. These technologies are now being adopted more frequently into the electric grid, both in large and small deployments.

Renewable power plants can now be found in the scale of hundreds of megawatts (MWs) to gigawatts (GWs) of potential power generation [4]. These generation plants are a composite of many small generation resources, all interconnected with an electrical network known as a collector system [5], [6], [7]. An example layout for a PV plant with a supplementary ES system is shown in Fig. 1. At each resource within the power plant, power electronic converter (PEC) systems with intelligent controllers are used to perform conversion and control of the power produced by both the PV modules and ES technology. These systems support several different operational modes and communications protocols via an integrated communications module. System coordination is performed through a plant supervisory control and data acquisition (SCADA) system.

As presented in [11], SCADA systems have been the target of many attacks that can impact a reliable and fast communications network. These attacks include eavesdropping, man-in-the-middle, masquerade, virus and worms, trojan horses, and denial-of-service. These attacks have targeted the various levels of SCADA networks including the application layer, session layer, network transport layer, data link layer, and physical layers, with varying success rates. Therefore, electric utilities and generation plants are applying many different approaches to secure the information flow. These methods...
include adopting considerations of privacy/confidentiality, integrity, authentication, and trusted computing [12], [13], [11].

Solutions for ensuring confidentiality and integrity of the communicated data include the utilization of encryption and authentication algorithms. Both encryption and authentication schemes use cryptographic algorithms and secret keys, however, the two general schemes are different: encryption converts a message plaintext to ciphertext to protect the information, whereas authentication is the attribute of confirming a message is genuine and has not been altered during transmission.

Currently, many popular cryptographic solutions, such as public-key cryptography, are based on hard-to-solve mathematics using assumptions based on potentially available computing resources [14], [15]. One of the major advantages of public-key cryptography is enabling messages to be encrypted and/or authenticated with a “public” key (i.e., known to all) which in turn can only be decrypted and/or signed with a “private” key (i.e., kept secret). The generation of the public-private key pair leverages the aforementioned mathematics. To continually improve security of this type of cryptography, the secret key size must increase with available computational capabilities [16]. This can be a challenge for devices deployed in the field as availability of computational resources (i.e., memory size and processing capability) is typically fixed. Hence, without detrimentally increasing latency or potentially being put out of service—as the processing demand increases—devices in the field must be replaced [17], [18].

In contrast, private-key cryptography—where a single key performs both encryption and decryption tasks—can be implemented very efficiently in hardware [19], while exhibiting low computational overhead with deterministic latency. However, the challenge is all keys must be securely distributed to all parties prior to use, typically by a trusted courier, resulting in all keys being at risk of discovery during transit. From this perspective, quantum key distribution (QKD) approaches offer considerable promise: keys for private-key cryptography schemes can be established between parties—even over communication channels controlled by an adversary—in a provably secure manner [20].

Arguably, QKD is one of the most mature quantum applications available [15]. The fundamental technology has already been observed to be transitioning from research laboratories to commercial products [21], [22]. Combined with information-theoretic security protocols [23], QKD offers future-proof security: proven to be safe regardless of the technological development in computing, quantum or otherwise [15].

Securing a simulated power grid communications network using QKD was presented in [24] and using real time digital simulator (RTDS) microgrid testbed in [25] while theoretical approaches to improve the power grid physical security using quantum computing was explored in [26]. Recently, QKD has been applied in a trusted relay [27], [28], [29], [30], [31], [32], [33] as well as a fiber loop-back on a utility network [34]. Following the initial utility demonstration, a four-node QKD trusted relay network on a utility fiber infrastructure showed the interoperability between diverse QKD systems that worked together to deliver secure keys across the critical energy infrastructure [35].

In this paper, QKD keys are used to authenticate communications of integrated power electronics energy resources in electric grid infrastructure. This work is the first time quantum keys have been used to authenticate smart grid communications. More specifically, (a) QKD has been applied over an internet of things protocol Message Queuing Telemetry Transport (MQTT) for supporting DER communications, (b) the developed software design pulls secret keys from a commercial Qubitekk quantum key distribution system to authenticate machine-to-machine (M2M) communications, and (c) the platform has been applied in a real utility setting. This paper is organized as follows: Section II presents a short background on quantum key distribution; Section III presents a proposed communications and control approach for distributed energy resources to support QKD application; Section IV discussed the QKD approach; Section V the experimental setup; Section VI the experimental results, and Section VII the conclusions.

II. BACKGROUND ON QUANTUM KEY DISTRIBUTION

In this section, a background on quantum key distribution is presented along with the requirements and importance of the system application specifics for the electric grid.

A. Quantum Key Distribution

Quantum key distribution relies on three of the main tenets of quantum physics: the uncertainty principle, the effect of measurements on quantum states, and the no-cloning theorem. The uncertainty principle prevents conjugate variables (e.g., position and momentum, energy and time, or amplitude and phase) of a quantum state from being measured with absolute precision—there will always be a minimum amount of error present. Performing such measurements upon a quantum state irreversibly modifies the quantum state during the measurement process. Finally, it is impossible to create an identical copy of an unknown quantum state without causing errors in the state. Furthermore, in a uniquely quantum mechanical aspect, it is possible to entangle quantum particles (in our case photons) such that the state is only described by its joint properties, each object on its own carries no local information on its state. Measurement on one of the particles modifies the quantum wavefunction of the entangled state, leading to definitive outcome on the other particle. Photons may be described in terms of their polarization, energy (wavelength), creation time, or spatial mode. Thanks to modern optical communications technology, one has the ability to encode quantum information in these photonic degrees of freedom.

Quantum Key Distribution describes a variety of techniques whereby quantum states are used to establish a shared random key between two spatially separated parties, commonly referred to as Alice and Bob in cryptographic parlance. BB84 [36] is the most well-known QKD protocol, yet others exist which leverage different encoding schemes [37], [38] as well as entanglement [39]. QKD is not a cryptographic mechanism—it is a method to distribute correlated random bit strings for later use in any application, including well-known symmetric cryptography schemes such as the Advanced Encryption Standard (AES), Blowfish, and others.
The commercial QKD system implemented an entanglement-based protocol [39]. In the specific Qubits ek system implementation, a polarization entangled photon pair is generated from spontaneous parametric downconversion, a nonlinear optical process where a pump photon weakly decays into two daughter photons. One photon is immediately measured in a random polarization basis; either horizontal/vertical basis (immediately measured in a random polarization basis; either horizontal/vertical basis, or diagonal/anti-diagonal basis \((D) = (|H⟩ + |V⟩)/\sqrt{2}, |A⟩ = (|H⟩ - |V⟩)/\sqrt{2})\), resulting in four possible and equally likely measurement outcomes \((|H⟩, |V⟩, |D⟩, \text{or} |A⟩)\). The second photon is sent across a fiber link, where it undergoes a measurement similar to the first photon, again resulting in four possible and equally likely measurement outcomes. For this particular entangled photon source, the state \((|HV⟩ - |VH⟩)/\sqrt{2}\), where the first photon’s polarization state is labeled with the leftmost polarization position and second photon is labeled immediately next. So, if the first photon is detected as \(|H⟩\), the second will be detected as \(|V⟩\), and vice versa. Likewise, whenever diagonal (anti-diagonal) is measured on photon one, the second photon will be measured as anti-diagonal (diagonal) due to the entanglement correlations. Since the basis of measurement is randomly chosen at the transmitter and receiver, they will only match half of the time, yielding “sifted” bits, which then must undergo error correction and privacy amplification to generate final, secure key material.

The presence of an eavesdropper (Eve) within the quantum channel is revealed during the error correction phase. Recall the essence of quantum mechanics: Eve cannot measure an unknown quantum state with precision, nor can she create an identical copy of it. Thus, attempts by Eve to intercept-and-resend single photons will result in larger than expected quantum bit error rates, which will be corrected and privacy amplified, reducing the size of a final key. If the error rate is too high, no final key may be distilled, which is equivalent to a denial of service. As a result, Eve gains no information of the final secret key, and the presence of Eve is revealed by the excess quantum bit error rate (QBER) metric [40].

**B. Requirements for QKD in an electrical network**

While it would be possible to deploy free-space terminals to perform QKD, the availability of fiber optic infrastructure makes for a much more convenient alternative, as one does not need to worry about objects (such as inclement weather) blocking the communications path. The presence of much higher power classical optical signals on the same fiber as the quantum signals, can introduce a considerable amount of noise in the quantum channel [41], [42]. As a result, it is highly advantageous to utilize a “dark” fiber dedicated to the quantum signal alone.

Fortunately, many utilities— as part of the power grid modernization— have been heavily investing in information technology, including deploying optical fibers between operations centers, substations, and the distributed energy resources. The investment in fiber offers the utility companies considerable bandwidth, which can be partially leased, as well as flexibility for grid operational communications [27].

The North American Electric Reliability Corporation (NERC) issued a set of Critical Infrastructure Protection (CIP) reliability standards [43] to ensure the security of Bulk Electric System (BES). The Physical Security Perimeter (PSP) standard (CIP-006-6) defines a physical security plan to safeguard BES cyber systems against any compromise that might cause improper BES behavior. For this reason, PSP access control requirements include key card access, special locks, security personnel, and authentication devices such as biometrics and tokens. Also, the standard outlines methods to monitor and log the physical access using alarm systems, human observation, computerized logging, and video surveillance and recording.

Typically, the connectivity between a utility SCADA system and devices is the current widespread networking suite called Transmission Control Protocol/Internet Protocol (TCP/IP) [44], [45]. While all communications for the electrical system must be trustworthy [46] and timely [47], transmitting data via a TCP/IP protocol is susceptible to cyber attacks including spoofing [18]. Such attacks include injecting malicious data during transmission that may result in poor control responses and outages could occur. For this reason, electrical systems connected to the internet are vulnerable to cyber-attacks [48], [47].

**C. Importance of QKD field evaluation on the electric grid**

Authentication of data and control messages is crucial for reliable, safe, and secure grid operations. Using an authentication protocol and secret keys known only to the sender and the receiver enable bi-directional message authentication. Moreover, an information-theoretic (meaning security is not based upon computing resource assumptions) authentication protocol based on private-key encryption comes without the latency penalty of public-key cryptosystems [15], [16]. For example, using the Carter-Wegman [49] authentication protocol requires fewer computational resources and thus provides a long-lasting and more resource-efficient authentication compared to the asymmetric public-key-based authentication protocols [17]. Demonstrating QKD technology in a real-world environment to verify the feasibility of quantum-based cybersecurity for power grid communications is a crucial way point towards wider adoption. A controlled laboratory setup dramatically reduces environmental impacts compared to field deployments. For example, environmental variables such as temperature and humidity, in addition to the electromagnetic emanations of specialized power equipment, can affect the quantum hardware, including optics, electronics, and electro-optics. Further, the fiber optic deployment mechanism in a real-world environment is another vital element to consider. The QKD key rate of an underground and aerial fiber will likely be affected in some QKD implementations and may require additional equipment/engineering compared to lab-based demonstrations.

**D. Message encryption and authentication in QKD - Galois message authentication**

Cyberattacks seeking to disrupt grid communications can have devastating consequences for grid operations. Therefore,
it is crucial to verify that the grid communications have originated from the authorized user. One way to authenticate information in transfer over a network is by employing an authenticator that can be used as a challenge to verify the authenticity of a message. Several methods are available to produce a message authenticator: message encryption, hash functions, or a message authentication code (MAC). Message encryption uses symmetric or asymmetric cryptographic algorithms. In light of the latency issues evident with public-key cryptography as outlined earlier, QKD-bootstrapped symmetric-key cryptography offers an attractive solution for the long-term security of grid communications. Message encryption hides information, and only users who know the secret key can encrypt and decrypt a message; for this reason, it also ensures the authenticity of the message. For power grid communications, the information in transit contains measurement data that needs to be examined if abnormalities are to be detected. As a result, authentication is preferable to encryption as data is usable even if there is a problem (such as a delay) with the authentication cryptographic operations. Hash functions map a variable-length message to a fixed-length hash value that can be used as an authenticator. On the other hand, a message authentication code (MAC) is similar to a hash function but uses a secret key for mapping. While hash functions guarantee the integrity of a message, a message authentication code using a secret key guarantees message integrity and authenticity.

The concept of provably secure authentication was introduced in [50] using a secret key that is longer than the message itself. Carter and Wegman showed it is possible to use a secret key shorter than the message to achieve information-theoretic authentication [51]. Later, using a block cipher, it was shown by Brassard that a shorter secret key can be expanded and used for the Carter-Wegman authentication scheme [52]. Galois/Counter Mode (GCM) is a state-of-the-art parallelizable symmetric-key cryptographic protocol based on the Carter-Wegman authentication scheme [53]; it offers information-theoretic encryption and authentication. The Galois Message Authentication Code (GMAC) is the GCM standalone authentication scheme, i.e., where the message does not need to be encrypted. The National Institute of Standards and Technology (NIST) approved GCM and GMAC as standards in 2007 via NIST SP 800-38D [54].

There are three inputs to the GMAC: (1) the message to be authenticated, (2) an initialization vector (IV), also referred to as a nonce, and (3) a secret key. The output is the message authentication code (MAC). As expected in symmetric-key algorithms, GMAC assumes a fundamentally secure key exchange between the sender and the receiver. GMAC allows reusing a secret key to authenticate more than one message; however, it prohibits using it with the same nonce [54]. Currently, the acceptable block ciphers recommended by NIST are AES-128, AES-192, and AES-256 [55]. For the nonce, the acceptable size is 96 and 128-bits. The length of the output message authentication code is 128 bits. The authentication process is initiated by a sender (Alice) who wants to send an authenticated message to a receiver (Bob). A new secret key, a nonce, and the original message are then supplied to the GMAC, which outputs the message authentication code. Alice sends the original message, the nonce, and the MAC to the Bob, but keeps the secret key a secret. Upon receipt, Bob then forwards Alice’s message, nonce, and MAC along with the corresponding secret key to the GCM verification algorithm, whose output is a simple statement: true if the message is authentic or false if not.

III. PROPOSED COMMUNICATION AND CONTROL ARCHITECTURE

In this work, the concept of operations is the communications between a single photovoltaic (PV) system and a SCADA system. In the following sections, a generalized architecture for supporting the quantum-bootstrapped authentication demonstration is discussed.

A. Photovoltaic and Energy Storage System Architecture

The integration of a power electronic controller (PEC) and integrated resource to construct a distributed energy resource (DER) can be performed through a multiple vendor “black-box” integration effort [56], [57]. The “black-box” designation signifies that only a communications interface to the system is present. This work proposes an architecture that utilizes an integration layer (or coordination controller) to couple systems and providers, as shown in Fig. 2. The proposed coordination controller can be placed directly within the hardware system and provides an opportunity to automatically enable QKD systems to be applied to many different PEC-type resources.

The coordination controller has been developed as a means to integrate many types of PECs and resources. The design utilizes a multi-agent architecture comprised of four agents: converter, source/load, interface, and intelligence. The Converter Agent interacts with the PEC then shares the status and data over a local messaging bus. The Source/load Agent interacts with the source/load then transmits data which includes control and status, with other agents. The Interface Agent interacts with the external agents to send and receive information, then relays the information to the local agents over the local message bus. Finally, the Intelligence Agent interacts with the interface agent to convert requested control signals into actionable signals for the separate resources. All
Moreover, we also assume that each user has a set of random Alice and secret keys with even serial numbers \( k \) presented in [60].

commissioning sequence through a registration process is coordination controller subscribes to control data published

lined as follows: the SCADA subscribes to measurement data SCADA system and the DER coordination controller is out-

topic-based filtering—in an event-based manner.

For this reason, the broker activities can be parallelized—using

published message must contain a required topic—that clients

ker, who then relays them to the interested subscribers. Each

The publisher role enables a client to send messages to the bro-

approach scales more effectively than the typical client-server

all incoming traffic and appropriately deliver messages to

broker's responsibility is to process all incoming traffic and appropriately deliver messages to the intended subscribers. As a result, this communications approach scales more effectively than the typical client-server architecture. An MQTT client can be a publisher or subscriber. The publisher role enables a client to send messages to the bro-

who then relays them to the interested subscribers. Each published message must contain a required topic—that clients subscribe to it’s relevant messages—and an optional payload. For this reason, the broker activities can be parallelized—using topic-based filtering—in an event-based manner.

In this work, an MQTT messaging approach between the SCADA and the DER coordination controller is out-

TABLE I

| Publisher   | Subscriber | Data |
|-------------|------------|------|
| SCADA       | Coordination Controller | Topic: PV/Control Payload: Control Information |
| Coordination Controller | SCADA | Topic: PV/Measurement Payload: Measurement Information |

IV. APPLIED QKD APPROACH

Assuming Alice and Bob share a set of QKD-based secret keys \( k_1, \ldots, k_n \) where \( n \) is an arbitrary serial number for each key. To guarantee that each secret key is used by only one user, we assign secret keys with odd serial numbers \( k_{odd} \) to Alice and secret keys with even serial numbers \( k_{even} \) to Bob. Moreover, we also assume that each user has a set of random initialization vectors \( iv_1, \ldots, iv_j \) privately generated from a quantum random number generator (QRNG) where \( j \) is an arbitrary serial number for each \( iv \). To publish an authenticated message \( m \) and its topic \( t \) using a secret key \( k_n \), the secret key serial number \( n \) used in this process need to be transmitted to indicate to the receiver which key was used (without disclosing any information about the key itself). In our case, we choose to set the key serial number to be part of the overall message to be authenticated. To avoid replay-attacks, an authenticated timestamp \( ts \) is used. Therefore, the total message \( tm_i \) where \( i \) is the number of the message and it’s related topic to be authenticated becomes:

\[
 tm_i = m_i + t_i + n + ts
\] (1)

In our software, we utilized the MQTT built-in publish() callback function to create the specific message authentication code from the sender \( mac_S \) for each \( tm_i \) being published using the GMAC encryption \( GMAC_E \) algorithm such that:

\[
 mac_S = GMAC_E(tm_i, k_n, iv_j)
\] (2)

where the total message \( tm_i \), secret key \( k_n \), and initialization vector \( iv_j \) inputs to the GMAC algorithm, and \( mac_S \) is a 16-byte string uniquely associated with the inputs. Once \( k_n \) and \( iv_j \) are retrieved for use with the GMAC, they are immediately flagged as used. To verify the authenticity of the \( tm_i \), Alice needs to share the \( mac_S \) and the initialization vector \( iv_j \) with Bob while keeping the secret key \( k_n \) secret. Thus, the payload \( p \) of each message being published becomes

\[
 p = tm_i + iv_j + mac_S
\] (3)

While the payload of a standard MQTT message contains only the message data \( m \), we employed a delimiting character between the components of the total message \( tm_i \) to construct the new payload (e.g. using dashes, \( m_i - t_i - n - ts - iv_j - mac_S \)) for convenient payload coding and decoding. We utilized the MQTT on_message() callback function to verify every received message. For each received message, we use the delimiting character to break the payload data—to retrieve all the components of the total message \( tm_i \)—and start the verification process. First, we verify using \( k_n \) that the secret key has not been used before. Second, comparing the last used \( k_n \) and the \( ts \), we verify that the message is not delayed or replayed by considering the typically expected delays in the network \( \delta \). While the \( ts \) will depend on classical network synchronization (e.g., Precision Time Protocol and Network Time Protocol), any anomalies detected in timing between the nodes will trigger further investigation. Third, we use the message topic and verify it is equal to the topic embedded in the \( tm_i \). Fourth, using the received \( tm_i, iv_j, mac_S \), and the corresponding \( k_n \), the receiver performs the verification GMAC decryption \( GMAC_D \) as follows:

\[
 mac_R = GMAC_D(tm_i, k_n, iv_j)
\] (4)

Bob compares the received 16-byte \( mac_S \) and the calculated \( mac_R \). If both match, then \( tm_i \) and subsequently the original
message $m_i$ are authentic, otherwise the authenticity cannot be established for this message, and further investigation is warranted. Upon successful verification, Bob flags the received key as used. Algorithm 1 and Algorithm 2 summarize the process of creating and verifying the MAC, respectively.

V. QKD-BITSTRAPPED AUTHENTICATION FOR GRID COMMUNICATIONS

We demonstrate the above QKD-enabled MQTT approach in a real-world electrical utility environment at the Electric Power Board (EPB) [67], Chattanooga, Tennessee. Two optical fibers are used to create a dedicated quantum communications link between a distribution center (DC) and an electrical substation (SUB). A Qubitekk [22] Industrial Control Systems (ICS) commercial QKD system was used for this demonstration. The link distance between DC and SUB was approximately 3.4 km, and exhibits an optical attenuation of 1.3 dB at 1550 nm, including patch panel connectors and splices. While the dedicated optical fiber is bundled with many other optical fibers used for utility operations, we note the quantum communications link, and all other classical communications links used for this work, are isolated from EPB’s operational network. This isolation is good practice for testing experimental technologies in operating power grid infrastructure. In this network, the bulk of the optical fiber link is deployed aerially between utility poles, and hence experiences environmental variables such as temperature changes and wind motion. This in turn has a slight effect on the quantum key generation rates, as would be expected with polarization encoded photons utilized by the Qubitekk system. In addition to the dedicated quantum optical fiber links, we also establish a typical TCP/IP local area network for the corresponding classical channels between virtual distributed energy storage systems located at DC and SUB.

A. Network Configuration

The QKD hardware was deployed in the utility distribution center (DC) and substation (SUB). At each location, a virtual distributed energy storage (vDES) machine on Raspberry Pi 3 B+ was deployed. Thus, we set up the Intelligence (Intel) Agent machine in DC and the Photovoltaic (PV) Agent machine in SUB. We connected each system to the private classical network via a network switch (see Fig. 3). Additionally, we configured two other supporting devices in the same network. One acts as a server to collect the network statistics in DC; the other device is in the substation and used for administration tasks including control and data monitoring.

B. QKD-BOOSTRAPPED AUTHENTICATION SOFTWARE

We developed software that handles network node secret key operations, including retrieving, verifying, and managing the QKD-generated key material. The key material is then used to authenticate the vDES communications. While performing these operations, each node is responsible for tracking and reporting statistics related to the key materials and the completed tasks. Communications in this network follow a publish-subscribe architecture; when the software starts, the receive node verifies the authenticity of the transmit node; then subscribes to topics of interest. In what follows, we will describe the basic functionalities of the network nodes.

1) Secret key management: Python-based software runs a background service in each device, retrieving secret keys from each QKD system over a serial cable. As the keys become available, the background software stores them in a local file. The software has two possible file operations: append to an existing secret key file or create a new one. If a secret key file exists in the directory, the background service appends the new keys to the existing file; otherwise, it creates a new one. Fig. 4 shows the format of the key materials retrieved from the vDES communications link.

Algorithm 1: Create MAC for each outgoing message

| Input: message, topic |
|-----------------------|
| Output: payload       |
| Function publish(message): |
| $m$ ← message         |
| $t$ ← topic           |
| $n$ ← number of next secret key |
| $ts$ ← timestamp      |
| $tm$ ← $m + t + n + ts$ |
| key ← the nth secret key from key table |
| $iv$ ← next random number from QRNG |
| $mac_S$ ← GMAC$_E$(tm, key, iv) |
| $p$ ← $tm + iv + mac_S$ |
| return $p$             |
| End Function           |

Algorithm 2: Verify MAC for each incoming message

| Input: payload, topic |
|-----------------------|
| Output: True/False    |
| Function on_message(payload, topic): |
| foreach $p$ in payload do |
| $tm$ ← $p[1]$         |
| $iv$ ← $p[2]$         |
| $mac_S$ ← $p[3]$      |
| end                   |
| foreach $q$ in $tm$ do |
| $m$ ← $q[1]$          |
| $t$ ← $q[2]$          |
| $n$ ← $q[3]$          |
| $ts$ ← $q[4]$         |
| end                   |
| $ct$ ← current time   |
| key ← the nth secret key from key table |
| $mac_R$ ← GMAC$_D$(tm, key, iv) |
| if $mac_S = mac_R$ & & $t == topic$ & & $ts < ct - \delta$ then |
| $result$ ← True       |
| else                  |
| $result$ ← False      |
| end                   |
| return $result$       |
| End Function           |
the QKD system and stored in the local file. One functionality of the software we have developed and integrated into the vDES system is to periodically monitor and retrieve new key materials from the local file as they become available. Next, the software checks every received key for the appropriate length: to avoid run-time errors caused by inadequate key length. In our case, we verify the key of length 32 bytes (256-bit) as a valid key. Finally, the software creates a record for each key, including a serial identification number and a Boolean status flag indicating the used and unused secret keys. After this point, each node should have an identical key table to use for secure communications.

2) Random number management: Like secret key management, each node has access to a list of local (initially private) random numbers generated from a quantum random number generator (QRNG) to use as initialization vectors. In our case, we use random numbers generated from a commercial IDQ QRNG [21]. The QRNG outputs a large string of random numbers that we chunk into smaller strings, each of length 16 bytes (128-bit) which the authentication algorithm accepts. Because these random numbers do not need to be identical between the network nodes, each node manages them locally. When a node plans to create a new MAC for a message, it retrieves a random number from the local list and sets its flag to “used”, so it never gets used again.

3) Authentication and verification: Authentication and verification are the core of the software described in the previous section. The software is called when the vDES systems want to publish a message to create its MAC. The original message gets appended with the MAC and other supporting information to enable a receiver that shares secret keys with the sender to verify the message’s authenticity. Additionally, the software asserts other security measures to prevent replay and delay attacks. For this reason, the timestamp, message topic, and secret key serial identification number are also set to be authenticated and verified by the receiver. Thus, a received message gets verified against replay and delay attacks. For example, the software verifies a timely message receipt by tracking the last secret key used, confirming the expected behavior of message sequence, in addition to checking the timestamps.

4) Statistics reporting: Similarly, information related to the random numbers, including those added, available, and used, is also reported. For example, statistics of created message authentication codes include the identification of the last key used. Additionally, the verification algorithm reports the number of successful and unsuccessful message verification instances.

VI. RESULTS

When the PV and Intel agents start, they proceed to perform the secret key and random number management described in section V. Then, they initialize a set of global variable objects of various classes needed to support the communications, interfaces, and measurements. Additionally, if enabled, they set the graphical user interface (GUI) parameters. After the initialization step, each agent requests to connect to the broker using the broker IP address and port number—the default MQTT port number is 1883—over the TCP/IP protocol. A successful connect request by an agent is responded with a connect acknowledgment flag by the broker. Individually,
each agent informs the broker about the list of topics they are interested in and each topic’s quality of service (QoS). The QoS indicates the level of reliability required based on the network and the application requirements. QoS 0 indicates a best-effort service—delivery is not guaranteed—a published message is transmitted to a subscriber once, and no acknowledgment of delivery is required. In QoS 1, a published message is generated to be delivered at least once. Therefore, an acknowledgment flag is required from the subscriber to confirm the delivery, or re-transmission of the same message is triggered: lost acknowledgment flags trigger re-transmission of previously delivered messages. Using a four-way handshake, QoS 2 guarantees that a message is published and delivered to a subscriber exactly once: it ensures that no duplicate messages are sent to the same client. Hereafter, agents in the vDES software are connected to the broker and subscribed to each other’s topics. Consequently, their published messages are authenticated and verified using the QKD secret keys. The published messages between the agents include slow and fast local periodic messages. The Intel agent publishes information about the system type, configuration, and forecast in the slow periodic messages, while the PV agent publishes configuration and forecast. On the other hand, in the fast period local messages, the Intel agent publishes information about the system types, measurements, operation status, errors, and faults, whereas the PV agent publishes the system status, measurements, and errors.

We collect data related to the number of added and available keys for each agent. Fig. 5 and Fig. 6 show the number of added keys for the Intel and the PV agent, respectively, as a function of time during the demonstration. The number of added keys is reported by the agent every 5 seconds—to conserve processing time required to check more frequently. If a more frequent key material update was used, both figures would be identical. Further, to avoid one node using a key that is not yet polled by the other node—due to polling delay, synchronization delay, or network disruption—we set a lower threshold $T$ on the number of keys each node keeps as a reserve pool. In this work, we set $T = 30$ as the minimum number of keys each node must keep before using a new key. Hence, we keep using the last known secret key—always with a new initialization vector—until the threshold $T > 30$, then a new key is used. The key reuse typically lasts for approximately 5 seconds until the subsequent key poll is complete. Fig. 7 and Fig. 8 show the number of available secret keys at each agent as a function of time. Each agent starts collecting keys from the QKD system before starting the energy storage system communications. When the agents start communicating, a reservoir of approximately 950 keys are available in the secret key file. Then, each begins authenticating their received messages using odd (or even) key identification number for the Intel (PV) agent. Fig. 7 and Fig. 8 shows comparatively slower key consumption by the Intel agent compared to the PV agent. This slower consumption is due to their functional differences resulting in a difference in the rate of sent messages.

![Fig. 5. Number keys added to the Intel Agent system (polled every 5 seconds).](image)

![Fig. 6. Number keys added to the PV agent system (polled every 5 seconds).](image)

![Fig. 7. The number of keys available to the intelligence agent. The inset figure shows the minimum keys available each agent must maintain (30 keys, which we chose arbitrarily) to keep the secret keys synchronized.](image)

VII. CONCLUSION

We have presented the first quantum key-based authentication of smart grid communications across an energy delivery infrastructure environment. Our system utilized a flexible and scalable smart grid communications protocol utilizing a publish-subscribe method. Further, we utilized keys from a commercial Qubitekk quantum key distribution system along with Carter-Wegman authentication protocol, which is in principle information-theoretic secure. With this demonstration, we show that quantum and classical security technologies can be integrated together in a way that can work in the energy
infrastructure to authenticate its data and control communications, providing long-term security, capable of exceeding the expected infrastructure service life.

ACKNOWLEDGMENT

We acknowledge the support of Steve Morrison, Tyler Morgan, and Ken Jones and Patrick Swingle. This work was performed at Oak Ridge National Laboratory (ORNL). ORNL is managed by UT-Battelle, LLC, under Contract No. DE-AC05-00OR22725 for the DOE. We acknowledge support from the DOE Office of Cybersecurity Energy Security and Emergency Response (CESER) through the Cybersecurity for Energy Delivery Systems (CEDS) program.

REFERENCES

[1] M. Yao and X. Cai, “An overview of the photovoltaic industry status and perspective in china,” IEEE Access, vol. 7, pp. 181 051–181 060, 2019. [Online]. Available: https://doi.org/10.1109/access.2019.2959399

[2] W. J. Cole, C. Marcy, V. K. Krishnan, and R. Margolis, “Utility-scale lithium-ion storage cost projections for use in capacity expansion models,” in 2016 North American Power Symposium (NAPS). IEEE, Sep. 2016. [Online]. Available: https://doi.org/10.1109/naps.2016.7747866

[3] M. Stecca, L. R. Elizondo, T. B. Soeiro, P. Bauer, and P. Palensky, “A comprehensive review of the integration of battery energy storage systems into distribution networks,” IEEE Open Journal of the Industrial Electronics Society, pp. 1–1, 2020. [Online]. Available: https://doi.org/10.1109/ojetes.2020.2981832

[4] Q. Aksuqarz, “Texas Renewable Energy Policy Sets an Example for the World,” 2019. [Online]. Available: https://spectrum.ieee.org/news-from-around-ieee/the-institute/ieee-member-news/texas-renewable-energy-policy-sets-an-example-for-the-world

[5] E. Camm, M. R. Behnke, O. Bolado, M. Bollen, M. Bradt, C. Brooks, W. Dilling, M. Edds, W. J. Hejdak, D. Houseman, S. Klein, F. Li, J. Li, P. Maibauch, T. Nicolai, J. Patino, S. Pasupulati, N. Samaan, S. Saylors, T. Siebert, T. Smith, M. Starke, and R. Walling, “Wind power plant substation and collector system redundacy, reliability, and economics,” in 2009 IEEE Power & Energy Society General Meeting. IEEE, Jul. 2009. [Online]. Available: https://doi.org/10.1109/pes.2009.5275333

[7] “IEEE guide for solar power plant grounding for personnel protection.” [Online]. Available: https://doi.org/10.1109/ieeesdt.2020.9068514

[8] K. Tomsovic, D. Bakken, V. Venkatasubramanian, and A. Bose, “Designing the next generation of real-time control, communication, and computations for large power systems.” Proceedings of the IEEE, vol. 93, no. 5, pp. 965–979, May 2005. [Online]. Available: https://doi.org/10.1109/jproc.2005.847249

[9] R. Ma, H.-H. Chen, Y.-R. Huang, and W. Meng, “Smart grid communication: Its challenges and opportunities,” IEEE Transactions on Smart Grid, vol. 4, no. 1, pp. 36–46, Mar. 2013. [Online]. Available: https://doi.org/10.1109/tsg.2012.2225851

[10] E. Hossain, J. Hossain, and F. Un-Noor, “Utility grid: Present challenges and their potential solutions,” IEEE Access, vol. 6, pp. 60 294–60 317, 2018. [Online]. Available: https://doi.org/10.1109/access.2018.2873615

[11] S. Ghosh and S. Sampalli, “A survey of security in SCADA networks: Current issues and future challenges.” IEEE Access, vol. 7, pp. 135 812–135 831, 2019. [Online]. Available: https://doi.org/10.1109/access.2019.2926441

[12] B. Zhu, A. Joseph, and S. Sastry, “A taxonomy of cyber attacks on scada systems,” in 2011 International conference on internet of things and 4th international conference on cyber, physical and social computing. IEEE, 2011, pp. 380–388.

[13] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, “A survey on cyber security for smart grid communications,” IEEE Communications Surveys & Tutorials, vol. 4, no. 4, pp. 998–1010, 2012. [Online]. Available: https://doi.org/10.1109/survt.2012.010912.00035

[14] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” SIAM Journal on Computing, vol. 26, no. 5, pp. 1484–1509, Oct. 1997. [Online]. Available: https://doi.org/10.1137/s0097539797925312

[15] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, “The security of practical quantum key distribution,” Reviews of Modern Physics, vol. 81, no. 3, pp. 1301–1350, Sep. 2009. [Online]. Available: https://doi.org/10.1103/reviewmodphys.81.1301

[16] E. B. Barker and Q. H. Dang, “Recommendation for key management part 3: Application-specific key management guidance,” National Institute of Standards and Technology, Tech. Rep., Jan. 2015. [Online]. Available: https://doi.org/10.6028/nist.sp.800-57pt3r1

[17] C. H. Hauser, T. Maniyanan, and D. E. Bakken, “Evaluating multicast message authentication protocols for use in wide area power grid data delivery services,” in 2012 45th Hawaii International Conference on System Sciences. IEEE, Jan. 2012. [Online]. Available: https://doi.org/10.1109/hicss.2012.253

[18] D. Wei, Y. Lu, M. Jafari, P. M. Skare, and K. Rohde, “Protecting smart grid automation systems against cyberattacks,” IEEE Transactions on Smart Grid, vol. 2, no. 4, pp. 782–795, Dec. 2011. [Online]. Available: https://doi.org/10.1109/tsg.2011.2159999

[19] S. Mangard, M. Aigner, and S. Dominikus, “A highly regular and scalable aes hardware architecture,” IEEE Transactions on Computers, vol. 52, no. 4, pp. 483–491, 2003.

[20] H. Lo, “Unconditional security of quantum key distribution over arbitrarily long distances,” Science, vol. 283, no. 5410, pp. 2050–2056, Mar. 1999. [Online]. Available: https://doi.org/10.1126/science.283.5410.2050

[21] “ID Quantique.” [Online]. Available: https://www.idquantique.com/

[22] “Qubitiekk Inc.” [Online]. Available: http://qubitiekk.com/

[23] C. E. Shannon, “A mathematical theory of communication,” Bell System Technical Journal, vol. 27, no. 3, pp. 379–423, Jul. 1948. [Online]. Available: https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

[24] R. J. Hughes, J. E. Nordholt, K. P. McCabe, R. T. Newell, C. G. Peterson, and R. D. Somma, “Network-centric quantum communications with application to critical infrastructure protection,” 2013.

[25] Z. Tang, Y. Qin, Z. Jiang, W. O. Krawiec, and P. Zhang, “Quantum-secure multicasting over quantum power systems,” Journal of Modern Power Systems and Clean Energy, vol. 6, no. 2, pp. 1250–1263, Mar. 2021. [Online]. Available: https://doi.org/10.1007/s40565-020-01107-1

[26] R. Eskandarpour, P. Gokhale, A. Khodaei, F. T. Chong, A. Passo, and S. Bahramirad, “Quantum computing for enhancing grid security,” IEEE Transactions on Power Systems, vol. 35, no. 5, pp. 4135–4137, Sep. 2020. [Online]. Available: https://doi.org/10.1109/tpwrs.2020.3004073

[27] C. Elliott, “Building the quantum network,” New Journal of Physics, vol. 4, pp. 46–46, Jul. 2002. [Online]. Available: https://doi.org/10.1088/1367-2630/4/1/346

Fig. 8. The number of keys available to the PV Agent. The PV agent consumes the secret keys relatively faster than the intelligence agent due to its higher messaging rate. The inset shows the minimum number of secret keys (30 keys) the PV agent must maintain to keep the secret keys synchronized with the intelligence agent.
modular software and hardware power electronic interfaces,” in 2019 IEEE Energy Conversion Congress and Exposition (ECCE). IEEE, Sep. 2019. [Online]. Available: https://doi.org/10.1109/ecce.2019.8912525

[61] M. Starke, M. Chinthavali, S. Zheng, S. Campbell, R. Zeng, M. Smith, T. Kuruganti, and B. Dean, “Agent-based framework for supporting behind the meter transactive power electronic systems,” in 2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT). IEEE, Feb. 2020. [Online]. Available: https://doi.org/10.1109/isgt45199.2020.9087687

[62] M. Starke, B. Xiao, A. Thapa, P. Bhowmik, R. S. K. Moorthy, S. Campbell, B. Dean, and M. Chinthavali, “Control and management of multiple converters in a residential smart grid,” in 2021 IEEE Applied Power Electronics Conference. IEEE, Jun. 2021.

[63] M. Starke, P. Bhowmik, B. Xiao, R. S. K. Moorthy, S. Campbell, B. Dean, A. Thapa, and M. Chinthavali, “A secondary use—plug-and-play energy storage system composed of multiple energy storage technologies,” in 2021 IEEE IEEE Innovative Smart Grid Technologies. IEEE, Jun. 2021.

[64] P. Jamborsalamati, E. Fernandez, M. Moghimi, M. J. Hossain, A. Heidari, and J. Lu, “MQTT-based resource allocation of smart buildings for grid demand reduction considering unreliable communication links,” IEEE Systems Journal, vol. 13, no. 3, pp. 3304–3315, Sep. 2019. [Online]. Available: https://doi.org/10.1109/jsyst.2018.2875537

[65] R. K. Kodali and S. Sorakal, “MQTT based home automation system using ESP8266,” in 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC). IEEE, Dec. 2016. [Online]. Available: https://doi.org/10.1109/r10-htc.2016.7906845

[66] K. Birman and T. Joseph, “Exploiting virtual synchrony in distributed systems,” in Proceedings of the eleventh ACM Symposium on Operating systems principles - SOSP. ACM Press, 1987. [Online]. Available: https://doi.org/10.1145/41457.37515

[67] “Electric Power Board.” [Online]. Available: https://epb.com/