Virtual-Device-Based Policy Enforcement in Multi-Admin Smart Environments

Yunping Fang[1], Chenglong Fu[2], and Xiaojiang Du[1]
[1]Dept. of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA
[2]Dept. of Software and Information Systems, Univ. of North Carolina at Charlotte, Charlotte, NC 28223, USA
Email: {yfang24,xdu16}@stevens.edu, chenglong.fu@uncc.edu

2023 IEEE 12th International Conference on Cloud Networking (CloudNet) | 979-8-3503-1306-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/CloudNet59005.2023.10490072

Abstract—**The Matter standard, formerly known as Connected Home over IP, has emerged as the preferred choice for most smart home IoT vendors and service providers for the next-generation smart home IoT systems. It enhances interoperability across different smart home ecosystems and introduces the multi-admin feature, allowing a device to be commissioned and managed by multiple platforms simultaneously. While this standard offers considerable convenience, it also presents challenges for security policy enforcement. Policy enforcement has been highlighted in various studies as a crucial countermeasure against smart home IoT system vulnerabilities. Existing smart home policy enforcement methods, designed for individual IoT admin platforms, operate under the assumption of having a global view and control over all IoT devices in a smart home. This assumption, however, is untenable in a multi-admin environment, where an admin may only have partial device access or a partial view. In this work, we thoroughly analyze these challenges in a multi-admin setting and propose the first cross-admin policy enforcement solution. Our solution can parse complex policies into deployable sub-policies for each admin, and create virtual device instances and virtual automation rules to interconnect various admins. We test our solution on a real-world testbed involving 12 IoT devices and three popular Matter-enabled IoT platforms. Our results show that our solution can enforce cross-admin policies with a 100% success rate and a very small delay.**

Index Terms—**IoT, Smart Home, Security, Matter, Policy Enforcement**

I. Introduction

The Internet of Things (IoT) has witnessed an increasing emergence of IoT platforms and device vendors, each offering its own set of smart applications and line of devices. However, the compatibility issue between platforms and devices has resulted in a complex landscape where homeowners often find themselves managing devices from different manufacturers on multiple smart home platforms. Multi-admin smart homes as such where multiple device controllers have administrative privileges impose restricted visibility and partial control for each administrator, that is, an administrator can only view and control a specific subset of devices at home. Due to these limitations, collaborative automations spanning across different administrative domains become infeasible.

Worse still, the existence of partial view and partial control also impedes cross-admin policy enforcement. While the convenience offered by smart homes is undeniable, the inherent security vulnerabilities cannot be ignored. First, the interconnected nature of IoT devices in smart homes creates a fertile ground for cyber attacks [7], [16], which can have severe consequences, including the compromise of user privacy [19], disruption of critical services [14], or even the exploitation of personal safety. Second, smart homes, though equipped with sophisticated security systems, are not immune to physical breaches, leading to theft and potential harm to occupants. Third, user negligence, such as leaving appliances unattended or disregarding safety guidelines, can trigger electrical faults or fire hazards or result in security risks. Given these circumstances, security policies are designed to establish guidelines and safeguards and their correct and timely enforcement plays a pivotal role in mitigating the risks associated with smart homes [17].

Though previous studies have made efforts to ensure enforcement of rules [8], [9], [13], [18] and policies [10], [12], [20], they did not consider cross-admin situations where deployment is a fundamental issue, let alone enforcement. In this regard, device access delegation mechanisms have emerged, allowing controlled access to devices to be granted between different cloud platforms. While this approach offers a potential solution to partial view and partial control, it also introduces a myriad of security concerns [21] that must be carefully attended to. In addition to cloud delegation, to save the fragmented IoT ecosystem, a general standard, Matter [5], is incubated to enable interoperability among diverse devices and platforms. It standardizes communication protocols across smart home devices with enhanced security and privacy measures. With its multi-admin support, Matter enables sharing of onboarded devices among various platforms [3]. Partial view and partial control cease to exist in a fully-equipped Matter home, yet most of the smart homes still contain devices that do not support Matter and direct device sharing in policy enforcement also spurs security issues.

In light of this, a pressing need arises for advanced measures that not only resolve the paradigm of partial view and partial control but also ensure the feasibility and security of cross-admin policy enforcement. In this paper, we propose a comprehensive approach to enable cross-admin policy enforcement in multi-admin smart homes, leveraging innovative features of Matter standard to overcome the partial view and partial control limitations. By establishing a secure and private framework,

ACKNOWLEDGMENT This work was supported in part by the US National Science Foundation (NSF) under grants CNS-2204785 and CNS-2205868.

we aim to enhance the overall security posture of smart homes by defending against domestic threats and safeguarding home information privacy. We contribute in the following ways:

- We identify of a multi-admin home structure the innate limitations that are lethal to cross-admin policy enforcement.
- We present a way to integrate non-Matter devices into Matter environments.
- We propose a secure and private approach for cross-admin policy enforcement in a multi-admin Matter smart home.
- We evaluate our approach on a testbed and the result proves it to be effective and efficient.

II. Background

A. Security Policies

Security policies serve as a proactive defense mechanism, enabling smart home systems to monitor and respond to critical events. By defining conditions, these policies can trigger automated actions that provide timely alerts and mitigate risks. For instance, a common security policy may involve sounding an alarm if a door is opened while the homeowner is away. This immediate response serves as a deterrent and helps to protect against unauthorized entry. By automating security policies as such, smart homes become more efficient, capable, and reliable in their ability to protect against threats.

B. Matter Standard

Introduced as a collaborative effort by a working group within the Connectivity Standards Alliance, Matter standardizes the connectivity mechanisms among smart home devices, offering a unified and seamless experience for homeowners [4]. Its key features are of unparalleled import to smart home technologies.

a) Multi-admin Support: Matter offers multi-admin support, which enables the sharing and control of onboarded Matter devices across supported platforms. This means that in a multi-admin smart home, Matter controllers with administrative privileges can conveniently access and manage devices in other fabrics [3]. This property fosters interoperability, allowing devices from different manufacturers to seamlessly communicate and interact with each other.

b) Enhanced Security: Matter prioritizes security and privacy, incorporating robust encryption and authentication mechanisms to safeguard user data and protect against potential vulnerabilities [15].

III. System and Threat Models

Security is always a crucial concern when it comes to smart homes. In this section, we introduce the smart home system considered in this paper and associated possible security threats.

A. System: A Multi-admin Matter Smart Home

A Matter smart home normally refers to a multi-device integrative ecosystem built upon Matter standard, where all smart devices work with Matter to jointly enable automated home operations. However, as many device manufacturers are still working in progress to get their devices to accommodate Matter specification, a smart home that consists of both regular smart devices and Matter devices is often the case with most home users as a transitional stage. Therefore, in this paper, the Matter smart home we discuss includes not only standard Matter homes but also Matter homes that contain a mixture of non-Matter and Matter devices.

B. Threat Model

Like any other intelligent system, smart homes could be exposed to various hazards.

a) Malicious Wireless Attack: Aggressive attackers are capable of compromising the security of a home network by targeting security weaknesses in smart devices.

- The seizure of security-sensitive devices like door locks and security systems can make foreign access to a home simple and unnoticed, aiding in unlawful break-in and domestic theft.
- The hijack of safety-warning devices like gas stoves and clothes dryers (those that consume high energy and produce massive heat) along with water valves and sprinklers (those that are common protective appliances in a home) can help deliberately induce residential fires and pose direct threats to user's well-being.

b) Physical Forced Entry: Less intelligently, threats in smart homes can also occur through traditional means. Intruders can exploit weak entry points such as doors and windows or bypass security measures to obtain illegal access to a smart home. The consequences could entail loss of belongings and even physical harm to home user.

c) User Negligence: Other than vicious agenda of exotic perpetrators, a home can be put in danger by oversights of its own members. It is quite likely, when leaving home, that the user forgot to shut the door, making it easy for breaking and entering, or carelessly kept the gas stove on, rendering the whole household in possible jeopardy of fire.

IV. Problem Statement

The existence of multiple admins introduces serious challenges to cross-admin rule and policy enforcement.

A. Policies

While rules favor the comfort of home user's daily life, policies are a hard guarantee for home security and user safety. For simple illustration, in our study, we treat rules as policies of a loosely-designed type that only realize automation but do not account for security defense, and set a specific format for general policies.

A policy conforms to a Condition-Action template, where a condition is a check on device's current attribute value and/or an event that indicates the change of device states, and an action is simply an event that is to happen when all conditions are met. Let event $\mathcal{E}_{S_A}^{A}$ denote each device a_i in a device set A reaches its matching state S_{a_i} in S_A, and state $\mathcal{S}_{S_A}^{A}$ suggest every $a_i \in A$ is currently at $S_{a_i} \in S_A$ respectively. Thereby a policy could be described as

$$\mathcal{E}_{S_{C_e}}^{C_e} \cap \mathcal{S}_{S_{C_s}}^{C_s} \rightarrow \mathcal{E}_{S_A}^{A} \tag{1}$$

where device group C_e and C_s make up condition device set C and at least one of them is not an empty set; elements in set A are dubbed as the action devices; and $\cap$ is the logic AND. When each device $ce_i \in C_e$ change into state $S_{ce_i} \in S_{C_e}$, if $cs_i \in C_s$ are exactly at state $S_{cs_i} \in S_{C_s}$, we say the conditions of this policy are satisfied and the action shall be performed thereafter turning every device $a_i \in A$ to its state $S_{a_i} \in S_A$.

Further, since we are aiming at cross-admin policy enforcement, devices involved in a policy should not all fall under the same admin(s). Suppose admins in a multi-admin home are signified as M and numbered in order, starting from 1 to N (the number of admins), i.e., there are admin M_1, M_2, $\cdots$, M_N. Devices under an admin are represented by a device set distinguished by the admin number, e.g., D_i for M_i, D_2 for M_2, etc. Accordingly, our policy dictates that

$$\begin{cases} N \geq 2 \\ D_i \subsetneq (C \cup A), \quad \forall\, i \in [1, N] \end{cases} \tag{2}$$

B. Partial View and Partial Control

Partial view and partial control problems are prompted by the multi-admin setting in a smart home, serving as a hard barrier to cross-admin policy enforcement.

a) Partial View: In a multi-admin smart home, each admin only has a partial view of all smart devices. That is, one can only see devices in its own fabric and view their attribute values. During the condition checking stage of policy enforcement, the admin that handles policy enforcement—we call it the policy enforcement admin M_P— needs to view either state changes or current states of condition devices. It not having eyes on any one of them could directly handicap its enforcement power. A formulaic explanation of the partial view paradigm is

$$(D_i \cap C) \subsetneq C, \quad 1 \leq i \leq N \tag{3}$$

b) Partial Control: Similarly, partial control states that an admin does not have the authority to alter attributes of all devices at a multi-admin home. While, in policy enforcement, the action performing phase requires that M_P have the authority to control every action device in order to complete the actions, having a portion of access to action devices grants it only limited capacity to carry out the task. The partial control hurdle of the execution of policy action commands can be expressed into

$$(D_i \cap A) \subsetneq A, \quad 1 \leq i \leq N \tag{4}$$

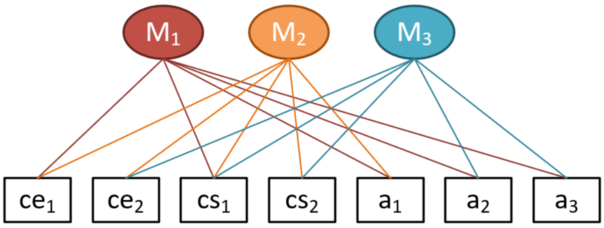


Fig. 1: Illustration of partial view and partial control. The ovals on the upper half represent admins and boxes on the lower half devices. Lines in the middle connecting ovals and boxes show admin-device links. Different colors imply different fabrics.

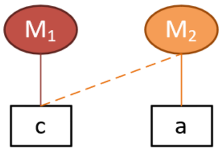


Fig. 2: Basic multi-admin home with the simplest solution. The dotted line indicates the sharing process of device c to admin M_2

c) Hybrid Case: Partial view and partial control are in regard to admins. While an admin can have a shortage of vision on condition devices or a lack of controlling privileges in view of action devices, it can also have them both, that is

$$\begin{cases} (D_i \cap C) \subsetneq C \\ (D_i \cap A) \subsetneq A \end{cases} \quad 1 \leq i \leq N \tag{5}$$

Take Fig. 1 as an example. The policy here is: $\mathcal{E}_{\{S_{ce_1}, S_{ce_2}\}}^{\{ce_1, ce_2\}} \cap \mathcal{S}_{\{S_{cs_1}, S_{cs_2}\}}^{\{cs_1, cs_2\}} \rightarrow \mathcal{E}_{\{S_{a_1}, S_{a_2}, S_{a_3}\}}^{\{a_1, a_2, a_3\}}$. Here,

$$\begin{cases} N = 3, D_1 = \{ce_1, cs_1, a_1, a_2, a_3\} \\ D_2 = \{ce_1, ce_2, cs_1, cs_2, a_1\}, D_3 = \{ce_2, cs_1, cs_2, a_2, a_3\} \\ C = \{ce_1, ce_2, cs_1, cs_2\}, A = \{a_1, a_2, a_3\} \end{cases}$$

Apparently, the policy satisfies Eq. (2), so it is a correct cross-admin policy; D_1 meets Eq. (3), which means admin M_1 has partial view; D_2 complies with Eq. (4), indicating M_2 having partial control issue; and D_3 complies with Eq. (5), which implies a hybrid situation for M_3.

C. Concerns for Device Sharing

One simple solution to partial view and partial control is device sharing. But it comes with potential security threats, privacy concerns, and other inconveniences. Take Fig. 2 for illumination.

a) Unauthorized Access: Complying with Matter's device-to-device communication, once shared to M_2, c could have a view of and even control a over Matter, resulting in security and privacy issues. This could be devastating if c is a Matter controller, such as a speaker [11] or a hub.

b) Invalid Subscription: When c is subscribed by M_1, sharing c to M_2 would possibly render the subscription invalid provided M_2 is malicious or acquired by an attacker. For example, if c is a sensor that sends status reports to M_1 regularly, after pairing with it, M_2 can change configurations of c through write transactions, e.g., extend its report intervals to a point where when a report is received, the status it describes is no longer valid. Then the sensor c is basically a useless device as it reads the wrong data.

c) Inconvenience to Users: Under the circumstance where M_1 and M_2 belong to different users in the same home (user U_1 has M_1 and U_2 owns M_2), if c is shared to M_2 just for the purpose of policy enforcement, over time, there is a possibility that U_2 could forget and at some time mistakenly think that c is his/her own device and set up automation rules using c, which is not what U_1 desires and in the meantime could also disrupt U_2's ideal automation and spell security threats.

V. System Design

The multi-admin feature of Matter standard allows one device to be shared and controlled by multiple admins, which properly fits into our design. Targeting partial view and partial control, we present a secure and private approach to enforce policies in a multi-admin Matter smart home while maintaining the original home structure (which helps avoid device sharing) and preserving home security and user safety in the meantime. In addition, we propose improvement measures to minimize the overhead to achieve better efficiency, and further reinforce home's secure status along with device information privacy.

A. Solution: Device Shadowing

1) Shadow Device: Sometimes, there would be some non-Matter devices in a Matter smart home. We need a way to integrate them into the Matter ecosystem so as to leverage Matter's unique highlights. There are also times where no device supports Matter in a home, calling for a method to create a Matter environment on the basis of a non-Matter home. As such, shadow device is proposed to heed those calls.

For a non-Matter device, we create its virtual Matter instance, referred to as the *shadow device*, and have it commissioned to the same admin as its real entity (named as *entity device*) is connected to. The virtual shadow shares the same device data model and synchronizes all attribute states with its physical counterpart. The entire process is termed *device shadowing*.

Through shadowing non-Matter devices, every non-Matter or half-Matter smart home could operate as a Matter one. Shadow devices talk Matter on behalf of the entities. When entities change states, shadows mimic the changes and transmit non-Matter communications to Matter fabrics. Oppositely, when Matter messages arrive, altering attributes of shadow devices, entities are forced to adjust to their shadows' "shapes" and Matter voices are thus heard by non-Matter networks.

Moreover, Matter devices also have to be shadowed when needed to enforce a policy so that there is no direct device sharing and current fabric status and home structure remain intact.

2) Solutions to Different Cases: On the basis of device shadowing, we present solutions to the aforementioned obstacles thwarting policy enforcement in Section III-B.

a) Shadow Condition Devices (for Partial View): Partial view happens when the policy enforcement admin M_P is able to perform the action yet cannot see some of the condition devices to check if the policy needs to be enforced. To crack the partial view dilemma, any condition device that is blocking the view is shadowed. Their shadows are then shared to M_P, giving it a *global view* of all condition devices mentioned in the policy. In consequence, M_P can check the conditions and send the action commands if needed.

b) Shadow Action Devices (for Partial Control): If M_P can get hold of current states of all condition devices but have trouble gaining control of part of action devices, this is where partial control comes into place. In a similar manner to fixing partial view, unraveling partial control requires the action devices that do not fall into M_P's control domain be shadowed and shared to M_P. As a result, M_P has a *global control* of action devices and can properly enforce the actions when conditions pass checking.

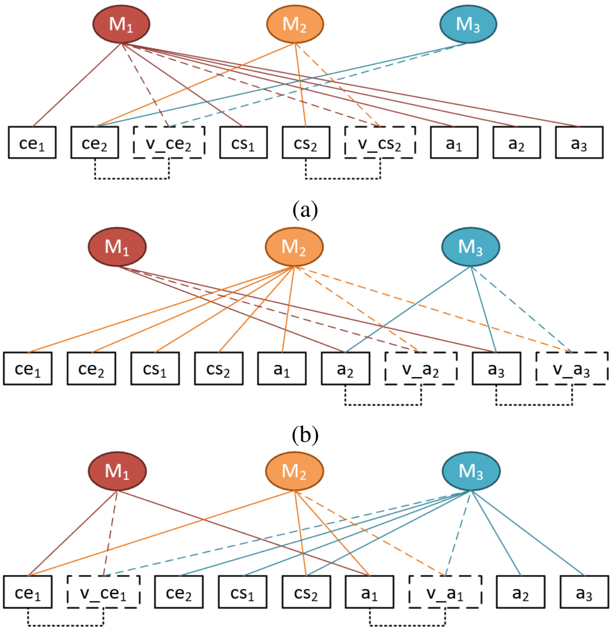
c) For Hybrid Case: When M_P has a partial view and a partial control at the same time, it is easy to conclude, from the previously-discussed solutions, that those condition devices that it does not see and action devices it cannot control should be shadowed and all shadow devices must be shared to M_P to allow a global view and global control of those devices pertaining to the policy.

To elaborate, referring to Fig. 1, for admin M_1 who is facing partial view problem, condition device ce_2 is shadowed in the fabric of either M_2 or M_3, cs_2 is shadowed under M_2, which is the only admin it is connected to, and their shadows v_ce_2 and v_cs_2 are shared to M_1; considering M_2 as the M_P, partial control could be relieved by shadowing action device a_2 in M_1 and a_3 in M_3 and sharing shadow devices' control privilege to M_2; as to M_3, condition device ce_1 and action device a_1 are shadowed and the shadows are shared to M_3 in order to solve the hybrid case. Example home structures after problem solving are shown in Fig. 3.

The solution of device shadowing does not alter the original home structure since shadows are shared instead of the entities, so it addresses issues introduced by direct device sharing in that respect. And for Matter standard is designed with baked-in security and privacy features, the solution is also geared with intrinsic security and privacy defense.

B. Overhead Minimization

Based on device shadowing, the preparation of policy enforcement comprises three main steps: shadow device building, commissioning, and sharing, which make up the overhead of our solution method. Shadow device commissioning is fixed once a shadow is built in that a device building process must be followed by a commissioning operation. And shadow device sharing is determined after M_P is set. Yet the first step,



(a)

(b)

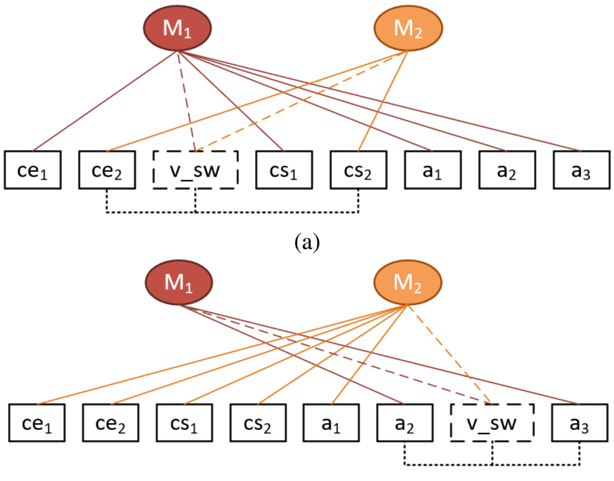
(c)

Fig. 3: Solution of partial view and partial control. Devices whose name starts with $v\_$ and box's contour is dotted are shadow devices. Dotted lines connecting devices indicate the entity-shadow relations and those between admins and devices show admin-shadow links. Connections unrelated to the solution are omitted. (a), (b) and (c) present the solution when M_1, M_2 and M_3 is M_P, respectively.

shadow device building, varies with different cases and highly depends on policies and home structures. Even for the same policy in the same home, entirely disparate solutions could be deduced, differing in the number of shadow devices. To achieve better efficiency, we put forth two approaches calculated on minimizing the workload brought by this procedure.

1) Shadow Switches: Shadow switches are proposed to reduce the number of shadow devices so as to cut down the overhead in shadow device building stage. Some of the devices that need to be shadowed might fall under the same admin, which leads to a circumstance that we call *multi-shadow*. To avoid appearance of multi-shadow, when an admin shall connect to excessive shadows, we replace all shadows with a shadow switch. The switch is an overlapped shadow of entity devices that are in the fabric of an identical admin, to which the switch is commissioned. After being commissioned, it is shared to M_P like a normal shadow device.

The entity devices of a shadow switch should be in the same device category, i.e., they are all either condition devices or action devices. When shadowing multiple condition devices, the switch serves as a signal of satisfaction status of conditions in correspondence to those devices. Only if all conditions are met will the switch turn on. If there would have been multi-shadowing of multiple action devices, a switch presents as an acting commander. When the switch is turned on, commands are sent to set action devices to the desired states as policy demands. And they are changed to the opposite states when switch turns off.



(a)

(b)

Fig. 4: Example solutions using shadow switches (represented by v_sw). (a) A switch shadowing two condition devices. (b) A switch shadowing two action devices.

Examples are exhibited in Fig. 4. In Fig. 4a, where M_1 is considered the M_P, two condition devices ce_2 and cs_2 that are both connected to M_2 need to be shadowed. Therefore, a shadow switch v_sw is constructed in M_2's fabric, shadowing both of them and then shared to M_1. It is set to *ON* only when ce_2 changes to state S_{ce_2} and cs_2 is at S_{cs_2}. In view of Fig. 4b, M_2 is M_P and a v_sw is shadowing action devices a_2 and a_3 since they are both controlled by admin M_1. When the switch changes to *ON*, a_2 is set to S_{a_2} and a_3 S_{a_3}. In turn, they are set to states as opposed to S_{a_2} and S_{a_3} correspondingly if v_sw switches to *OFF*. Each improved solution, compared to the initial one in Fig. 3a and Fig. 3b, is one shadow device short. In cases where redundant shadows are numerous, using shadow switches instead can significantly decrease the number of shadows and lower the overhead.

2) Multi-M_P: As a home contains multiple admins, the choice of the policy enforcement admin M_P is also of pivotal import to overhead efficiency. Aside from elaborate selection of an M_P that has a minimal number of out-of-view condition devices and out-of-control action devices, there could be multiple M_Ps working in collaboration to enforce one policy.

While in the single M_P case, the M_P needs to get rid of both partial view and partial control, in the multi-M_P solution, M_Ps are allowed to have partial control. However, it requires that the union of all action device sets of M_Ps contains every action device in the policy. For instance, as in Fig. 1, since $A \subset (D_2 \cup D_3)$, M_2 and M_3 can act as the cooperative M_Ps if ce_1 is shadowed and the shadow is shared to M_3 to lift M_3 out of partial view. For policy enforcement, M_2 is responsible for setting a_1 to S_{a_1} and M_3 for changing a_2 and a_3 to their destination states when conditions are checked and confirmed

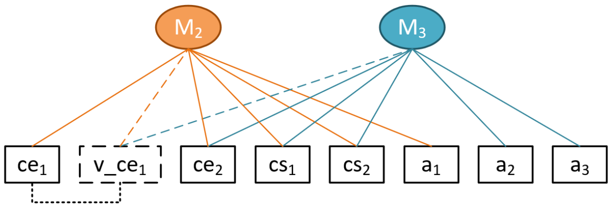


Fig. 5: Example solution using multiple M_Ps.

satisfied in their own fabric separately. The multi-M_P solution is presented in Fig. 5. In contrast to the solutions in Fig. 3b and Fig. 3c where M_2 and M_3 is the sole M_P, one less shadow device is constructed when there are multiple M_Ps in action, diminishing the overhead to some degree.

C. Special Cases

There are special occasions concerning security and privacy that request special attention.

1) Privacy: When there are several users in a home system and they refuse to credit admins held by others, during policy enforcement, they may be reluctant to share even shadow devices to other admins on the grounds that shadow devices would reveal current states of their real entities such that device information privacy is corrupted.

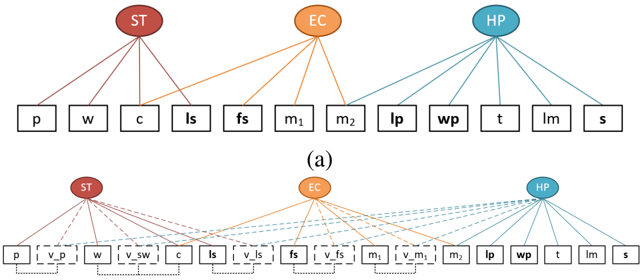
We suggest, to enhance user privacy in this context, that a virtual Matter switch be built when a device needs to be shadowed instead of a virtual device of its same type. Here, the switch does not play the same role as shadow switches in Section V-B1 to solve multi-shadow but operates in the same logic, i.e., it only changes to *ON* when the condition is satisfied if its entity is a condition device or when an action is to be performed on its action device entity. A shadow switch as such does not disclose actual device types or their states, so it addresses the privacy concern when it is to be shared to other admins.

2) Security: In a smart home, devices are typically of different security levels. High-security-level devices, e.g., security-sensitive devices like door lock, garage door controller and security system, are preferred to not be shared to other admins that are not trusted by the device owner. So are their shadows. If they are cognate shadow devices, admins they are shared to can easily view states of real entity devices and control them. If they are shadow switches, by pooling all policies and analyzing the traffic during policy enforcement, an untrustworthy admin can conclude which devices they shadow and take control over them.

Yet the controller privilege of high-security-level devices is critical to home security and cannot be handed over presumptuously. We regulate the solution when policy contains devices of different security levels with a rule that higher-security-level devices shall not be shadowed by any chance and admins that control them shall be the M_Ps to enforce the actions.

VI. Evaluation

A case study is performed to validate the effectivity and test the efficiency of our approach. The one-week experiment is performed in a multi-admin smart home with simulated security eventualities and natural environment that triggers automation. Devices in the testbed are nearly half Matter-enabled and half non-Matter. Our devised approach is employed along with improvement measures to get the optimized solution for every policy. Device event logs on different platform assist in the evaluation, upon which we draw our final conclusion.



(a)

(b)

Fig. 6: Original home structure and that intended for policy enforcement. Devices in bold indicate actuators while others are sensors. (a) The smart home structure in our experiment. (b) Final solution derived for the experiment.

A. Experimental Setup

The experiment is set up in a multi-admin home that contains both non-Matter devices and Matter devices. Most of the policies are designed to secure the premises and a minority to realize automation for convenience in everyday life. The shadow devices we propose in our approach are also built beforehand and kept running in the background.

1) Home Structure: Our experimental testbed contains three popular admins (from three different vendors) with both Matter controller and Thread Border Router capabilities, which determine that the admins can create a Thread environment at home and onboard a Matter device or pair with one that is shared by other admins. Our selection of vendor platforms is based on whether use can specify a personal automation rule on the vendor app and preferred device vendors are those that store device history, which aids in the evaluation of our approach. Chosen admins are SmartThings Hub v3 (abbreviated to *ST*), Echo 4th generation (*EC*) and HomePod mini (*HP*), each operating on SmartThings [6], Alexa [1] and HomeKit [2] platforms respectively.

The multi-admin Matter smart home structure in this experiment is presented in Fig. 6a. Device information and deployment are shown in Table. I where devices whose name is italic are Matter devices. Testbed devices are displayed in Fig. 7a and their deployments are depicted in Fig. 7b.

2) Policies: A set of cross-admin policies are added to the home as described in Table. II. There are rule-like policies (those start with *R*) that realize home automation, and security-guarding policies (those start with *P*) as part of defensive mechanisms at home. We design the latter with respect to the three sorts of threats proposed in our threat model. During the seven

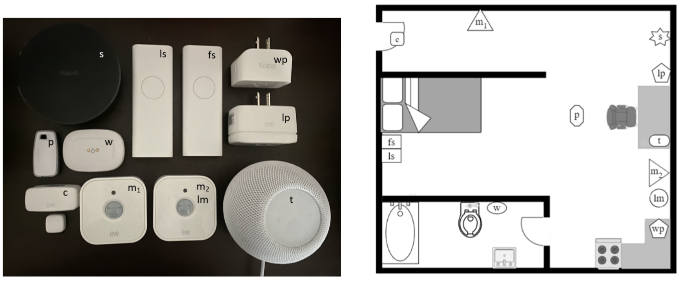
TABLE I: Device information and deployment in our testbed.

Devices				
Label	**Name**	**Type**	**Connected Admins**	**Deployment**
p	SmartThings Arrival Sensor	presence sensor	SmartThings Hub	detect user's presence
w	SmartThings Water Leak Sensor	water sensor	SmartThings Hub	detect leak of the toilet
ls	ThirdReality Smart Switch	light switch	SmartThings Hub	control top light
c	*Eve Door & Window*	contact sensor	SmartThings Hub, Echo	detect contact of the front door
fs	ThirdReality Smart Switch	fan switch	Echo	control ceiling fan
m_1	*Eve Motion*	motion sensor	Echo	detect motion in the foyer
m_2	*Eve Motion*	motion sensor	Echo, HomePod mini	detect room motion
lp	*Eve Energy*	light plug	HomePod mini	control desk lamp
wp	*Tapo Mini Smart Wi-Fi Plug*	waffle maker plug	HomePod mini	control waffle maker
t	HomePod Mini	temperature sensor	HomePod mini	detect room temperature
lm	*Eve Motion*	illuminance sensor	HomePod mini	detect room light level
s	*Aqara Hub M2*	security system	HomePod mini	sound an alarm and push alarm notifications

TABLE II: Description of policies together with formula, M_P, enforcement result and delay of transformed policies in the experiment.

Policies		Transformed Policies			Enforcement		Delay
Label	**Description**	**Label**	**Formula**	$\mathbf{M_P}$	**Correct**	**Wrong**	**(s)**
P1	While user is away, any occurrence of water leak, contact open and motion at home will trigger the alarm.	P1.1	$\mathcal{E}^{w}_{wet} \cap \mathcal{S}^{p}_{away} \rightarrow \mathcal{E}^{s}_{alarm}$	HP	80	0	0.616
		P1.2	$\mathcal{E}^{c}_{open} \cap \mathcal{S}^{p}_{away} \rightarrow \mathcal{E}^{s}_{alarm}$	HP	87	0	1.054
		P1.3	$\mathcal{E}^{m_1}_{detected} \cap \mathcal{S}^{p}_{away} \rightarrow \mathcal{E}^{s}_{alarm}$	HP	64	0	1.234
		P1.4	$\mathcal{E}^{m_2}_{detected} \cap \mathcal{S}^{p}_{away} \rightarrow \mathcal{E}^{s}_{alarm}$	HP	47	0	0.813
P2	When user is at home and asleep, if there's a water leakage or the door is open, sound the alarm.	P2.1	$\mathcal{E}^{w}_{wet} \cap \mathcal{S}^{\{p,m_2\}}_{\{away,\, not\, detected\}} \rightarrow \mathcal{E}^{s}_{alarm}$	HP	37	0	0.794
		P2.2	$\mathcal{E}^{c}_{open} \cap \mathcal{S}^{\{p,m_2\}}_{\{away,\, not\, detected\}} \rightarrow \mathcal{E}^{s}_{alarm}$	HP	39	0	0.516
P3	If user left home with door open, leaking water or waffle maker on, send the alarm notification.	P3.1	$\mathcal{E}^{p}_{away} \cap \mathcal{S}^{c}_{open} \rightarrow \mathcal{E}^{s}_{alarm}$	HP	44	0	1.266
		P3.2	$\mathcal{E}^{p}_{away} \cap \mathcal{S}^{w}_{wet} \rightarrow \mathcal{E}^{s}_{alarm}$	HP	44	0	1.206
		P3.3	$\mathcal{E}^{p}_{away} \cap \mathcal{S}^{wp}_{on} \rightarrow \mathcal{E}^{s}_{alarm}$	HP	41	0	1.133
P4	When user is at home, if temperature rises above 77° F, turn on the fan; if it drops below 73° F, turn off the fan.	P4.1	$\mathcal{E}^{t}_{>77^\circ F} \cap \mathcal{S}^{p}_{present} \rightarrow \mathcal{E}^{fs}_{on}$	HP	39	0	0.933
		P4.2	$\mathcal{E}^{t}_{<73^\circ F} \cap \mathcal{S}^{p}_{present} \rightarrow \mathcal{E}^{fs}_{off}$	HP	39	0	1.183
P5	If user left home or user is at home but falls asleep, turn off the lights.	P5.1*	$\mathcal{E}^{p}_{away} \rightarrow \mathcal{E}^{ls}_{off}$	ST	50	0	1.023
		P5.2	$\mathcal{E}^{p}_{away} \rightarrow \mathcal{E}^{lp}_{off}$	HP	50	0	0.650
		P5.3	$\mathcal{E}^{m_2}_{not\, detected} \cap \mathcal{S}^{p}_{present} \rightarrow \mathcal{E}^{ls}_{off}$	HP	64	0	0.300
		P5.4	$\mathcal{E}^{m_2}_{not\, detected} \cap \mathcal{S}^{p}_{present} \rightarrow \mathcal{E}^{lp}_{off}$	HP	64	0	0.100
P6	When user is at home and not asleep, if light level at home drops below 25 lux, turn on the lights.	P6.1	$\mathcal{E}^{lm}_{<25lux} \cap \mathcal{S}^{\{p,m_2\}}_{\{present,\, detected\}} \rightarrow \mathcal{E}^{ls}_{on}$	HP	64	0	1.466
		P6.2	$\mathcal{E}^{lm}_{<25lux} \cap \mathcal{S}^{\{p,m_2\}}_{\{present,\, detected\}} \rightarrow \mathcal{E}^{lp}_{on}$	HP	64	0	1.316

* indicate the policy is not cross-admin.



(a) (b)

Fig. 7: Test bed devices and floorplan. (a) All devices in use. (b) Floorplan of the testbed with device placement.

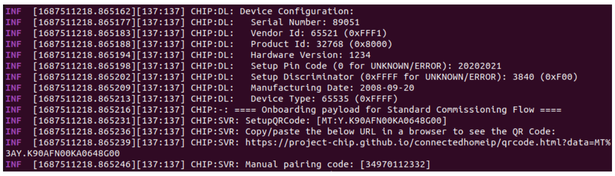
days of experiment, rule-like policies are triggered (namely, the conditions are met) as natural environment changes. While for policies designed for security preservation, security threats that can trigger the policies are simulated several times on a daily basis.

3) Shadow Devices: In our experiment, virtual Matter shadow devices run on Ubuntu 22.04.2 LTS in Oracle VM VirtualBox on a laptop with 64-bit operating system, AMD Ryzen 5 5625U with Radeon Graphics 2.30 GHz processor, and 8.00 GB RAM. The basic information is displayed in terminal as in Fig. 8 when a shadow device is set up, which is used for Matter commissioners (i.e., vendor apps) to onboard the device.

B. Solution

Here, we explain how to get a valid and optimized solution regarding a specific smart home with multiple policies to enforce.

a) Policy Formatting: Since user-specified policies usually come in natural languages [17] as in Table. II, we parse and translate them into the policy format we defined in Eq. (1). To



```
INF  [1687511218.865162][137:137] CHIP:DL: Device Configuration:
INF  [1687511218.865177][137:137] CHIP:DL:   Serial Number: 89051
INF  [1687511218.865183][137:137] CHIP:DL:   Vendor Id: 65521 (0xFFF1)
INF  [1687511218.865188][137:137] CHIP:DL:   Product Id: 32768 (0x8000)
INF  [1687511218.865194][137:137] CHIP:DL:   Hardware Version: 1234
INF  [1687511218.865198][137:137] CHIP:DL:   Setup Pin Code (0 for UNKNOWN/ERROR): 20202021
INF  [1687511218.865202][137:137] CHIP:DL:   Setup Discriminator (0xFFFF for UNKNOWN/ERROR): 3840 (0xF00)
INF  [1687511218.865209][137:137] CHIP:DL:   Manufacturing Date: 2008-09-20
INF  [1687511218.865213][137:137] CHIP:DL:   Device Type: 65535 (0xFFFF)
INF  [1687511218.865216][137:137] CHIP:-: ==== Onboarding payload for Standard Commissioning Flow ====
INF  [1687511218.865231][137:137] CHIP:SVR: SetupQRCode: [MT:Y.K90AFN00KA0648G00]
INF  [1687511218.865236][137:137] CHIP:SVR: Copy/paste the below URL in a browser to see the QR Code:
INF  [1687511218.865239][137:137] CHIP:SVR: https://project-chip.github.io/connectedhomeip/qrcode.html?data=MT%
3AY.K90AFN00KA0648G00
INF  [1687511218.865246][137:137] CHIP:SVR: Manual pairing code: [34970112332]
```

Fig. 8: Virtual Matter shadow device's basic information.

simplify the results and facilitate solution generation, we add that the sets in Eq. (1) can be replaced with their element if their size is 1 and that if action device set A contains more than one element, the equation is decomposed into several formulae (in the number of A's size) that are identical on the left of the arrow $\rightarrow$, while on the right each has only one of the action device in a non-repeat manner, for instance, if $A = \{a_1, a_2\}$, then Eq. (1) is decomposed into

$$\begin{cases} \mathcal{E}^{C_e}_{S_{C_e}} \cap \mathcal{S}^{C_s}_{S_{C_s}} \rightarrow \mathcal{E}^{a_1}_{S_{a_1}} \\ \mathcal{E}^{C_e}_{S_{C_e}} \cap \mathcal{S}^{C_s}_{S_{C_s}} \rightarrow \mathcal{E}^{a_2}_{S_{a_2}} \end{cases}$$

From one's formula, it can clearly be seen whether a simplified policy is cross-admin merely by checking if there is a condition device that does not belong to any of the fabric the action device is in. The resulting policies we get are shown in Table. II.

b) Attend to Security Concerns: It is worth noting that P1.1 through P3.3 all involve security system as the action device, which is a highly security-sensitive device. Referring to Section V-C2, it cannot be shadowed and the admin (i.e., *HP*) that has it should be the policy enforcement admin M_P. Therefore, condition devices (i.e., p, w, c and m_1) that do not belong to *HP* in P1.1 to P3.3 should be shadowed and their shadows are shared to *HP*,

c) Shadow Minority Devices: Bearing in mind that those shadowed devices are now under *HP*, we turn to look at other policies. It is obvious that P5.2, P5.4, and P6.2 no longer cross domain. For the rest cross-admin policies, we develop their solutions separately. To reach a minimal number of shadow devices, we regulate that in a policy, if devices that connect to a specific admin are a majority, the admin should be M_P and other devices should be indirectly shared to it through shadowing. Take P4.1 as an example. It includes three devices, t, p and fs, two (t and p) of which are under admin *HP*, thus *HP* is the M_P and fs's shadow should be shared to it. After going through every policy that is left, ls and fs are shared to *HP*.

d) Replace with Shadow Switches: At last, we attend to further overhead minimization by deploying shadow switches. In a multi-policy scenario, a shadow switch can only substitute shadow devices when those shadows are linked to the same admin and they relate to the same state throughout every policy. Therefore, despite of the fact that shadow devices of p, w, c, ls, fs and m_1 are all under *HP*, only w and c can be shadowed altogether by a switch. The switch turns *ON* at $\mathcal{E}^{w}_{wet} \cap \mathcal{E}^{c}_{open}$ and *OFF* at $\mathcal{E}^{w}_{dry} \cup \mathcal{E}^{c}_{closed}$.

In light of above discussions, the final solution is deduced and demonstrated in Fig. 6b. Policies are enforced under their specific M_Ps. The policy-M_P relations are shown in Table. II.

C. Results

Implementing the optimized solution, we collect data from the testbed with a continuous running of policies for a span of seven days. Analysis of the results reveals solid effectivity and moderate efficiency of our proposed approach.

1) Effectivity: Effectivity is evaluated with device history data. SmartThings and Alexa have a device event log for every device, while HomeKit does not. Luckily, Eve and Aqara keep track of their device states. Therefore, to get device history for devices connected to HomeKit, we build a virtual Matter shadow switch for each device that is neither Eve nor Aqara—waffle maker plug wp and temperature sensor t—and have it shared to SmartThings. For wp, the switch is completely synchronized with it, while as to t, the shadow switch is turned *On* when its relevant condition in P4.1, $E^t_{>77^\circ F}$, is satisfied and to *OFF* if $E^t_{<73^\circ F}$ in P4.2. In this way, SmartThings indirectly records its past states, to the extent that policy effectivity can be logically examined.

After getting experiment data, we check the timelines to see if there are times when a policy is triggered but not enforced later on (i.e., no enforcement) and when a policy is enforced but was not triggered (i.e., false enforcement). We categorize both of the circumstances to be *wrong enforcement* and others as *correct enforcement*. The enforcement result is displayed in Table. II. It can be seen that, with our approach, the percentage of correct enforcement is as high as 100%, which proves its reliable effectivity of threat detection and home automation.

2) Efficiency: We assess the efficiency of our method by latency of policy enforcement. Device history could not be of use inasmuch as these platforms log events with low granularity. Time records are saved in seconds with no more precision. Yet enforcement gap of various policies, there is usually no difference in seconds, but in smaller units. As an expedient, the time gap between policy triggering and policy enforcement is measured manually, producing an estimated time of delay with a $0.1s$ margin of error.

The latency figures are shown in Table. II. With different policies, the delay ranges from $0s$ to $1.5s$. Note that the policy P5.1 which does not cross domain produces a delay of $1.023s$, which surpasses that of half of the cross-admin policies. It evidences that the latency of our approach is within fine limits and that it can successfully secure the household as well as swiftly achieve automation, which confirms the ideal efficiency of our solution.

VII. Conclusion

The application of the newly-emerged Matter standard brings the difficulty of enforcing security policies in a multi-admin coexisting smart environment. In this work, we systematically analyzed and formalized these new challenges. We designed a novel cross-admin policy enforcement solution by using virtual devices and virtual automation rules, which resolves the partial view and partial control problem. We conducted an experiment of one week on a real smart home testbed that contains a dozen of Matter and non-Matter devices and is protected by several policies and automated by rules. The results showed that every triggered policy was correctly enforced and the latency is within $1.5s$, which demonstrated the feasibility and efficiency of our approach.

REFERENCES

[1] “Amazon alexa, 2015,” https://alexa.amazon.com.

[2] “Apple homekit, 2020,” https://www.apple.com/shop/accessories/all/homekit.

[3] “Matter primer, google developer center,” https://developers.home.google.com/matter/primer/fabric.

[4] “Matter specification, connectivity standards alliance,” https://csa-iot.org/wp-content/uploads/2022/11/22-27349-001_Matter-1.0-Core-Specification.pdf.

[5] “project-chip/connected home over ip: Matter,” https://github.com/project-chip/connectedhomeip, (Accessed on 06/10/2023).

[6] “Smartthings, 2013,” https://www.smartthings.com.

[7] O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose, “Sok: Security evaluation of home-based iot deployments,” in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 1362–1380.

[8] S. Birnbach, S. Eberz, and I. Martinovic, “Peeves: Physical event verification in smart homes,” in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1455–1467. [Online]. Available: https://doi.org/10.1145/3319535.3354254

[9] Z. B. Celik, P. McDaniel, and G. Tan, “Soteria: Automated IoT safety and security analysis,” in *2018 USENIX Annual Technical Conference (USENIX ATC 18)*. Boston, MA: USENIX Association, Jul. 2018, pp. 147–158. [Online]. Available: https://www.usenix.org/conference/atc18/presentation/celik

[10] Z. B. Celik, G. Tan, and P. Mcdaniel, “Iotguard: Dynamic enforcement of security and safety policy in commodity iot,” *Proceedings 2019 Network and Distributed System Security Symposium*, 2019.

[11] Y. Chen, X. Yuan, J. Zhang, Y. Zhao, S. Zhang, K. Chen, and X. Wang, “Devil’s whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices,” in *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Aug. 2020, pp. 2667–2684. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/chen-yuxuan

[12] W. Ding, H. Hu, and L. Cheng, “Iotsafe: Enforcing safety and security policy with real iot physical interaction discovery,” *Proceedings 2021 Network and Distributed System Security Symposium*, 2021. [Online]. Available: https://www.ndss-symposium.org/ndss-paper/iotsafe-enforcing-safety-and-security-policy-with-real-iot-physical-interaction-discovery/

[13] C. Fu, Q. Zeng, and X. Du, “HAWatcher: Semantics-Aware anomaly detection for appified smart homes,” in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 4223–4240. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/fu-chenglong

[14] N. I. Haque, M. Ngouen, Y. Al-Wahadneh, and M. A. Rahman, “Poster: A novel formal threat analyzer for activity monitoring-based smart home heating, ventilation, and cooling control system,” in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’22. New York, NY, USA: Association for Computing Machinery, 2022, p. 3359–3361. [Online]. Available: https://doi.org/10.1145/3548606.3563547

[15] G. Jiacheng, “Matter security model, the esp journal,” https://blog.espressif.com/matter-security-model-37f806d3b0b2.

[16] D. Kumar, K. Shen, B. Case, D. Garg, G. Alperovich, D. Kuznetsov, R. Gupta, and Z. Durumeric, “All things considered: An analysis of IoT devices on home networks,” in *28th USENIX Security Symposium (USENIX Security 19)*. Santa Clara, CA: USENIX Association, Aug. 2019, pp. 1169–1185. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/kumar-deepak

[17] S. Manandhar, K. Moran, K. Kafle, R. Tang, D. Poshyvanyk, and A. Nadkarni, “Towards a natural perspective of smart homes for practical security and safety analyses,” in *2020 IEEE Symposium on Security and Privacy (SP)*, 2020, pp. 482–499.

[18] D. T. Nguyen, C. Song, Z. Qian, S. V. Krishnamurthy, E. J. M. Colbert, and P. D. McDaniel, “Iotsan: Fortifying the safety of iot systems,” *CoRR*, vol. abs/1810.09551, 2018. [Online]. Available: http://arxiv.org/abs/1810.09551

[19] Z. Wang, Y. Yan, Y. Yan, H. Chen, and Z. Yang, “CamShield: Securing smart cameras through physical replication and isolation,” in *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022, pp. 3467–3484. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/wang-zhiwei

[20] M. Yahyazadeh, P. Podder, E. Hoque, and O. Chowdhury, “Expat: Expectation-based policy analysis and enforcement for appified smart-home platforms,” in *Proceedings of the 24th ACM Symposium on Access Control Models and Technologies*, ser. SACMAT ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 61–72. [Online]. Available: https://doi.org/10.1145/3322431.3325107

[21] B. Yuan, Y. Jia, L. Xing, D. Zhao, X. Wang, and Y. Zhang, “Shattered chain of trust: Understanding security risks in Cross-Cloud IoT access delegation,” in *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Aug. 2020, pp. 1183–1200. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/yuan