An IoT Crypto Gateway for Resource-Constrained IoT Devices

Ahmed Alqattaa

Department of Electrical Engineering,
Media and Computer Science
OTH – Technical University of Applied Sciences
Amberg, Germany
Email: a.alqattaa@oth-aw.de

Daniel Loebenberger

Department of Electrical Engineering,
Media and Computer Science
OTH – Technical University of Applied Sciences
Amberg, Germany
Email: d.loebenberger@oth-aw.de

***Abstract*—One of the biggest challenges for the Internet of Things (IoT)-Security is to implement high-end asymmetric cryptography while at the same time meeting the requirements of IoT devices due to their constrained resources. Instead of reducing the security level (e.g., by employing lightweight cryptographic primitives), this paper presents a work-in-progress project and specifies the overall architecture of an IoT cryptographic gateway "*IoT crypto gateway*", which sits in-between attached IoT devices and the cloud. The gateway communicates with the cloud implementing the Message Queuing Telemetry Transport (MQTT) protocol over a TLS (Transport Layer Security) connection employing up-to-date asymmetric cryptography at a high security level. On the other hand, the gateway allows the IoT devices to connect to the network by implementing MQTT over the Quick UDP Internet Connections (QUIC) protocol, which is at the moment still being developed by IETF. Since on transport layer, the gateway is fully transparent, the (logical) TLS connection in QUIC between the IoT devices and the gateway may save time, power and computation on the IoT device's side without compromising security.**

***Keywords*–gateway; IoT; TLS; QUIC; MQTT.**

I. Introduction

Through huge technological advances, society is moving towards an "always connected" paradigm. One wide concept associated with the "future Internet" is the *Internet of Things (IoT)*. The IoT is a network where all kinds of electronic devices are connected to each other and provide the capability to interact. The "Thing" in IoT can be any device, for instance a phone or a small sensor node that is able to connect, transfer, receive or exchange data with the network [1].

Developers as well as companies have started to increasingly introduce numerous IoT-based products and services. Furthermore, practitioners increasingly view the IoT as a real business opportunity, and expect that it could grow to USD 949.42 billion by 2025 [2]. The IoT converts the everyday world into a more flexible and accessible one. Thing, place and time do not matter anymore as long as there is access to the Internet. However, if the IoT devices are connected to the Internet without being protected properly, they may become vulnerable to attacks on the devices and the network itself.

Thus, IoT security is a relevant aspect in the design of IoT protocols. For instance, in 2015, the Federal Bureau of Investigation published a public service announcement to warn against the potential vulnerabilities of IoT devices [3]. In addition, the German Federal Office for Information Security (BSI) continuously warns against the potential attacks on the IoT and gives users possible countermeasures at hand in order to limit serious attacks against IoT devices [4] [5] [6].

As an example, Wenxiang et al. presented how to use multiple vulnerabilities to achieve a remote attack on some of the most popular smart speakers. The attack effects include silent listening, control of speaker speaking content, and other demonstrations, while offering no clue to the user that the device has been compromised [7].

This paper is structured as follows: we first discuss in Sections I-A and I-B different security requirements and challenges of the IoT, respectively. Afterwards, in Section II, related work of the past few years is identified and discussed. In Section III, the contribution is stated. Finally, the proposed architecture of this paper is described in Section IV.

A. Security Requirements for the IoT

Various hardware mechanisms and software parameters must be taken in consideration in order to secure IoT devices. We list here the most important cryptographic ones most of which can also be found in the surveys [8] [9] [10].

1) Confidentiality: the tunnel is private. Encryption is used for all messages after a simple handshake. Thus, the data is only visible to the endpoints (end-to-end encryption). A proper encryption mechanism is required to ensure the confidentiality of data in IoT [4] [5].

2) Integrity: the channel is reliable. It ensures that data contained in the device is not changed unnoticed during the transmission. Because of the constrained resources of IoT devices and network, the data, which is stored on an IoT node, could be vulnerable to integrity violation by compromising it [9].

3) Authentication and Authorization: the tunnel is authenticated. A proper implementation of authentication and authorization results in a trustworthy environment, which ensures a secure environment for communication. The variety of authentication mechanisms for the IoT exists mainly because of the different heterogeneous underlying architectures and environments that support IoT devices. These environments pose a challenge for the definition of a global standard protocol for authentication in the IoT [4] [5] [9].

Additionally, there are non-cryptographic requirements for IoT devices, such as availability, which are not addressed here.

B. *IoT Security Challenge*

IoT devices are often resource-constrained, low-power, and have small storage. Thus, attacks on IoT architectures may result in an increase in energy consumption by flooding the network and exhausting IoT resources through redundant or forged service requests [11]. Moreover, cryptographic functionalities can be realized by implementing one of the two schemes: symmetric key algorithms or public key algorithms. In comparison, public key algorithms offer a totally different set of security features such as digital signatures and key exchange mechanisms, however at higher computational cost. Taking the constrained resources of IoT devices into account, the high overhead of public key cryptography has become a major bottle-neck and triggered the use of lightweight cryptography. This, however, comes at the cost of a reduced security level [12] [13].

In order to understand the overall approach to data security, there is a need to know about the security requirements for all key components of IoT systems, i.e., IoT devices, IoT users, the IoT gateway, communication channels and cloud applications. For instance, public key infrastructure may not be suitable for IoT environments as it becomes a computationally expensive task to calculate ciphertexts because of the high computational cost for asymmetric cryptography. On the other hand, asymmetric cryptography provides additional security functionalities against attacks [13] [14].

II. Related Work

Two years ago, the Internet Engineering Task Force (IETF) finished the development of a new version of TLS, 1.3 [15]. Furthermore, the IETF is recently working on deprecating TLS 1.0 and 1.1 because these versions lack support for current and recommended cipher suites [16]. The primary goal of TLS is to secure the communication between two peers (client and server) by providing three basic properties: confidentiality, integrity and authentication. Note that other requirements, such as privacy, are not addressed by TLS and are typically not met when using TLS for IoT devices [10] [15].

Currently, the IETF is working on developing the security of the QUIC protocol by integrating TLS 1.3 in it [17] [18]. Quick UDP Internet Connections (QUIC) is a transport protocol developed by google, which reduces latency compared to TCP [19]. QUIC is a TCP-like protocol, which supports congestion control and loss recovery. It reduces a number of transport and application layers problems that occur in modern web applications, while requiring little or no modification from application writers [20] [17]. In addition, QUIC was the first protocol that can create a secure connection implementing a 0-RTT handshake between the peers, which has been later adopted in TLS 1.3 with some improvements [15] [18] .

The DTLS protocol is based on TLS and provides security for UDP-based applications. The purpose of DTLS is to make only the minimal changes to TLS required to fix loosing or reordering the packets when implementing TLS over UDP (DTLS) [21]. Currently, IETF is working on developing a new version of DTLS, 1.3 [22]. However, the UDP-Based multiplexed and secure transport (QUIC) is different from DTLS. QUIC combines multiple data streams into a single flow of UDP packets and necessarily has to handle reordering and loosing packets, like TCP [17].

The Message Queuing Telemetry Transport (MQTT) protocol is a lightweight messaging protocol, which works over the transmission protocol TCP/IP and is one of the most used protocols for IoT devices [23] [24]. For embedded devices, MQTT is highly recommended because it can work with limited processor and memory resources. In addition, through the Publish/Subscribe message pattern, the protocol provides one-to-many message distribution. The MQTT protocol itself supports only a username and a password to secure the communication between a server and clients. Any additional security has to be added into the protocol individually by employing a suitable transport protocol [25].

The mitmproxy project is a free and open source interactive HTTPS proxy, which differs from the gateway proposed here in several points, since it has the ability to communicate with different peers using different layer protocols. Furthermore, mitmproxy has been developed for other purposes, such as modifying and intercepting data between the peers [26].

NGINX published the technology preview of HTTP/3 (QUIC+HTTP), which is at the moment still being developed by IETF, at an open source repository [27] [28]. The project is a pre-release software, which is based on the IETF QUIC draft and maintained in a development branch, which is isolated from the stable and mainline branches. The release is an initial development and available for interoperability testing, feedback and code contributions. Notably, QUIC also incorporates TLS as an integral component, not as an additional layer as with HTTP/1.1 and HTTP/2 (see Figure 1) [17] [29]. Moreover, OpenSSL as well as wolfSSL have just started adding QUIC to their libraries [30] [31].

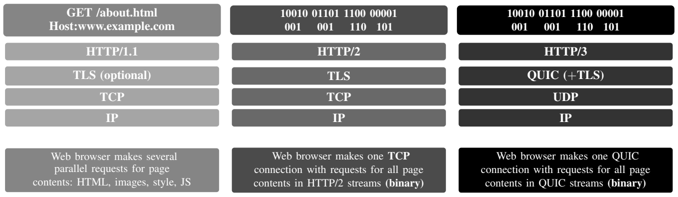

GET /about.html
Host:www.example.com
10010 01101 1100 00001
001 001 110 101
10010 01101 1100 00001
001 001 110 101
HTTP/1.1
HTTP/2
HTTP/3
TLS (optional)
TLS
QUIC (+TLS)
TCP
TCP
UDP
IP
IP
IP
Web browser makes several parallel requests for page contents: HTML, images, style, JS
Web browser makes one **TCP** connection with requests for all page contents in HTTP/2 streams (**binary**)
Web browser makes one QUIC connection with requests for all page contents in QUIC streams (**binary**)

Figure 1. High-level overview of HTTP transport stacks [29].

Recently, many authentication schemes for IoT have been proposed. For instance, Tewari et al. [32] suggested a robust anonymity preserving authentication protocol for IoT devices that provides mutual authentication between tag and reader through the server. This scheme uses Elliptic Curve Cryptography to implement authentication. As a method to provide the user the access to sensors or sensor data, the user is usually authenticated through the gateway.

Research by King et al. [33] attempted to reduce the energy consumption of IoT devices by performing lightweight protocols on the IoT device side and with minimal resource requirements, while heavier tasks are performed in the gateway side. The proposed mechanism utilizes a symmetric encryption for data objects combined with the native wireless security to offer a layered security mechanism between the device and the gateway.

In addition, Razouk et. al. [34] suggested a security middleware architecture based on fog computing and cloud to support resource constrained devices for authentication. The middleware acts as a smart gateway in order to pre-process data at the edge of the network. Thus, data is either processed and stored locally on fog or sent to the cloud for further processing.

As a result, all of the stated approaches either use expensive concepts of public key cryptography in order to establish a high security level or reduce the security level by employing cheaper lightweight methods. As it turns out, constrained IoT devices which communicate through proposed middleware, have access to more computing power and have thus enhanced capabilities to perform secure communications at a high security level [13] [34].

III. CONTRIBUTION

We present here a work-in-progress IoT crypto gateway, which has the ability to reduce the required security computations for IoT devices based on low-power System-on-a-Chip (SoC). The IoT crypto gateway stands between the cloud and the IoT devices and communicates with the cloud as a client and with the IoT devices as a cloud (see Figure 2).

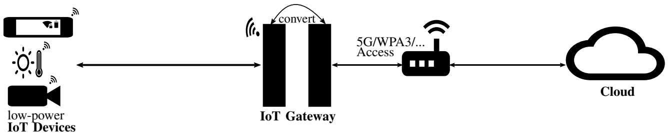


Figure 2. Establishing a connection between the IoT crypto gateway and the IoT device.

Precisely, this project aims to reduce the required security computations for the IoT devices by implementing MQTT over IETF QUIC in the IoT devices and developing an IoT crypto gateway, which has the ability to convert the communication from TCP-TLS-MQTT, which is the actual/common case, to QUIC-MQTT and vice versa.

IV. PROPOSED ARCHITECTURE

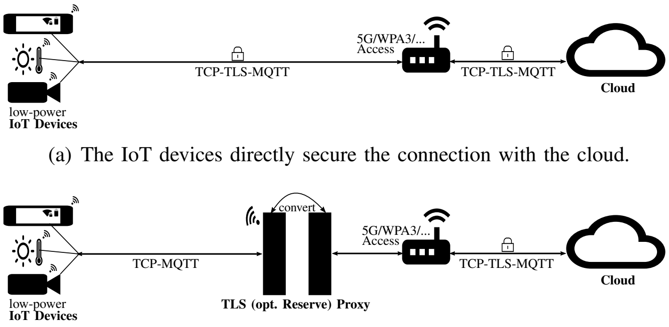
The gateway developed should perform as a translator between the IoT devices and the cloud using common protocols with the cloud and more efficient/suitable protocols with the IoT devices in order to save energy and improve performance.

As mentioned earlier in Section I, one of the biggest challenge of securely attaching IoT devices to cloud services is to achieve a high security level using only low resources. The storage and processing capabilities of an IoT device are restricted by the resources available, which are, for example, constrained due to size limitation, energy, and computational capability. Thus, these systems rely on IoT middleware to provide needed capabilities [34].

Traditionally, IoT devices may be connected to the cloud implementing two ways (see Figure 3). First, the IoT devices may have the ability to securely communicate via TLS directly with the cloud (see Figure 3(a)). In this way, both peers can perform a direct TLS-handshake between each other. Hence, the data can be secured in the private as well as the public network.

Second, the IoT devices might be connected to the cloud through a TLS (opt. reserve) proxy implementing a web server (e.g., NGINX), which only secures the data before emerging out to the public network (see Figure 3(b)). Thus, the connection between the proxy and the cloud is secured via TLS, and data between the proxy and the IoT devices is transmitted without TLS. The proxy aims to reduce the risks on the IoT devices by securing the data only in the public network and to save the resources of the IoT devices by decrypting the data before emerging in the private network [35].

Both communication scenarios have their drawbacks. By implementing the one in Figure 3(a), the IoT devices have to establish an (expensive) secured tunnel which is – at a high security level – not suitable for constrained IoT devices [15] [36]. Furthermore, by implementing the scheme in Figure 3(b), the connection between the proxy and the IoT devices does not provide the security requirements mentioned in Section I-A. Additionally, some attacks, such as DDoS and MITM, are possible on the network [37] [38].



(a) The IoT devices directly secure the connection with the cloud.

(b) A TLS (opt. reserve) proxy between the IoT devices and the cloud.

Figure 3. Illustrations of how IoT devices may secure the connection with a cloud service.

To circumvent both problems, we present the following architecture: the IoT crypto gateway stands between the cloud and the IoT devices and communicates with the cloud as a client and with the IoT devices as a cloud, as shown in Figure 2. When an IoT device attempts to connect to a cloud service in order to send or request some data, it first connects to the Internet using one of the Internet access protocols, such as 5G or WPA3. The IoT crypto gateway creates an Internet connection with the IoT device and starts to establish it in order to receive the data from the IoT device and transmit it to the cloud. Since QUIC does not support all TLS versions, the gateway is restricted to secure the communication with the IoT devices using TLS 1.3 and above. On the other side, the gateway secures the communication implementing TLS 1.2 and above. However, for the reason that the transport layer (TCP-like) and TLS are integrated in QUIC, the IoT devices exchange less packets with the gateway. Hence, the battery life, the CPU computations and the resource usage in the IoT devices side may be better optimized. We summarize the benefits of our approach in TABLE I.

TABLE I. THE PROPOSED IMPLEMENTATION COMPARED WITH TRADITIONAL CONNECTIONS SECURED DIRECTLY WITH TLS.

#	IoT devices secured via QUIC	via TLS directly
Security	high	high
Latency	lower	longer
Resource usage	lower	higher
Battery life	longer	shorter
Computations	lower	higher

The IoT crypto gateway establishes the connection using TCP and communicates with the cloud implementing MQTT over TLS (see Figure 4). At the same time, the IoT crypto gateway communicates with the IoT devices implementing MQTT over IETF QUIC (+ TLS). Thus, the crypto gateway should perform with both peers and transmit the packets almost simultaneously.

Assuming an MQTT-Publish message must be sent from one of the IoT devices to the cloud. Since TLS is integrated in

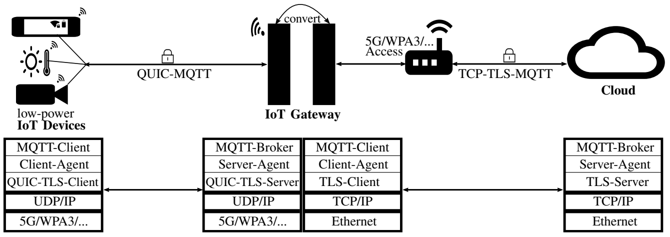
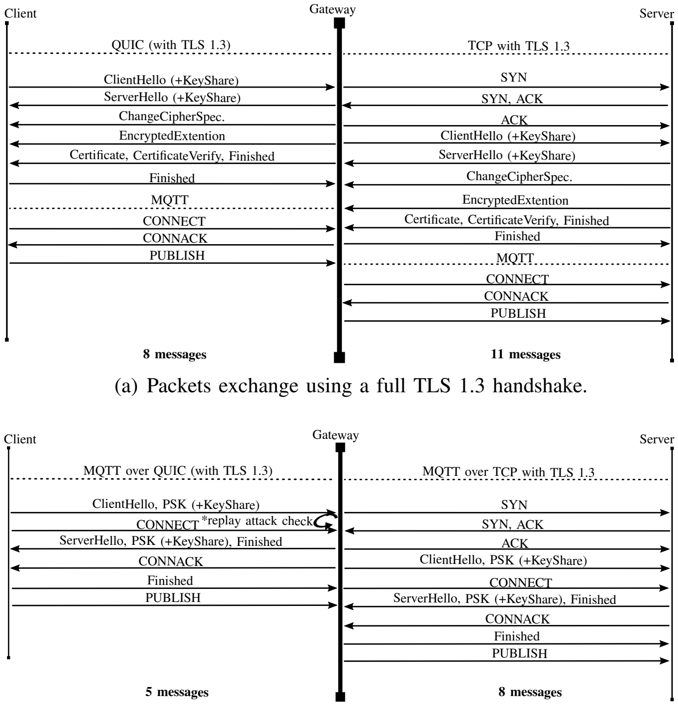


Figure 4. The IoT crypto gateway secures the connection between both peers implementing different layer protocols.

QUIC, the client can start to communicate with the gateway by sending its first packet ClientHello (CH), which is contained in the first QUIC message and should then be resent to the cloud. The gateway initiates establishing a TCP connection with the cloud and sends its CH message. Additionally, the gateway checks the CH packet sent from the client and performs a full TLS handshake if there was no previous connection with the peers before and a resumed TLS handshake using PSKs if the peers have connected with each other before. As a server, the gateway completes establishing the QUIC connection with the IoT device. Furthermore, as a client, the gateway completes the connection with the server implementing TLS over TCP (see Figure 5). The gateway may perform mutual authentication with both peers in order to hand high security for the IoT devices and may frequently use PSKs (TLS-PSK) with the IoT devices in order to optimize their performance. In addition, the gateway communicates with both peers individually and may therefore use different TLS versions, parameters and RTTs at the same time. Finally, the IoT device can communicate MQTT and send its MQTT-Publish message to the gateway, which will be sent to the server.



(a) Packets exchange using a full TLS 1.3 handshake.

(b) Packets exchange using a TLS-PSK 1.3 handshake.

Figure 5. Illustrations of the IoT crypto gateway packets exchange between the IoT client and the cloud service assuming only one side authentication.

In case of using the TLS-PSK mechanism, the IoT gateway should check if the connection is a replay attack against the cloud and interrupt the connection/return back to a full TLS handshake if it is needed. In order to discover a replay attack, the IoT crypto gateway should implement one of the following three mechanisms: saving the session tickets which can be used once only and rejecting duplicates, recording a unique value (e.g., the random value) derived from the CH packets and refusing duplicates, or refusing old packets by checking the time in the CH packets to efficiently determine whether a CH was sent recently or it was an old packet. Furthermore, the IoT crypto gateway may check the validation of the PSKs, HMACs and signatures and interrupt/retry the connection if it is needed [15] [18].

V. Conclusion and Future Work

This paper proposed a cryptographic gateway between low-power IoT devices and a cloud service, which connects the device to the cloud service with a high security level while at the same time saving considerable resources on the side of IoT devices by using a transparent cryptographic gateway.

The proposed gateway opens in direction of the cloud a fully-fledged authenticated TLS tunnel and in direction of the IoT device a TLS connection using the new (IETF) QUIC protocol which exchanges less packets and employs after the first handshake a PSK. As a result, peers are able to establish a TLS connection with less resources for the IoT devices. Thus, the gateway may save time, power and computation on the IoT device's side without compromising security.

The QUIC protocol is still a work in progress by IETF, which forces adding changes in this project continously and makes the implementation of it difficult. Cases, such as authentication and certificates handling between the peers, are still under research and development. Nevertheless, as a next step, a proof of concept implementation is in plan.

Acknowledgement

This paper is a part of the BMBF-funded project "Lernlabor Cybersicherheit" at Ostbayerische Technische Hochschule (OTH) Amberg-Weiden in cooperation with Fraunhofer AISEC.

References

[1] F. Alkhabbas, R. Spalazzese, M. Cerioli, M. Leotta, and G. Reggio, "On the Deployment of IoT Systems: An Industrial Survey," in 2020 IEEE International Conference on Software Architecture Companion (ICSA-C), May 2020, pp. 17–24, retrieved: August 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9095740

[2] Research and Markets (The world's largest market research store), "Industrial Internet of Things (IIoT) Market Size, Share & Trends Analysis Report by Component, by End Use (Manufacturing, Energy & Power, Oil & Gas, Healthcare, Logistics & Transport, Agriculture), and Segment Forecasts, 2019 - 2025," June 2019, retrieved: August 2020. [Online]. Available: https://www.researchandmarkets.com/reports/4240418/industrial-internet-of-things-iiot-market-size

[3] Federal Bureau of Investigation, "Internet of Things Poses Opportunities for Cyber Crime," Federal Bureau of Investigation, September 2015.

[4] BSI, "BSI - SYS: IT-Systeme - SYS.4.4 Allgemeines IoT-Gerät," retrieved: August 2020. [Online]. Available: https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKompendium/bausteine/SYS/SYS_4_4_Allgemeines_IoT-Ger%C3%A4t.html

[5] BSI, "BSI - IT-Grundschutz-Kompendium - Umsetzungshinweise zum Baustein SYS.4.4 Allgemeines IoT-Gerät," retrieved: August 2020. [Online]. Available: https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKompendium/umsetzungshinweise/

SYS/Umsetzungshinweise_zum_Baustein_SYS_4_4_Allgemeines_IoT-Ger%C3%A4t.html

[6] BSI, “Sicherheit von Geräten im Internet der Dinge,” retrieved: August 2020. [Online]. Available: https://www.allianz-fuer-cybersicherheit.de/ACS/DE/_/downloads/BSI-CS/BSI-CS_128.pdf?__blob=publicationFile&v=10

[7] W. HuiYu, Q. Wenxiang, and L. Yuxiang, “Breaking Smart Speaker: We are Listening to You,” August 2018, Tencent Blade Team.

[8] A. Hameed and A. Alomary, “Security Issues in IoT: A Survey,” in 2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), 2019, pp. 1–5, retrieved: August 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8910320

[9] I. Alqassem and D. Svetinovic, “A Taxonomy of Security and Privacy Requirements for the Internet of Things (IoT),” in 2014 IEEE International Conference on Industrial Engineering and Engineering Management. Selangor Darul Ehsan, Malaysia: IEEE, December 2014, pp. 1244–1248, retrieved: August 2020. [Online]. Available: http://ieeexplore.ieee.org/document/7058837/

[10] S. Oh and Y. Kim, “Security Requirements Analysis for the IoT,” in 2017 International Conference on Platform Technology and Service (PlatCon), March 2017, pp. 1–6, retrieved: August 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7883727

[11] U. Singh and I. Chana, “Enhancing Energy Efficiency in IoT (Internet of Thing) Based Application,” in Inventive Computation Technologies, ser. Lecture Notes in Networks and Systems, S. Smys, R. Bestak, and A. Rocha, Eds. Cham: Springer International Publishing, November 2019, pp. 161–173.

[12] Z. Zhang et al., “IoT Security: Ongoing Challenges and Research Opportunities,” in 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications, 2014, pp. 230–234, retrieved: August 2020. [Online]. Available: https://ieeexplore.ieee.org/document/6978614

[13] G. Matsemela, S. Rimer, K. Ouahada, R. Ndjiongue, and Z. Mngomezulu, “Internet of Things Data Integrity,” in 2017 IST-Africa Week Conference (IST-Africa), November 2017, pp. 1–9, retrieved: August 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8102332

[14] R. Roman, C. Alcaraz, J. Lopez, and N. Sklavos, “Key Management Systems for Sensor Networks in the Context of the Internet of Things,” Computers & Electrical Engineering, vol. 37, no. 2, March 2011, pp. 147–159.

[15] E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.3,” Internet Engineering Task Force, August 2018, RFC 8446, retrieved: August 2020. [Online]. Available: https://tools.ietf.org/html/rfc8446

[16] K. Moriarty and S. Farrell, “Deprecating TLSv1.0 and TLSv1.1,” Internet Engineering Task Force, Internet-Draft, January 2020, work in Progress,retrieved: August 2020. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-tls-oldversions-deprecate

[17] J. Iyengar and M. Thomson, “QUIC: A UDP-Based Multiplexed and Secure Transport,” Internet Engineering Task Force, Internet-Draft, June 2020, Work in Progress, retrieved: August 2020. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29

[18] M. Thomson and S. Turner, “Using TLS to Secure QUIC,” Internet Engineering Task Force, Internet-Draft, June 2020, work in Progress, retrieved: August 2020. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-quic-tls-29

[19] P. Kumar and B. Dezfouli, “Implementation and analysis of QUIC for MQTT,” Computer Networks, vol. 150, February 2019, pp. 28–45.

[20] The Chromium Projects, “QUIC, a multiplexed stream transport over UDP - The Chromium Projects,” library Catalog: www.chromium.org, retrieved: August 2020. [Online]. Available: https://www.chromium.org/quic

[21] E. Rescorla and N. Modadugu, “Datagram Transport Layer Security Version 1.2,” RFC 6347, Internet Engineering Task Force, June 2012, retrieved: August 2020. [Online]. Available: https://rfc-editor.org/rfc/rfc6347.txt

[22] E. Rescorla, H. Tschofenig, and N. Modadugu, “The Datagram Transport Layer Security (DTLS) Protocol Version 1.3,” Internet Engineering Task Force, Internet-Draft, May 2020, work in Progress, retrieved: August 2020. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-tls-dtls13-38

[23] M. B. Yassein, M. Q. Shatnawi, and D. Al-zoubi, “Application Layer Protocols for the Internet of Things: A Survey,” in 2016 International Conference on Engineering MIS (ICEMIS), September 2016, pp. 1–4.

[24] A. Talaminos-Barroso, M. A. Estudillo-Valderrama, L. M. Roa, J. Reina-Tosina, and F. Ortega-Ruiz, “A Machine-to-Machine Protocol Benchmark for eHealth Applications – Use Case: Respiratory Rehabilitation,” Computer Methods and Programs in Biomedicine, vol. 129, June 2016, pp. 1–11.

[25] A. Banks and R. Gupta, “MQTT Version 3.1.1,” October 2014, retrieved: August 2020. [Online]. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html

[26] Mitmproxy, “mitmproxy - an interactive HTTPS proxy,” retrieved: August 2020. [Online]. Available: https://mitmproxy.org/

[27] NGINX, “nginx-quic: log,” retrieved: August 2020. [Online]. Available: https://hg.nginx.org/nginx-quic/shortlog

[28] M. Bishop, “Hypertext Transfer Protocol Version 3 (HTTP/3),” Internet Engineering Task Force, Internet-Draft, June 2020, work in Progress, retrieved: August 2020. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-quic-http-29

[29] NGINX, “Introducing a Technology Preview of NGINX Support for QUIC and HTTP/3,” June 2020, library Catalog: www.nginx.com, retrieved: August 2020. [Online]. Available: https://www.nginx.com/blog/introducing-technology-preview-nginx-support-for-quic-http-3/

[30] OpenSSL, “QUIC and OpenSSL,” February 2020, retrieved: August 2020. [Online]. Available: https://www.openssl.org/blog/blog/2020/02/17/QUIC-and-OpenSSL/

[31] D. Stenberg, “QUIC with wolfSSL,” June 2020, retrieved: August 2020. [Online]. Available: https://daniel.haxx.se/blog/2020/06/18/quic-with-wolfssl/

[32] A. Tewari and B. B. Gupta, “A Robust Anonymity Preserving Authentication Protocol for IoT Devices,” in 2018 IEEE International Conference on Consumer Electronics (ICCE), January 2018, pp. 1–5, retrieved: August 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8326282

[33] J. King and A. I. Awad, “A Distributed Security Mechanism for Resource-Constrained IoT Devices,” Informatica, vol. 40, no. 1, February 2016, retrieved: August 2020. [Online]. Available: http://www.informatica.si/index.php/informatica/article/view/1046

[34] W. Razouk, D. Sgandurra, and K. Sakurai, “A New Security Middleware Architecture based on Fog Computing and Cloud to Support IoT Constrained Devices,” October 2017, pp. 1–8.

[35] NGINX, “Improve IoT Security with NGINX Plus: Encrypt & Authenticate MQTT,” March 2017, library Catalog: www.nginx.com, retrieved: August 2020. [Online]. Available: https://www.nginx.com/blog/nginx-plus-iot-security-encrypt-authenticate-mqtt/

[36] O. Rajaee, “IoT, Resource Constrained Devices, Security,” February 2017, conference: RSA 2017, at: San Francisco, CA.

[37] S. Andy, B. Rahardjo, and B. Hanindhito, “Attack Scenarios and Security Analysis of MQTT Communication Protocol in IoT System,” in 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), 2017, pp. 1–6.

[38] G. Perrone, M. Vecchio, R. Pecori, and R. Giaffreda, “The Day After Mirai: A Survey on MQTT Security Solutions After the Largest Cyber-attack Carried Out through an Army of IoT Devices:,” in Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security. Porto, Portugal: SCITEPRESS - Science and Technology Publications, 2017, pp. 246–253.