MPO: MQTT-Based Privacy Orchestrator for Smart Home Users

Ahmed Alhazmi, Khulud Alawaji and TJ OConnor
Computer Engineering and Sciences
Florida Institute of Technology
Melbourne, United States of America
aalhazmi2017@my.fit.edu, kalawaji2017@my.fit.edu, toconnor@fit.edu

***Abstract*—Security and privacy concerns present the most significant obstacles to consumer adoption of Internet-of-Things (IoT) devices. A lack of transparency and control complicates user trust in IoT. Additionally, a growing history of misuse and abuse exists in IoT. Notably, smart TVs have periodically scanned and collected users' private information without consent. Due to a hybrid of distributed ecosystems within IoT, users cannot easily implement traditional access control over their devices as data flows within different nodes for storage and processing. We propose MQTT-Based Privacy Orchestrator (MPO) to implement traditional access control on IoT devices. MPO enforces privacy preferences by implementing access control at an MQTT broker. We open-source and provide MPO as an add-on to the popular Home Assistant open-source home automation framework to support widespread adoption. We conducted experimental evaluations to validate the functionality and examine the performance of MPO. Our performance evaluation generated more than 16,686 messages, which MPO delivered in under a second. Our work demonstrates a practical solution to facilitate users' privacy preferences and enforce access control for MQTT-based devices.**

***Index Terms*—Privacy, Access Control, MQTT, IoT**

I. Introduction

The Internet of Things (IoT), which connects users to a wide array of sensors and actuators, has grown exponentially over the last decade, with 75 billion devices estimated in use by 2025 [1], [2]. Security and privacy concerns present the largest obstacles to consumers' adoption of IoT devices [3], [4]. Smart home IoT sensors collect our most personal and intimate moments, magnifying this concern [5]. The opaque nature of IoT obscures device actions and leaves the user unable to control access to devices. IoT-enabled attacks are growing in a market rife with vulnerabilities [6].

We present the MQTT-Based Privacy Orchestrator (MPO) to address a lack of control over privacy. Our approach translates consumers' privacy preferences into access control rules for MQTT-enabled IoT devices. MPO preserves users' privacy preferences from unauthorized access by enforcing access control at the Message Queuing Telemetry Transport (MQTT) broker. We make the following contributions in this paper:

1) We propose MPO to assist users in enforcing their privacy preferences within a few steps. MPO grants users the ability to easily control the flow of their IoT-generated data in their smart homes. MPO prevents unauthorized access to IoT devices from both users and other IoT devices.
2) Through our approach, we offer an open-source access control implementation mechanism to preserve users' privacy preferences. We implement MPO on top of the popular Home Assistant open-source home automation framework and provide the source at *https://github.com/AKEHZ/MPO.*
3) We conduct experimental evaluations of MPO. Our evaluation demonstrates that MPO can enforce access control to preserve privacy preferences without significantly impacting performance.

II. Motivation And Problem

The distributed nature of IoT ecosystems abstracts transparency and control away from consumers, leading to misuse and abuse. To motivate this problem, consider the case of the Skyworth smart TV that was scanning its users' network periodically, collecting a ton of personal information, and sending it back to a data analytics company [7]. This incident affected millions of users who remained unaware of the smart TV's actions performed without users' consent. Shortly after a user exposed the TV's behaviors, the TV company issued a statement apologizing for what happened and blaming the data analytics company for violating users' privacy [8].

This growing abuse of trust is indicative of the opaque nature of IoT devices. Due to the distributed nature of IoT, consumers are unaware of how their devices are controlled or by whom. This incident motivates our work by identifying a need to return transparency and control to consumers, who should have the right to control how their IoT devices collect, share, and offer access to data. MPO addresses this lack of control by enforcing access control at an MQTT broker. Under our proposed solution, the consumer controls *who* and *how* data is accessed. MPO enforces fine-grained access control at the user and device levels.

III. Overview

MPO translates users' privacy preferences into Access Control Lists (ACLs). MPO enforces the resulting access control rules at the local network layer over the lightweight MQTT protocol. MPO implements access control rules as a series of privacy profiles. As depicted in Fig. 1, MPO utilizes

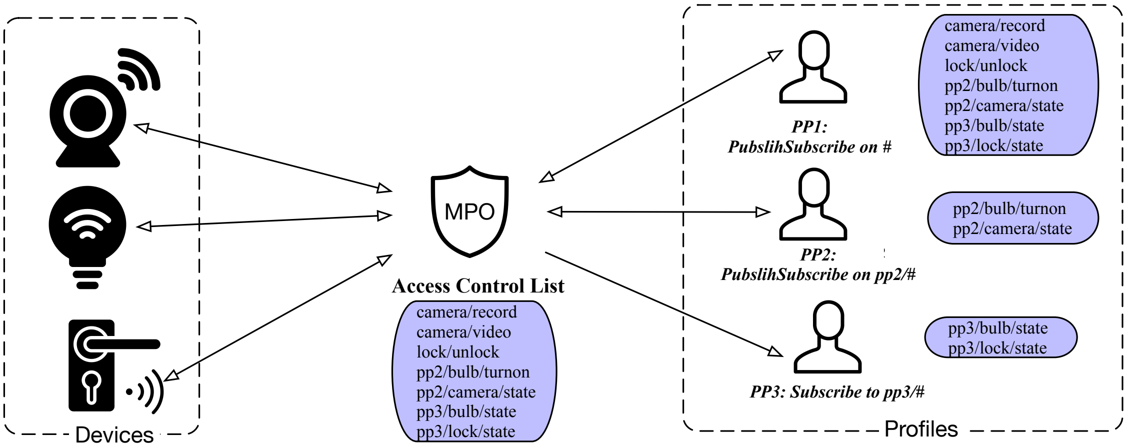


Fig. 1: MPO implements access control for IoT ecosystems, allowing users control of who and how devices and their data are accessed.

three privacy profiles to protect against unauthorized access to IoT devices and data. The device owner controls the first and primary profile, which contains full access to the device and permits both publish and subscribe actions. The device owner may allow discretionary access through secondary and tertiary profiles. The secondary profile grants both publish and subscribe access for a limited number of IoT devices. In contrast, the tertiary profile grants only subscribe access to an IoT device(s). By default, MPO denies any access not explicitly granted. In Fig. 1, MPO grants the primary profile owner access to all MQTT topics for the smart camera, smart bulb, and smart door lock devices. The secondary profile user gets access to the smart camera's state topic and can send a command to turn on the smart bulb via "pp2/bulb/turnon" topic. The tertiary privacy profile user gets access to the state of the smart bulb and smart door lock and cannot control any of the three IoT devices. Further, MPO is fully implemented on Home Assistant [9].

A. Challenges

Access Control: Ensuring each user gets the appropriate access to IoT data presents a challenge. A lack of access control can result in attackers compromising users' private information, leading to harmful consequences such as burglary or domestic violence [10], [11]. Another challenging component in an IoT system is the devices' access, where uncontrolled access may lead to private information leakage. Conversely, deprived IoT device(s) cannot provide the type of functionality that their owner seeks.

User Management: IoT devices must authenticate themselves when connecting to the MQTT broker in a secure MQTT environment. Generating and managing the credentials for these devices can be challenging to users, which can lead to poor security practices such as using the same password frequently as a way to mitigate the challenge of having several credentials to manage [12], [13]. Prior work [12], [14], [15], [16] also showed that users tend to have a *unique number* of passwords that they use across all their accounts.

TABLE I: Privacy profiles and their access levels.

	Access Level
Privacy Profile 1	*topic readwrite #*
Privacy Profile 2	*topic readwrite pp2/#*
Privacy Profile 3	*topic read pp3/#*

IV. Design

A. Authentication and User Management

The Mosquitto broker add-on can successfully authenticate clients that utilize existing users' credentials stored on Home Assistant or its local database [17]. Therefore, MPO requires users to create three user accounts for each privacy profile on Home Assistant because MPO does not need to acquire or maintain the credentials of these profiles. On the other hand, the user/owner needs to provide the credentials of privacy profile 2 or 3 to other users to whom they would like to give data access.

We automate the process of generating a unique credential for each IoT device that connects to the MQTT broker to address the user management challenge we explained in section III-A. In other words, MPO generates and manages the credentials of all IoT devices that will connect to the Mosquitto broker. MPO stores these credentials in Mosquitto's local database to allow the broker to authenticate the IoT devices. The user will utilize the generated credentials for the new smart device to authenticate and connect it to the MQTT broker. The user can recall the credentials of any IoT device they have via MPO's configuration section. Once an IoT device is no longer in use, MPO will delete the device's credentials, ensuring the device cannot connect to the MQTT broker. MPO creates device labels based on distinct user-assigned device names. For example, when the primary profile owner of the three smart devices in Fig. 1 has a new smart lock they want to add to their smart home system, the user will edit MPO's configuration for adding this device and giving it a unique name, e.g. lock3. MPO will use the "lock3" name as a username for the new door lock. Once the user runs MPO, it will generate and maintain a random password for the new "lock3" door lock in case the user resets the device and wishes to reconnect it with the Mosquitto broker.

B. MPO's Configuration

Accessible Data for PP2 Users: MPO's owner can use the "pp2_two_publish" and "pp2_two_subscribe" options to determine the MQTT data and control they feel comfortable in sharing with a secondary profile user.

Accessible Data for PP3 Users: The option "pp_three_subscribe" allows MPO's owner to identify the data a tertiary profile user can access. This option ensures MPO's owner that tertiary profile users will be denied from accessing any other data and cannot control any IoT device as well.

Accessible Data for IoT Devices: When adding a new device to the smart home system, MPO's owner defines the topics this device can access/publish to via "IoTDevices_Publish" and "IoTDevices_Subscribe". These options ensure MPO's owner that the new IoT device will be unable to leak potentially private information about the user as its access level is limited to its assigned topics. Identifying the ACL for each IoT device addresses the under/over-privileges challenge that we described in section III-A.

Device_Credentials: MPO's owner uses this option to retrieve or acquire the credentials of their IoT devices.

C. Privacy Profiles

MPO harnesses three privacy profiles to provide access control capabilities to users. Table I demonstrates the ACL level that MPO assigns for each privacy profile. The owner of MPO can control all IoT devices via the primary profile. They can also determine the type of data and control that the other privacy profiles have. For instance, MPO's owner can authorize the secondary profile users to access data sent to the "bulb1/state" topic. MPO will then deliver this data to "pp2/bulb1/state" through the utilization of its *delivery agents*. These agents are MQTT clients that connect to the MQTT broker, deliver IoT-generated data to the authorized privacy profile (2, 3, or both) topics, and then disconnect from the MQTT broker. We leveraged "Eclipse's Paho" [18] to generate MPO's delivery agents. Besides transporting data to the secondary privacy profile users, delivery agents forward these users' commands to control IoT devices. Typically, the secondary user sends these commands to the respected topics that begin with "pp2/...", and the delivery agents will forward them to the right destinations. A second privacy profile user may be allowed to turn the light on in the living room. The user will send an *on* command to the topic (*pp2/livingRoom/light/set*), which will be relayed by the delivery agent to the smart bulb's subscribed topic (*livingRoom/light/set).* The need for delivery agents to relay these commands is due to how various IoT devices are designed to subscribe to a predefined set of MQTT topics [19], [20]. Such devices will not be able to subscribe to secondary profile (PP2) and tertiary profile (PP3) topics.

TABLE II: Devices that were used in the real-world experiment part.

Vendor	Type	Model
Athom	Smart Bulb	7W Color Bulb
Yale	Smart Door Lock	YRD226-NR-619
SONOFF	Smart Temperature and Humidity Device	SNZB-02
INSTAR	Smart Camera	IN-9008 Full HD
SONOFF	Smart Plug	S-31

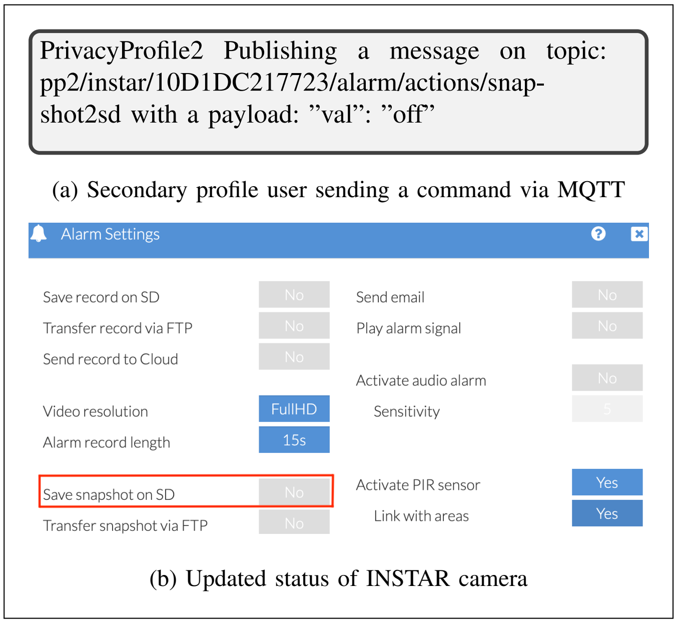
V. Evaluation

A. Real-World Experiment

We tested MPO in a real-world environment where we experimented with five different types of IoT devices: *smart camera, smart door lock, smart temperature and humidity device, smart plug, and smart bulb*. We chose these devices as they can be representative of the most popular types of IoT devices in current use in users' smart homes [21], [22], [23]. These devices are available at the stores of major US retailers such as Walmart, Best Buy, and Amazon. Table II demonstrates the list of the devices we used for this experiment. In terms of their MQTT support, Athom [24] and INSTAR Camera [25] were designed to support MQTT communication out-of-the-box. We used the Zigbee2MQTT[1] tool to be able to communicate with the Yale door lock [26] and SONOFF temperature and humidity device [27] through MQTT. We installed the Zigbee2MQTT add-on to use this tool in our experiment. In addition, although it was possible to use Zigbee2MQTT with SONOFF S-31 smart plug [28] for MQTT communications, we decided to flash this device with Tasmota Firmware [29] to include all the possible ways through which a user can utilize an IoT device to connect via MQTT and communicate with MPO to ensure that MPO works properly with all IoT devices that can support local MQTT communications.

Our goal was to validate MPO's functionality with on-the-market already-in-use IoT devices. We began the experiment as any MPO user would by adding the five devices and their MQTT topics to MPO's configuration section. We included all the topics for these devices except for INSTAR's topics. INSTAR camera uses more than 245 topics. Various topics customize the camera's functionality, such as adjusting the quality of the photos/videos taken by the camera to the users' liking. Hence, we selected 60 topics for the INSTAR camera and added them in MPO. Then, we ran MPO to allow for the generation of credentials that we are going to use when connecting each device to the Mosquitto broker. When MPO starts, it adds the appropriate ACL implementation for all the devices that are available in its configuration section. We allow secondary and tertiary profile users to subscribe to five topics; each belongs to one of the five smart devices. We also allow the secondary profile users to have limited control over the

[1] https://www.zigbee2mqtt.io/



PrivacyProfile2 Publishing a message on topic: pp2/instar/10D1DC217723/alarm/actions/snapshot2sd with a payload: ”val”: ”off”

(a) Secondary profile user sending a command via MQTT

Alarm Settings

Save record on SD	No
Transfer record via FTP	No
Send record to Cloud	No
Video resolution	FullHD
Alarm record length	15s
Save snapshot on SD	No
Transfer snapshot via FTP	No
Send email	No
Play alarm signal	No
Activate audio alarm	No
Sensitivity	5
Activate PIR sensor	Yes
Link with areas	Yes

(b) Updated status of INSTAR camera

Fig. 2: Successful command published by the secondary profile user to control the INSTAR camera’s option of saving snapshots of future detected motion incidents into the SD card.

INSTAR camera and smart plug devices through publishing a few topics to control these devices.

In this experiment, we utilized Eclipse’s Paho [18] to generate three MQTT clients representing a user for each privacy profile: primary, secondary, and tertiary. As a result, MPO successfully implemented the correct ACL implementation for all three privacy profile users. The primary profile user was able to access all MQTT topics published by the five smart IoT devices. The secondary profile user received all the data published on the topics they were allowed to access. Similarly, the tertiary profile user was able to acquire all information published on topics they could access. The secondary profile user was successfully able to have only a little control over the INSTAR camera and SONOFF plug as the user was allowed to turn on/off the plug, turn on/off manual recording on the camera, and control whether the camera would save a snapshot of any detected motion into the external SD card. Fig. 2 shows how the secondary profile user successfully controlled the save-snapshot-to-SD option in the INSTAR camera and turned it off.

B. Performance

Measuring the performance of an MQTT-based tool such as MPO entails focusing on two aspects that can be critical to smart home users: the successful delivery and the transmission time of MQTT messages. Researchers in [30] were able to capture and label 16,686 events generated in their IoT system during one week. We designed our performance test to include at least 16,686 MQTT messages as this number represents events in an IoT system.

We have conducted two experiments for MPO’s performance evaluation. In the first experiment, we did not include MPO’s enforcement of TLS connection between clients and the broker. We wanted to understand whether enforcing connections via TLS would negatively affect MPO’s performance. To do so, we replicated the same experiment but with the inclusion of MPO’s TLS enforcement. In both experiments, we simulated the five IoT devices and their MQTT topics from the real-world experiment in section V-A except for the Athom bulb. Athom bulb and SONOFF plug share similar MQTT topics’ structures. Thus, we simulated Shelly [31] smart bulb topics to include different MQTT topic’s structures in this experiment. We programmed the five simulated devices to send 16,686 MQTT messages. Each experiment took almost 28 hours to complete. We used eight MQTT clients in these experiments. We classified them as follows: five MQTT clients to simulate the IoT devices, one client for the primary profile user, one client for the secondary profile user, and one client for the tertiary profile user. As a result, we published 25,056 MQTT messages in the first experiment and 25,141 in the second experiment. We analyzed the transmission time each MQTT message took to reach its recipient. The goal was to examine MPO’s performance and assess whether it caused any delay to the MQTT system. We also wanted to analyze the time MPO’s agents took to deliver the messages to the secondary and tertiary profile users and the IoT devices, which can occur when a secondary profile user sends a command to control these devices. Table III records the analysis of the transmission time of all MQTT messages for both experiments.

When examining the results of the first MPO’s performance experiment shown in Table III, we made a few observations. First, more than 95% of the messages to which secondary and tertiary profile users subscribed were delivered almost instantly by MPO’s agents. Similarly, MPO’s agents instantly delivered the messages published by the secondary profile user to control the IoT devices. We observed instantaneous delivery of more than 96% of the 16,686 messages published by the simulated IoT devices. Moreover, less than 0.04% of the 25,056 MQTT messages took longer than one second to reach their intended recipients. We suspected the reason behind messages taking more than one second of transmission time to be related to a sleep function we utilized in our scripts for primary profile and secondary profile users to let them wait before publishing MQTT messages. Subsequently, we adjusted the timing of this function for these clients before performing the second experiment. As a result, all MQTT messages in the second experiment were successfully delivered within one second. Implementing TLS did not cause any negative impact on MPO’s performance. In fact, more messages were successfully delivered in less than a second during the second experiment than in the first experiment. Therefore, these observations showed that MPO does not pose any performance overhead on the MQTT system that is used with, as all messages were delivered successfully to the intended recipients within one second.

TABLE III: An analysis of the transmission time (in seconds) of all 25,056 (1st experiment) and 25,141 (2nd experiment) MQTT messages that were published during MPO's performance evaluation.

TLS Enforced?	**MQTT Messages**	**Publisher**	**Transmission Time**		
			<1s	=1s	>1s and ≤ 6s
×	16,686	Five IoT devices	16,089 (96.42%)	591 (3.54%)	6 (0.04%)
	4157	MPO's agents on pp2 topics	3989 (95.96%)	167(4.02%)	1(0.02%)
	4198	MPO's agents on pp3 topics	4027 (95.93%)	170 (4.05%)	1 (0.02%)
	15	MPO's agents on IoT devices' topics (5) PP1 user (5) PP2 user (5)	15 (100%)	0 (0%)	0 (0%)
✓	16,686	Five IoT devices	16,623 (99.62%)	63 (0.38%)	0 (0.00%)
	4194	MPO's agents on pp2 topics	4058 (96.76%)	136(3.24%)	0(0.00%)
	4246	MPO's agents on pp3 topics	4122 (97.08%)	124 (2.92%)	0 (0.00%)
	15	MPO's agents on IoT devices' topics (5) PP1 user (5) PP2 user (5)	15 (100%)	0 (0%)	0 (0%)

VI. Limitations

MPO is limited to the enforcement of IoT devices that leverage the MQTT protocol. We designed our implementation to support MQTT due to widespread support and adoption as a messaging protocol by both cloud-based IoT platform providers and IoT developers [32], [33]. As illustrated in our experiments with Zigbee2MQTT, we can extend support to unsupported devices by using several MQTT available bridges.

VII. Related Work

The closest work to MPO was proposed in [34] where the researchers built a stand-alone home automation system in which they included an ACL to work with their MQTT-based system. However, their work depends on accessing the MQTT data, storing it, and then delivering it encrypted to the authorized users. They relied on the security of MQTT protocol, which previous works have challenged [35], [36], [37]. Their approach does not prevent a malicious MQTT client from connecting to the broker and then acquiring the IoT-generated data. MPO controls the access to the MQTT broker through its implemented ACL and the credentials it creates and manages for the MQTT clients connected to the broker (see section IV-A and IV-B). In addition, their proposed system does not stop an adversary from accessing the special $SYS topics that can disclose the broker's specific information an adversary can use to compromise the broker. MPO prevents that by denying all clients from accessing these special topics except for the client that belongs to MPO's owner.

Using privacy profiles to capture users' privacy preferences has been presented in prior work [38]. Liu proposed a privacy assistant in [38] to develop privacy profiles and select the suitable profile for each user based on users' current settings and answers to the assistant's questions. Bahirat et al. [39] proposed a layered-based settings interface containing pre-built privacy profiles from which a user would choose one profile that suits their privacy needs. He [40] extended this approach with a new extension aimed at smart home users. The new layered-based interface follows the exact mechanism where a user will select a pre-built profile and then make further adjustments to its privacy settings. Unlike these approaches, we view and propose the idea of "privacy profiling" differently because we do not pre-build these profiles or ask users questions to build them. Our goal of these profiles is to help the users be the only entity accessing all devices' data while controlling and limiting the access level that other users and devices have. Other related work includes Kratos [41], an ACL system designed for multi-user smart homes. Kratos assumes that homeowners, occupants, and visitors can access the same smart hub controller/framework. This assumption allows for a malicious adversary to probe the smart home system and look for vulnerabilities to exploit. Kratos also does not account for domestic violence, where the victim can be tracked and monitored by their partner who has a user account with the same privileges. Kratos introduces a policy language for its users to specify their privacy and security needs, which can be overwhelming for users [42].

VIII. Future Work

Our plan for future work involves conducting a study to investigate the usability aspect of our tool. We hope to gain insightful feedback from users that would allow us to evaluate and improve MPO's usability. In addition, researchers have associated complex privacy settings with the emergence of the privacy fatigue phenomenon [43], [44]. Consequently, our future plan involves investigating how the design of MPO's privacy configuration can help in mitigating privacy fatigue among smart home users.

IX. Conclusion

In this paper, we presented MPO, a tool that takes control and transparency back to the end-users. MPO implements ACL on the device and user levels and enforces its ACL on the MQTT broker. MPO also ensures that users can enforce their privacy preferences in a few steps. We conducted experimental evaluations to test MPO's functionality and performance capabilities. MPO functions properly with on-the-market devices used by users in their smart homes. The performance evaluation concluded with all messages delivered successfully within one second.

Acknowledgements

This material is based upon work supported in whole or in part with funding from the Office of Naval Research (ONR) contract #N00014-21-1-2732. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ONR and/or any agency or entity of the United States Government.

References

[1] D. Kim, K. Park, Y. Park, and J.-H. Ahn, "Willingness to provide personal information: Perspective of privacy calculus in iot services," *Computers in Human Behavior*, vol. 92, pp. 273–281, 2019.

[2] "• Number of IoT devices 2015-2025 — Statista." [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[3] D. K. Alferidah and N. Jhanjhi, "A review on security and privacy issues and challenges in internet of things," *International Journal of Computer Science and Network Security IJCSNS*, vol. 20, no. 4, pp. 263–286, 2020.

[4] N. N. Thilakarathne, "Security and privacy issues in iot environment," *International Journal of Engineering and Management Research*, vol. 10, 2020.

[5] R. Xu, Q. Zeng, L. Zhu, H. Chi, X. Du, and M. Guizani, "Privacy leakage in smart homes and its mitigation: Ifttt as a case study," *IEEE Access*, vol. 7, pp. 63 457–63 471, 2019.

[6] "The Cybersecurity 202: Smart home devices with known security flaws are still on the market, researchers say - The Washington Post." [Online]. Available: https://www.washingtonpost.com/politics/2021/02/04/cybersecurity-202-smart-home-devices-with-known-security-flaws-are-still-market-researchers-say/

[7] "Millions of Chinese Smart TVs Scanned WiFi Every 10 Minutes, Sent Personal Data to Company." [Online]. Available: https://www.newsweek.com/millions-chinese-smart-tvs-scanned-wifi-every-10-minutes-sent-personal-data-company-1588322

[8] "COMPANY STATEMENT." [Online]. Available: https://www.skyworth.net/global/news-list/news12.html

[9] P. Schoutsen *et al.*, "Home assistant," 2017.

[10] "How your smart home devices can be turned against you - BBC Future." [Online]. Available: https://www.bbc.com/future/article/20200511-how-smart-home-devices-are-being-used-for-domestic-abuse

[11] S. Dong, Z. Li, D. Tang, J. Chen, M. Sun, and K. Zhang, "Your smart home can't keep a secret: Towards automated fingerprinting of iot traffic," in *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, 2020, pp. 47–59.

[12] R. Wash, E. Rader, R. Berman, and Z. Wellmer, "Understanding password choices: How frequently entered passwords are re-used across websites," in *Twelfth Symposium on Usable Privacy and Security ({SOUPS} 2016)*, 2016, pp. 175–188.

[13] G. Notoatmodjo and C. Thomborson, "Passwords and perceptions," in *Proceedings of the Seventh Australasian Conference on Information Security-Volume 98*. Citeseer, 2009, pp. 71–78.

[14] S. Fahl, M. Harbach, Y. Acar, and M. Smith, "On the ecological validity of a password study," in *Proceedings of the Ninth Symposium on Usable Privacy and Security*, 2013, pp. 1–13.

[15] D. Florencio and C. Herley, "A large-scale study of web password habits," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 657–666.

[16] D. Wang, D. He, H. Cheng, and P. Wang, "fuzzypsm: A new password strength meter using fuzzy probabilistic context-free grammars," in *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2016, pp. 595–606.

[17] "addons/mosquitto at master · home-assistant/addons · GitHub." [Online]. Available: https://github.com/home-assistant/addons/tree/master/mosquitto

[18] "Eclipse Paho — The Eclipse Foundation." [Online]. Available: https://www.eclipse.org/paho/

[19] "Availability and announces – API Reference." [Online]. Available: https://shelly-api-docs.shelly.cloud/#availability-and-announces

[20] "INSTAR MQTT Server for INSTAR Full HD Camera Models — INSTAR Wiki 2.0 — INSTAR Deutschland GmbH." [Online]. Available: https://wiki.instar.com/en/Advanced_User/INSTAR_MQTT_Broker/MQTT_API/

[21] R. Trimananda, J. Varmarken, A. Markopoulou, and B. Demsky, "Packet-level signatures for smart home devices," *Signature*, vol. 10, no. 13, p. 54, 2020.

[22] "Amazon Best Sellers: Best Home Automation Devices." [Online]. Available: https://www.amazon.com/Best-Sellers-Home-Improvement-Automation-Devices/zgbs/hi/6478739011

[23] "Amazon Best Sellers: Best Smart Home." [Online]. Available: https://www.amazon.com/Best-Sellers/zgbs/smart-home

[24] "Powered by ATHOM — Tasmota — ESPHome — Home Assistant — ESP8266 ESP32 —Smart Home." [Online]. Available: https://www.athom.tech//

[25] "IN-9008 Full HD WiFi white." [Online]. Available: https://www.instar.com/produkte/wlan-aussenkameras-poe-ip-kameras-wetterfeste-netzwerkkameras-outdoor-uberwachungskameras-fur-den-aussenbereich/in-9008-full-hd-series/in-9008-full-hd-wifi-white.html

[26] "Yale Assure Lock Touchscreen, Standalone - Yale Home." [Online]. Available: https://shopyalehome.com/collections/keypad-locks/products/yale-assure-lock-touchscreen-standalone?variant=39341912162436

[27] "SONOFF SNZB-02 - ZigBee Temperature And Humidity Sensor." [Online]. Available: https://sonoff.tech/product/smart-home-security/snzb-02/

[28] "SONOFF S31/S31LITE - Power Usage Monitor Plug Wifi Smart Socket." [Online]. Available: https://sonoff.tech/product/smart-plug/s31-s31lite/

[29] "Tasmota." [Online]. Available: https://tasmota.github.io/docs/

[30] D. Campos and T. OConnor, "Towards labeling on-demand iot traffic," in *Cyber Security Experimentation and Test Workshop*, 2021, pp. 49–57.

[31] "Shelly Duo." [Online]. Available: https://shelly.cloud/products/shelly-duo-smart-home-automation-bulb/

[32] "IoT Surveys — IoT development made simple - iot.eclipse.org." [Online]. Available: https://iot.eclipse.org/community/resources/iot-surveys/assets/iot-developer-survey-2020.pdf

[33] E. Al-Masri, K. R. Kalyanam, J. Batts, J. Kim, S. Singh, T. Vo, and C. Yan, "Investigating messaging protocols for the internet of things (iot)," *IEEE Access*, vol. 8, pp. 94 880–94 911, 2020.

[34] Y. Upadhyay, A. Borole, and D. Dileepan, "Mqtt based secured home automation system," in *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*. IEEE, 2016, pp. 1–4.

[35] A. Palmieri, P. Prem, S. Ranise, U. Morelli, and T. Ahmad, "Mqttsa: A tool for automatically assisting the secure deployments of mqtt brokers," in *2019 IEEE World Congress on Services (SERVICES)*, vol. 2642. IEEE, 2019, pp. 47–53.

[36] D. Dinculeană and X. Cheng, "Vulnerabilities and limitations of mqtt protocol used between iot devices," *Applied Sciences*, vol. 9, no. 5, p. 848, 2019.

[37] M. Harsha, B. Bhavani, and K. Kundhavai, "Analysis of vulnerabilities in mqtt security using shodan api and implementation of its countermeasures via authentication and acls," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2018, pp. 2244–2250.

[38] B. Liu, M. S. Andersen, F. Schaub, H. Almuhimedi, S. A. Zhang, N. Sadeh, Y. Agarwal, and A. Acquisti, "Follow my recommendations: A personalized privacy assistant for mobile app permissions," in *Twelfth Symposium on Usable Privacy and Security ({SOUPS} 2016)*, 2016, pp. 27–41.

[39] P. Bahirat, Y. He, A. Menon, and B. Knijnenburg, "A data-driven approach to developing iot privacy-setting interfaces," in *23rd International Conference on Intelligent User Interfaces*, 2018, pp. 165–176.

[40] Y. He, P. Bahirat, B. P. Knijnenburg, and A. Menon, "A data-driven approach to designing for privacy in household iot," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 10, no. 1, pp. 1–47, 2019.

[41] A. K. Sikder, L. Babun, Z. B. Celik, A. Acar, H. Aksu, P. McDaniel, E. Kirda, and A. S. Uluagac, "Kratos: multi-user multi-device-aware access control system for the smart home," in *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2020, pp. 1–12.

[42] K. Rannenberg, "How much negotiation and detail can users handle? experiences with security negotiation and the granularity of access control in communications," in *European Symposium on Research in Computer Security*. Springer, 2000, pp. 37–54.

[43] H. Choi, J. Park, and Y. Jung, "The role of privacy fatigue in online privacy behavior," *Computers in Human Behavior*, vol. 81, pp. 42–51, 2018.

[44] M. J. Keith, C. Maynes, P. B. Lowry, and J. Babb, "Privacy fatigue: The effect of privacy control complexity on consumer electronic information disclosure," in *International Conference on Information Systems (ICIS 2014), Auckland, New Zealand, December*, 2014, pp. 14–17.