Received August 23, 2021, accepted September 7, 2021, date of publication September 9, 2021,
date of current version September 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3111477

Defining the Behavior of IoT Devices Through the MUD Standard: Review, Challenges, and Research Directions

JOSÉ L. HERNÁNDEZ-RAMOS[1], SARA N. MATHEU[2], ANGELO FERAUDO[3], GIANMARCO BALDINI[1], (Senior Member, IEEE), JORGE BERNAL BERNABE[2], POONAM YADAV[4], ANTONIO SKARMETA[2], (Member, IEEE), AND PAOLO BELLAVISTA[3], (Senior Member, IEEE)

[1]Joint Research Centre, European Commission, 21027 Ispra, Italy
[2]Department of Information and Communication Engineering, University of Murcia, 30100 Murcia, Spain
[3]DISI, University of Bologna, 40126 Bologna, Italy
[4]Computer Science Department, University of York, York YO10 5DD, U.K.

Corresponding author: José L. Hernández-Ramos (jose-luis.hernandez-ramos@ec.europa.eu)

This work was supported in part by the European Commission under Project H2020-830929 CyberSec4Europe, Project H2020-780139 SerIoT, and Project H2020-952702 BIECO.

ABSTRACT With the strong development of the Internet of Things (IoT), the definition of IoT devices' intended behavior is key for an effective detection of potential cybersecurity attacks and threats in an increasingly connected environment. In 2019, the Manufacturer Usage Description (MUD) was standardized within the IETF as a data model and architecture for defining, obtaining and deploying MUD files, which describe the network behavioral profiles of IoT devices. While it has attracted a strong interest from academia, industry, and Standards Developing Organizations (SDOs), MUD is not yet widely deployed in real-world scenarios. In this work, we analyze the current research landscape around this standard, and describe some of the main challenges to be considered in the coming years to foster its adoption and deployment. Based on the literature analysis and our own experience in this area, we further describe potential research directions exploiting the MUD standard to encourage the development of secure IoT-enabled scenarios.

INDEX TERMS MUD, Internet of Things, security, IETF standards.

I. INTRODUCTION

With the increasing deployment of the Internet of Things (IoT), cybersecurity issues may have a broader scope and impact [1]. Indeed, the interconnection of physical devices to the Internet (which is one of the underlying aspects of IoT) may lead to an increase of the attack surface, as well as a more significant impact derived from potential threats and attacks. This aspect has been exploited by well-known attacks (e.g., Mirai or Hajime botnets [2]) that leverage vulnerable IoT devices to launch cyberattacks on other Internet devices and services. In spite of the prominent advances enhancing IoT security in recent years [3], these attacks highlight the need to improve existing attack detection and mitigation mechanisms in IoT-enabled environments.

The realization of an effective detection of security attacks in a specific IoT system or network requires identifying the expected behavior of each device composing such environment [4], [5]. Indeed, most of existing approaches based on machine learning techniques to improve IoT security [6] require the proper definition of devices' intended operation and behaviour to train the corresponding model. The concept is that events or communications, which are not part of the IoT device's *normal* behavior, can be considered as a potential threat or attack. From another point of view, a legitimate behaviour may be imposed on IoT devices. For example, rules can be defined and applied to determine how a device is deployed or connected to a network. For that purpose, specific network components may require adapting their operation to

The associate editor coordinating the review of this manuscript and approving it for publication was Ilsun You.

enforce restrictions associated with the intended operation of a new device. However, the application of these concepts is challenging due to the current heterogeneity of IoT devices, which are based on various technologies and communication protocols. Furthermore, the restrictions inherent to certain IoT devices (e.g., the lack of user interface) make management of IoT devices cumbersome for non-expert users. Therefore, the use of standard approaches to control the behaviour of IoT devices and networks is key to promote a secure and automated deployment and management of IoT technologies.

Moreover, some IoT operational environments and deployment scenarios have specific features that can make more difficult the detection of security attacks, and can increase the potential negative impact of such attacks or can require support for a long period. For example, Industrial IoT is characterized by systems and infrastructures, which are more complex and with a longer lifecycle than in the IoT consumer market [7], [8]. Then, the deployment of automated security solutions to improve the infrastructure management and extend the lifetime of IoT systems can be beneficial. Another example is the automotive sector (e.g., Intelligent Transport Systems [9]) where security risks can become safety risks (e.g., car accidents) if they are not properly managed [8]. In this context, the deployment of security solutions to control the behaviour of IoT devices and networks is essential to improve the reaction time by the infrastructure managers and the fast mitigation of vulnerabilities.

To cope with these challenges, the Manufacturer Usage Description (MUD) was standardized in 2019 within the scope of the Internet Engineering Task Force (IETF) [10]. MUD defines an architecture and data model to restrict the communication to/from a certain device. In particular, it provides manufacturers with the possibility to define network behavior profiles for their devices. Each profile is defined around a list of policies or Access Control Lists (ACLs) that define the endpoints of the intended communication to reduce the attack surface. Additionally, the proposed architecture allows obtaining this profile to be enforced by the network domain where the device is deployed. Since its adoption, MUD has received a significant interest from the research community and standardization bodies [11]. In particular, the National Institute of Standards and Technology (NIST) proposes the MUD standard as a promising approach to mitigate security threats [12], and to cope with denial-of-service (DoS) attacks in IoT environments, including home and small-business networks [13]. Additionally, the European Union Agency for Cybersecurity (ENISA) considers the use of MUD as part of IoT security good practices to improve, allowing devices to advertise their supported and intended functionality [14].

Based on the growing interest in the MUD standard from industry and academia, we provide a comprehensive analysis of the current landscape related to this emerging standard. It should be noted that existing surveys on MUD only cover partially some of the IoT security aspects addressed by the standard. In particular, [15] analyzed existing MUD research proposals for intrusion detection and prevention based on Software-Defined Networks (SDN) [16]. Furthermore, [17] describes the main implementation scenarios, applications and limitations of the MUD standard in relation to IoT devices' identification. Our analysis covers the study of current research proposals as well as existing MUD implementations and tools. Unlike previous works, we provide an up-to-date classification of existing proposals around the different stages of MUD profiles' lifecycle to help cybersecurity researchers to identify the different requirements and challenges for each process. Indeed, based on our own experience working with the MUD standard [18]–[21], we identify a set of challenges for the adoption of MUD, and provide potential research directions to be considered in the coming years. In fact, although MUD is widely considered as a promising approach, recent works address different limitations of the standard, or define new MUD-based applications to improve security in IoT scenarios. Due to the significant market growth of IoT devices, we believe that the use of a standardized approach such as MUD will be crucial to face existing and new security threats, as well as the heterogeneity of existing devices and technologies. Our work analyzes existing MUD-related proposals and provides insights on the potential deployment in the coming years. Therefore, it can be used as a reference for future research and standardization activities to evolve this standard. In particular, the contributions of this work are:

- Description of the MUD standard and analysis of the main stages of MUD behavioral profiles in relation to IoT devices' lifecycle
- Comprehensive analysis of existing research proposals, implementations and tools related to the MUD standard based on a proposed taxonomy
- Definition of the main challenges and future trends to be considered in the future years to cope with security issues in IoT-enabled scenarios based on the MUD standard and for the definition of IoT devices' behavioral profiles

The structure of this paper is the following: Section II describes the MUD standard, and the main processes associated to a MUD file throughout its lifecycle. Then, Section III is the main core of this work, in which we provide a comprehensive analysis of existing proposals based on the phases of such lifecycle. Furthermore, we analyze existing MUD-related implementations in Section IV with a specific focus on MUD tools. Based on the previous analysis and authors' experience on the area, Section V describes some of the main challenges and gaps in the MUD standard as well as potential research directions to be considered in the coming years to address such challenges and gaps. Finally, Section VI concludes this paper.

II. MUD STANDARD

As already described, the MUD standard is intended to define the expected behavior of a given IoT device by restricting its communications and/or network functions. Based on the

MUD specification [10], we provide an overview of the main architectural components and data models for defining MUD profiles. Furthermore, we define the main stages of a MUD profile throughout its lifecycle that will be used to classify the existing MUD research works in the following sections of this paper.

A. MUD ARCHITECTURE

The MUD architecture defines basic components for the deployment and use of a MUD file, which describes the device's behavior and is assumed to be defined by the device's manufacturer. As mentioned in [10], the notion of *manufacturer* is defined in a loose way in this context to refer to the entity or organization that will state how a device is intended to be used. Figure 1 shows such components, as well as the main interactions for obtaining a MUD file. The architecture includes a *Thing*, which represents the IoT device (in the rest of this paper the two terms are used as synonyms), and it is responsible for generating and transmitting a MUD URL; a *router* that provides network access to the device; the *MUD Manager*, which makes requests to obtain a MUD file based on the MUD URL received; and the *MUD File Server*, which hosts MUD files.

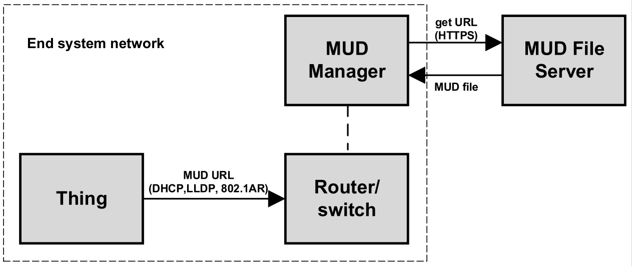

FIGURE 1. MUD architecture [10].

According to the MUD specification, the *Thing* or IoT device sends a MUD URL to the MUD Manager indicating where the corresponding MUD file is hosted. This communication is performed by the router, which forwards the MUD URL to the MUD Manager. Toward this end, the standard defines several options (e.g., through the 802.1AR standard [22]) depending on the scenario being considered. Then, the MUD Manager uses the MUD URL to request the MUD file server in order to retrieve the MUD file and its associated signature. After receiving it, the MUD Manager validates and parses the MUD file, and configures the corresponding network components (e.g., a router) with the network restrictions included in such file.

B. MUD DATA MODEL

The MUD standard restricts the communications of IoT devices through the definition of Access Control Lists (ACLs), which are defined using the *Yet Another Next Generation* (YANG) [23] standard to model network restrictions, and JavaScript Object Notation (JSON) [24] as the serialization format. Indeed, the MUD model contains extensions to the YANG data model for ACLs [25] to represent network access conditions with a high level of expressiveness. In particular, the model defines the container "mud" that provides information about the MUD file, such as where it is stored ("mud-url") or when it was generated ("last-update"). Additionally, the MUD data model describes the "acls" container based on [25], including additional restrictions, such as allowing or denying the communication with certain IP addresses or ports, as well as with devices from the same manufacturer ("manufacturer" and "same-manufacturer" fields).

```
{
"ietf-mud:mud": {
   "mud-version":1,
   "mud-url":"https://manufacturer1.org/deviceD/modelM.json",
   "mud-signature": "https://manufacturer1.org/deviceD/modelM.p7ss",
   "last-update": "2020-07-08T12:23:23+00:00",
    ...
   "to-device-policy": {
     "access-lists": {
       "access-list": [
         {"name": "mud-92140-v6to"}
       ]}},
    ...
 }
 "ietf-access-control-list:acls": {
    "acl": [
        "name": "mud-92140-v6to",
        "type": "ipv6-acl-type",
        "aces": {
         "ace": [
            {
              "name": "c10-todev",
              "matches": {
                "ipv6": {
                  "ietf-acldns:src-dnsname": "allowedhost.org",
                  "protocol": 17
                }
              },
              "actions": {
                "forwarding": "accept"
              }
            }
          ]}]}}
```

LISTING 1. Example of MUD file allowing the communication to a given host.

Listing 1 shows an example of MUD file, which has been generated through the MUD Maker tool [26]. In addition to the fields "mud-url" and "last-update", the "mud" container includes the version of the standard specification ("mud-version") and the value of the file signature in the form of URI ("mud-signature"). It also contains the name of the ACLs to restrict the communication to/from the device. In this case, the "to-device-policy" field indicates the "mud-92140-v6to" ACL to define restrictions that must be enforced on the traffic going to the device. Thus, the "acl" container defines this ACL through a set of rules in the form of access control entries ("ace"). In particular, the "c10-todev" ace allows traffic from the "allowed.host.org" host through the UDP protocol ("17").

C. THE LIFECYLE OF MUD FILES

As previously described, the MUD specification defines a data model to define a device's intended behavioral profile, as well as an architecture for obtaining these profiles that are included in a MUD file. However, it does not define the processes and components required to manage the MUD file during the lifecycle of a certain IoT device. This aspect is key to reflect the possible behavioral changes of a device (e.g., due to software updates to mitigate new discovered vulnerabilities not identified in the initial or previous testing phase), as well as to enforce the restrictions included in a MUD file. Figure 2 provides an overview of the relationship between a MUD file and an IoT device's lifecycle according to the main phases defined by [27].

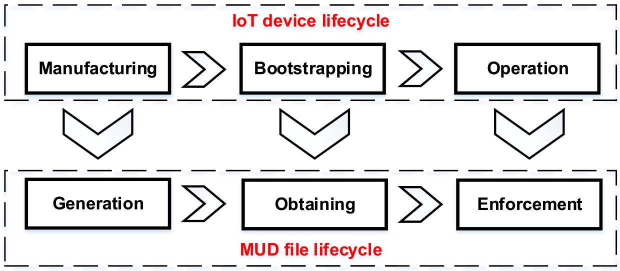

FIGURE 2. Overview of the MUD files' lifecycle.

The lifecycle of an IoT device begins when it is created during the *manufacturing* process. During this process, the manufacturer is expected to generate a MUD file with the network access control restrictions for such device. Then, the IoT device is installed and commissioned on a certain network during the *bootstrapping* process, in which the generated MUD file is obtained to adapt the network components to the restrictions included in the profile. It should be noted that the obtaining of the MUD file could include the processes defined in Figure 1 involving MUD components. Once bootstrapped, the device starts with the provision of its intended functionalities during the *operation* phase. At this point, the restrictions embedded in the MUD file are enforced through the corresponding technologies, such as SDN [16]. Furthermore, although it is not shown for simplicity, a device can be modified during its lifecycle through software updates or configuration changes. In this case, the device may need to be rebootstrapped and, depending on the changes, it could require an update of the MUD file. Additionally, the discovery of a new vulnerability or attack associated with the device could also require updating the MUD restrictions to guarantee a secure behavior of the device.

Based on the main stages of the MUD files' lifecycle, the next section provides an exhaustive analysis of existing MUD-related proposals, which are classified according to such phases.

III. ANALYSIS OF MUD-BASED LITERATURE

For the classification of the existing MUD-based research proposals, we use the taxonomy presented in Figure 3, which includes a category for research proposals defining applications based on the MUD standard.

In particular, for the *MUD profiles generation* phase, the existing literature is mainly based on the use of *manual* approaches in which MUD restrictions are defined by users for testing purposes, or based on network traffic traces, which are used as an input to create MUD profiles. In the case of the *MUD profiles obtaining*, proposals are classified according to the protocols intended to transport the MUD URL, which indicates where the MUD file is hosted. In particular, we classify existing proposals based on the three alternatives defined by the MUD standard, namely: 1) the Dynamic Host Configuration Protocol (DHCP) [28], 2) Link Layer Discovery Protocol (LLDP) [29], or 3) included in an X.509 certificate [22]. Then, for the *MUD profiles enforcement* phase, existing approaches are mostly based on SDN to satisfy MUD restrictions, in addition to more static solutions that use common network components. Furthermore, we describe MUD-based proposals that define applications derived from the MUD standard. Based on this classification, the following subsections describe the existing MUD-related literature and provide a detailed analysis.

A. MUD PROFILES GENERATION

As described in the previous sections, the *generation* process refers to the steps required to generate a MUD file and is intended to be performed by the manufacturer according to the MUD specification. Indeed, a significant number of current MUD-related proposals assume the existence of a MUD file associated to each IoT device. However, this assumption is not supported in current IoT deployments, where the MUD standard is not widely deployed yet. For this reason, several recent approaches propose additional mechanisms to generate MUD files through different tools to help in this process. Indeed, as already mentioned, the MUD Maker tool [26] allows the creation of MUD files by providing a simple interface to specify the different MUD data model's fields. Users of this tool need to define the value for each field according to the MUD data model; however, this information could be unknown for most of users. MUD Maker is used by recent works, such as [30] to create MUD files for a network access control framework, and [31], which is intended to detect flooding attacks. Furthermore, [32] uses the same tool to create a botnet detection and mitigation system. The authors of [33] also employ MUD Maker to generate MUD files associated to devices that are submitted to a vulnerability assessment process before getting network access. Furthermore, [21] uses this tool as an example to show how MUD restrictions in a network could be modified to mitigate privacy concerns.

As an alternative of manual approaches to generate MUD files, other proposals use network traffic traces from a certain device to identify the values to define its network profile. One of the most widely used approaches is represented by the *MUDgee* tool, which was described by [34]. MUDgee is an open-source tool [35] that allows the generation of MUD files using network traffic traces contained in pcap files. In particular, the authors capture traffic from 28 IoT devices for 6 months and use MUDgee to generate a MUD file for each device based on their traffic traces. Additionally, the approach is complemented by a framework called *MUDdy*, which allows a formal semantic validation of MUD profiles and the compatibility check of MUD policies with the restrictions defined by an organization where the

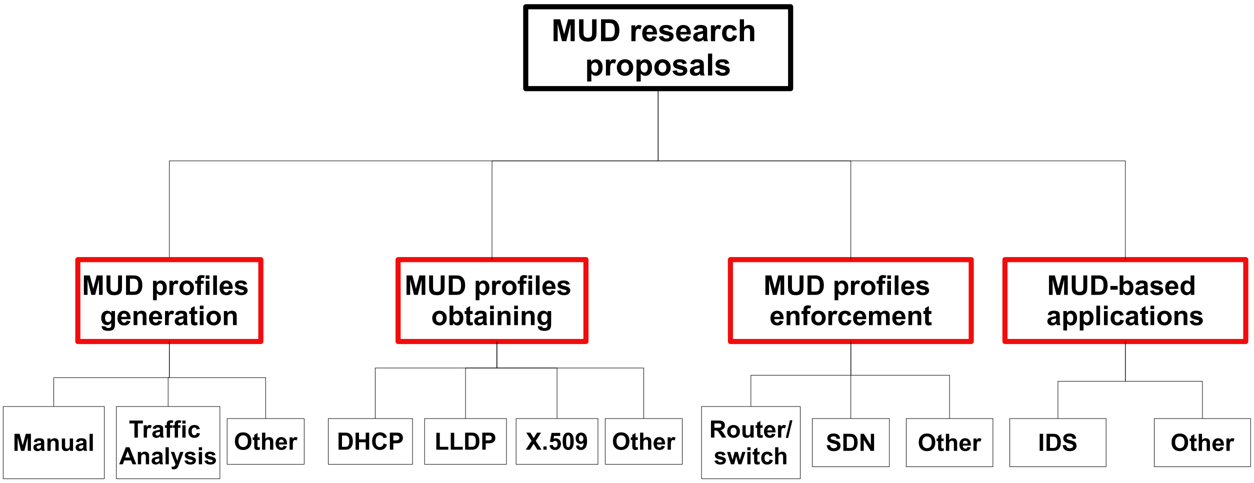

FIGURE 3. Classifying aspects for MUD-based proposals.

device is deployed. Furthermore, MUDgee is used by [36] and [37], which proposes an IoT device classification framework, and by [38], in which the generated MUD profiles are translated into flow rules. Moreover, [39] analyzes the use of MUD in general-purpose devices, (e.g., smart TVs or cameras) in the scope of smart homes. The authors use MUDgee to create MUD files associated with these devices and analyze the limitations of the standard to specify the behavior of such devices. The same tool is used to generate the network profile of a smart doorbell in [40], which also identifies similar limitations of the standard to define more fine-grained aspects of the communication of IoT devices. Another approach using MUDgee is proposed by [41] in which network flows are obtained from observing traffic on a network component (e.g., a router), which receives the network packets from connected devices. Furthermore, [42] uses MUDgee to create an anomaly detection system.

Related to the generation of MUD files, other recent works address the limitations of the data model by considering additional security aspects to generate augmented behavioral profiles. For example, [20] defines an augmented MUD profile including properties such as key sizes or cryptographic primitives, to characterize the intended behavior of a device. To obtain such information, authors use a security testing methodology by using Model-Based Testing (MBT) techniques [43]. This methodology includes tests to calculate the maximum number of simultaneous connections supported by a certain device and protocol to identify DoS attacks. The limitations of the MUD data model are also addressed by [44], which extends the model by considering dynamic security aspects in the context of smart buildings. Additionally, [19] defines an extension of the MUD model based on the Medium-level Security Policy Language (MSPL) language that has been used within the scope of the European project H2020 ANASTACIA [45]. In particular, the authors of [19] extend the model to define policies for network filtering, channel protection (e.g., based on DTLS [46]), data privacy, and application-layer authorization. Other works extend the model to consider additional information such as Quality of Service (QoS) [47] to detect attacks based on overuse of resources. Furthermore, the definition of Human Usage Descriptions (HUD) is proposed by [48] to describe users' behavior and interactions with their devices. Recent works also integrate traffic analysis with extensions to the MUD data model, such as [49], which includes physical layer parameters and flow statistics that depend on the environment where the device is deployed. To do this, authors use a learning-based system to extract the features and create an augmented model associated to a device's intended behavior. In particular, the approach is based on the use of hierarchical clustering [50], which is applied to LoRaWAN devices [51]. Furthermore, [52] proposes similar profiles to the MUD standard that are generated from direct observations of network traffic. During the packet analysis phase, device's information (e.g., model, type, or firmware) is obtained to create a feature vector, which is used to classify the device using machine learning algorithms like Decision Tree (DT) [53] and Support Vector Machine (SVM) [54].

1) ANALYSIS

Table 1 provides an overview of the different works previously analyzed. It should be noted that the definition of network behavioral profiles for IoT devices has attracted a significant interest in recent years for an effective identification of security attacks affecting such devices. Even though the MUD standard is not widely deployed today, it offers a standard representation to define such behavior. For this reason, many of the current proposals use traffic analysis techniques to obtain the parameters defined by MUD profiles. However, a simple traffic analysis is not enough to obtain additional parameters linked to network structure and characteristics that would help identify more sophisticated

TABLE 1. Research proposals for MUD profiles generation.

	Manually	Traffic Analysis	Other
[31]	✓(MUD Maker)	✗	✗
[33]	✓(MUD Maker)	✗	✗
[34]	✗	✓(MUDgee)	✗
[36]	✗	✓(MUDgee)	✗
[37]	✗	✓(MUDgee)	✗
[38]	✗	✓(MUDgee)	✗
[39]	✗	✓(MUDgee)	✗
[21]	✓(MUD Maker)	✗	✗
[40]	✗	✓(MUDgee)	✗
[41]	✗	✓(MUDgee)	✗
[20]	✗	✗	✓(security testing results [55])
[44]	✗	✗	✓(devices' changing behavior in smart buildings)
[32]	✓(MUD Maker)	✗	✗
[19]	✗	✗	✓(Medium-level Security Policy Language (MSPL) [45])
[47]	✗	✗	✓(QoS parameters)
[42]	✗	✓(MUDgee)	✗
[48]	✗	✗	✓(human usage aspects)
[49]	✗	✓	✓(physical layer parameters and flow statistics)
[52]	✗	✓	✓(devices' fingerprinting data)
[30]	✓(MUD Maker)	✗	✗
[42]	✗	✓(MUDgee)	✗

security attacks (e.g., based on the application layer [56]). In this direction, some of the previously described works have proposed extensions to the MUD data model to represent additional communication aspects such as QoS parameters, or cryptographic algorithms being used. Indeed, such extensions to the MUD model will be likely proposed in the coming years to represent specific aspects associated with certain use cases (e.g., e-health devices), as well as to address emerging 5G-based scenarios. These extensions should be considered in the scope of SDOs' working groups to foster interoperability and a large-scale deployment of behavioral profiles for connected devices. These aspects are further discussed in Section V.

B. MUD PROFILES OBTAINING

For obtaining a MUD file, the standard specification assumes the use of the MUD URL parameter, which indicates where the MUD file is hosted. This parameter can be sent by the end device to the MUD Manager through three alternative approaches: Dynamic Host Configuration Protocol (DHCP) [28], Link Layer Discovery Protocol (LLDP) [29] or included in an X.509 certificate by using the 802.1AR standard [22]. In the case of DHCP, the MUD URL is contained in the DHCP option 161. This approach is used by [57], in which MUD URLs are associated to devices' MAC address. This association is learned by the SDN controller through the DHCP message exchange. DHCP is also considered by [31], which extends the functionality of current DHCP implementations to transport the MUD URL. Furthermore, some MUD-related reports from NIST propose high-level architectures based on DHCP for obtaining the MUD URL where a certain MUD file is hosted [12]. The second alternative is based on a ''vendor-specific'' extension of the LLDP type-length-value design in which the MUD URL is carried. This approach is only considered by a recent NIST report [13], which defines several architectures for securing small-business and home IoT networks by using the MUD standard. However, it should be noted that both DHCP and LLDP alternatives could represent a security issue, as the end device could spoof its identity to obtain additional network resources [10].

The security issues derived from the use of DHCP and LLDP are addressed by the third alternative based on X.509 certificates. In this case, the MUD URL is included in the certificate, so that this information is linked with the device's identity. For example, the already mentioned NIST report [13] considers a scenario where devices are provisioned with certificates to associate device authentication with MUD files. Furthermore, the proposed architecture uses an Authentication, Authorization and Accounting (AAA) infrastructure [58] based on the Remote Authentication Dial-In User Service (RADIUS) protocol [59], so that the router/switch can communicate the MUD URL to the MUD Manager. A similar approach is also proposed by [60], where MUD Manager's functionality is integrated into fog nodes [61]. In the case of [33], the device's certificate is generated before the MUD profile is created for that device. Furthermore, [62] proposes a certificate-based approach by using a Bootstrapping Remote Secure Key Infrastructure (BRSKI) [63] and AAA for obtaining MUD files through the manufacturer. In particular, BRSKI messages are transported through the Tunnel Extensible Authentication Protocol (TEAP) [64]. BRSKI is also considered by [65], which uses a certificate-based approach for device's authentication.

As an alternative to the use of certificates, recent proposals use Pre-Shared Key (PSK) authentication to associate the device's identity with its MUD profile. The main purpose is to provide a lightweight authentication mechanism that can be used by resource-constrained devices and Low-power and Lossy Networks (LLN). Indeed, although recent works have proposed the deployment of a Public Key Infrastructure (PKI) based on lightweight X.509 certificates [66], the use of PKI might have a significant impact in certain low-power wide-area network (LPWAN) technologies (e.g., LoRAWAN [51]). Thus, [18] proposes the use of the Protocol for Carrying Authentication for Network Access (PANA) standard [67] for the transport of Extensible Authentication Protocol (EAP) messages [68] using EAP-PSK [69] as authentication method. A similar approach is proposed by [19], in which the PANA protocol is replaced by the Constrained Application Protocol (CoAP) [70] to come up with a more efficient and lightweight approach to be used in constrained environments using the RADIUS protocol to transport the MUD URL. In both proposals, the process of obtaining the MUD file is associated with the initial authentication of the device joining the network (i.e., bootstrapping [27]), so that MUD restrictions can be enforced before granting network access to the device. Like in the previous proposal, IoT devices do not have network access until they are authenticated and an associated MUD file is obtained.

1) ANALYSIS

Table 2 provide an overview of the different proposals related to the MUD profiles obtaining phase. Based on our analysis, most of the approaches are based on the alternatives proposed by the standard MUD specification, namely: DHCP, LLDP and certificates. However, DHCP and LLD based approaches pose security concerns, as the process for obtaining the MUD file is not linked to the device's authentication, so that a malicious device could spoof its MUD URL. Although the use of certificates avoids this problem, the deployment of PKI may not be an efficient solution for devices and networks with tight resource constraints. Furthermore, according to the MUD architecture, the MUD File Server can become a bottleneck for obtaining MUD files, and it could generate serious security problems in case it is compromised. As will be described in Section V, the standard specification also does not consider updating MUD files to reflect the devices' changing behavior (e.g., due to a software update). This process will require scalable solutions that allow manufacturers to update the intended network behavior of their end devices where they are deployed.

TABLE 2. Research proposals for MUD profiles obtaining.

	DHCP	LLDP	X.509	Other
[57]	✓	✗	✗	✗
[31]	✓	✗	✗	✗
[12]	✓	✗	✗	✗
[13]	✓	✓	✓	✓(AAA)
[60]	✗	✗	✓	✓(AAA)
[62]	✗	✗	✓	✓(BRSKI) [63]
[33]	✗	✗	✓	✗
[65]	✗	✗	✓	✓(BRSKI)
[18]	✗	✗	✗	✓(PANA-EAP (PSK), AAA)
[19]	✗	✗	✗	✓(CoAP-EAP (PSK), AAA)

C. MUD PROFILES ENFORCEMENT

As already described in Section II, the MUD standard does not define specific mechanisms to enforce the restrictions defined within a MUD file. Furthermore, these restrictions need to be translated into network rules to be deployed in the corresponding network components, such as routers and switches. In this direction, [49] generates extended MUD profiles, which are directly enforced in a network component called *gateway* to filter the communication with certain external services. Similarly, [71] presents a deployment scenario for MUD files that are directly deployed on a local router as part of a home network scenario. Also in a similar home setting, [33] proposes an extended MUD architecture based on blockchain technology to obtain vulnerabilities associated with devices on a certain network. In the case of devices getting over a vulnerability assessment process, MUD restrictions are obtained and enforced in routers deployed throughout the network. Furthermore, [41] proposes an architecture to enforce MUD constraints by using common routers and switches. Similarly, MUD restrictions are also enforced in typical network components in the work proposed by [21] for the detection of privacy threats.

Beyond the use of typical network components, in recent years the use of SDN techniques has been strongly considered for the protection of IoT systems [72], [73]. Indeed, as described by [74], the use of SDN represents an effective tool for the dynamic protection against certain types of attacks in IoT networks, such as DDoS attacks. When SDN techniques are used in conjunction with the MUD standard, network restrictions in the MUD files are translated by an SDN controller into flow rules that are deployed on several SDN switches. Indeed, recent efforts by NIST [13] are intended to use SDN and Virtual LAN (VLAN) techniques to secure IoT devices in home and small-business networks. Furthermore, [31] uses an SDN approach to translate and enforce MUD rules using the OpenFlow protocol [75]. In particular, authors implement an SDN component that is responsible for creating OpenFlow rules using the IoT devices' MAC address. Moreover, [57] proposes an enforcement scheme based on SDN by using three flow tables and the OpenFlow implementation. Specifically, flow tables are used to map the source and destination MAC addresses, as well as MUD access control entries (see Section II-B). The main purpose of this implementation is to address the problem associated with access control entries that are defined for device classes (e.g., ''same-manufacturer'') that can lead to an explosion of network rules and, consequently, scalability issues. The use of OpenFlow is also proposed by [30], which integrates MUD restrictions with user policies to restrict local communications that cannot be defined by the manufacturer. Furthermore, [76] builds a system to detect anomalous patterns based on OpenFlow and the Faucet SDN controller [77]. Also based on OpenFlow, a similar approach is followed by [52] that enforces MUD restrictions on an SDN controller called Home Area Network Zero Operations (HANZO) by using the Open vSwitch implementation [78]. Additionally, [38] and [36] also consider SDN to enforce MUD restrictions translating the MUD policies into flow rules. In particular, [36] considers flow rules to be enforced on network switches using SDN. Toward this end, authors create an SDN simulator that uses PCAP traces to inspect device behavior so that only traces corresponding to suspicious behavior are sent for further inspection in an Intrusion Detection Systems (IDS) based on Snort [79].

Additional approaches for the enforcement of MUD profiles are based on the integration of several technologies, including SDN. In this direction, [18] proposes an SDN-based system to enforce MUD policies as part of the security framework developed in the context of the EU H2020 ANASTACIA project [80]. The approach is based on the translation of MUD restrictions into security policies represented in the MSPL language, which was developed during the project. Then, these policies are translated into specific rules to be deployed in network components using OpenFlow. It should be noted that network restrictions are deployed before the device obtains network access. Furthermore, the same authors propose an extension [19] of this architecture by using extended MUD profiles that are enforced by using an authorization approach based on the eXtensible Access Control Marlup Language (XACML) and CBOR Web Tokens (CWT) [81] that are extended with the Authorization Information Format (AIF) format [82], which is an IETF standardization effort. These technologies are also used by [20] for the enforcement of extended MUD profiles with application-layer authorization. In addition, [83] extends the MUD architecture considering a Local MUD Manager (LMM) that is deployed on an SDN controller, as well as several Mobile MUD Enforcement Engines (MMEEs) running on smartphones to enforce MUD restrictions.

TABLE 3. Research proposals for MUD profiles enforcement.

	Router/ switch	SDN	Other
[49]	✓	✗	✗
[71]	✓	✗	✗
[33]	✓	✗	✗
[41]	✓	✗	✗
[21]	✓	✗	✗
[13]	✓	✓(OpenFlow)	✗
[31]	✗	✓(OpenFlow)	✗
[57]	✗	✓(OpenFlow)	✗
[30]	✗	✓(OpenFlow)	✗
[76]	✗	✓(OpenFlow)	✗
[52]	✗	✓(OpenFlow)	✗
[38]	✗	✓	✗
[36]	✗	✓(SDN simulator)	✗
[18]	✗	✓(OpenFlow)	✗
[20]	✗	✗	✓(CWT [81]-AIF [82] tokens and XACML [84])
[19]	✗	✓(OpenFlow)	✓XACML, CWT-AIF credentials
[83]	✗	✓	✓(smartphones)

1) ANALYSIS

Table 3 summarizes the approaches for the enforcement of MUD profiles that were previously analyzed. Based on our analysis, the use of SDN is widely considered for the enforcement of network restrictions that are defined in MUD profiles. Indeed, the deployment of SDN techniques represents a key trend in IoT scenarios to offer a scalable approach for an effective and dynamic management approach for IoT networks and devices [72]. However, it should be noted that the enforcement of MUD profiles requires an intermediate process to translate the MUD restrictions into flow rules to be deployed in network components. Although most of MUD rules can be easily translated into flow rules, the use of high-level terms (e.g., "same-manufacturer") can lead to an explosion of rules and therefore scalability issues, as mentioned by [57]. Furthermore, as demonstrated by [18], [19], the definition of extended behavioral profiles (i.e., beyond the network level) requires the use of additional mechanisms (e.g., XACML policies) to satisfy devices' restrictions at the application layer. The definition of these extended profiles should take into account user-defined usage restrictions in addition to the constraints defined through MUD-based approaches. This aspect is only considered by a recent work [85], which defines the User Policy Server (UPS) component to provide network administrators and end-users the ability to interact with MUD components through a user-friendly interface.

D. PROPOSED APPLICATIONS OF MUD IN LITERATURE

In addition to the previous works focusing on the main phases of the MUD profile's lifecycle, there are recent efforts using the MUD standard to propose different related applications. Most of these works are focused on the development of IDS based on the monitoring of the rules defined in MUD profiles. In this direction, [31] proposes an SDN architecture to detect flooding attacks. Specifically, an SDN component is implemented to periodically monitor network flows considering certain features (e.g., bytes per second) by using the Exponentially Weighted Moving Average (EWMA) technique [86]. Authors use MUD rules as whitelists, so the traffic from/to a device must match with such restrictions. A similar approach is proposed by [32], which developed an IDS based on the Snort system [79]. The proposed approach is intended to monitor network traffic and compare with MUD rules to find suspicious traces that are sent to the IDS. Additionally, the system is enabled with an alarm mechanism for end users, and packet filtering as a mitigation mechanism. Also based on Snort, [36] uses MUD rules to check the compliance of actual traffic with such restrictions to detect different types of attacks, such as reflection/amplification, flooding, Address Resolution Protocol (ARP) spoofing and port scanning. The same authors extend this approach for the detection of volumetric attacks based on machine learning techniques and a feature analysis to evaluate its impact on the detection of different attacks [76]. Furthermore, [49] also proposes an IDS based on clustering techniques using extended behavioral profiles. Specifically, authors use hierarchical clustering [87] to create *normal* behavioral profiles, which are defined as MUD profiles. Based on this, if an anomalous behavior is detected, the device is isolated and network restrictions are applied as a mitigation mechanism. A similar approach is also proposed by [42] in which MUD profiles are generated as *normal* traffic to detection potential anomalies. The detection of DDoS attacks is considered by [60], which defines an architecture for integrating MUD managers in fog nodes, although implementation details are not provided. In this case, MUD restrictions are used as mitigation mechanism in case an attack is detected. Moreover, [88], [89] proposes the use of MUD and Network Function Virtualization (NFV) [90] to detect suspicious behavior in an Internet Service Provide (ISP) domain. The system is used to monitor traffic so that, if a deviation is detected by comparing with MUD rules, the communication is blocked. The authors also propose the use of *packet marking rules* to avoid the problems

TABLE 4. MUD-based applications.

	Intrusion Detection System (IDS)	Other	Description
[31]	✓	✗	Monitoring of MUD rules to identify traffic deviations that could represent a DDoS attack
[32]	✓	✗	Snort-based IDS in which MUD rules are monitored to identify Mirai botnets
[36]	✓	✗	Snort-based IDS in which MUD rules are monitored to identify different attacks, including reflection/amplification, flooding and port scanning
[76]	✓	✗	Monitoring of MUD rules to identify volumetric attacks by using machine learning techniques
[49]	✓	✗	Generation of MUD profiles based on hierarchical clustering and monitoring for attack detection in LoRaWAN devices
[60]	✓	✗	Deployment of MUD profiles in fog nodes when detecting a DDoS attack
[89]	✓	✗	Monitoring of MUD rules and use of NFV to detect potential attacks in an ISP architecture
[12]	✓	✗	Definition of a high-level architecture integrating MUD with threat signaling and software updates
[13]	✓	✗	Definition of MUD-based network architectures to protect IoT devices in home and small-business networks
[91]	✗	✓ (identification of IoT devices)	Application of machine learning techniques to classify devices into categories to which MUD restrictions are mapped
[37]	✓	✓ (identification of IoT devices)	Framework to generate, verify and monitor MUD profiles
[93]	✗	✓ (secure federated learning)	Architecture and mechanism to ensure only MUD-compliant devices participate in the training process of federated learning
[21]	✓	✗	Integration of MUD restrictions and a machine learning approach for the identification and mitigation of privacy threats
[96]	✗	✓ (energy optimization)	Use of MUD files as an input to optimize the target wake time of battery-powered 802.11ax devices
[98]	✗	✓ (security certification)	Testing framework to pre-certify an IoT device's security level based on MUD profiles
[48]	✗	✓ (dynamic resource management)	Integration of human behavioral aspects into MUD profiles to adapt dynamically network resources
[42]	✓	✗	Checking compliance of MUD rules to detect malware based on machine learning techniques

derived from the use of Network Address Translation (NAT) in home networks. Additionally, the use of MUD profiles to protect IoT devices against different attacks is also considered by recent NIST works, such as [12], which proposes an architecture integrating threat signaling and update servers together with MUD profiles to increase the protection of IoT devices in the context of home and small-business networks. This document is extended by [13] that defines four different implementations based on the standard to protect IoT devices in such networks.

Besides the use of MUD profiles to detect different types of attacks, other works describe additional applications. For example, [91] proposes the use of MUD files to identify categories of IoT devices. In this way, MUD restrictions of different devices could be grouped in a single policy for that category of devices. For this purpose, authors make use of different machine learning techniques, such as Support Vector Machine (SVM) [92]. A similar application is proposed by [37], which sets out the problem of associating existing MUD files to devices that do not use these profiles and monitoring their behavior to detect potential changes. In particular, authors generate and update the behavior profiles in real time to verify their similarity to previously created MUD profiles. Moreover, [93] uses MUD files to limit the attack surface in federated learning scenarios [94]. In particular, only devices satisfying the restrictions of their MUD files are allowed to participate during the training process. Also, [21] proposes a machine learning approach and policy-based security framework by using Seckit [95] for the analysis of encrypted traffic to detect potential privacy threats. In this case, authors use MUD profiles for privacy threats that cannot be mitigated by other approaches, such as obfuscation of certain features, which are used by machine learning techniques. Other applications derived from the use of the MUD standard are proposed by recent approaches. For example, [96] develops a system based on machine learning that uses MUD files as input to optimize the target wake time in 802.11ax devices [97], that is, the time when they *wake up* to send or receive data. Furthermore, [98] uses MUD files as a basis to certify the level of cybersecurity provided by a certain device. In particular, authors define an architecture for automated testing to check if the actual behavior of the device complies with the restrictions specified in the MUD file. In addition, an extension of MUD profiles is proposed by [48] to describe users' behavior and interactions in order to dynamically adapt the Quality of Experience (QoE) and other network resources.

1) ANALYSIS

Table 4 provides an overview of the previous works. As described, most of the applications derived from the MUD standard are intended to monitor the compliance of the actual traffic of IoT devices in relation to their MUD profiles to come up with IDS for IoT. However, such proposals do not consider potential behavioral changes that could affect to the *normal* operation of such devices. Even if this aspect was highlighted by [12], new approaches need to be proposed in this direction considering the lifecycle of IoT devices. Moreover, the different approaches being analyzed only consider certain network attacks, such as DDoS. This is mainly due to the limited expressiveness of MUD profiles, as was described in Section III-A. Therefore, potential extensions to the MUD data model could foster the development of new applications to improve IoT security. In turn, these extensions could be also based on the specific requirements of certain IoT-enabled scenarios, such as autonomous driving or smart cities, which could trigger new extensions of the MUD standard.

Our previous analysis provides several insights about the challenges associated to the MUD standard, its implementation and deployment. For example, in the case of the *generation* of MUD files, several proposals point out the lack of expressiveness as one of the main aspects to be considered. Furthermore, the obtaining phase should be designed taking into account the need for secure protocols to link the device's identity with its associated MUD profile. While some of these challenges were already mentioned in our analysis, Section V provides a more detailed overview of the current challenges that need to be considered to foster the adoption of the MUD standard in the market and the research community.

IV. MUD IMPLEMENTATIONS

Based on the increasing interest in the MUD standard, in recent years, several implementations have been proposed to foster the deployment of its main architecture components. In this section, we extend our previous analysis of existing implementations [85] and we describe recent tools that are currently considered by researchers and industry designing MUD-related solutions to protect IoT devices [13].

A. OPEN SOURCE MUD MANAGER

Open Source MUD manager [99] (osMUD) is a C based open-source solution[1] that aims to make home/small-business networks compliant with the MUD standard. It is designed to be easily built on OpenWRT routers [100], and since it is based on C language, it can be compiled to run in any firewall machine having a C compiler. Nevertheless, the implementation provided by osMUD designers is strictly tied to a customised version of dnsmasq [101] and to an OpenWRT firewall service (Figure 4). The former runs a DHCP server able to extract MUD-URL from DHCP requests, thus no generic DHCP servers are admitted, while the latter limits deployment choices to only OpenWRT-compliant devices ([102]). It is worth mentioning, however, that osMUD designers provided all necessary tools to build osMUD outside OpenWRT environments, by leaving the task of the firewall implementation and configuration to developers.

Finally, the current osMUD release ignores MUD rules for lateral movement (e.g. same-manufacturer, controller, my-controller etc.). As a consequence, malicious actors can progressively move inside the network looking for valuable key data and assets.

B. MICRONETS MUD

The Micronets MUD implementation [103] is composed by a Micronets manager,[2] which represents an application that enables to run the MUD manager as utility server by using the *systemd* service control. The Micronets MUD manager communicates with the Micronets Gateway service[3] for enforcing MUD rules. The Micronets Gateway service provides REST endpoints for direct or websocket-based invocation, configuring the DHCP server (dnsmasq or ISC DHCP software), configuring network resources for the hostapd service, and issuing openVSwitch/OpenFlow commands to enforce Micronet- and device-level policy. The main flow is shown in Figure 5. The Micronets MUD manager is easy to deploy as individual services are provided as a docker container.

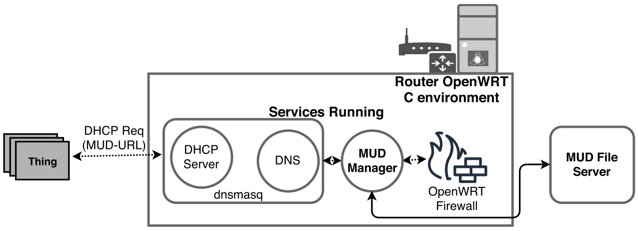

FIGURE 4. osMUD architecture.

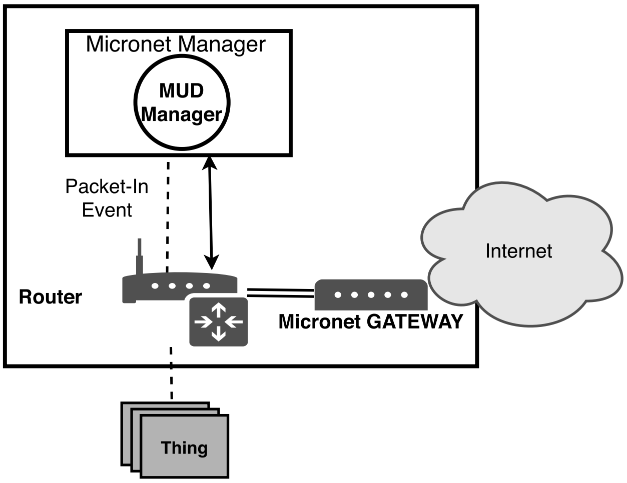

FIGURE 5. Logical architecture of micronets MUD implementation.

C. CISCO MUD MANAGER

Cisco MUD Manager is an open-source software [104] resulting from a collaboration between different service providers, which lays the foundation for a MUD manager implementation in real case scenarios. The logical architecture, illustrated in Figure 6, demonstrates that its environment includes a single device serving as MUD manager. Moreover, this runs an open-source implementation of an Authentication, Authorization, and Accounting (AAA) server based on FreeRADIUS [105], which interacts with the MUD manager to authenticate MUD-URLs received from the Cisco Catalyst 3850-S switch. The switch was customised to support MUD-URLs extraction from DHCP and LLDP messages and to interpret MUD file constructs. In particular, it

[1] https://github.com/osmud/osmud
[2] https://github.com/cablelabs/micronets-manager
[3] https://github.com/cablelabs/micronets-gw/tree/master/micronets-gw-service

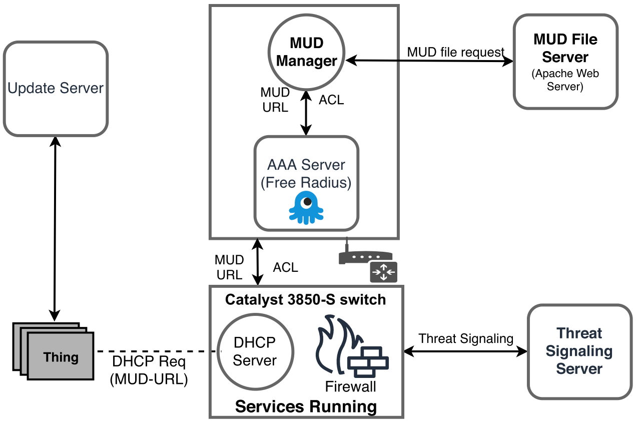

FIGURE 6. Logical architecture of Cisco MUD implementation.

implements IP access list policy and uses Virtual Local Area Networks (VLANs) to support MUD-oriented policies, such as my-controller and my-manufacturer.

Although this solution provides all the necessary tools to make a network MUD compliant, it is designed as a way for researchers and network engineers to familiarise themselves with the MUD concept, thus presenting some limits in real case scenarios. For example, the Cisco Catalyst 3850-S switch does not support ingress dynamic ACL [106], consequently it may occur that a MUD-capable device can receive traffic from a not authorised external domain.

D. NIST MUD

The National Institute of Standards and Technology (NIST) MUD manager[4] (Fig.7) relies on Software Defined Networking (SDN) concept, thus involving switches and controllers that observe the OpenFlow protocol. The controller uses OpenDaylight [107] software to manage and monitor the wireless SDN switch, which forms and manages the network hosting and connecting to IoT devices. Furthermore, the SDN controller hosts a MUD manager written in Java and implemented as an OpenDaylight application. In addition to parsing, verifying and injecting MUD rules, the MUD manager extracts the MUD-URL from MUD compliant DHCP requests. Hence, the SDN switch forwards each DHCP request to the controller, thus allowing the usage of a generic DHCP server.

To achieve rules scalability, the SDN switch adopts six flow tables: the first two tables classify source and destination MAC address; the next two flow tables manage from-device and to-device policies; finally, the pipeline table provides two flow tables, Pass-Through and Drop table, to apply SDN switch decisions. Thus, each of them adds metadata information to packets, which is used to support Switch decision making.

Nevertheless, the table pipeline may block a packet while its flow rules are being created and injected into the switch,

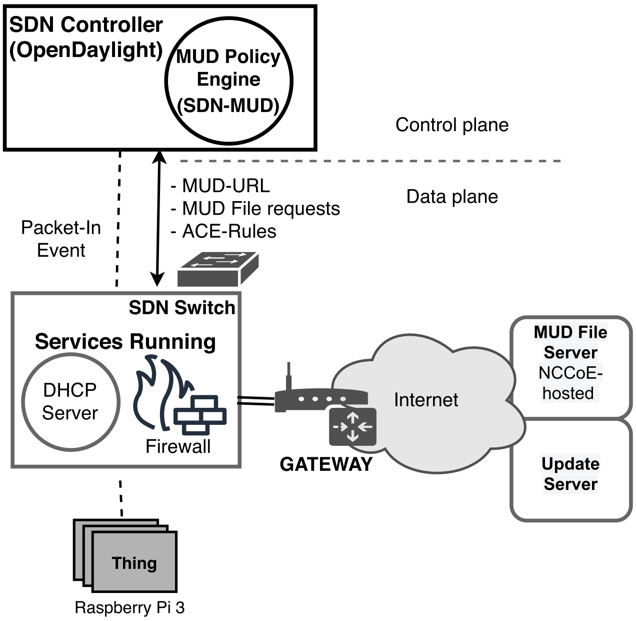

FIGURE 7. NIST MUD architecture.

which may cause a switch failure. For instance, no packets from a newly connected device can go through the first table until rules have been installed. Hence, authors [57] designed a *relaxed mode* to cope with these situations, so that packets can proceed in the pipeline while classification flow rules are being installed. These packets could get through before the classification rule is installed at the switch, which can result in the temporary violation of MUD ACEs.

E. UMU MUD IMPLEMENTATION

The University of Murcia (UMU)-MUD proof-of-concept (PoC) implementation [18], developed in the scope of the ANASTACIA EU project [45], follows an SDN approach to dynamically enforce MUD behavioural profiles in the deployed and managed IoT systems, thereby (re-)configuring the network flow-rules in the IoT SDN switches through OpenFlow. This approach allows to adapt the IoT network traffic in real time according to the context and behavioural profiles hosted by the MUD Server, thus, dropping, forwarding or redirecting IoT traffic. In the PoC, the SDN controller uses the Open Network Operating System (ONOS) [108], which manages either physical IoT devices, or virtual devices in a Cooja simulator [109]. The implementation has been tested for IoT devices with Contiki operating system [110].

Regarding authentication aspects, the UMU-MUD PoC is empowered by an AAA server (FreeRadius 2.0.2) and EAP server [68] implemented in C aimed to manage the authentication in the IoT network using the PANA protocol [67]. Meanwhile, the PANA authenticator (PAA) component, deployed in the IoT local network, uses a OpenPANA implementation [111] to authenticate the IoT devices during bootstrapping and interacts with the remote EAP server for remote authentication. Unlike traditional systems based on

[4] https://github.com/usnistgov/nist-mud

the usage of certificates, the UMU-MUD implementation also supports the usage of Pre-Shared Keys (PSKs) [69] to be considered in constrained IoT environments where the deployment of PKI might not be appropriate.

The MUD Manager is responsible for contacting the remote MUD file server to obtain the MUD files. In addition, the MUD Manager implementation is endowed with a policy-based management system in charge of storing and handling the network restrictions. Thus, it features a local security repository to store both home-domain security policies and remote MUD policies originating from the MUD server. Besides, it is endowed with a Policy Interpreter component for translating the MUD profiles to specific configuration enablers based on an ONOS controller [108], which enforces the filtering policies in the IoT system being managed.

F. MUD TOOLS

Besides the implementations previously described, several tools have been recently developed addressing specific aspects of the lifecycle of MUD files that are described below.

1) MUD MAKER

The MUD specification's authors developed the web application MUD Maker [26] for the creation of MUD files. Specifically, it provides manufacturers with guidelines to build their own MUD-enabled infrastructure. The tool provides a GUI to generate MUD files by requesting some basic information such as the MUD URL, manufacturer, the URL for the documentation and a short description of the device. The MUD restrictions are built through a simple questionnaire related to the device's intended communication, including data on Internet communication, controller data, or devices from certain manufacturers. Depending on the type of communication, the tools ask for more specific data, such as ports, protocols (UDP, TCP) or information related with the controller and local hosts. This information is used to generate a MUD file, which can be visualized either in JSON or in the form of ACL, and it can be downloaded.

2) MUDdy

MUDdy [112] is an open source tool based on Python to ease the creation of MUD files. While MUDdy's functionality is similar to MUD Maker, it only provides a command-line interface. In particular, the tool gives users with the possibility to provide information associated to a device's intended communication, including data about the manufacturer, associated URL and ACLs. As a result, the tool generates a MUD file according to the standard specification.

3) MUD VISUALIZER

MUD Visualizer [113], [114] is an open source tool also developed by MUD specification's authors to visually represent the content of a MUD file by showing the communications to/from a certain device. The tool is integrated with MUD Maker, so a user can visualize the MUD file previously created through such tool. In particular, MUD Visualizer supports six MUD abstractions including the terms *domain-names*, *manufacturer*, *same-manufacturer*, *local-networks*, *controller* and *my-controller*. The tool can be used to represent both the incoming and outgoing traffic of a device. The visualization is not restricted to a single MUD file; indeed, it can be used to visualize multiple files to provide a complete overview of the communications in a certain network.

4) MUD-URL-VALIDATOR

This open source tool [115] was developed by Cisco, and it consists of a Python script to obtain the Ethernet frame from a pcap file, in order to get the MUD URL. In particular, MUD-URL-Validator is intended to get the MUD URL in an LLDP message, a DHCP Discover message, and a DHCP Request message. Then, the tool checks if the MUD URL is properly formed according to the MUD specification. As the tool is only intended to get MUD URLs and check their validity, users need to provide MUD-URL-Validator with pcap files by using additional tools such as Wireshark [116]. Then, this pcap file can be used as the input for the MUD-URL-Validator tool.

5) MUDgee

MUDgee [35] is a tool to generate MUD files from network traces. Indeed, it uses pcap files as the input that must be obtained previously by the user. For the generation of the MUD file, the tool must be properly configured by specifying the location of the corresponding pcap file, information related to the default gateway (e.g., IP or MAC address) for the device, and data about the device itself. As already described in the previous sections, this tool has been widely used in recent MUD-related research proposals.

6) MUD-PD

MUD-PD [117], [118] is an open source tool based on Python developed by the NIST's National Cybersecurity Center of Excellence (NCCoE). Its main objective is to help manufacturers, researchers and developers to implement MUD-enabled scenarios, assisting in the generation of a MUD file according to the standard specification. The tool requires Python, MySQL and improves MUDgee tool by providing an intuitive GUI and supporting additional features related to the MUD specification, such as the automatic generation of ‘‘same-manufacturer’’ and ‘‘controller’’ classes. Furthermore, it supports merging network traffic captures to create an overview of a specific network infrastructure. MUD-PD also provides the functionality to connect to databases containing network information, as well as to generate MUD files and reports for a particular device.

Based on the description of the different MUD-related implementations and tools, Table 5 provides an overview of the classification of such solutions according to the phases of the MUD files' lifecycle, which was defined in Section II-C.

TABLE 5. MUD implementations and tools in relation to MUD profiles' lifecycle.

	MUD Generation	MUD Obtaining	MUD Enforcement
osMUD [99]	✗	✓	✓
Micronets MUD [103]	✗	✓	✗
Cisco MUD Manager [104]	✗	✓	✓
NIST MUD [119]	✗	✓	✓
UMU MUD [18]	✗	✓	✓
MUD Maker [26]	✓	✗	✗
MUDdy [112]	✓	✗	✗
MUD Visualizer [113]	✓	✗	✗
MUD-URL Validator [115]	✗	✓	✗
MUDgee [35]	✓	✗	✗
MUD-PD [117]	✓	✗	✗

V. CHALLENGES AND RESEARCH DIRECTIONS

Based on the previous analysis of MUD-related research proposals and implementations, in this section we describe the main challenges and potential research directions to be considered in the coming years to foster the adoption of the MUD standard. Furthermore, we provide an analysis of recent proposals addressing such challenges that is summarized in Table 6.

A. MUD DATA MODEL EXPRESSIVENESS

As described in Section III-A, one of the main limitations associated with the MUD standard is the lack of expressiveness for the definition of access restrictions beyond the network layer. This aspect has led to the development of research proposals to extend the MUD data model considering additional security aspects, such as channel protection or application-layer authorization [19], [49]. Furthermore, the identification of authorized endpoints to communicate with a certain device is not enough to protect against certain attacks. As mentioned by [49], a compromised service may hijack devices by increasing the data rate, as this aspect is not considered in the definition of MUD profiles. Indeed, the definition of enriched behavioral profiles could be used to detect a broader range of potential security attacks, including application layer threats such as slow DDoS attacks [120]. However, possible extensions to the MUD data model could require the extension of network deployments to detect and mitigate other types of attack. Furthermore, such extensions must be addressed through joint efforts in the scope of Standards Developing Organizations (SDOs) to foster the adoption of the MUD approach. In this direction, [121] represents a standardization initiative to extend the MUD model using (D) TLS parameters that represents an excellent starting point for the development of standardized efforts to extend MUD in the coming years.

B. SCALABILITY

Another main challenge associated with the deployment of the MUD standard in large-scale IoT infrastructures is the potential explosion of network rules, which could be derived from the restrictions defined in a MUD file. This is mainly due to the use of terms such as *manufacturer* and *same-manufacturer*, which need to be translated into the corresponding flow rules to be deployed in switches and routers. Indeed, as described by [57], if the restrictions associated with the same manufacturer are implemented as MAC flow rules, this could require N^2 rules, where N is the number of devices associated with a certain switch or router. Although the same authors of [57] propose an approach to mitigate this problem based on SDN and several flow tables, it is not yet clear whether the use of these terms can represent an obstacle in deployments with a potentially large number of devices. Another approach to mitigate scalability issues is the use of classification frameworks, so that MUD files are associated to class of devices instead a single component [37], [52]. While these solutions could reduce the number of flow rules, still there is a need to evaluate their behavior in large-scale deployments. Indeed, based on our analysis of the literature, we note that most of the MUD-related proposals consider home and small-business networks [13]. Therefore, the deployment of MUD in large-scale environments needs to be evaluated considering real-world scenarios, including Industry 4.0 and smart cities use cases.

C. CONFLICT DETECTION

In most of the current real-world deployments, organizations use security policies that are defined by an administrator to control the access from/to certain devices and services. These policies may be in conflict with the MUD restrictions, which are defined by manufacturer for their devices. For example, while a MUD file may allow a device to communicate with a certain service, the access to this service may be banned by the access policies on the domain where the device is deployed. Indeed, as described by [49], network restrictions for certain devices could depend on the operational environment where they will be deployed, so that they cannot be specified in the manufacturing phase. In these cases, end users or network administrators could be required to configure the device; therefore, user-friendly interfaces should be considered to enable non-expert users to modify their devices' behavior [13]. Furthermore, while conflict detection has been widely analyzed in access control policies [122], [123], in the case of the MUD standard, only a few papers have addressed this issue. In particular, [34] proposes a framework for the syntactic and semantic validation of MUD profiles using the MUDdy tool to check its compatibility with an organization's security policies. However, taking into account the scale and heterogeneity of IoT devices, conflict detection approaches still need to be complemented with automated solutions that allow the dynamic re-configuration of security policies in a given deployment.

D. MUD PROFILES UPDATING

The intended behavior of an IoT device can evolve during its lifecycle due to software updates and security patches (e.g., to mitigate a new security threat), as well as potential

TABLE 6. Analysis of MUD challenges and existing proposals Legend: Exp = Expressiveness, Sca = Scalability, Conf = Conflict Detection, Upd = MUD profiles updating, Light-Auth = Lightweight Authentication, Traf = Traffic analysis, Sha = MUD profiles sharing, Apps = Potential applications, 5G = MUD profiles for 5G systems, Strd = Standardization.

	Exp	Sca	Conf	Upd	Light	Traf	Sha	Apps	5G	Strd
[49]	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗
[19]	✓	✗	✗	✗	✓	✗	✗	✗	✗	✓
[121]	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
[57]	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
[52]	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
[37]	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
[13]	✗	✗	✗	✓	✗	✓	✗	✗	✗	✓
[34]	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
[18]	✗	✓	✗	✗	✓	✗	✗	✗	✗	✓
[129]	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
[93]	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
[136]	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗
[137]	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗
[20]	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗
[98]	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
[48]	✓	✗	✗	✗	✗	✗	✗	✓	✓	✗
[151]	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓
[152]	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
[153]	✓	✗	✗	✓	✗	✗	✗	✗	✗	✓
[154]	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓
[154]	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

changes in the network where it is deployed [27]. These behavioral changes may require updating the MUD file associated with such device to reflect the new intended behavior. However, the standard specification [124] only considers the information returned by the MUD file server to be valid for as long as the device is connected. In particular, as described by [13], the MUD specification defines the *cache-validity* timer that indicates when the MUD manager should check new MUD files, so that the current profile is used while such timer has not expired. Therefore, the network restrictions contained in such profile can become obsolete, so that security breaches could arise. In this context, the extension of the standard MUD architecture may be required to perform periodic checks on possible potential changes in the behavior of a device. Also, the use of publish/subscribe communication patterns between the MUD Manager and the MUD file server could be considered to receive potential updates of devices' behavioral changes. This process might additionally require a re-authentication of the device to bind the updated network restrictions with the device's identity. However, connectivity issues of such devices could hinder the realization of this process.

E. LIGHTWEIGHT AUTHENTICATION FOR MUD FILES OBTAINING

As described in Section III-B, the MUD standard describes three alternatives for obtaining the MUD URL parameter based on DHCP, LLDP, and X.509 certificates. However, as described in the standard specification [124], the use of DHCP and LLDP could represent a security issue, since compromised devices may send fake MUD URLs. Indeed, as described by [13], the process to send the MUD URL parameter should be linked with the authentication of the device. Furthermore, while X.509 certificates address the issues associated with DHCP and LLDP approaches, the use of PKI might be infeasible on resource-constrained devices. This aspect is not addressed by existing MUD implementations (except the UMU-MUD approach), which are usually based on DHCP and LLDP protocols to obtain the MUD URL. Furthermore, the authentication of existing commercial IoT devices is usually based on simple login/password mechanisms instead of device certificates. In this direction, recent proposals, such as [18], [19] implement pre-shared key (PSK) based authentication using an EAP-based authentication framework. However, the use of EAP might also entail performance issues on constrained networks with limited Maximum Transmission Unit (MTU) sizes, such as LPWAN (e.g., LORaWAN [51]). Furthermore, the use of PSK approaches could pose scalability issues as devices would need to share such key with the authenticator endpoint. Therefore, there is a need to evaluate lightweight authentication mechanisms for obtaining the MUD URL considering the heterogeneity and constraints of the underlying network technologies. For this purpose, the use of emerging alternative application-level protocols, such as the Ephemeral Diffie-Hellman Over Cose (EDHOC) [125], [126] and Object Security for Constrained RESTful Environments (OSCORE) need to be explored in the deployment of the MUD standard. These technologies have been specifically designed for resource-constrained environments by using lightweight representation formats (i.e., Concise Binary Object Representation (CBOR) [127]) and could mitigate performance issues of existing mechanisms for obtaining MUD files.

F. TRAFFIC ANALYSIS BASED ON MUD RESTRICTIONS

The network restrictions contained in a MUD file can be used to compare the actual behavior of a device with its MUD profile by using traffic analysis techniques. As described in Section III-D, several proposals use tools to monitor devices' behavior to detect potential attacks using

MUD restrictions [31], [32]. Furthermore, as described by recent NIST efforts [13], these tools could be included in MUD-enabled environments to analyze network behavior. The MUD concept could be used to mitigate the issue in traffic analysis that network changes can invalidate the traffic analysis models already created. In MUD-enabled environments, the network manager can easily instantiate and deploy new MUD profiles to address these changes. Additionally, the restrictions contained in the MUD files of devices can be aggregated to create a complete network model to detect potential attacks with different targets. For this purpose, a potential approach could be based on the use of graph-based techniques (e.g., graph embedding [128]) where communication endpoints are represented as graph nodes and the interactions are described as the edges. This approach was proposed by [129], which uses techniques based on graph kernels [130] to represent MUD restrictions, or [131], which proposes the use of graph neural networks [132]. Another potential approach to be explored in the coming years is represented by the use of federated learning [94], in which end devices do not share their network traces for traffic analysis purposes, but updates of the model to be learned. Besides the use of MUD restrictions as an input for machine learning based traffic analysis, such rules could also be considered to mitigate potential adversarial attacks [133], as described by [93] in the case of federated learning. Depending of the scenario, the use of these techniques could also mitigate the privacy issues associated to the MUD manager component, which could be aware of all the traffic of MUD-capable IoT devices in a network [13].

G. MUD FILES SHARING

As discussed in Section V-D, a device's intended behavior can change during its lifecycle that could, in turn, require the update of the associated MUD file. According to the MUD standard specification, the process for obtaining a MUD file is carried out through the MUD file server entity, which is contacted by the MUD manager using the MUD URL sent by a device. However, it does not define any mechanism to obtain updated versions of MUD files or approaches for manufacturers to communicate possible changes in the behavior of their devices. Furthermore, the MUD file server can be considered as a single point of failure in the architecture. Indeed, as described by [13], there is a need to protect the MUD manager in the case that the MUD file server is compromised. To address this issue, the use of distributed ledger technologies (DLTs) [134] (e.g., blockchain [135]) could be considered to create a platform for sharing behavioral profiles of IoT devices. The use of blockchain could mitigate the issue associated to the MUD file server as single point of failure, and would allow to keep track of the different MUD file's versions associated with a certain device. Furthermore, the establishment of a blockchain platform could be used to link MUD profiles with threats discovered on those devices, as described by our previous works [136], [137]. This platform would allow to provide the intended behavior of a device during its lifecycle and increase the transparency in the use of new IoT devices and technologies. An additional blockchain-related application is the creation of smart contracts based on MUD profiles to ensure the intended behavior of IoT devices as proposed by [138].

H. MUD-BASED POTENTIAL APPLICATIONS

In 2019, the new EU cybersecurity regulation "Cybersecurity Act" [139] was adopted to define a cybersecurity certification framework, so that any ICT product, service or process can be evaluated in terms of cybersecurity. In general, cybersecurity certification refers to the evaluation process to verify the conformity of a system with a certain set of requirements [140]. In the case of IoT devices [141], some of these requirements could be represented by the restrictions defined in a MUD profile that could be used to design a set of tests in order to assess whether a device operates according to its intended behavior. These aspects were addressed by recent works [20], [55], [142], which propose an automated testing for the security assessment of IoT devices, as well as [98], which describes a testing methodology to verify the compliance with MUD profiles. Based on these works, the use of MUD profiles would favor the creation of automated tests to obtain an accurate assessment of the security level of a device, and would foster the dialogue between device manufacturers and policy-makers to provide a transparent view on the cybersecurity level of devices in the market.

Other potential applications could be derived from the use of the MUD standard in specific sectors. In particular, the automotive sector could leverage the standardized approach of MUD to cope with security threats. Indeed, vehicles are becoming increasingly connected and automated but they can also be more vulnerable to cybersecurity attacks [143], [144]. In this context, cybersecurity threats can become safety hazards because a compromised Artificial Intelligence (AI) component in an automated vehicle can lead to car accidents with potential loss of human lives. In addition, the increased connectivity of cars can increase the attack surface as demonstrated in [145]. There is the need to enhance the security of vehicles by giving greater control to the vehicle manufacturers even after the vehicle is deployed in the market. From this point of view, the MUD concept can be useful to mitigate the problem of updating the security measures in automotive vehicles without issuing recalls, which can be quite costly for the vehicle manufacturer. In fact, new MUD profiles can be downloaded to the vehicles to mitigate new found vulnerabilities to minimize the need of a recall with the related action to bring the vehicle to the workshop. We recognize that the MUD standard [124] was defined for a completely different context and some tailoring will be needed. For example, the most common in-vehicle network standard in vehicles (i.e., CAN-bus) is quite old and it was not designed with security functions in mind [143]. Then, some functions needed by the MUD standard like authentication must be re-designed. In addition, there may be limitations in connectivity (since continuous connectivity may not be always ensured with moving vehicles due to lack of wireless coverage). Then, the challenge described before in V-D and related to dynamism is relevant to this context.

I. DEPLOYMENT OF MUD IN THE 5G ERA

With the advent of 5G technology [146], the definition of behavioral profiles for the next generation of 5G-enabled devices can be a key factor to reduce the attack surface of such components. The definition of these profiles would promote a common understanding of the risks associated with 5G systems, as well as the creation of well-established and recognized testing methodology to verify the compliance of these systems against existing cybersecurity best practices for 5G [147]. However, the 5G ecosystem could integrate general-purpose systems with more complex functionality than in the case of IoT devices. This aspect could require the extension of the current MUD data model, as well as a more complex architecture to enforce the behavioral profiles of such systems. Some preliminary approaches have considered the definition of MUD profiles for 5G systems like in [48] or in contexts where 5G is deployed in combination with other concepts like NFV or Fog in [148] to support security aspects. Both references have pointed out that further research is required to come up with suitable behavioral profiles to reflect the complexity of the 5G ecosystem. Additionally, the process for obtaining MUD files must be integrated in the existing 5G authentication mechanisms (e.g., 5G-AKA [149]).

Another recent use of the MUD standard in 5G technology has been proposed by the authors in [150], in relation to the 5G slicing concept where several logical networks are deployed on the top of the same infrastructure and each 5G slice is optimized to fulfill certain objectives imposed by the specific use case. In a multi-party and multi-layer 5G architecture, the definition of liabilities and responsibilities in case of a security breach may be complex to manage, but they are still essential to support confidence between parties and compliance with regulation. In this complex environment, MUDs can be used to enforce controls in the network components to ensure that their characteristics and functioning comply with their obligations (e.g., Service Level Agreement, regulations) and capabilities in order to keep the threat and liability levels at an acceptable level.

J. MUD STANDARDIZATION EFFORTS

Using a standardized approach to describe IoT devices' behavioral profiles can support the development of more secure and interoperable IoT scenarios. Therefore, the development of possible MUD extensions should be considered within the scope of SDOs to foster a large-scale adoption. In addition to NIST initiatives to promote the use of MUD in IoT environments [13], in recent years, several efforts have been proposed at the IETF to extend the MUD data model for different purposes. In particular, [151] proposes an extension to characterize the traffic of a device indicating the bandwidth and the time required to use a certain service. Furthermore, [121] describes an extension to describe connection parameters related to the use of the TLS and DTLS protocols. With a similar motivation, [152] is intended to include software information, such as versioning and dependencies to MUD profiles. Related to device software information, [153] proposes to integrate MUD behavior information with information about software/firmware updates. In addition to these approaches addressing the need to extend the MUD data model (see Section V-A), other ongoing efforts address additional aspects related to the use of MUD profiles. For example, [154] proposes an approach to define the possible update of the MUD URL for a certain device. Additionally, [155] proposes an alternative method to obtain the MUD URL parameter using QR codes. In addition to these initiatives, other aspects of the MUD standard may be subject to standardization efforts in the coming years, such as the communication between the MUD manager and the router, the extension of the architecture to consider software updates, as well as lightweight authentication mechanisms for constrained devices (see Section V-E). These aspects are mentioned by [12], which proposes a high-level architecture intended to link MUD profiles with threat signaling and software updates. Furthermore, [13] highlights that the lack of standardized approaches for the communication between MUD components could inhibit the interoperability of MUD-based implementations. The development of these proposals by using standard mechanisms could also foster the development of more interoperable MUD implementations that would favor their deployment in the market.

Previous challenges represent some of the current obstacles for a wide adoption of the MUD standard. It should be noted that in addition to NIST's interest in defining several MUD-based scenarios to protect home and small-business networks [12], [13], other initiatives have emerged in recent years. For example, IoTivity [156] represents an open source project implementing the Open Connectivity Foundation (OCF) standards to promote secure communication of IoT devices. Indeed, the recent OCF specification [157] proposes the use of MUD by using a possible extension for X.509 certificates associated with end devices. Furthermore, the Internet Society has developed a document on security recommendations for IoT that promotes the development and deployment of MUD [158]. In addition, the use of MUD has recently been reported by several IoT security practitioners as a promising approach for IoT security [159]. These initiatives should address additional challenges inherent to IoT, including usability requirements through the development of security labels [160] to increase the cybersecurity awareness of end users in the coming years.

VI. CONCLUSION

In a still fragmented landscape of IoT security protocols and mechanisms, the development of standardized approaches is key for the development of secure and interoperable applications. In this work, we provided a comprehensive analysis of the MUD standard, which has been recently proposed to define the intended network behavior of IoT devices. This standard is intended to reduce the attack surface of current IoT environments by providing controls and limiting the communication of end devices. We provided a thorough analysis of existing research proposals, which were classified according to MUD profiles' lifecycle. Based on such analysis and our own experience on this area, we described the main challenges as well as a list of potential research directions to be considered in the coming years for the implementation and deployment of the MUD standard in different contexts and applications. Although MUD is not widely deployed yet, we believe that the adoption of a standard in the directions identified in this study could foster the collaboration between device manufacturers, policy makers and Standard Developing Organizations (SDO)s for the creation of a more secure ecosystem of IoT devices to be leveraged by end users.

REFERENCES

[1] J. H. Ramos and A. Skarmeta, *Security and Privacy in the Internet of Things: Challenges and Solutions*, vol. 27. Amsterdam, The Netherlands: IOS Press, 2020.

[2] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, ''DDoS in the IoT: Mirai and other botnets,'' *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[3] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, ''Internet of Things security: A top-down survey,'' *Comput. Netw.*, vol. 141, pp. 199–221, Aug. 2018.

[4] D. Barrera, I. Molloy, and H. Huang, ''IDIoT: Securing the Internet of Things like it's 1994,'' 2017, *arXiv:1712.03623*. [Online]. Available: http://arxiv.org/abs/1712.03623

[5] D. Barrera, I. Molloy, and H. Huang, ''Standardizing IoT network security policy enforcement,'' in *Proc. Workshop Decentralized IoT Secur. Standards*, 2018, p. 6.

[6] K. A. P. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, ''Internet of Things: A survey on machine learning-based intrusion detection approaches,'' *Comput. Netw.*, vol. 151, pp. 147–157, Mar. 2019.

[7] M. Serror, S. Hack, M. Henze, M. Schuba, and K. Wehrle, ''Challenges and opportunities in securing the industrial Internet of Things,'' *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 2985–2996, May 2021.

[8] I. Stellios, P. Kotzanikolaou, M. Psarakis, C. Alcaraz, and J. Lopez, ''A survey of IoT-enabled cyberattacks: Assessing attack paths to critical infrastructures and services,'' *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3453–3495, 4th Quart., 2018.

[9] A. Sładkowski and W. Pamuła, *Intelligent Transportation Systems—Problems and Perspectives*, vol. 303. Cham, Switzerland: Springer, 2016. [Online]. Available: https://www.springer.com/gp/book/9783319191492

[10] E. Lear, D. Romascanu, and R. Droms, *Manufacturer Usage Description Specification*, document RFC 8520, 2019.

[11] J. Melzer, J. Latour, M. Richardson, A. Ali, and W. Almuhtadi, ''Network approaches to improving consumer IoT security,'' in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2020, pp. 1–6.

[12] T. Polk, M. Souppaya, and W. C. Barker. (2017). *Mitigating IoT-Based Automated Distributed Threats*. [Online]. Available: https://www.nccoe.nist.gov/sites/default/files/library/project-descriptions/iot-ddos-project-description-draft.pdf

[13] *Securing Small-Business and Home Internet of Things Devices: NIST SP 1800-15*, NIST, Gaithersburg, MD, USA, 2019.

[14] *Good Practices for Security of IoT—Secure Software Development Lifecycle*, ENISA, Athens, Greece, 2019.

[15] N. Mazhar, R. Salleh, M. Asif, and M. Zeeshan, ''SDN based intrusion detection and prevention systems using manufacturer usage description: A survey,'' *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 12, pp. 717–737, 2020.

[16] J. H. Cox, J. Chung, S. Donovan, J. Ivey, R. J. Clark, G. Riley, and H. L. Owen, ''Advancing software-defined networks: A survey,'' *IEEE Access*, vol. 5, pp. 25487–25526, 2017.

[17] N. Mazhar, R. Salleh, M. Zeeshan, and M. M. Hameed, ''Role of device identification and manufacturer usage description in IoT security: A survey,'' *IEEE Access*, vol. 9, pp. 41757–41786, 2021.

[18] S. N. M. García, A. M. Zarca, J. L. Hernández-Ramos, J. B. Bernabé, and A. S. Gómez, ''Enforcing behavioral profiles through software-defined networks in the industrial Internet of Things,'' *Appl. Sci.*, vol. 9, no. 21, p. 4576, Oct. 2019.

[19] S. N. Matheu, A. R. Enciso, A. M. Zarca, D. Garcia-Carrillo, J. L. Hernández-Ramos, J. B. Bernabe, and A. F. Skarmeta, ''Security architecture for defining and enforcing security profiles in DLT/SDN-based IoT systems,'' *Sensors*, vol. 20, no. 7, p. 1882, Mar. 2020.

[20] S. N. Matheu, J. L. Hernandez-Ramos, S. Perez, and A. F. Skarmeta, ''Extending MUD profiles through an automated IoT security testing methodology,'' *IEEE Access*, vol. 7, pp. 149444–149463, 2019.

[21] G. Baldini, J. L. Hernandez-Ramos, S. Nowak, R. Neisse, and M. Nowak, ''Mitigation of privacy threats due to encrypted traffic analysis through a policy-based framework and MUD profiles,'' *Symmetry*, vol. 12, no. 9, p. 1576, Sep. 2020.

[22] IEEE. (2018). *802.1AR—Secure Device Identity*. [Online]. Available: https://1.ieee802.org/security/802-1ar/

[23] M. Bjorklund, *The YANG 1.1 Data Modeling Language*, document RFC 7950, 2016. [Online]. Available: https://tools.ietf.org/html/rfc7950

[24] T. Bray, *The JavaScript Object Notation (JSON) Data Interchange Format*, document RFC 8259, 2017. [Online]. Available: https://tools.ietf.org/html/rfc8259

[25] M. Jethanandani, D. Blair, L. Huang, and S. Agarwal, *YANG Data Model for Network Access Control Lists*, document RFC 8519, 2019. [Online]. Available: https://tools.ietf.org/html/rfc8519

[26] *MUD Maker*. Accessed: Aug. 20, 2021. [Online]. Available: https://mudmaker.org/

[27] O. Garcia-Morchon, S. S. Kumar, and M. Sethi, *Internet of Things Security: State of the Art and Challenges*, document RFC 8576, 2019. [Online]. Available: https://tools.ietf.org/html/rfc8576

[28] R. Droms, *Dynamic Host Configuration Protocol*, document RFC 2131, 1997. [Online]. Available: https://tools.ietf.org/html/rfc2131

[29] *Link Layer Discovery Protocol*. Accessed: Aug. 20, 2021. [Online]. Available: http://www.ieee802.org/1/files/public/docs2002/lldp-protocol-01.pdf

[30] M. Al-Shaboti, I. Welch, A. Chen, and M. A. Mahmood, ''Towards secure smart home IoT: Manufacturer and user network access control framework,'' in *Proc. IEEE 32nd Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, May 2018, pp. 892–899.

[31] L. Chang. *A Proactive Approach to Detect IoT Based Flooding Attacks by Using Software Defined Networks and Manufacturer Usage Descriptions*. Accessed: Aug. 20, 2021. [Online]. Available: https://repository.asu.edu/attachments/207561/content/Chang_asu_0010N_18188.pdf

[32] H. J. Hadi, S. M. Sajjad, and K. U. Nisa, ''BoDMitM: Botnet detection and mitigation system for home router base on MUD,'' in *Proc. Int. Conf. Frontiers Inf. Technol. (FIT)*, Dec. 2019, pp. 139–1394.

[33] S. M. Sajjad, M. Yousaf, H. Afzal, and M. R. Mufti, ''EMUD: Enhanced manufacturer usage description for IoT botnets prevention on home WiFi routers,'' *IEEE Access*, vol. 8, pp. 164200–164213, 2020.

[34] A. Hamza, D. Ranathunga, H. H. Gharakheili, M. Roughan, and V. Sivaraman, ''Clear as MUD: Generating, validating and applying IoT behavioral profiles,'' in *Proc. Workshop IoT Secur. Privacy*, Aug. 2018, pp. 8–14.

[35] *Mudgee*. Accessed: Aug. 20, 2021. [Online]. Available: https://github.com/ayyoob/mudgee

[36] A. Hamza, H. H. Gharakheili, and V. Sivaraman, ''Combining MUD policies with SDN for IoT intrusion detection,'' in *Proc. Workshop IoT Secur. Privacy*, New York, NY, USA, Aug. 2018, pp. 1–7.

[37] A. Hamza, D. Ranathunga, H. H. Gharakheili, T. A. Benson, M. Roughan, and V. Sivaraman, ''Verifying and monitoring IoTs network behavior using MUD profiles,'' Feb. 2019, *arXiv:1902.02484*. [Online]. Available: http://arxiv.org/abs/1902.02484

[38] H. H. Gharakheili, A. Sivanathan, A. Hamza, and V. Sivaraman, ''Network-level security for the Internet of Things: Opportunities and challenges,'' *Computer*, vol. 52, no. 8, pp. 58–62, Aug. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8780392/

[39] M. H. Mazhar and Z. Shafiq, ''Characterizing smart home IoT traffic in the wild,'' 2020, *arXiv:2001.08288*. [Online]. Available: http://arxiv.org/abs/2001.08288

[40] R. Fontein and E. Khan. *For Whom the IoT-Bell Tolls*. Accessed: Aug. 20, 2021. [Online]. Available: https://telluur.com/utwente/master/SSI%20-%20Security%20Services%20for%20the%20IoT/Project/For_Whom_the_IoT_Bell_Tolls.pdf

[41] C. Schutijser. *Towards Automated DDoS Abuse Protection Using MUD Device Profiles*. Accessed: Aug. 20, 2021. [Online]. Available: https://www.sidnlabs.nl/downloads/theses/towards_automated_ddos_abusse_protection_cschutijser.pdf

[42] M. Nakahara, N. Okui, Y. Kobayashi, and Y. Miyake. (2021). *Malware Detection for IoT Devices Using Automatically Generated White List and Isolation Forest*. [Online]. Available: https://www.scitepress.org/Papers/2021/103949/103949.pdf

[43] J. Zander, I. Schieferdecker, and P. J. Mosterman, *Model-Based Testing for Embedded Systems*. Boca Raton, FL, USA: CRC Press, 2017. [Online]. Available: https://www.routledge.com/Model-Based-Testing-for-Embedded-Systems/Zander-Schieferdecker-Mosterman/p/book/9781138076457

[44] Z. Jin, Y. M. Lee, C. H. Copass, and Y. Park, "Building system with dynamic manufacaturer usage description (MUD) files based on building model queries," U.S. Patent 16 666 005, Apr. 30, 2020.

[45] A. M. Zarca, J. B. Bernabe, R. Trapero, D. Rivera, J. Villalobos, A. Skarmeta, S. Bianchi, A. Zafeiropoulos, and P. Gouvas, "Security management architecture for NFV/SDN-aware IoT systems," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8005–8020, Oct. 2019.

[46] E. Rescorla and N. Modadugu, *Datagram Transport Layer Security Version 1.2*, document RFC 6347, Request for Comments, IETF, Jan. 2012. [Online]. Available: https://tools.ietf.org/html/rfc6347

[47] E. Lear, J. Henry, and R. Barton. *Determining Nominal Quality of Service Needs of a Device*. Accessed: Aug. 20, 2021. [Online]. Available: https://www.tdcommons.org/dpubs_series/1625

[48] M. Hanes, C. Byers, J. Clarke, and G. Salgueiro. *Human Usage Description for 5G Networks Endpoints*. Accessed: Aug. 20, 2021. [Online]. Available: https://www.tdcommons.org/dpubs_series/1254

[49] S. Singh, A. Atrey, M. L. Sichitiu, and Y. Viniotis, "Clearer than mud: Extending manufacturer usage description (MUD) for securing IoT systems," in *Internet of Things—ICIOT*, vol. 11519, V. Issarny, B. Palanisamy, and L.-J. Zhang, Eds. Cham, Switzerland: Springer, 2019, pp. 43–57.

[50] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: An overview, II," *WIREs Data Mining Knowl. Discovery*, vol. 7, no. 6, p. e1219, Nov. 2017.

[51] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of LoRaWAN," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 34–40, Sep. 2017.

[52] A. Singh, S. Murali, L. Rieger, R. Li, S. Hommes, R. State, G. Ormazabal, and H. Schulzrinne, "HANZO: Collaborative network defense for connected things," in *Proc. Princ., Syst. Appl. IP Telecommun. (IPTComm)*, Oct. 2018, pp. 1–8.

[53] J. Tanha, M. van Someren, and H. Afsarmanesh, "Semi-supervised self-training for decision tree classifiers," *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 1, pp. 355–370, Feb. 2015.

[54] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, Jun. 1999.

[55] S. N. Matheu, J. L. Hernandez-Ramos, and A. F. Skarmeta, "Toward a cybersecurity certification framework for the Internet of Things," *IEEE Security Privacy*, vol. 17, no. 3, pp. 66–76, May/Jun. 2019.

[56] A. Praseed and P. S. Thilagam, "DDoS attacks at the application layer: Challenges and research perspectives for safeguarding web applications," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 661–685, 1st Quart., 2019.

[57] M. Ranganathan, "Soft MUD: Implementing manufacturer usage descriptions on OpenFlow SDN switches," in *Proc. Int. Conf. Netw. (ICN)*, 2019, pp. 49–54. [Online]. Available: https://thinkmind.org/index.php?view=article&articleid=icn_2019_4_20_30014

[58] J. Vollbrecht, M. Holdrege, C. Laat, P. Calhoun, L. Gommans, S. Farrell, B. D. Bruijn, G. Gross, and D. Spence, *AAA Authorization Framework*, document RFC 2904, 2000.

[59] B. Aboba and P. R. Calhoun. (2003). *RADIUS (Remote Authentication Dial in User Service) Support for Extensible Authentication Protocol (EAP)*. [Online]. Available: https://tools.ietf.org/html/rfc3579

[60] V. Andalibi, D. Kim, and L. J. Camp. *Throwing MUD Into the FOG: Defending IoT and Fog by Expanding MUD to Fog Network*. Accessed: Aug. 20, 2021. [Online]. Available: https://www.usenix.org/conference/hotedge19/presentation/andalibi

[61] J. Ni, K. Zhang, X. Lin, and X. S. Shen, "Securing fog computing for Internet of Things applications: Challenges and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 601–628, 1st Quart., 2018.

[62] R. B. Shah, B. E. Weis, K. Kumar, and M. K. Nayak, "Zero-touch IoT device provisioning," U.S. Patent 10 298 581, May 21, 2019.

[63] M. Pritikin, M. Richardson, M. Behringer, S. Bjarnason, and K. Watsen, "Bootstrapping remote secure key infrastructures (BRSKI)," IETF, Fremont, CA, USA, Tech. Rep. IETF RFC 8995, 2020. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8995.pdf

[64] H. Zhou, N. Cam-Winget, J. Salowey, and S. Hanna, *Tunnel Extensible Authentication Protocol (TEAP) Version 1*, document RFC 7170, 2014.

[65] E. Lear, B. Weis, and E. Nilsen-Nygaard. *Automatic Access-Control Admission and Management of Controllers for Things Using Manufacturer Usage Description*. Accessed: Aug. 20, 2021. [Online]. Available: https://www.tdcommons.org/dpubs_series/1557

[66] J. Höglund, S. Lindemer, M. Furuhed, and S. Raza, "PKI4IoT: Towards public key infrastructure for the Internet of Things," *Comput. Secur.*, vol. 89, Feb. 2020, Art. no. 101658.

[67] Y. Ohba, B. Patil, D. Forsberg, H. Tschofenig, and A. E. Yegin, *Protocol for Carrying Authentication for Network Access*, document RFC 5191, 2008.

[68] B. Aboba, D. Simon, and P. Eronen, *Extensible Authentication Protocol (EAP) Key Management Framework*, document RFC 5247, 2008. [Online]. Available: https://tools.ietf.org/html/rfc5247

[69] F. Bersani and H. Tschofenig, *The EAP-PSK Protocol: A Pre-Shared Key Extensible Authentication Protocol Method*, document RFC 4764, 2007.

[70] Z. Shelby, K. Hartke, and C. Bormann, *The Constrained Application Protocol (CoAP)*, document RFC 7252, 2014. [Online]. Available: https://tools.ietf.org/html/rfc7252

[71] P. Yadav, V. Safronov, and R. Mortier, "Enforcing accountability in smart built-in IoT environment using MUD," in *Proc. 6th ACM Int. Conf. Syst. Energy-Efficient Buildings, Cities, Transp. (BuildSys)*, Nov. 2019, pp. 368–369, doi: 10.1145/3360322.3361004.

[72] I. Farris, T. Taleb, Y. Khettab, and J. Song, "A survey on emerging SDN and NFV security mechanisms for IoT systems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 812–837, 1st Quart., 2019.

[73] S. Bera, S. Misra, and A. V. Vasilakos, "Software-defined networking for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1994–2008, Dec. 2017.

[74] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 602–622, 1st Quart., 2016.

[75] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow—Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, p. 69, 2008.

[76] A. Hamza, H. H. Gharakheili, T. A. Benson, and V. Sivaraman, "Detecting volumetric attacks on loT devices via SDN-based monitoring of MUD activity," in *Proc. ACM Symp. SDN Res.*, San Jose, CA, USA, Apr. 2019, pp. 36–48.

[77] J. Bailey and S. Stuart, "Faucet: Deploying SDN in the enterprise," *Commun. ACM*, vol. 60, no. 1, pp. 45–49, Dec. 2016.

[78] B. Pfaff, J. Pettit, T. Koponen, E. Jackson, A. Zhou, J. Rajahalme, J. Gross, A. Wang, J. Stringer, and P. Shelar, "The design and implementation of open vSwitch," in *12th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2015, pp. 117–130.

[79] B. Caswell and J. Beale, *Snort 2.1 Intrusion Detection*. Amsterdam, The Netherlands: Elsevier, 2004. [Online]. Available: https://www.elsevier.com/books/snort-21-intrusion-detection-second-edition/caswell/978-1-931836-04-3

[80] *H2020 ANASTACIA EU Project*. Accessed: Aug. 20, 2021. [Online]. Available: http://anastacia-h2020.eu/

[81] S. Erdtman, E. Wahlstroem, H. Tschofenig, and M. Jones. (2018). *CBOR Web Token (CWT)*. [Online]. Available: https://tools.ietf.org/html/rfc8392

[82] C. Bormann. (2020). *An Authorization Information Format (AIF) for ACE*. [Online]. Available: https://tools.ietf.org/html/draft-ietf-ace-aif-00

[83] I. B. Fink, M. Serror, and K. Wehrle. (2020). *Extending MUD to Smartphones*. [Online]. Available: https://www.comsys.rwth-aachen.de/fileadmin/papers/2020/2020-fink-lcn-mud-smartphone.pdf

[84] OASIS. (2013). *Extensible Access Control Markup Language Version 3.0*. [Online]. Available: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

[85] A. Feraudo, P. Yadav, R. Mortier, P. Bellavista, and J. Crowcroft, "SoK: Beyond IoT MUD deployments—challenges and future directions," 2020, *arXiv:2004.08003*. [Online]. Available: http://arxiv.org/abs/2004.08003

[86] H. J. Stuart, "The exponentially weighted moving average," *J. Quality Technol.*, vol. 18, no. 4, pp. 203–210, 1986.

[87] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: An overview," *Wiley Interdiscipl. Rev. Data Mining Knowl. Discovery*, vol. 2, no. 1, pp. 86–97, 2012.

[88] Y. Afek, A. Bremler-Barr, D. Hay, L. Shafir, and I. Zhaika, "Demo: NFV-based IoT security at the ISP level," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Apr. 2020, pp. 1–2.

[89] Y. Afek, A. Bremler-Barr, D. Hay, R. Goldschmidt, L. Shafir, G. Abraham, and A. Shalev, "NFV-based IoT security for home networks using MUD," 2019, *arXiv:1911.00253*. [Online]. Available: http://arxiv.org/abs/1911.00253

[90] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 90–97, Feb. 2015.

[91] M. D. Thomsen, A. Giaretta, and N. Dragoni, "Smart lamp or security camera? Automatic identification of IoT devices," in *Proc. Int. Netw. Conf.* Cham, Switzerland: Springer, 2020, pp. 85–99.

[92] Y. Ma and G. Guo, *Support Vector Machines Applications*, vol. 649. Cham, Switzerland: Springer, 2014. [Online]. Available: https://www.springer.com/gp/book/9783319022994

[93] A. Feraudo, P. Yadav, V. Safronov, D. A. Popescu, R. Mortier, S. Wang, P. Bellavista, and J. Crowcroft, "CoLearn: Enabling federated learning in MUD-compliant IoT edge networks," in *Proc. 3rd ACM Int. Workshop Edge Syst., Anal. Netw.*, Apr. 2020, pp. 25–30.

[94] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[95] R. Neisse, G. Steri, I. N. Fovino, and G. Baldini, "SecKit: A model-based security toolkit for the Internet of Things," *Comput. Secur.*, vol. 54, pp. 60–76, Oct. 2015.

[96] V. Lade, A. Mohan, and S. Patil. *802.11AX for the Internet of Things—Machine Learning Assisted Optimized Power Save Techniques for IoT Devices Using 802.11AX Target Wake Time*. Accessed: Aug. 20, 2021. [Online]. Available: https://www.tdcommons.org/dpubs_series/1546

[97] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A tutorial on IEEE 802.11 ax high efficiency WLANs," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 197–216, 1st Quart., 2019.

[98] C. Gangurde. *Automation of IoT Pre-Certification Security Testing Environment Based on the Manufacturing Usage Description*. Accessed: Aug. 20, 2021. [Online]. Available: https://research.tue.nl/en/studentTheses/automation-of-iot-pre-certification-security-testing-environment-

[99] (2018). *Open Source Manufacture Usage Specification*. [Online]. Available: https://osmud.org

[100] *Openwrt Project*. Accessed: Aug. 20, 2021. [Online]. Available: https://openwrt.org/

[101] osMUD. (2018). *Dnsmasq*. [Online]. Available: https://github.com/osmud/dnsmasq

[102] OpenWRT. *OpenWRT Table of Hardware*. Accessed: Aug. 20, 2021. [Online]. Available: https://openwrt.org/toh/start

[103] *Micronets Manufacturer Usage Description (MUD) Tools*. Accessed: Aug. 20, 2021. [Online]. Available: https://github.com/cablelabs/micronets-mud-tools

[104] *Mud-Manager*. Accessed: Aug. 20, 2021. [Online]. Available: https://github.com/CiscoDevNet/MUD-Manager

[105] D. Van der Walt, *FreeRADIUS Beginner's Guide*. Birmingham, U.K.: Packt, 2011.

[106] D. Dodson *et al.*, "Securing small business and home Internet of Things (IoT) devices: Mitigating network-based attacks using manufacturer usage description (MUD)," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep., 2020. [Online]. Available: https://www.nist.gov/publications/securing-small-business-and-home-internet-things-iot-devices-mitigating-network-based

[107] J. Medved, R. Varga, A. Tkacik, and K. Gray, "OpenDaylight: Towards a model-driven SDN controller architecture," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw.*, Jun. 2014, pp. 1–6.

[108] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, and G. Parulkar, "ONOS: Towards an open, distributed SDN OS," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw.*, 2014, pp. 1–6.

[109] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with COOJA," in *Proc. 31st IEEE Conf. Local Comput. Netw.*, Nov. 2006, pp. 641–648.

[110] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki—A lightweight and flexible operating system for tiny networked sensors," in *Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw.*, Nov. 2004, pp. 455–462.

[111] P. Moreno-Sanchez, R. Marin-Lopez, and F. Vidal-Meca, "An open source implementation of the protocol for carrying authentication for network access: OpenPANA," *IEEE Netw.*, vol. 28, no. 2, pp. 49–55, Mar. 2014.

[112] *MUDdy Tool*. Accessed: Aug. 20, 2021. [Online]. Available: https://github.com/lstn/muddy

[113] *Mud Visualizer*. Accessed: Aug. 20, 2021. [Online]. Available: https://mudmaker.org/mudvisualizer.php

[114] V. Andalibi, J. Dev, D. Kim, E. Lear, and L. J. Camp, "Making access control easy in IoT," in *Proc. Int. Symp. Hum. Aspects Inf. Secur. Assurance*. Cham, Switzerland: Springer, 2021, pp. 127–137.

[115] *MUD-URL-Validator*. Accessed: Aug. 20, 2021. [Online]. Available: https://github.com/CiscoDevNet/MUD-URL-Validator

[116] C. Sanders, *Practical Packet Analysis: Using Wireshark to Solve Real-world Network Problems*. San Francisco, CA, USA: No Starch Press, 2017.

[117] *MUD-PD*. Accessed: Aug. 20, 2021. [Online]. Available: https://github.com/usnistgov/MUD-PD

[118] P. Watrobski, J. Klosterman, W. Barker, and M. Souppaya, "Methodology for characterizing network behavior of Internet of Things devices (draft)," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep., 2020. [Online]. Available: https://csrc.nist.gov/publications/detail/white-paper/2020/04/01/methodology-for-characterizing-network-behavior-of-iot-devices/draft

[119] *NIST MUD*. Accessed: Aug. 20, 2021. [Online]. Available: https://github.com/usnistgov/nist-mud

[120] K. Hong, Y. Kim, H. Choi, and J. Park, "SDN-assisted slow HTTP DDoS attack defense method," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 688–691, Apr. 2018.

[121] T. Reddy, D. Wing, and B. Anderson. (2020). *MUD (D)TLS Profiles for IoT Devices*. Accessed: Aug. 20, 2021. [Online]. Available: https://tools.ietf.org/html/draft-reddy-opsawg-mud-tls-05

[122] G. Liu, W. Pei, Y. Tian, C. Liu, and S. Li, "A novel conflict detection method for ABAC security policies," *J. Ind. Inf. Integr.*, vol. 22, Jun. 2021, Art. no. 100200.

[123] B. Stepien and A. Felty, "Resolving XACML rule conflicts using artificial intelligence," in *Proc. 3rd Int. Conf. Inf. Sci. Syst.*, Mar. 2020, pp. 121–127.

[124] E. Lear, R. Droms, and D. Romascanu, *Manufacturer Usage Description Specification*, document RFC 8520, Mar. 2019.

[125] G. Selander, J. Mattsson, and F. Palombini. (2021). Ephemeral Diffie–Hellman over COSE (EDHOC). IETF. [Online]. Available: https://tools.ietf.org/html/draft-ietf-lake-edhoc-05

[126] S. Pérez, J. L. Hernández-Ramos, S. Raza, and A. Skarmeta, "Application layer key establishment for end-to-end security in IoT," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2117–2128, Mar. 2020.

[127] C. Bormann and P. Hoffman, *Concise Binary Object Representation (CBOR)*, document RFC 8949, 2020. [Online]. Available: https://tools.ietf.org/html/rfc8949

[128] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowl.-Based Syst.*, vol. 151, pp. 78–94, Jul. 2018.

[129] S. McKinney. (2019). *Graph-Based Analysis for IoT Devices With Manufacturer Usage Descriptions—An ScM Research Project*. [Online]. Available: https://cs.brown.edu/research/pubs/theses/masters/2019/mckinney.samuel.pdf

[130] N. M. Kriege, F. D. Johansson, and C. Morris, "A survey on graph kernels," *Appl. Netw. Sci.*, vol. 5, no. 1, pp. 1–42, Dec. 2020.

[131] E. Gelenbe, P. Fröhlich, M. Nowak, S. Papadopoulos, A. Protogerou, A. Drosou, and D. Tzovaras, "IoT network attack detection and mitigation," in *Proc. 9th Medit. Conf. Embedded Comput. (MECO)*, Jun. 2020, pp. 1–6.

[132] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 793–803.

[133] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*. [Online]. Available: http://arxiv.org/abs/1706.06083

[134] A. Sunyaev, "Distributed ledger technology," in *Internet Computing*. Cham, Switzerland: Springer, 2020, pp. 265–299.

[135] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[136] R. Neisse, J. L. Hernández-Ramos, S. N. Matheu, G. Baldini, and A. Skarmeta, ''Toward a blockchain-based platform to manage cybersecurity certification of IoT devices,'' in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Oct. 2019, pp. 1–6.

[137] R. Neisse, J. L. Hernández-Ramos, S. N. Matheu-Garcia, G. Baldini, A. Skarmeta, V. Siris, D. Lagutin, and P. Nikander, ''An interledger blockchain platform for cross-border management of cybersecurity information,'' *IEEE Internet Comput.*, vol. 24, no. 3, pp. 19–29, May 2020.

[138] P. Krishnan, K. Jain, K. Achuthan, and R. Buyya, ''Software-defined security-by-contract for blockchain-enabled MUD-aware industrial IoT edge networks,'' *IEEE Trans. Ind. Informat.*, early access, Jun. 2, 2021, doi: 10.1109/TII.2021.3084341.

[139] European Parliament. (2019). *Regulation (EU) 2019/881 of the European Parliament and of the Council of 17 April 2019 on ENISA (the European Union Agency for Cybersecurity) and on Information and Communications Technology Cybersecurity Certification (Cybersecurity Act)*. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2019/881/oj

[140] CNSSI. (2015). *CNSSI No. 4009: Committee on National Security Systems (CNSS) Glossary*. [Online]. Available: https://cryptosmith.files.wordpress.com/2015/08/glossary-2015-cnss.pdf

[141] S. N. Matheu, J. L. Hernández-Ramos, A. F. Skarmeta, and G. Baldini, ''A survey of cybersecurity certification for the Internet of Things,'' *ACM Comput. Surv.*, vol. 53, no. 6, pp. 1–36, Feb. 2021.

[142] S. N. Matheu-García, J. L. Hernández-Ramos, A. F. Skarmeta, and G. Baldini, ''Risk-based automated assessment and testing for the cybersecurity certification and labelling of IoT devices,'' *Comput. Standards Interfaces*, vol. 62, pp. 64–83, Feb. 2019.

[143] J. Petit and S. E. Shladover, ''Potential cyberattacks on automated vehicles,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 546–556, Apr. 2015.

[144] G. De La Torre, P. Rad, and K.-K.-R. Choo, ''Driverless vehicle security: Challenges and future research opportunities,'' *Future Gener. Comput. Syst.*, vol. 108, pp. 1092–1111, Jul. 2020.

[145] A. Greenberg, ''Hackers remotely kill a jeep on the highway—With me it,'' *Wired*, vol. 7, p. 21, Jul. 2015.

[146] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, ''Internet of Things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios,'' *IEEE Access*, vol. 8, pp. 23022–23040, 2020.

[147] ENISA. (2020). *ENISA Threat Landscape for 5G Networks Report*. [Online]. Available: https://www.enisa.europa.eu/publications/enisa-threat-landscape-report-for-5g-networks

[148] F. van Lingen, M. Yannuzzi, A. Jain, R. Irons-Mclean, O. Lluch, D. Carrera, J. L. Perez, A. Gutierrez, D. Montero, J. Marti, R. Maso, and A. J. P. Rodriguez, ''The unavoidable convergence of NFV, 5G, and fog: A model-driven approach to bridge cloud and edge,'' *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 28–35, Aug. 2017.

[149] D. Basin, J. Dreier, L. Hirschi, S. Radomirovic, R. Sasse, and V. Stettler, ''A formal analysis of 5G authentication,'' in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 1383–1396.

[150] C. Gaber, J. S. Vilchez, G. Gür, M. Chopin, N. Perrot, J.-L. Grimault, and J.-P. Wary, ''Liability-aware security management for 5G,'' in *Proc. IEEE 3rd 5G World Forum (5GWF)*, Sep. 2020, pp. 133–138.

[151] E. Lear and J. Henry. (2020). *Bandwidth Profiling Extensions for MUD*. [Online]. Available: https://tools.ietf.org/html/draft-lear-opsawg-mud-bw-profile-01

[152] E. Lear and S. Rose. (2020). *Discovering and Accessing Software Bills of Materials*. [Online]. Available: https://tools.ietf.org/html/draft-lear-opsawg-sbom-access-00

[153] B. Moran and H. Tschofenig. *Strong Assertions of IoT Network Access Requirements*. Accessed: Aug. 20, 2021. [Online]. Available: https://tools.ietf.org/html/draft-moran-suit-mud-00

[154] M. Richardson, W. Pan, and E. Lear. (2021). *Authorized Update to MUD URLs*. [Online]. Available: https://tools.ietf.org/html/draft-ietf-opsawg-mud-acceptable-urls-00

[155] M. Richardson, J. Latour, and H. Habibi. (2020). *On Loading MUD URLs From QR Codes*. [Online]. Available: https://tools.ietf.org/html/draft-richardson-opsawg-securehomegateway-mud-05

[156] (2020). *IoTivity*. [Online]. Available: https://iotivity.org/about

[157] OCF Security Foundation. (2021). *OCF Security Specification—Version 2.2.2*. [Online]. Available: https://openconnectivity.org/specs/OCF_Security_Specification.pdf

[158] *Canadian Multistakeholder Process: Enhancing IoT Security Report: Final Outcomes and Recommendations Report*. [Online]. Available: https://iotsecurity2018.ca/wp-content/uploads/2019/05/Enhancing-IoT-Securit-Report-2019-Final-EN.pdf

[159] J. Chen and L. Urquhart, '''They're all about pushing the products and shiny things rather than fundamental security' mapping socio-technical challenges in securing the smart home,'' 2021, *arXiv:2105.11751*. [Online]. Available: http://arxiv.org/abs/2105.11751

[160] P. Emami-Naeini, Y. Agarwal, L. F. Cranor, and H. Hibshi, ''Ask the experts: What should be on an IoT privacy and security label?'' 2020, *arXiv:2002.04631*. [Online]. Available: http://arxiv.org/abs/2002.04631

JOSÉ L. HERNÁNDEZ-RAMOS received the Ph.D. degree in computer science from the University of Murcia, Spain. He is currently a Scientific Project Officer with the European Commission—Joint Research Centre. He has participated in different European research projects, such as SocIoTal, SMARTIE, and SerIoT, and coauthored more than 60 peer-reviewed publications. His research interests include the application of security and privacy mechanisms in the Internet of Things and transport systems scenarios, including blockchain and machine learning. He has served as a member of a technical program committee and the Chair Member of different international conferences.

SARA N. MATHEU received the B.S. degree in mathematics and the B.S. and M.S. degrees in computer science from the University of Murcia, Spain, in 2015 and 2016, respectively, and the Ph.D. degree, in 2020. She is currently a Postdoctoral Researcher with the University of Murcia. She has participated in several projects, such as ARMOUR and CyberSec4Europe or BIECO. Her main research interests include the security certification for the Internet of Things.

ANGELO FERAUDO received the master's degree in computer engineering from the University of Bologna, Italy, in 2020. He carried out his master's degree thesis focused on privacy preserving methodologies for the IoT devices, such as MUD and federated learning. Furthermore, he participated in different projects focused on improving the IoT privacy, in 2020. Currently, he is working as a Research Fellow with the University of Bologna. His research interests include distributed learning, the industrial IoT, edge computing, and vehicular networks.

GIANMARCO BALDINI (Senior Member, IEEE) received the Laurea degree in electronic engineering from the University of Rome, in 1993, and the Ph.D. degree in computer science from the University of Insubria, in 2019. He worked with the research and development departments in the field of wireless communications in Italy, Ireland, and USA, before joining the European Commission—Joint Research Centre (JRC), in 2007. In the JRC, he has worked in wireless communications, security, positioning, and machine learning. He has contributed to the formulation of European policies in the areas of radio frequency spectrum, road transportation, and cybersecurity.

JORGE BERNAL BERNABE received the B.S., M.S., and Ph.D. degrees in computer science and the M.B.A. degree from the University of Murcia, Spain. He is currently an Assistant Professor with the University of Murcia. He has been a Visiting Researcher with Hewlett Packard Laboratories and the University of the West of Scotland. He has authored several book chapters and more than 60 articles in international top-level conferences and journals. During the last years, he has been working in several European research projects, such as SocIoTal, ARIES, OLYMPUS, ANASTACIA, INSPIRE-5G, and CyberSec4EU.

POONAM YADAV received the M.Tech. degree from IIIT Allahabad, Prayagraj, India, and the Ph.D. degree in computing from Imperial College London, London, U.K. She is currently an Assistant Professor with the Computer Science Department, University of York, U.K. She is an active reviewer of many top tier ACM/IEEE IoT and networking conferences and journals. Her research interests include making the Internet of Things (IoT) and edge computing-based distributed systems resilient, reliable, and robust. She leads ACM-W U.K. Professional Chapter and is featured as "People of ACM Europe" and among the top ten N2Women Rising Star in computer networking and communications, in 2020.

ANTONIO SKARMETA (Member, IEEE) received the Ph.D. degree in computer science from the University of Murcia. He is currently a Full Professor with the Department of Information and Communications Engineering, University of Murcia. He has published more than 200 international papers. His research interests include the integration of security services, identity, the Internet of Things, and smart cities. He has been a member of several program committees.

PAOLO BELLAVISTA (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science engineering from the University of Bologna, Italy. He is currently a Full Professor of distributed and mobile systems with the University of Bologna. His research interests include from pervasive wireless computing to online big data processing under quality constraints and from edge cloud computing to middleware for industry 4.0 applications. He serves on several editorial boards, including IEEE COMMUNICATIONS SURVEYS AND TUTORIALS (Associate EiC), *ACM CSUR*, *JNCA* (Elsevier), and *PMC* (Elsevier). He is the Scientific Coordinator of the H2020 BigData Project IoTwins.