scientific reports

OPEN

Authentication of smart grid communications using quantum key distribution

Muneer Alshowkan[1]✉, Philip G. Evans[1], Michael Starke[2], Duncan Earl[3] & Nicholas A. Peters[1]

Smart grid solutions enable utilities and customers to better monitor and control energy use via information and communications technology. Information technology is intended to improve the future electric grid's reliability, efficiency, and sustainability by implementing advanced monitoring and control systems. However, leveraging modern communications systems also makes the grid vulnerable to cyberattacks. Here we report the first use of quantum key distribution (QKD) keys in the authentication of smart grid communications. In particular, we make such demonstration on a deployed electric utility fiber network. The developed method was prototyped in a software package to manage and utilize cryptographic keys 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.

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

Renewable power plant installations can now be found on the scale of hundreds of kilowatts(kW) to megawatts (MWs) of potential power generation. These generation plants are a composite of many small generation resources, all interconnected with an electrical network known as a collector system[4–6]. An example layout for a PV plant with a supplementary ES system is shown in Fig. 1a. 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 operational modes and communications protocols via an integrated communications module. System coordination is performed through a plant supervisory control and data acquisition (SCADA) system. Key to the deployment of these renewable plants is the ability for the SCADA system to communicate with the resources to establish operational capabilities and optimization strategies. Hence, secure and reliable two-way communications are critical to these systems[7–9].

Within a conventional SCADA system, a supervisory system, a human-machine interface (HMI), a communications network, a master terminal unit (MTU), remote terminal units (RTUs), and field devices. Hence, the communications network enables connectivity between the systems. Moreover, a SCADA communications network can be divided into four types: (1) monolithic systems that are isolated and do not interact with one another, (2) distributed systems that communicate over a local area network (LAN), (3) networked systems that operate in multiple sites and communicate over a wide area network (WAN), and (4) Internet of things (IoT) systems that are connected to cloud computing for widescale implementation and computational resource availability. Furthermore, the need for reliable, efficient, and continuous connectivity between the SCADA elements has led to the development of many different communications protocols. Some protocols have been designed to consider the processing power and communications requirements of industrial applications, while others focused on speed. Consequently, many protocols were designed without integrated security services such as authentication and encryption. While the SCADA system in the monolithic and distributed models can operate in isolation on private links, utilities are looking to use available or existing communication infrastructure such as WANs and IoT to reduce costs which are often shared with other entities or service providers. Consequently,

[1]Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA. [2]Electrification and Energy Infrastructures Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA. [3]Qubitekk Inc., Vista, CA 92081, USA. ✉email: alshowkanm@ornl.gov

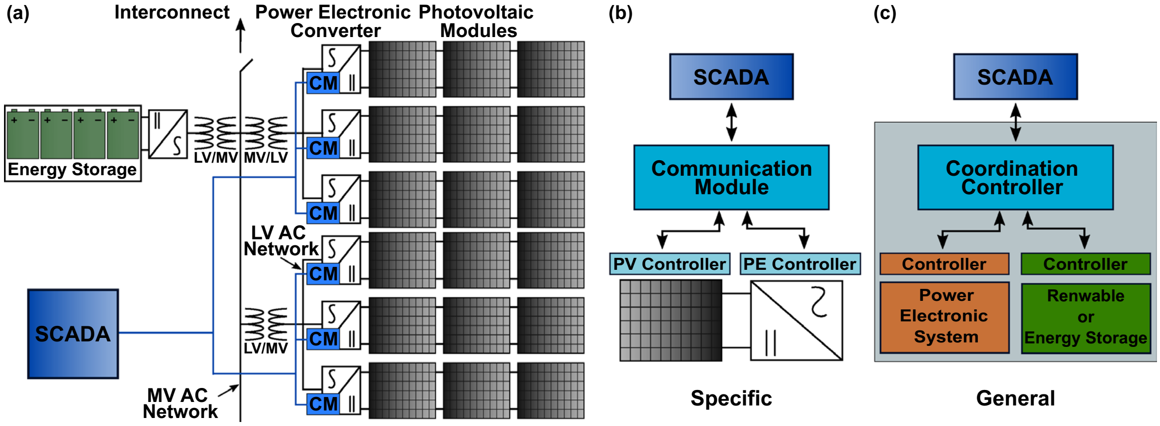

Figure 1. (**a**) Example of photovoltaic plant construction with voltage collector system (black) and communications network (blue). Architecture concept for (**b**) Specific and (**c**) General communications and control. CM: Communications module. LV: Low voltage. MV: Medium voltage. PE: Power electronic. PV: Photovoltaic. SCADA: Supervisory control and data acquisition.

Protocol	DNP3(-SA)	EtherCat	FF HSE	IEC-60870	IEC-61850	Modbus	Powerlink	Profinet
Authentication	No (DS)	No	No	DS	DS	No	No	DS
Open Source	Yes	No	Yes	No	YES	Yes	Yes	No
Comm. Model	C/S	C/S	C/S	C/S	C/S	C/S	C/S & P/C	P/C
TCP/UDP	Both	Both	Both	TCP	TCP	Both	Both	Both

Table 1. SCADA communications protocols and their characteristics. C/S: Client/Server communications model. DS: Digital Signature. DNP3: Distributed Network Protocol 3. DNP3-SA: Distributed Network Protocol 3 Secure Authentication. FF HSE: Foundation Fieldbus High Speed Ethernet. IEC: International Electrotechnical Commission. P/C: Producer/Consumer communication model. SA: Secure Authentication.

communications in these models are vulnerable to cyberattacks. For instance, the well-known ethernet-based SCADA communication protocols such as DNP3, EtherCat, Powerlink, Foundation Fieldbus HSE, and Modbus do not offer any authentication security mechanism. On the other hand, protocols such as DNS3-SA, IEC-60870, IEC-61850, and PROFINET implement security measures based on digital signatures. Table 1 shows the characteristics of these protocols, and a comprehensive review of SCADA communication protocol and their security can be explored in[10].

In addition to these standard communication protocols, IoT protocols such as message queuing telemetry transport (MQTT), data distribution service (DDS), hypertext transfer protocol (HTTP), constrained application protocol (CoAP), and advanced message queuing protocol (AMQP) can be implemented in SCADA systems for machine-to-machine (M2M) communications. MQTT[11] is a valuable protocol in the context of the IoT. MQTT has been utilized by companies such as IBM, Microsoft, and Amazon to operate as a message server that connects cloud applications and IoT devices. In comparison to SCADA systems, this protocol is similar to those often used in that data is frequently sought from other stations. One advantage of MQTT is that the protocol can be used with edge devices to integrate with older systems. Control stations and remote devices may be detached and communicate only over MQTT. Therefore, this simplifies peer-to-peer communications and relieves control stations of middleware duties. For this reason, MQTT has been recently explored and prototyped for SCADA systems[12–18].

As presented in[19], SCADA systems have been the target of many attacks that can impact the reliability of the 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[19–21].

Solutions for ensuring the privacy and integrity of the communicated data include utilizing encryption and authentication. Both encryption and authentication schemes use cryptographic algorithms and secret keys. However, the two general schemes are different: encryption converts a message plaintext to a 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[22,23]. 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[24]. This can be a challenge for devices deployed in the field as the availability of computational resources (i.e., memory size and processing capability) is typically fixed during deployment or when the device is constructed. Hence, without detrimentally increasing latency or potentially being put out of service—as the processing demand increases—devices in the field must be replaced[25,26].

In contrast, private-key cryptography—where a single key performs both encryption and decryption tasks—can be implemented very efficiently in hardware[27], 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[28]. Arguably, QKD is one of the most mature quantum applications available[23]. The fundamental technology has already been observed to be transitioning from research laboratories to commercial products. Combined with information-theoretic security protocols[29], QKD offers future-proof security: proven to be safe regardless of the technological development in computing, quantum or otherwise[23].

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[30] is the most well-known QKD protocol, yet others exist which leverage different encoding schemes[31,32] as well as entanglement[33]. 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 used in this paper implements an entanglement-based protocol[33]. It generates keys that are pulled into a higher layer to authenticate smart grid communications.

Securing a simulated power grid communications network using QKD was presented in[34] and using real time digital simulator (RTDS) microgrid testbed in[35] while theoretical approaches to improve the power grid physical security using quantum computing were explored in[36]. Previously, QKD has been applied in a trusted relay testbeds[37–43] as well as a fiber loop-back on a utility network[44]. 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[45] using the one-time-pad encryption technique. In[43] the secret keys were further used to encrypt banking communication systems via the AES-128 protocol. Hence, authentication—which is a fundamental cryptographic security service—of typical network communications was not demonstrated in any previous work to secure the power grid communications as the secret keys in the trusted relay experiments were used only for encryption of distributed keys to relay them between the network nodes.

Our main objective is to achieve in principle information-theoretic authentication in smart grid communications. Our specific implementation uses the publish-subscribe paradigm, which is popular for smart grid data, and in particular the MQTT protocol. We develop a detailed methodology, practical design, and integrate several heterogeneous components on each publisher-subscriber link in the deployed energy delivery infrastructure. The major challenges to realizing authentication are the commodity SCADA microcontrollers' limited resources, as well as their integration with a QKD system and the quantum random number generators (QRNG). Additionally, a further challenge we solve is how to manage the random numbers and the secret keys over the distributed devices.

While a review of the challenges of using QKD in the context of smart grid communications has been explored in[46], here we highlight the challenges related to securing the SCADA communications and the concepts developed to accomplish this task in our demonstration. One challenge with using public networks like WANs in the smart grid is that the networking infrastructure is often shared. A challenge arises when data leaves the utility network and becomes vulnerable to cyberattacks. A network design must be developed to provide authentication and verification services to real-time outgoing and incoming communications messages. The lack of integrated security services—such as authentication and encryption—is another challenge associated with many existing SCADA communications protocols. As a result, these protocols are also susceptible to cyberattacks. Although some protocols rely on computationally intensive public-key digital signatures for authentication, the length of their secret keys must be increased to maintain their security over time. Devices in the field often face this challenge because the computational resources available after deployment are often fixed. Moreover, SCADA systems utilize specialized microcontrollers with limited resources that may be incapable of performing the intensive calculations required for public-key cryptography as key sizes increases. Therefore, equipment in the field must be upgraded to prevent communications delays and outages. This is a challenge for devices that are deployed in remote locations and are intended to operate for a long time.

To overcome these challenges, we present specialized and generalized architectures in which QKD secret keys protect SCADA communications. The generalized approach can be applied for proprietary protocols, including many-to-many communications scenarios. The specialized network architecture intends to operate effectively for open-source point-to-point communication protocols. Utilizing the open-source MQTT protocol—which can be used for an edge device and can be integrated with older systems—is a concept that provides flexibility in terms of communications and security. Consequently, a compatible, lightweight, and information-theoretic authentication protocol can be incorporated into MQTT and operated on the SCADA microcontrollers, reliably performing authentication and verification services. Furthermore, we solve the latency challenges with private-key cryptography, in which a single key performs encryption and decryption functions with minimal computing overhead and delays. Using quantum key distribution (QKD) techniques, secure keys for private-key cryptography schemes can be established between participants. We integrate QKD keys in information-theoretically secure protocols to provide a future-proof authentication that is secure and independent of the advancement of classical or quantum computing technology. Therefore, our computationally efficient approach is able to overcome the challenges associated with limited computing resources as the key size increases in public-key cryptography. We compare the execution time of our technique to the public-key cryptography counterpart, demonstrating its feasibility for smart grid applications and showing how QKD can benefit grid communications.

In this paper, we achieve our objective by using QKD secret keys to authenticate communications of integrated power electronics energy resources in electric grid infrastructure. This work is the first time quantum secret keys have been used to authenticate smart grid communications. More specifically, (a) QKD secret keys have been applied over the IoT protocol MQTT for supporting DER communications, (b) the developed software design to utilize and manages secret keys established by a commercial Qubitekk quantum key distribution system to authenticate M2M communications, and (c) the platform has been applied in a real utility setting (at EPB in Chattanooga Tennessee, between a data center and an electrical substation connected via an optical fiber). We first lay the foundation of our developed approach in the next section and then provide a detailed description of our system and methods used to solve the challenges in the following sections.

Message encryption and authentication in QKD - Galois message authentication.

The concept of provably secure authentication was introduced in[47] 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[48]. Later, using a block cipher, it was shown by Brassard that a shorter secret key could be expanded and used for the Carter-Wegman authentication scheme[49]. Galois/Counter Mode (GCM) is a state-of-the-art parallelizable symmetric-key cryptographic protocol based on the Carter-Wegman authentication scheme[50]; 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 in 2007 via NIST SP 800-38D standard[51] which is also part of the federal information processing standards (FIPS).

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[51]. Currently, the acceptable block ciphers recommended by NIST are AES-128, AES-192, and AES-256[52]. 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 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.

Communications 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 authentication of smart grid communications using quantum key distribution demonstration is discussed.

The integration of a power electronic controller (PEC) and an energy resource to construct a distributed energy resource (DER) can be performed through a multiple vendor "black-box" integration effort[53,54]. The "black-box" designation signifies that only a communications interface to the system is present, as shown in Fig. 1b. This work proposes an architecture that utilizes an integration layer (or coordination controller) to couple systems and providers, shown in Fig. 1c. 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 communications between agents and message buses utilize the MQTT protocol. As an example, a start-up request is broken into manageable steps between the resource and the PEC to complete the task. These operations must be tightly synchronized and often autonomous to avoid errors and protect the energy infrastructure. This approach has been demonstrated in the development of energy storage systems and PV from residential[14–17], to commercial-scale[18] systems in both hardware and controller hardware-in-the-loop platforms. We note that other MQTT-based work for autonomous resource allocation systems was explored in[12] and for automation systems in[13].

In this work, an MQTT messaging approach between the SCADA system and the DER coordination controller is outlined as follows: the SCADA subscribes to measurement data published by the DER coordination controller and the DER coordination controller subscribes to control data published by SCADA as presented in Table 2. An example of an auto-commissioning sequence through a registration process is presented in[15].

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

Table 2. SCADA and DER communications using publisher-subscribe model

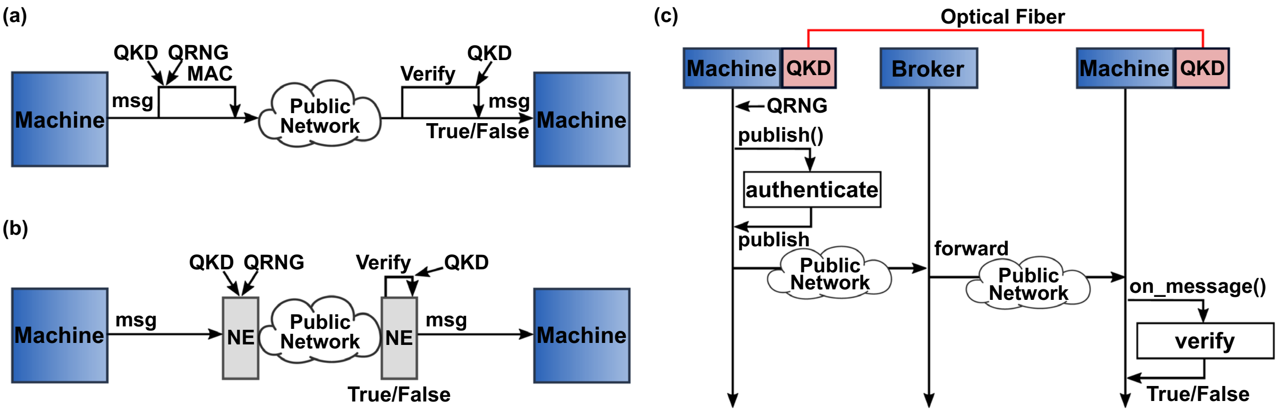

Figure 2. Showing (one-way for simplicity) generalized authentication approach using QKD secret keys and QRNG that can be implemented in: (**a**) Open-source and (**b**) Proprietary SCADA communication protocols to authenticate outgoing messages by one machine and verify incoming messages by another. (**c**) Implementation of the authentication approach using QKD secret keys and QRNG in the MQTT publisher/subscriber protocol operating for SCADA communications. MAC: Message authentication code. msg: SCADA message. NE: Network encryption card. on_message(): MQTT specific on received message callback function. publish(): MQTT specific on publish message callback function. QKD: Quantum key distribution. QRNG: Quantum random number generator.

Methods: Applied QKD key authentication approach

This section describes the authentication methods integrated into the MQTT-based protocol M2M SCADA communications system. We show the applied QKD keying approach to achieve information-theoretic authentication communications between publisher and subscriber.

Generalized authentication approach.

In general, machine-to-machine authentication can be accomplished by creating a cryptographic challenge using secret keys that are only known to the sender and the receiver. Ideally, through information-theoretic concepts combined with secret keys distributed over communication networks via QKD. Assuming the SCADA communication protocol is open-source, it is possible to implement such an authentication protocol for each outgoing message by sending the original message accompanied by its challenge (e.g., MAC). Then, the receiver uses a verification function to check the authenticity of each received message. This verification function will enable the SCADA receiving machine to accept or reject the received message, as shown in Fig. 2a. For proprietary SCADA communication protocols, QKD secret keys can be used in network encryptors modules as shown in Fig. 2b that perform end-to-end encryption and authentication services[42,55,56]. A Benefit of this approach is solving challenges in the system scalability, as described in[46]. In this case, the traditional point-to-point QKD system—including long-distance deployment via satellite—can be facilitated for many-to-many communications models.

Specific authentication approach to MQTT.

Assuming Alice and Bob share a set of QKD-based secret keys $k_1, ..., k_n$ where n is an arbitrary serial number for each key. To guarantee that only one user uses each secret key, we give each secret key a serial number. Then 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 $iv_1, ..., iv_j$ privately generated from a quantum random number generator where j is an arbitrary serial number for each iv. To publish an authenticated message m and its topic t —which is an MQTT specific variable and part of every packet—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:

$$tm_i = m_i + t_i + n + ts \tag{1}$$

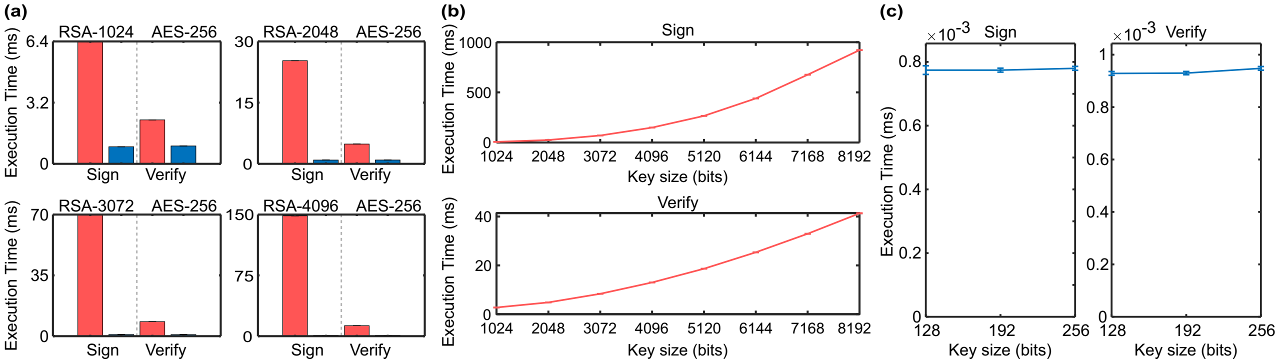

Figure 3. Measurement results for GMAC (blue) and Digital Signature (red) for (**a**) Execution times to authenticate a 256-byte message using GMAC and digital signature based on RSA with 1024, 2048, 3072, and 4096-bit and AES-256 bit keys. The vertical dotted lines are for visualization only to separate signing from verifying bars. (**b**) An extended measurement to show the trend of execution time for larger key sizes for RSA-based digital signature. (**c**) Authentication computation time of a 256-byte message using AES-based GMAC with s key sizes of 128, 192, and 256-bits.

In our software, we utilized the MQTT built-in *publish()* callback function as shown in Fig. 2(c) 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:

$$mac_S = GMAC_E(tm_i, k_n, iv_j) \tag{2}$$

where the total message tm_i, secret key k_n, and initialization vector iv_j are 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

$$p = tm_i + iv_j + mac_S \tag{3}$$

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 δ. 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:

$$mac_R = GMAC_D(tm_i, k_n, iv_j) \tag{4}$$

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. Supplementary Algorithm 1 and 2 summarize the process of creating and verifying the MAC, respectively.

Device execution time measurement. Theoretical information on the complexity of the underlying cryptographic algorithms has already been explored and can be found in[51,57,58]. Therefore, in this section, we characterize the device running the Python-based DER system by measuring the authentication execution time in the same programming language. Each DER machine runs on Raspberry Pi 3b+, which is equipped with a 1.4GHz Cortex-A53 quad-core processor and 1GB LPDDR2 SDRAM. We compare the proposed authentication using GMAC to the digital signature available in some SCADA communications protocols. Because SCADA messages are typically short, we set the message length to 256 bytes in the following measurements.

Figure 3a shows the execution time to sign and verify a message using digital signatures based on RSA 1024, 2048, 3072, and 4096-bit keys compared to GMAC based on AES with a 256-bit key and 128-bit nonce—the longest recommended secret key and nonce by NIST[52]. The average execution times in milliseconds (ms) to sign (verify) a message using GMAC with AES-256: 0.8895 ± 0.0072 (0.9309 ± 0.0088), RSA-1024: 6.3507 ± 0.0137 (2.2864 ± 0.0037), RSA-2048: 25.2802 ± 0.0214 (4.8489 ± 0.0057), RSA-3072: 69.9515 ± 0.0450 (8.3635 ± 0.0071) and, RSA-4096: 148.4858 ± 0.0207 (12.9215 ± 0.0078). The uncertainties are reported as the standard deviation of the mean of 512 samples. While the currently recommended RSA key sizes are 2048 and

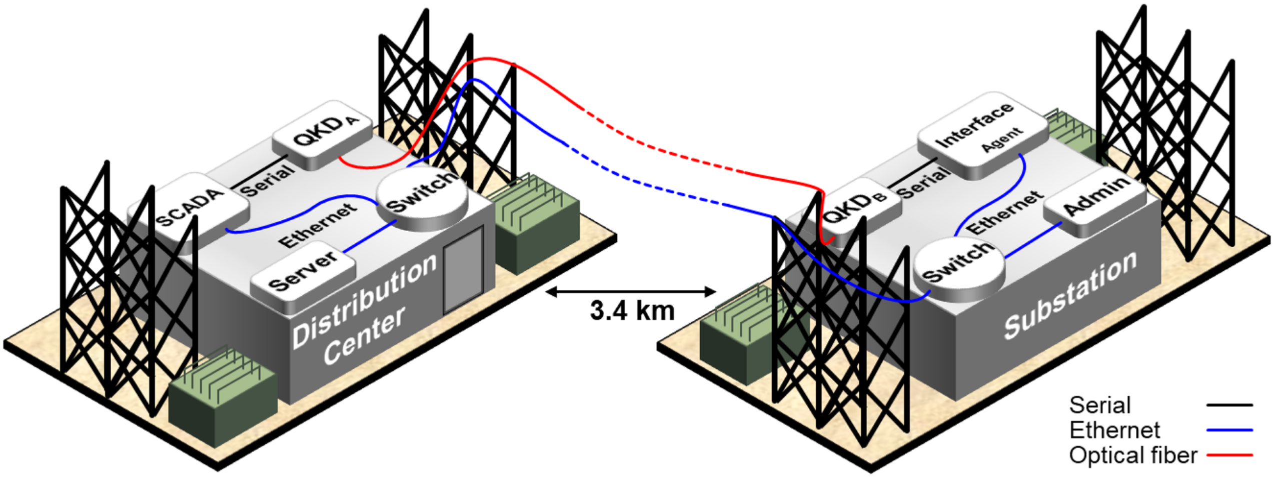

Figure 4. Illustration of the network configuration. The distribution center contains the QKD Alice system, SCADA (Supervisory control and data acquisition) system, and a server to collect network statistics. The substation contains the QKD Bob system, the energy storage system Interface agent, and an admin computer for real-time monitoring.

3072-bit, we are showing the measurement results for RSA 1024 and 4096-bit to illustrate the execution time of the previous and possibly future RSA standards[24], respectively. We notice that, as the RSA key size increases, the execution time significantly increases and note that at a delay of 160 ms (total time to sign and verify a message for the RSA 4096) it is possible to get electric grid synchronization errors. An extended measurement to show the trend of execution time for larger key sizes for RSA up to 8192-bit is shown in Fig. 3b. Figure 3c shows the GMAC authentication execution time of a message using AES with key sizes of 128, 192, and 256-bit (the maximum possible key size to measure), with a maximum delay of less than 2 ms to sign and verify. In contrast with the RSA results, the GMAC results show a negligible increase in the signing and verification times, indicating that increasing the key size in the future is feasible with negligible added delay.

Demonstration on the electric grid.

We demonstrate the above QKD-enabled MQTT approach in a real-world electrical utility environment at the Electric Power Board (EPB), 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 Industrial Control Systems (ICS) commercial QKD system is used in this demonstration. The link distance between DC and SUB is 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 that 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.

Network configuration.

The QKD hardware is deployed at the utility between DC and SUB. At each location, a virtual distributed energy resource (vDER) machine on Raspberry Pi 3 B+ was deployed: the Intelligence (Intel) Agent machine is set up in DC, and the Photovoltaic (PV) Agent machine in SUB. Each system is connected to the private classical network via a network switch (see Fig. 4). Additionally, two other supporting devices in the same network are connected: (1) a server to collect the network statistics in DC and (2) a device in the substation, used for administration tasks, including control and data monitoring.

Authentication software.

Software that handles network node secret key and random number operations, including retrieving, verifying, and managing the materials, has been developed for this demonstration. Then, we utilize these materials to authenticate the vDER communications. While performing these operations, each node is responsible for tracking and reporting statistics related to the secret keys, random numbers, and the completed tasks. Because the communications in this network follow a publish-subscribe architecture; when the software starts, the receiving node verifies the authenticity of the transmitting node; then subscribes to topics of interest. In what follows, the basic functionalities of the network nodes are described.

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

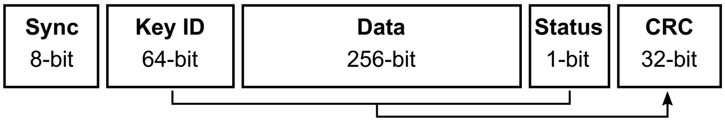

Figure 5. The format of the secret keys retrieved from the QKD system. Sync: Synchronization, a 8-bit to determine the start of a frame. Key ID: 64-bit key counter. Key Data: 256-bit Secret key. Key Status: 1-bit status of the key. CRC: Cyclic Redundancy Check, a 32-bit for key id, key data, and key status.

stores them in a local file. Figure 5 shows the format of the key materials retrieved from the QKD system and stored in the local file. One functionality of the software that has been developed and integrated into the vDER system is the periodic monitoring and data retrieval of new key materials from the local file as they become available. The software checks every received key for the appropriate length to avoid run-time errors caused by inadequate key length. In this case, the key of length 32 bytes (256-bit) is verified 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 MQTT protocol communications authentication.

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 to use as initialization vectors. In our case, we use random numbers generated from a commercial IDQ QRNG. 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, a random number from the local list is retrieved and a corresponding flag is set to "used" to never be used again.

Authentication and verification.

Authentication and verification are the core of the software integrated in MQTT-based SCADA system described in the previous section. The software is called when the vDER 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.

Statistics reporting.

For monitoring, each node periodically reports general information to the statistics server. For example, information reported related to the key management includes the number of available, added, and used secret keys. Similarly, information related to the random numbers, including those added, available, and used, is also reported. Additionally, the verification algorithm reports the number of successful and unsuccessful message verification instances.

Results

Using the developed authentication approach in the MQTT protocol operating in the SCADA system and the software described above, we authenticate the communications between the PV and Intel agents using secret keys from the deployed QKD system. When the PV and Intel agents start, they proceed to perform the secret key and random number management described in the previous section. A set of global variable objects are initialized of various classes needed to support the communications, interfaces, and measurements. Additionally, if enabled, the agents 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 acknowledged with a message containing a connect flag by the broker. Individually, each agent informs the broker about the list of topics of interest and each 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 retransmission of the same message is triggered: lost acknowledgment flags trigger retransmission 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 vDER 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. For example, the Intel agent publishes control and request information related to the type of the system on the identification of need, such as control and setpoints, while the PV agent publishes configuration and forecast using a slow periodic

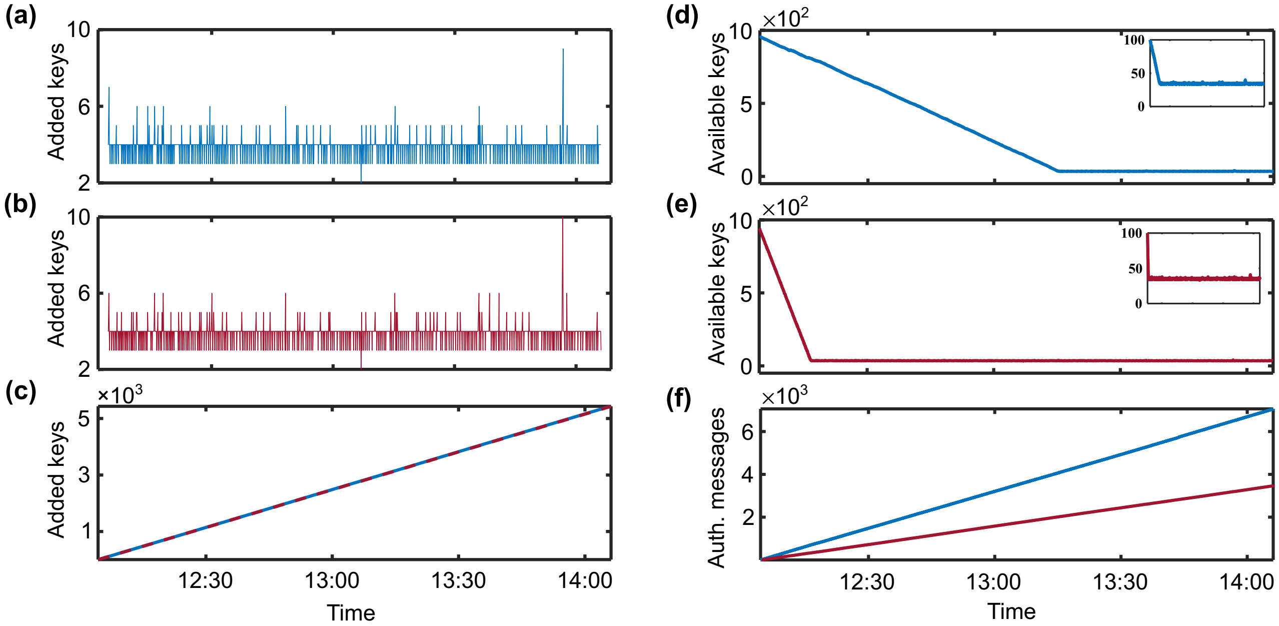

Figure 6. Polled every 5 seconds (**a**) Number keys added to the Intel Agent system. (**b**) Number keys added to the Pv Agent system. (**c**) Total number of the added keys to the Intel (blue) and Pv (red) agents. The number of keys available to (**d**) The Intel agent and (**e**) The Pv agent. The inset figure shows the minimum keys that each agent must maintain (30 keys, which we chose arbitrarily) to synchronize the secret keys. (**f**) The number of authenticated messages by the Intel (blue) and PV (red) agents.

timing (still in seconds). On the other hand, the PV agent publishes the system status, measurements, and errors in the fast periodic local messages.

We collect data related to the number of added and available keys for each agent. Figures 6a,b 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. The drop after 13:00 and spike before 14:00 in the added keys are likely the results of environmental changes, including wind gusts affecting the aerially deployed fiber. Figure 6c shows the total number of keys added by each agent. 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. Figures 6d,e show the number of available secret keys at each agent as a function of time. Before starting the energy storage system communications, each agent starts collecting keys from the QKD system. When the agents start communicating, a reservoir of approximately 950 keys is available in the secret key file. Then, each begins authenticating their received messages using an odd (or an even) key identification number for the Intel (PV) agent. Figures 6d,e 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. Consequently, as shown in Fig. 6f, the PV agent authenticates messages at a slower rate.

This paper presents the first demonstration of quantum key-based authentication of smart grid communications across an energy delivery infrastructure environment. The developed system utilizes a flexible and scalable smart grid communications protocol: a publish-subscribe method. Further, keys from a commercial Qubitekk quantum key distribution system along with the Carter-Wegman authentication protocol are used, which in principle offer information-theoretic security. With this demonstration, quantum and classical security technologies have been shown to work in the energy infrastructure to authenticate data and control communications, providing long-term security, capable of exceeding the expected infrastructure service life. Future development of the reported techniques could include full hardware integration via smart grid manufacturers. In addition, hardware platforms with fully integrated power electronics systems are in development today in a new facility called the Grid Research Integration and Deployment Center (GridC). This facility provides an avenue to fully scale the presented implementation into multiple power electronics systems and integration demonstrations. On the other hand, in terms of cybersecurity, previous work demonstrated the trusted relay on the power grid but stopped short of showing how to use the distributed secret keys[45], which is the focus of this work. Future work could concentrate on developing scalable secure communications including a wider range of power infrastructure devices.

Discussion

Cyberattacks seeking to disrupt grid communications can have devastating consequences for grid operations. Therefore, verifying that the grid communications have originated from the authorized user is crucial. 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 secret keys for symmetric cryptography offer an attractive solution for long-term secure grid communications authentication. On the other hand, message encryption hides information, and only users who know the secret key can encrypt and decrypt a message. Furthermore, for smart grid communications, the information in transit contains typical measurement data such as voltage, current, frequency, and phase that need to be examined for correctness—but not necessarily encrypted. As a result, in some applications such as the distribution automation system[59] authentication is preferable to encryption as data is usable during troubleshooting (such as a delay) with the cryptographic operations. Additionally, authentication has a further advantage in requiring fewer random bits from the QKD than full data encryption.

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[60,61]. 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 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[37].

The North American Electric Reliability Corporation (NERC) issued a set of Critical Infrastructure Protection (CIP) reliability standards[62] 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, which guarantees the physical security of the systems. However, the connectivity between a utility SCADA system and devices is the current widespread networking suite called Transmission Control Protocol/Internet Protocol (TCP/IP)[63,64]. While all communications for the electrical system must be trustworthy[65] and timely[66], transmitting data via a TCP/IP protocol is susceptible to cyber-attacks including spoofing[26]. 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 potentially vulnerable to cyber-attacks[66,67].

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[23,24]. For example, using the Carter-Wegman[68] 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[25]. Thus, 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.

MQTT[11] is a communications protocol based on the publish-subscribe model (instead of the typical client-server architecture) developed in 1999 to minimize power and bandwidth requirements[69]. In the publish-subscribe communications paradigm, the publishers and subscribers never communicate directly but utilize a third-party intermediary, commonly referred to as a *broker*. The 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 broker, who then relays them to the interested subscribers. Each published message must contain a required topic—that clients subscribe to its relevant messages—and an optional payload. For this reason, the broker activities can be parallelized—using topic-based filtering—in an event-based manner, making it an ideal protocol for IoT services.

Data availability

The published article contains all of the data that was collected or analyzed throughout the course of this study.

Received: 9 March 2022; Accepted: 4 July 2022
Published online: 26 July 2022

References

1. Yao, M. & Cai, X. An overview of the photovoltaic industry status and perspective in China. *IEEE Access* **7**, 181051–181060. https://doi.org/10.1109/access.2019.2959309 (2019).
2. Cole, W. J., Marcy, C., Krishnan, V. K. & Margolis, R. Utility-scale lithium-ion storage cost projections for use in capacity expansion models. In *2016 North American Power Symposium (NAPS)*, https://doi.org/10.1109/naps.2016.7747866 (IEEE, 2016).
3. Stecca, M., Elizondo, L. R., Soeiro, T. B., Bauer, P. & Palensky, P. A comprehensive review of the integration of battery energy storage systems into distribution networks. *IEEE Open J. Ind. Electr. Soc.* https://doi.org/10.1109/ojies.2020.2981832 (2020).
4. Camm, E. *et al.* Wind power plant collector system design considerations: IEEE PES wind plant collector system design working group. In *2009 IEEE Power & Energy Society General Meeting*, https://doi.org/10.1109/pes.2009.5275322 (IEEE, 2009).
5. Camm, E. *et al.* Wind power plant substation and collector system redundancy, reliability, and economics. In *2009 IEEE Power & Energy Society General Meeting*, https://doi.org/10.1109/pes.2009.5275333 (IEEE, 2009).
6. IEEE guide for solar power plant grounding for personnel protection, https://doi.org/10.1109/ieeestd.2020.9068514.
7. Tomsovic, K., Bakken, D., Venkatasubramanian, V. & Bose, A. Designing the next generation of real-time control, communication, and computations for large power systems. *Proc. IEEE* **93**, 965–979. https://doi.org/10.1109/jproc.2005.847249 (2005).
8. Ma, R., Chen, H.-H., Huang, Y.-R. & Meng, W. Smart grid communication: Its challenges and opportunities. *IEEE Trans. Smart Grid* **4**, 36–46. https://doi.org/10.1109/tsg.2012.2225851 (2013).
9. Hossain, E., Hossain, J. & Un-Noor, F. Utility grid: Present challenges and their potential solutions. *IEEE Access* **6**, 60294–60317. https://doi.org/10.1109/access.2018.2873615 (2018).
10. Pliatsios, D., Sarigiannidis, P., Lagkas, T. & Sarigiannidis, A. G. A survey on scada systems: Secure protocols, incidents, threats and tactics. *IEEE Commun. Surv. Tutor.* **22**, 1942–1976. https://doi.org/10.1109/COMST.2020.2987688 (2020).
11. Mqtt-v5.0. MQTT Version 5.0. Tech. Rep. March, Oasis-Open (2019).
12. Jamborsalamati, P. *et al.* MQTT-based resource allocation of smart buildings for grid demand reduction considering unreliable communication links. *IEEE Syst. J.* **13**, 3304–3315. https://doi.org/10.1109/jsyst.2018.2875537 (2019).
13. Kodali, R. K. & Soratkal, S. MQTT based home automation system using ESP8266. In *2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, https://doi.org/10.1109/r10-htc.2016.7906845 (IEEE, 2016).
14. Starke, M. *et al.* A multi-agent system concept for rapid energy storage development. In *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, https://doi.org/10.1109/isgt.2019.8791563 (IEEE, 2019).
15. Starke, M. *et al.* Residential (secondary-use) energy storage system with modular software and hardware power electronic interfaces. In *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*, https://doi.org/10.1109/ecce.2019.8912525 (IEEE, 2019).
16. Starke, M. *et al.* Agent-based framework for supporting behind the meter transactive power electronic systems. In *2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, https://doi.org/10.1109/isgt45199.2020.9087687 (IEEE, 2020).
17. Starke, M. *et al.* Control and management of multiple converters in a residential smart grid. In *2021 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 668–674, https://doi.org/10.1109/APEC42165.2021.9487327 (2021).
18. Starke, M. *et al.* Secondary use-plug-and-play energy storage system composed of multiple energy storage technologies. In *2021 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 1–5, https://doi.org/10.1109/ISGT49243.2021.9372177 (2021).
19. Ghosh, S. & Sampalli, S. A survey of security in SCADA networks: Current issues and future challenges. *IEEE Access* **7**, 135812–135831. https://doi.org/10.1109/access.2019.2926441 (2019).
20. Zhu, B., Joseph, A. & Sastry, S. 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*, 380–388, https://doi.org/10.1109/iThings/CPSCom.2011.34 (2011).
21. Yan, Y., Qian, Y., Sharif, H. & Tipper, D. A survey on cyber security for smart grid communications. *IEEE Commun. Surv. Tutor.* **14**, 998–1010. https://doi.org/10.1109/surv.2012.010912.00035 (2012).
22. Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* **26**, 1484–1509. https://doi.org/10.1137/s0097539795293172 (1997).
23. Scarani, V. *et al.* The security of practical quantum key distribution. *Rev. Mod. Phys.* **81**, 1301–1350. https://doi.org/10.1103/revmodphys.81.1301 (2009).
24. Barker, E. B. & Dang, Q. H. Recommendation for key management part 3: Application-specific key management guidance. Tech. Rep., National Institute of Standards and Technology. https://doi.org/10.6028/nist.sp.800-57pt3r1 (2015).
25. Hauser, C. H., Manivannan, T. & Bakken, D. E. Evaluating multicast message authentication protocols for use in wide area power grid data delivery services. In *2012 45th Hawaii International Conference on System Sciences*, https://doi.org/10.1109/hicss.2012.253 (IEEE, 2012).
26. Wei, D., Lu, Y., Jafari, M., Skare, P. M. & Rohde, K. Protecting smart grid automation systems against cyberattacks. *IEEE Trans. Smart Grid* **2**, 782–795. https://doi.org/10.1109/tsg.2011.2159999 (2011).
27. Mangard, S., Aigner, M. & Dominikus, S. A highly regular and scalable aes hardware architecture. *IEEE Trans. Comput.* **52**, 483–491. https://doi.org/10.1109/TC.2003.1190589 (2003).
28. Lo, H.-K. Unconditional security of quantum key distribution over arbitrarily long distances. *Science* **283**, 2050–2056. https://doi.org/10.1126/science.283.5410.2050 (1999).
29. Shannon, C. E. A mathematical theory of communication. *Bell Syst. Tech. J.* **27**, 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x (1948).
30. Bennett, C. H. & Brassard, G. Quantum cryptography: Public key distribution and coin tossing. *Theor. Comput. Sci.* **560**, 7–11. https://doi.org/10.1016/j.tcs.2014.05.025 (2014).
31. Bennett, C. H., Brassard, G. & Mermin, N. D. Quantum cryptography without bell's theorem. *Phys. Rev. Lett.* **68**, 557–559. https://doi.org/10.1103/physrevlett.68.557 (1992).
32. Scarani, V., Acín, A., Ribordy, G. & Gisin, N. Quantum cryptography protocols robust against photon number splitting attacks for weak laser pulse implementations. *Phys. Rev. Lett.* https://doi.org/10.1103/physrevlett.92.057901 (2004).
33. Ekert, A. K. Quantum cryptography based on bell's theorem. *Phys. Rev. Lett.* **67**, 661–663. https://doi.org/10.1103/physrevlett.67.661 (1991).
34. Hughes, R. J. *et al. Network-centric quantum communications with application to critical infrastructure protection* arXiv preprint arXiv:1305.0305 (2013).
35. Tang, Z., Qin, Y., Jiang, Z., Krawec, W. O. & Zhang, P. Quantum-secure microgrid. *IEEE Trans. Power Syst.* **36**, 1250–1263. https://doi.org/10.1109/tpwrs.2020.3011071 (2021).
36. Eskandarpour, R. *et al.* Quantum computing for enhancing grid security. *IEEE Trans. Power Syst.* **35**, 4135–4137. https://doi.org/10.1109/tpwrs.2020.3004073 (2020).
37. Elliott, C. Building the quantum network. *New J. Phys.* **4**, 46–46. https://doi.org/10.1088/1367-2630/4/1/346 (2002).
38. Peev, M. *et al.* The SECOQC quantum key distribution network in vienna. *New J. Phys.* **11**, 075001. https://doi.org/10.1088/1367-2630/11/7/075001 (2009).
39. Chen, T.-Y. *et al.* Metropolitan all-pass and inter-city quantum communication network. *Opt. Exp.* **18**, 27217. https://doi.org/10.1364/oe.18.027217 (2010).

40. Stucki, D. *et al.* Long-term performance of the SwissQuantum quantum key distribution network in a field environment. *New J. Phys.* **13**, 123001. https://doi.org/10.1088/1367-2630/13/12/123001 (2011).
41. Sasaki, M. *et al.* Field test of quantum key distribution in the tokyo QKD network. *Opt. Exp.* **19**, 10387. https://doi.org/10.1364/oe.19.010387 (2011).
42. Dynes, J. F. *et al.* Cambridge quantum network. *NPJ Quantum Inf.* https://doi.org/10.1038/s41534-019-0221-4 (2019).
43. Chen, Y.-A. *et al.* An integrated space-to-ground quantum communication network over 4, 600 kilometres. *Nature* **589**, 214–219. https://doi.org/10.1038/s41586-020-03093-8 (2021).
44. Evans, P. *et al.* Demonstration of a quantum key distribution trusted node on an electric utility fiber network. In *2019 IEEE Photonics Conference (IPC)*, https://doi.org/10.1109/ipcon.2019.8908470 (IEEE, 2019).
45. Evans, P. G. *et al.* Trusted node QKD at an electrical utility. *IEEE Access* **9**, 105220–105229. https://doi.org/10.1109/access.2021.3070222 (2021).
46. Kong, P.-Y. A review of quantum key distribution protocols in the perspective of smart grid communication security. *IEEE Syst. J.* **16**, 41–54. https://doi.org/10.1109/JSYST.2020.3024956 (2022).
47. Gilbert, E. N., MacWilliams, F. J. & Sloane, N. J. A. Codes which detect deception. *Bell Syst. Tech. J.* **53**, 405–424. https://doi.org/10.1002/j.1538-7305.1974.tb02751.x (1974).
48. Wegman, M. N. & Carter, J. L. New classes and applications of hash functions. In *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)*, https://doi.org/10.1109/sfcs.1979.26 (IEEE, 1979).
49. Brassard, G. On computationally secure authentication tags requiring short secret shared keys. In *Advances in Cryptology*, 79–86, https://doi.org/10.1007/978-1-4757-0602-4_7 (Springer US, 1983).
50. McGrew, D. & Viega, J. The galois/counter mode of operation (gcm). *submission to NIST Modes of Operation Process* **20**, 0278–0070 (2004).
51. Dworkin, M. J. *Sp 800-38d. recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac* (National Institute of Standards & Technology, 2007).
52. Barker, E. & Roginsky, A. Transitioning the use of cryptographic algorithms and key lengths. Tech. Rep., National Institute of Standards and Technology https://doi.org/10.6028/nist.sp.800-131ar2 (2019).
53. Arnedo, L., Burgos, R., Boroyevich, D. & Wang, F. System-level black-box dc-to-dc converter models. In *2009 Twenty-Fourth Annual IEEE Applied Power Electronics Conference and Exposition*, https://doi.org/10.1109/apec.2009.4802861 (IEEE, 2009).
54. Valdivia, V., Barrado, A., Lazaro, A., Zumel, P. & Raga, C. Easy modeling and identification procedure for “black box” behavioral models of power electronics converters with reduced order based on transient response analysis. In *2009 Twenty-Fourth Annual IEEE Applied Power Electronics Conference and Exposition*, https://doi.org/10.1109/apec.2009.4802675 (IEEE, 2009).
55. Choi, I. *et al.* Field trial of a quantum secured 10 gb/s dwdm transmission system over a single installed fiber. *Opt. Exp.* **22**, 23121–23128. https://doi.org/10.1364/OE.22.023121 (2014).
56. Alshowkan, M. *et al.* Advanced architectures for high-performance quantum networking. *J. Opt. Commun. Netw.* **14**(6), 493–499. https://doi.org/10.1364/JOCN.450201 (2022).
57. Boneh, D. *et al.* Twenty years of attacks on the rsa cryptosystem. *Notices AMS* **46**, 203–213 (1999).
58. Bogdanov, A., Khovratovich, D. & Rechberger, C. Biclique cryptanalysis of the full aes. In *Advances in Cryptology - ASIACRYPT 2011 (Springer)* (eds Lee, D. H. & Wang, X.) 344–371 (Berlin Heidelberg, Berlin, Heidelberg, 2011).
59. Lim, I. H. *et al.* Security protocols against cyber attacks in the distribution automation system. *IEEE Trans. Power Deliv.* **25**, 448–455. https://doi.org/10.1109/TPWRD.2009.2021083 (2010).
60. Peters, N. A. *et al.* Dense wavelength multiplexing of 1550 nm QKD with strong classical channels in reconfigurable networking environments. *New J. Phys.* https://doi.org/10.1088/1367-2630/11/4/045012 (2009).
61. Mao, Y. *et al.* Integrating quantum key distribution with classical communications in backbone fiber network. *Opt. Exp.* **26**, 6010. https://doi.org/10.1364/oe.26.006010 (2018).
62. North American Electric Reliability Corporation (NERC)-Critical Infrastructure Protection (CIP).
63. Cerf, V. & Kahn, R. A protocol for packet network intercommunication. *IEEE Trans. Commun.* **22**, 637–648. https://doi.org/10.1109/tcom.1974.1092259 (1974).
64. R. Braden, “Requirements for Internet Hosts - Communication Layers,” RFC 1122 (1989).
65. Metke, A. R. & Ekl, R. L. Security technology for smart grid networks. *IEEE Trans. Smart Grid* **1**, 99–107. https://doi.org/10.1109/tsg.2010.2046347 (2010).
66. Kansal, P. & Bose, A. Bandwidth and latency requirements for smart transmission grid applications. *IEEE Trans. Smart Grid* **3**, 1344–1352. https://doi.org/10.1109/tsg.2012.2197229 (2012).
67. Wang, W., Xu, Y. & Khanna, M. A survey on the communication architectures in smart grid. *Comput. Netw.* **55**, 3604–3629. https://doi.org/10.1016/j.comnet.2011.07.010 (2011).
68. Wegman, M. N. & Carter, J. New hash functions and their use in authentication and set equality. *J. Comput. Syst. Sci.* **22**, 265–279. https://doi.org/10.1016/0022-0000(81)90033-7 (1981).
69. Birman, K. & Joseph, T. Exploiting virtual synchrony in distributed systems. In *Proceedings of the eleventh ACM Symposium on Operating systems principles - SOSP*https://doi.org/10.1145/41457.37515 *(1987) (***ACM Press**).

Acknowledgements

We acknowledge the support of Steve Morrison, Tyler Morgan, and Ken Jones and Patrick Swingle. This work was partially 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.

Author contributions

M.A. performed the experiment, data analysis, and co-wrote the manuscript. P.E. guided the QKD implementation and co-wrote the manuscript M.S. developed the agent system and co-wrote the manuscript. D.E. provided support to Qubitekk QKD system. N.P. devised original experimental concept, led the project and co-wrote the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-022-16090-w.

Correspondence and requests for materials should be addressed to M.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.