2023 IEEE International Conference on Smart Computing (SMARTCOMP) | 979-8-3503-2281-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/SMARTCOMP58114.2023.00043

A Prototype for QKD-secure Serverless Computing with ETSI MEC

Claudio Cicconetti[+], Marco Conti[+], Eufemia Lella[*], Pietro Noviello[*], Gennaro Davide Paduanelli[*], Andrea Passarella[+], Elisabetta Storelli[*]
[+] *Institute of Informatics and Telematics – National Research Council – Pisa, Italy*
[*] *Marketing & Technology — Innovation Lab – Exprivia S.p.A. – Molfetta, Italy*

***Abstract*—In this demonstration, we showcase the realization of a prototype of an edge computing network, where the client and edge domains both host simulated Quantum Key Distribution devices, for a hospital use case. In particular, digital health applications using the Function-as-a-Service (FaaS) paradigm will invoke remote functions provided by an Apache OpenWhisk cluster deployed in the edge infrastructure, where the arguments and return value are encrypted using keys generated through an underlying simulated QKD point-to-point network. All the interactions in the control/management plane are handled through standard interfaces defined by the ETSI MEC and QKD industry study groups.**

***Index Terms*—quantum key distribution, QKD, edge computing, serverless computing, function-as-a-service, FaaS, ETSI MEC, ETSI QKD**

I. Introduction

Quantum Key Distribution (QKD) enables the exchange of keys between nodes exploiting the no-cloning property of quantum information theory to achieve unconditional security [1]. It is the most mature of the quantum information technologies [2], which are receiving significant attention due to recent breakthroughs and a raise in public/private investments. Indeed, QKD is close to becoming ready for massive deployment [3], hence it is urgent to assess its integration with practical communication protocols and software architectures.

The research activity that led to the realization of the demonstration illustrated in this abstract was carried out within the project QUANCOM, funded by the European Union through the Italian Ministry of University and Research and coordinated by the National Research Council (CNR). The project has the objective of demonstrating practical use cases of QKD; in particular, this activity has been focused on digital health services, with the collaboration of Exprivia S.p.A., a leading Italian company that provides innovative software and services and a partner of the project.

II. Background

We briefly introduce in this section the key technologies that are involved.

A. Quantum Key Distribution

As already mentioned in section I, QKD enables the secure exchange of cryptographic keys between two nodes directly connected through fiber optic cables [4] or free-space satellite links [5]. Such an exchange is made *unconditionally secure* through the so-called Prepare&Measure technique, which relies on the impossibility to copy an unknown quantum state according to the laws of physics. In QKD, the quantum state is encoded in single photons or weak laser pulses, which are subject to attenuation. For this reason, the maximum distance between nodes is limited, for instance with fiber optic cables, to a few tens of kilometers. To span across wider distances, intermediate relay nodes can be used, which have to be trusted because they have access to the secret material being shared, thus leading to *QKD networks*.

The European Telecommunications Standards Institute (ETSI) has since long established an Industry Specification Group (ISG) on QKD, which defined standard interfaces for the exchange of keys between the Secure Application Entity (SAE) and the Key Management Entity (KME), which is the component in charge of managing the secret keys continuously generated by the underlying QKD devices in the network. In this demonstration, we make use of the `014` interface [6], which is a REST Application Programming Interface (API) that allows acquiring new keys towards a given destination node via simple HTTP invocations.

B. ETSI MEC

Edge computing is a network architecture that envisions the execution of back-end services on nodes within the distributed infrastructure of a telco operator, as opposed to public cloud services available from remote and centralized data centers [7]. For the demonstration, we assume that the telco operator adopts the architecture and interfaces defined by ETSI Multi-access Edge Computing (MEC), which has the ambition to provide vendor-neutral APIs for edge-native applications [8]. In particular, we use the `Mx2` interface, which allows a *device app* to interact with an Life Cycle Management Proxy (LCMP) to i) query the list of available *MEC apps*, and ii) establish a new application session, which in ETSI MEC terminology is done through *context creation*. The MEC apps are executed within the security perimeter of the telco operator infrastructure, and we assume that they can access the corresponding KME and QKD devices. The device app, instead, is a broker that mediates communication with the edge computing domain and the *client app*. Similarly, we assume that the device and client apps are within the same security perimeter on the client side, which also includes KME and QKD devices.

2693-8340/23/$31.00 ©2023 IEEE
DOI 10.1109/SMARTCOMP58114.2023.00043

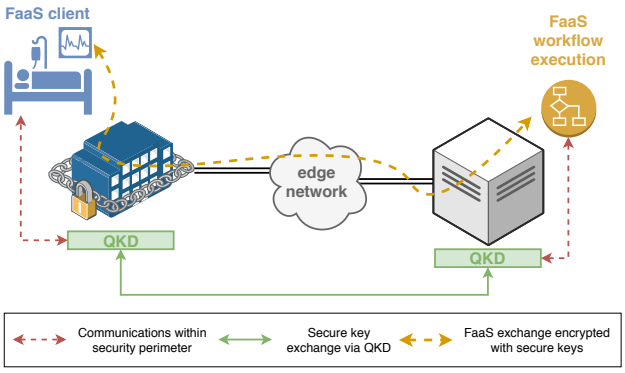

Fig. 1. Target use case: a client invokes remote functions, using a FaaS model, executed in the edge domain of a telco operator. The function arguments and return value, which may contain sensitive data, are encrypted using secure keys exchanged via QKD.

C. Serverless computing

Serverless computing [9] is an emerging trend in cloud technologies, which enables fast reactivity and high scalability through a programming model called Function as a Service (FaaS). The latter enables an application to invoke remote functions in a stateless fashion, relieving the app developer from the burden of managing both servers and platforms in the back-end. The serverless computing operators typically scale up/down the number of executors for a given function type to closely follow the actual demands, which saves resources and supports a flexible pay-as-you-go billing model.

D. Use case

The target use case is illustrated in fig. 1. It involves clients in a hospital being provided with digital health services by an edge computing domain. In the demonstration, we focus on two Artificial Intelligence (AI) applications: i) a continuous monitoring system of patients with cardiovascular risks, and ii) a personalized COVID-19 risk assessment tool. Both applications require that batches of sensitive data are transmitted asynchronously to the back-end to evaluate the respective health risk relying on pre-trained neural networks.

III. Logical Architecture of the Prototype

In fig. 2 we illustrate the software architecture of the prototype that is demonstrated. In the diagram, we also report the following sequence, which will be showcased during the demonstration as explained in section IV:

- **1.** When the service is enabled for a new user, the client app registers to the device app.
- **2.–3.** The device app performs a look-up on the directory of MEC apps through the LCMP to check that the given application (e.g., cardiovascular risk assessment) is available. This is done through the `Mx2` API.
- **4.** The device app requests to the LCMP the creation of a new application context, also via `Mx2`.
- **5.** The LCMP forwards such request to the MEC Orchestrator (MEO), which is the logical component in the edge domain responsible for allocating the resources in the MEC hosts, i.e., the edge nodes. We assume that each MEC host runs a serverless platform, which the MEO selects based on the client characteristics and current system conditions.
- **6.–7.** The MEO deploys the given MEC app on the target serverless platform/MEC host and returns to the LCMP the Uniform Resource Identifier (URI) for the client app to reach it.
- **8.–9.** The URI is returned as part of the `AppContext` standard message to the device app, which forwards it to the client app and SAE app.
- **10.–12.** At this point, the client app can send the arguments of the function, e.g., data from wearable health sensors on the patient, to the SAE app, which retrieves one key from the KME, encrypts the arguments, and invokes the remote function on the assigned serverless platform.
- **13.–14.** After retrieving the companion key from its KME, the serverless platform decrypts the arguments, executes the function, encrypts the return value, and makes this output available to the SAE app.
- **15.** Finally, the SAE app decrypts the return value, e.g., the risk assessment evaluation, and forwards it to the client app.

IV. Demo Experience

We now describe the planned demonstration experience.

A. Overview

The demo architecture is shown in fig. 3. It is realized by means of five Virtual Machines (VMs): two of them (`qkd1` and `qkd2`) simulate a QKD network using the open source software released by the OpenQKDNetwork project[1]; `quancom1` hosts the client, device, and SAE apps, which have been developed by Exprivia; `quancom2` hosts a serverless platform using Apache OpenWhisk[2]; finally, `quancom3` implements the ETSI MEC components, i.e., LCMP and MEO, using open-source software under development at CNR[3].

B. Operation

On-site, the demo will have one laptop, one tablet, and printed versions of the diagrams in the abstract. The demo will be interactive and participants will be able to:

- create/tear-down application contexts;
- trigger via a web-based interface on the tablet the execution of the cardiovascular or COVID-19 risk assessment procedures, with simulated data;
- follow the sequence of actions in fig. 2 by means of logs available on the laptop;
- inspect with tcpdump/Ethereal the content of messages exchanged at any point of the system;
- monitor the status of the ETSI MEC and serverless platforms via command-line utilities.

[1]https://github.com/Open-QKD-Network/qkd-net
[2]https://openwhisk.apache.org/
[3]https://github.com/ccicconetti/etsi-mec-qkd

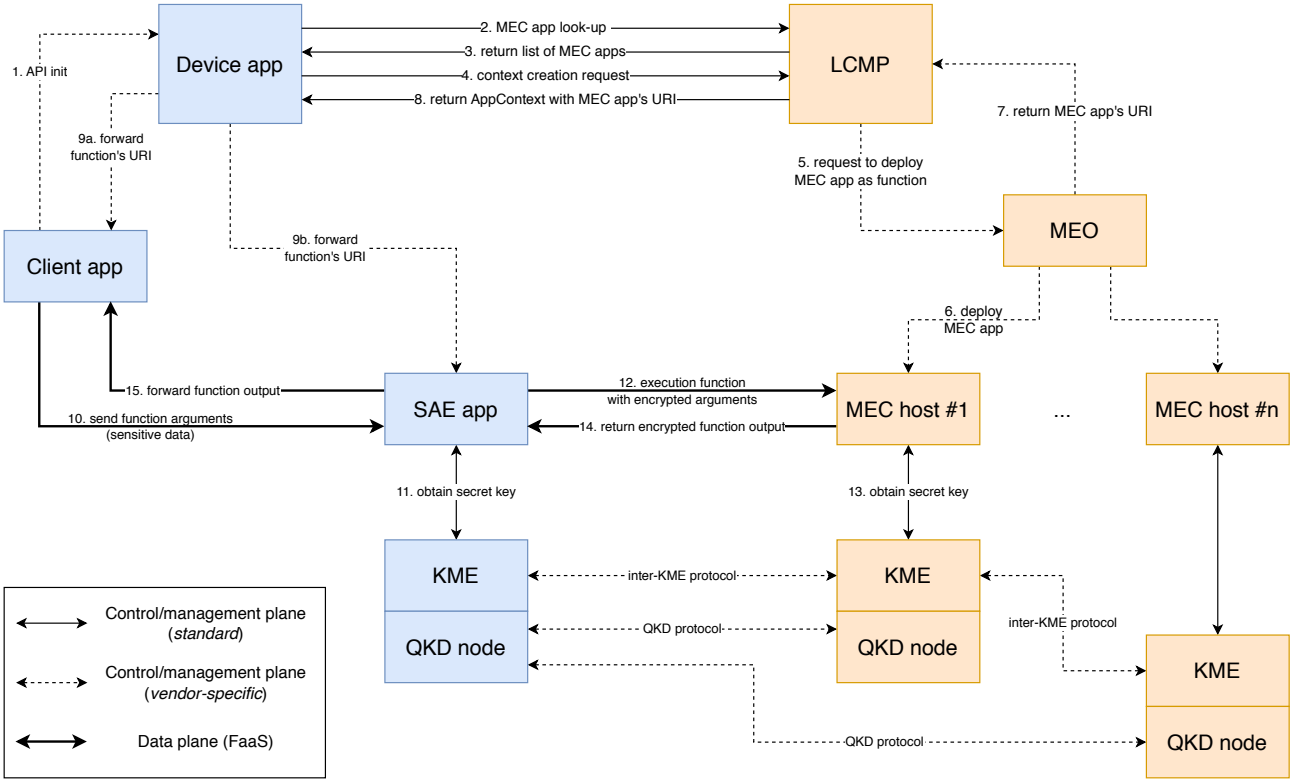

Fig. 2. Software architecture (slightly simplified) of the prototype illustrating the sequence of the main actions performed when a new client app enters the system; the direct interaction between QKD nodes can be substituted by a multi-hop QKD network with trusted relay nodes to extend the geographical range.

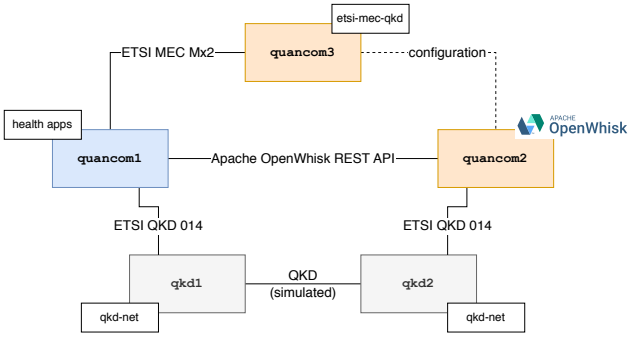

Fig. 3. Physical architecture of the prototype, showing the software installed on each (virtual) server and the relations between them.

V. Conclusions

In this abstract, we have illustrated the prototype of a system to deploy FaaS-based digital health services at the edge with QKD-secure communications, and we have described the related demo experience that the participants will find on-site.

The future steps in the roadmap of our research activity include: substituting the simulated QKD with commercial QKD devices, which are being deployed in the CNR facilities in Rome and Naples; validating the scalability of the solution by adding more serverless platforms orchestrated by a MEO.

Acknowledgment

Work co-funded by EU, *PON Ricerca e Innovazione* 2014–2020 FESR/FSC Project ARS01_00734 QUANCOM.

References

[1] L. Gyongyosi and S. Imre, "Advances in the quantum internet," *Communications of the ACM*, vol. 65, no. 8, pp. 52–63, 2022.

[2] "Industry quantum computing applications," vol. 8.

[3] Y. Cao, Y. Zhao, Q. Wang, J. Zhang, S. X. Ng, and L. Hanzo, "The Evolution of Quantum Key Distribution Networks: On the Road to the Qinternet," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 839–894, 2022, conference Name: IEEE Communications Surveys & Tutorials.

[4] D. Bacco, I. Vagniluca, B. Da Lio, N. Biagi, A. Della Frera, D. Calonico, C. Toninelli, F. S. Cataliotti, M. Bellini, L. K. Oxenløwe, and A. Zavatta, "Field trial of a three-state quantum key distribution scheme in the Florence metropolitan area," *EPJ Quantum Technology*, vol. 6, no. 1, p. 5, Dec. 2019.

[5] J. Yin *et al.*, "Satellite-based entanglement distribution over 1200 kilometers," *Science*, vol. 356, no. 6343, pp. 1140–1144, Jun. 2017.

[6] ETSI, "ETSI GS QKD 014 V1.1.1 (2019-02). Quantum Key Distribution (QKD); Protocol and data format of REST-based key delivery API," Tech. Rep., 2019.

[7] M. Campbell, "Smart Edge : The Center of Data Gravity Out of the Cloud," *Computer*, vol. 52, no. December, pp. 99–102, 2019.

[8] ETSI, "ETSI GS MEC 003 V3.1.1 (2022-03). Multi-access Edge Computing (MEC); Framework and Reference Architecture," Tech. Rep., 2022.

[9] A. Khandelwal, J. Carreira, N. J. Yadwadkar, R. A. D. A. Popa, J. E. Gonzalez, I. O. N. Stoica, and D. A. Patterson, "What Serverless Computing Is and Should Become: The Next Phase of Cloud Computing," *Communications of the ACM*, vol. 64, no. 5, pp. 76–84, 2021.