Dissecting Operational Cellular IoT Service Security: Attacks and Defenses

Sihan Wang, Tian Xie, Min-Yue Chen, Guan-Hua Tu, *Member, IEEE*, Chi-Yu Li, *Senior Member, IEEE*, Xinyu Lei, *Member, IEEE*, Po-Yi Chou, Fucheng Hsieh, Yiwen Hu, Li Xiao, and Chunyi Peng, *Senior Member, IEEE*

***Abstract*— More than 150 cellular networks worldwide have rolled out LTE-M (LTE-Machine Type Communication) and/or NB-IoT (Narrow Band Internet of Things) technologies to support massive IoT services such as smart metering and environmental monitoring. Such cellular IoT services share the existing cellular network architecture with non-IoT (e.g., smartphone) ones. When they are newly integrated into the cellular network, new security vulnerabilities may happen from imprudent integration. In this work, we explore the security vulnerabilities of the cellular IoT from both system-integrated and service-integrated aspects. We discover several vulnerabilities spanning cellular standard design defects, network operation slips, and IoT device implementation flaws. Threateningly, they allow an adversary to remotely identify IP addresses and phone numbers assigned to cellular IoT devices, interrupt their power saving services, and launch various attacks, including data/text spamming, battery draining, device hibernation against them. We validate these vulnerabilities over five major cellular IoT carriers in the U.S. and Taiwan using their certified cellular IoT devices. The attack evaluation result shows that the adversary can raise an IoT data bill by up to $226 with less than 120 MB spam traffic, increase an IoT text bill at a rate of $5 per second, and prevent an IoT device from entering/leaving power saving mode; moreover, cellular IoT devices may suffer from denial of IoT services. We finally propose, prototype, and evaluate recommended solutions.**

***Index Terms*— Cellular IoT, security, service charging, power saving mode.**

I. Introduction

THE market of cellular IoT is projected to reach 7.31 billion in 2025, growing at a CAGR of 23.34% since 2015 [1]. To support massive IoT devices that focus on low cost, low energy, and small data volumes, two cellular network technologies have been proposed: LTE-M (LTE-Machine Type Communication) [2] and NB-IoT (Narrow Band IoT) [3]. They can extend the battery life of cellular IoT devices up to 10 years by reducing modem complexity [4]; they support not only data service, which is the only service offered by other IoT technologies, but also voice and text services.

In practice, most massive cellular IoT users demand only small data volumes and long battery life, so carriers provide them with service plans that have small data volumes with low prices but higher data unit prices. For example, the cheapest monthly data service plan from AT&T for a non-IoT (e.g., smartphone) user is $30 for 5 GB data ($0.0059 per MB), whereas that for an IoT user is $0.99 for 0.5 MB ($1.98 per MB). Moreover, it is more expensive for the IoT user to receive text than the non-IoT one. For instance, the IoT user at Verizon needs to pay $0.05 for sending or receiving a text message, but the non-IoT one with a data service plan does not need to pay for the text service. Moreover, carriers also deploy the IoT-specific power saving mode (PSM) [5] to sustain the IoT battery life for years.

We are thus motivated to study whether those new IoT-specific charging policies, together with new cellular IoT power saving features, may create new security issues. Although there have been many security studies of the cellular network charging [6], [7], [8], [9], [10], [11], [12], the charging security issues of the massive cellular IoT have not been explored yet. Any security loopholes of the cellular IoT charging can impact on a huge amount of current and upcoming cellular IoT devices/users.

At first glance, cellular IoT users are more vulnerable to conventional charging attacks (e.g., overbilling attacks [8]) than non-IoT users since they have small data volumes with much higher data unit prices in cellular IoT service plans. However, launching data spamming attacks against cellular IoT devices is challenging, since adversaries need to remotely identify the IP addresses used by them. It is far from trivial due to two reasons. First, the carrier network may not adopt different IP assignment mechanisms for cellular IoT and non-IoT devices, so no difference can be observed from their IP addresses. Second, an IP address may be used by not only cellular IoT and non-IoT devices but also other kinds of IoT devices, e.g., Wi-Fi IoT devices which connect to Wi-Fi-to-Cellular gateways, so profiling IoT traffic may not be able to clearly differentiate cellular IoT devices from the other IoT ones. Furthermore, cellular IoT devices may suffer the text spamming that can cause overbilling since they are charged for receiving text messages. The prerequisite of the text spamming

Manuscript received 27 November 2022; revised 21 May 2023; accepted 30 August 2023; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor D. Pei. This work was supported in part by the National Science Foundation under Grant CNS-1815636, Grant CNS-1814551, Grant CNS-2112471, Grant CNS-2246050, Grant CNS-2246051, and Grant CNS-2153393; and in part by the National Science and Technology Council (NSTC) in Taiwan under Grant 112-2628-E-A49-016-MY3, Grant 110-2221-E-A49-031-MY3, Grant 112-2218-E-A49-021, Grant 112-2218-E-A49-023, and Grant 112-2634-F-A49-001-MBK. *(Corresponding author: Guan-Hua Tu.)*

Sihan Wang, Min-Yue Chen, Guan-Hua Tu, Yiwen Hu, and Li Xiao are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: wangsih3@msu.edu; chenmi33@msu.edu; ghtu@msu.edu; huyiwen3@msu.edu; lxiao@msu.edu).

Tian Xie is with the Department of Computer Science, Utah State University, Logan, UT 84322 USA (e-mail: tian.xie@usu.edu).

Chi-Yu Li, Po-Yi Chou, and Fucheng Hsieh are with the Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu 300093, Taiwan (e-mail: chiyuli@cs.nctu.edu.tw; bob2006tw@gmail.com; fredhs@cs.nctu.edu.tw).

Xinyu Lei is with the Department of Computer Science, Michigan Technological University, Houghton, MI 49931 USA (e-mail: xinyulei@mtu.edu).

Chunyi Peng is with the Department of Computer Science, Purdue University, West Lafayette, IN 47907 USA (e-mail: chunyi@purdue.edu).

Digital Object Identifier 10.1109/TNET.2023.3313557

TABLE I
SUMMARIZING THE DEVISED ATTACKS AND IDENTIFIED SECURITY VULNERABILITIES OF OPERATIONAL CELLULAR IOT SERVICES

Category	Attack			Vulnerability			
	Name	Description and Threat	Involved Protocol	Vulnerability Description	Type	Party Involved	Applicable IoT
Data service	Data spamming	Adversaries can identify cellular IoT devices remotely and launch an IoT-device-unaware data spamming attack, thereby increasing victims' monthly bills by $0.3-$3 per MB.	TCP	V1: Cellular IoT IP addresses can be identified remotely (§4.1).	Implementation flaw	UE	LTE-M NB-IoT
				V2: Cellular IoT PSM-unaware charging (§4.2).	Design defect	Infrastructure	LTE-M NB-IoT
Text service	Text spamming	Adversaries can identify the phone numbers assigned to cellular IoT devices and transmit unwanted text spam to them, thereby increasing their monthly bills by $5 per second.	SIP	V3: Leakage of phone-number device type from VoLTE signaling (§5.1).	Design defect	Infrastructure	LTE-M
				V4: Leakage of phone-number status from SMS (§5.2).	Operational issue	Infrastructure	LTE-M
				V5: Insecure pushed text service (§5.3).	Operational issue	Infrastructure	LTE-M
PSM service	Battery draining	Adversaries can prevent cellular IoT devices from entering the sleep mode, thereby draining their batteries sooner by up to 20 times.	EMM*	V6: No secure confirmation of PSM configuration (§6.1). V7: No anomaly detection on PSM configuration (§6.2).	Design defect	Infrastructure (V6) & UE (V6 and V7)	LTE-M NB-IoT
	Device hibernation	Adversaries can guide the cellular infrastructure to assign an extremely long sleep time to PSM-enabled cellular IoT devices; once they enter the sleep mode, they will not respond to any requests for a long time.					

*: The EMM (EPS Mobility Management) message tampered by adversaries is EMM Attach Request that is transmitted in plain-text without NAS integrity and confidentiality protection [5].

attack is to identify the phone numbers assigned to cellular IoT users, but it is even more challenging than identifying their IP addresses.

Unfortunately, we find that the above challenges that inherently build security defense against the data/text spamming attacks can be resolved. The problematic interactions between newly deployed IoT devices and the conventional core network lead us to discover seven vulnerabilities from two major aspects, system-integrated and service-integrated, for breaking the security defense. Specifically, for the system-integrated aspect integrating cellular IoT devices into the cellular network, we discover two vulnerabilities, namely remote identification of cellular IoT IP addresses (V1) and cellular IoT PSM-unaware charging (V2). V1 is an observed common implementation flaw that roots in the vertical integration across layers on cellular IoT devices, whereas V2 is a design defect of the horizontal integration between cellular IoT devices and the core network. For the service-integrated aspect, we investigate the security of the services used by cellular IoT and then uncover five vulnerabilities: leakage of phone-number device type from VoLTE (Voice over LTE [13]) signaling (V3), leakage of phone-number status from SMS (Short Message Service) signaling (V4), insecure pushed text service (V5), no secure confirmation of PSM configuration (V6), and no anomaly detection on PSM configuration (V7); V3, V6 and V7 are design defects from the 3GPP standards, whereas V4 and V5 are operational issues and operator-dependent. Note that V3, V4, and V5 are specific to LTE-M, since NB-IoT does not support voice or text services; the others are applicable to both of them. Table I summarizes these vulnerabilities.

We further devise four proof-of-concept attacks: (1) data spamming, (2) text spamming, (3) battery draining, and (4) device hibernation, against cellular IoT users based on the discovered vulnerabilities. We evaluate the attacks using various cellular IoT and non-IoT devices in operational cellular networks. The result shows that an adversary can increase an IoT data bill by up to $226 with less than 120 MB spam data traffic, increase an IoT text bill at up to a rate of $5 per second, increase an IoT's power consumption by up to 20 times, and get an IoT device stuck in hibernation for a long time. Note that the attack cost of sending data and text spam is not high, since many Internet service providers (e.g., Xfinity [14]) offer unlimited Internet data plans, and most carriers provide inexpensive unlimited text services.

Finally, we propose a suite of solutions to address the discovered vulnerabilities and confirm their effectiveness based on a prototype and its evaluation. Notably, although the security vulnerabilities are discovered from LTE-M and/or NB-IoT in 4G networks, they will still exist in 5G networks, since LTE-M and NB-IoT have been confirmed to be continuously supported in 5G networks [15].

This paper makes the following key contributions:

- We identify seven vulnerabilities of the cellular IoT from standard design defects, network operation slips, and device implementation flaws. We validate them experimentally and analyze root causes.
- We devise four proof-of-concept attacks by exploiting the identified vulnerabilities and assess their real-world impact on five major U.S. and Taiwan IoT carriers.
- We propose a suite of standard-compliant solutions and evaluate them based on a prototype. The lessons learned can secure and facilitate the global deployment of cellular IoT services, especially for upcoming 5G networks.

The rest of this paper is structured as follows. §II introduces the primer of the cellular IoT service. We analyze and discover vulnerabilities of cellular IoT data service, text service, and power saving service in §IV, §V, and §VI, respectively. We propose the standard-compliant solutions and evaluate it in §VII. Finally, we present discussion, related work, and conclusion in §VIII, §IX, and §X, respectively.

II. CELLULAR IOT SERVICE PRIMER

Cellular IoT is an emerging solution for connecting IoT devices over cellular networks. Cellular IoT devices share network infrastructure with non-IoT devices (e.g., smartphones), but require special supports, such as PSM [2], [3], [5], [16]. We target cellular massive IoT applications (e.g., smart agriculture and location tracking) with the requirements of low cost, low energy, and small data volumes; they are mainly supported by LTE-M [2], [16] and NB-IoT [3], [17], where LTE-M supports the data, voice, and text services, whereas NB-IoT has the data service only.

Cellular IoT network architecture. Figure 1 shows the 4G LTE cellular IoT network architecture. It consists of Radio Access Network (RAN) and core network. The RAN connects IoT devices to the core network. The core network comprises eight main entities as follows. The MME (Mobility Management Entity) is responsible for user mobility, user

TABLE II
COMPARISON OF NON-IOT AND IOT SERVICE PLANS FOR THREE MAJOR U.S. CARRIERS (STUDIED IN OCT. 2022)

Carrier	Service	Non-IoT Devices‡			IoT Devices		
		Limited Plan		Unlimited Plan	Limited Plan		Unlimited Plan
		Monthly fee	Overage	Monthly fee	Monthly fee	Overage	Monthly fee
AT&T	Data	5GB ($30), 15 GB($40)	Reduce to 128kbps	$65	0.5MB ($0.99), 1MB ($1.5), 2MB ($2),..	Auto renew	$30
	Voice	$0*	$0*	$0*	NA	NA	NA
	Text	$0*	$0*	$0*	NA	NA	NA
Verizon⋆	Data	5GB($40), 25GB($60)	Reduce to 128kbps	$65	1MB($3), 50MB($6), 100MB($9)	⋄: 1 MB($0.2∼$1.25)	NA
	Voice	$0*	$0*	$0*	NA	NA	NA
	Text	$0*	$0*	$0*	†: $0.05 per text	NA	NA
T-Mobile⋆	Data	2GB($10), 5GB($20), 10GB($30)	Reduce to 128kbps	$50	†: $0.1 per MB	NA	NA
	Voice	$0*	$0*	$0*	NA	NA	NA
	Text	$0*	$0*	$0*	NA	NA	NA

‡: The non-IoT service plans studied in this table are individual smartphone user plans but not family plans. *: Included in the data service plan.
†: No minimal subscription is required; IoT users are charged by their service usage amount.⋄: $1.25/MB for a 1 MB plan, $0.4/MB for a 50 MB plan, $0.2/MB for an 100 MB plan.
⋆: Verizon and T-Mobile do not directly sell IoT plans to individual users; however, users can still subscribe to IoT services through the operators' collaborators, such as DigiKey and Twillio; all the SIM cards purchased from the collaborators still come with official Verizon or T-Mobile logos.
Note that the text and voice services presented in this table are cellular IMS-based services, rather than those from Internet, such as Skype and WhatApps.

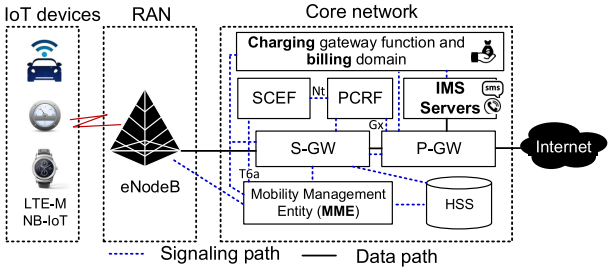


Fig. 1. Cellular IoT network architecture.

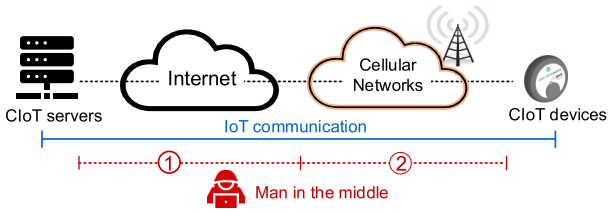


Fig. 2. MiTM attacks in the threat model.

authentication, and resource reservation. The HSS (Home Subscriber Server) stores user information and subscription data. The S-GW (Serving Gateway) forwards data between the RAN and the P-GW (Packet Data Network Gateway), whereas the P-GW assigns IP addresses to cellular IoT devices, routes data between the S-GW and the Internet or IMS (IP Multimedia Subsystem) server, and keeps track of data usage of the IoT devices. The IMS server provides the IoT devices with the voice service, VoLTE [13], and text service [18]. The SCEF (Service Capability Exposure Function) monitors the desired events (e.g., connection status) regarding IoT devices and provides notifications. The PCRF (Policy and Charging Rules Function) mainly mandates the S-GW and the P-GW to detect service data flows, enforce flow policies, and collect service usage statistics. The CGF (Charging Gateway Function) collects data usage from the 4G gateways and forwards it to a billing system to generate bills based on the operator's charging policies.

Cellular IoT-specific functions. There are two major IoT-specific functions, which are supported by both LTE-M and NB-IoT. The first is the half duplex (HDX) communication [2], [3], where an IoT device cannot transmit and receive data simultaneously. With the HDX, the maximum downlink speeds of LTE-M and NB-IoT are only 300 Kbps and 26 Kbps, respectively. The second is the PSM [2], [3], which can increase the battery life of massive IoT devices. It allows an IoT device to enter the sleep mode to save power; it needs to inform the MME of its desirable sleep and active time periods. By cellular IoT standards [2], [3], the minimum and maximum sleep times for the PSM are 4 hours and 413 days, respectively. For the length of active time, there are three kinds: (1) from 2 to 62 seconds in a sequence with a difference of 2, (2) from 1 to 31 minutes in a sequence, and (3) from 6 to 186 minutes in a sequence with a difference of 6. A sleeping IoT device is unreachable and cannot receive any signaling messages or data, but still keeps its registration state and IP address with the core network.

Note that conventional cellular devices have only *active* and *inactive* modes, since the *sleeping* mode can prevent them from receiving incoming calls, text, or data. In the active mode, the devices have established radio connections with the infrastructure for immediate signaling/data transmission; in the inactive mode, they have no radio connections but can timely reestablish them for the infrastructure's `Paging` requests [16].

Operational cellular IoT charging policies. The current service charges of cellular IoT devices from three major U.S. carriers, AT&T, Verizon, and T-Mobile, are summarized in Table II. We have three observations. First, the IoT data service plans are cheaper than non-IoT ones, e.g., $0.99 (500 KB) v.s. $30 (5 GB) in AT&T. Second, the data unit prices of IoT services are 13∼1,304 times higher than those of non-IoT services, e.g., an IoT user needs to pay $0.1∼$3 for 1 MB data, whereas a non-IoT user is charged only 0.23∼0.78 cents with a limited data plan. Third, non-IoT users subscribing to data service plans are offered free voice and text services; however, IoT users are charged for their usage amounts, e.g., $0.05 per sent/received text message.

III. THREAT MODEL, METHODOLOGY, AND ETHICAL CONSIDERATION

Threat Model. In this work, victims are cellular IoT users attacked remotely by adversaries who are organizations or individuals. Assume that neither cellular IoT networks nor IoT devices are compromised; the adversaries adhere to all cryptographic assumptions, e.g., encrypted messages cannot be decrypted without decryption keys. There are three different threat models for the proposed attacks below.

leftmargin=0.1in

- Data spamming attack (Section IV-C): the adversaries can launch a MiTM (Man in The Middle) attack (e.g., [19], [20], [21]) against victims by sitting between the victims' cellular IoT devices and their IoT servers. As shown in Figure 2, the communication path between the IoT devices and servers can be divided into two segments: Segment 1 includes the network routes/facilities between

outside the cellular network and the IoT servers, whereas Segment 2 indicates the inside of the cellular network. In this attack, the adversaries are assumed to sit somewhere on public communication channels from Segment 1, so they can intercept and modify messages exchanged between the IoT devices and servers, and inject messages into their communication. There have been several techniques exposed to achieve such a MiTM attack. For example, the adversaries can leverage the DNS spoofing or the ARP spoofing, or compromise a host at the IXP (Internet Exchange Point) [22].

- Text spamming attack (Section V-D): the adversaries have full control of rooted smartphones that have the VoLTE and text services enabled, and launch the attack from end devices instead of sitting on the communication path.
- Battery draining and device hibernation attacks (Section VI-C): the adversaries are assumed to be present somewhere on the path of cellular signal transmission from Segment 2. They can eavesdrop on the communication between IoT devices and servers, overshadow the devices' cellular signals, and inject messages into the communication. Several techniques [19], [23] have been proposed to launch such the MiTM attack.

Experimental Methodology. We validate the discovered vulnerabilities and attacks of the cellular IoT service in the networks of three U.S. carriers and two Taiwan carriers, which are denoted as US-I, US-II, US-III, TW-I, and TW-II; they together take more than 80% and 50% of market share in U.S. and Taiwan, respectively. We test three kinds of devices: (1) various carrier-certified cellular IoT devices supporting both LTE-M and NB-IoT technologies, including Wio CIoT Tracker [24], Pycom FiPy [25], mangOH Yellow [26], Sixfab CIoT HAT [27], Arduino MKR NB 1500 [28], Telit Charlie Evaluation Kit [29], and Waveshare CIoT kit [30]; (2) non-cellular IoT devices, including 2 Wi-Fi-connected smart sockets, Geekbes YM-WS-5 and TECKIN SP10; (3) cellular non-IoT devices with four smartphones, including Google Pixel 5, Apple iPhone XS MAX, and Samsung S5/S10. They are connected to the operational cellular networks in the experiment. Note that we had reported the vulnerabilities to the affected cellular IoT operators.

Ethical Consideration. We understand that some feasibility tests and attack evaluations might be detrimental to cellular network operators and users. We thus proceed with this study in a responsible manner by running controlled experiments. Specifically, two approaches are adopted. First, in all the experiments, we use our own devices as the victims, and no human subjects are involved. Second, the vulnerability validation and attack experiments are conducted with small-scale tests on the principle that aims to disclose cellular IoT security issues instead of aggravating damages.

IV. Vulnerable Data Service of Cellular IoT

The data service of the cellular IoT may be vulnerable to traffic spam, since its subscriptions have only a small amount of data yet are with much higher unit prices than those of non-IoT subscriptions (see Table II). That small data amount available to cellular IoT devices can be easily exhausted under a spamming attack. It may cause the owners of the cellular IoT devices to either pay high overage fees for data usage or suffer from the IoT service termination.

Seemingly, it is challenging to spam cellular IoT devices even by a MiTM attack, since various IoT and non-IoT traffic flows can be observed. While observing traffic coming from the cellular network, the adversary needs to identify the IP addresses used by cellular IoT devices so that (s)he can spam them. Identifying the IP addresses can be difficult, since carrier networks do not adopt different IP assignment mechanisms for IoT and non-IoT devices according to our study on three U.S. carriers. Although cellular IoT devices may have specific IoT traffic patterns with sparse data transmissions, which may enable the identification of their IP addresses, those IoT traffic patterns can be also observed from the Wi-Fi IoT devices that connect to the cellular network through Wi-Fi-to-Cellular home gateways. Such mixed usage scenario including both cellular and Wi-Fi IoT devices in the cellular network makes it more difficult to identify the IP addresses used by the cellular IoT.

However, after studying whether the existing device/network operations conflict with the new cellular IoT power saving service, i.e., PSM, we discover two vulnerabilities that make the spamming attack possible. The first vulnerability (V1) comes from inconsistent states between transport-layer communication and the underlying PSM at cellular IoT devices. It allows the adversary to remotely probe whether an IP address is used by a cellular IoT device. The second one (V2) is from a mismatch between the PSM and some core network operations. That is, the spam traffic sent to a sleeping cellular IoT device can be accepted and charged at the core network, but the sleeping device is unaware of it and cannot take any immediate defense. More threateningly, the device owner needs to pay for the spam.

We next elaborate on each vulnerability with experimental validation and then present the spamming attack.

A. V1: Cellular IoT IP Addresses Can Be Identified

We can identify the IP addresses used by massive cellular IoT devices by probing whether they have the PSM (TS24.301 [5], CLP.28 [3], TS36.331 [16]) or not, since most of them enable the PSM to extend battery life but the other cellular devices do not have it. For the probing, based on the proposed threat model with the MiTM attack, the adversary can observe the traffic coming from an IP address and interact with the device owning the IP by sending packets to the IP and intercepting the device's response. Once there is a kind of probing packets to which each non-*sleeping* device (i.e., in the *active* or *inactive* mode) has to reply, no response from a device implies that the device is offline or *sleeping* with the PSM. Moreover, the offline case can be excluded when probing an IP address is only triggered at the observation of the traffic coming from the IP, which represents its device is active. Thus, no response observed for an IP address can be used to infer that its device is a PSM-enabled cellular IoT device. Note that although there is still a possibility that an active device with outgoing traffic suddenly becomes offline during the probing (e.g., the device is powered off or enters a non-signal zone) and then no response is observed, the probability can be small.

To this end, we develop a probing mechanism based on the cellular IoT PSM, designated as CIoT-Prober. It sends a series of probe messages to each given IP address by carefully

TABLE III
CONFUSION MATRIX OF THE CLASSIFICATION OF PSM-ENABLED CELLULAR IOT DEVICES BASED ON CIOT-PROBER

Actual class \ Predicted class	PSM-enabled Cellular IoT Devices					Non PSM-enabled Cellular IoT Devices							
PSM-enabled Cellular IoT Devices	100% (150/150)					0%							
Non PSM-enabled Cellular IoT Devices	0%					100% (240/240)							
Devices	Arduino MKR	Botletics SIM7000	RAKWireless RAK2011	Sixfab CIoT HAT	Pycom FiPy	**Cellular IoT without PSM** Wio CIoT Tracker	MangoH Yellow	**Non-cellular IoT Devices** Geekbes YM-WS-5	TECKIN SP10	**Non-IoT Devices** Galaxy S5	Galaxy S10	Google Pixel 5	iPhone XS Max
Probing time	2m14s	3m18s	2m06s	2m39s	1m46s	9m49s	9m50s	9m46s	9m46s	9m49s	9m49s	9m50s	9m48s

selecting probing intervals that ensure at least a probe occurs during the sleep time of each probed IoT device. Therefore, when all the cellular devices without PSM can acknowledge all the probes, a PSM-enabled IoT device can be identified based on a failed probe caused by its sleep. Note that a probe may contain multiple probing messages to cover packet loss cases.

One prerequisite for the probing is that cellular IoT devices need to have a service running independently of the PSM so that CIoT-Prober can probe the service to determine whether the probed devices go to sleep. According to the observation obtained from our tested cellular IoT devices, the TCP connection between each cellular IoT device and its server keeps staying alive no matter whether the device is sleeping; that is, the IoT devices do not close TCP connections before going to sleep. Therefore, CIoT-Prober can probe each ongoing TCP connection, and expect that active cellular IoT devices and the other cellular devices can always be probed successfully but the sleeping cellular IoT devices make the probing fail. Note that TCP has been broadly used by IoT messaging protocols (e.g., MQTT and HTTP) in practice; a recent study [31] shows that top two IoT communication protocols are HTTP/HTTPS (51%) and MQTT (41%), which are TCP-based protocols.

There are still two major challenges to be addressed. First, *which kind of TCP packets can be used for the probing to make all non-sleeping devices reply but does not affect their ongoing TCP connections?* We discover one kind of TCP ACK packets is suitable for the probing; the TCP ACK packets acknowledge the sequence number that has not been used yet by the other TCP connection end. On receipt of such ACK packet, the recipient needs to reply to it with another ACK packet using a correct sequence number and then discards it [32]. Thus, it does not affect the state of the ongoing TCP connection.

Second, *how to make sure that at least one probe can proceed while the probed device is sleeping if it is a PSM-enabled cellular IoT device?* According to the cellular IoT standards [2], [3], each PSM-enabled cellular IoT device must be configured with a length for each of its active time periods, and the length is limited to three kinds of values (see Section II). Thus, multiple probes can be scheduled with a set of intervals where for each possible active time length, at least one interval value is larger than the active time but smaller than the sum of the active time and the minimum sleep time; it can ensure that at least one of the consecutive probes with that interval happens while the probed device is sleeping.

In practice, carriers may set their specific constraints on the minimum active time; the value of 16s is observed from AT&T and Verizon. Also, most device vendors restrict active times for longer battery life; specifically, 80% of massive cellular IoT devices [33] are with average active times less than 5mins. Based on the above two practical observations, the possible values of the active time lengths can be greatly pruned; they are in a range of 16s and 5mins.

Algorithm 1 Cellular IoT Probing Mechanism

```
Launch an ARP spoofing attack against the router that the IoT server connects.
Set time intervals list T[0, 15, 30, 60, 180, 300]; i = 0;
while (1) do
    if Observe a new TCP connection from cellular network then
        while i ≤ 5 do
            Sleep T_i seconds;
            Send a probing packet while spoofing the source IP;
            if No response is received within 5s for the initial packet then
                Retransmit the probing packet;
            end if
            if No response is received within 5s for the retransmission then
                The PSM-enabled device is identified;
                Break;
            end if
            i++;
        end while
    end if
end while
```

1) Validation: We experimentally validate the effectiveness of CIoT-Prober by examining whether it can successfully identify IP addresses used by cellular IoT devices. CIoT-Prober is deployed to sit on the communication paths between all the test devices and the IoT server by launching an ARP spoofing attack against our router to which the IoT server connects. Notably, some remedies, such as ARP inspection [34] and Secure Neighbor Discovery (SEND) [35], have been proposed to defend against the ARP spoofing attack. However, they are optional and thus not broadly deployed in practice; any of them is not observed in our experiments.

We conduct the experiment using 13 test devices in our campus network: 7 cellular IoT devices, 2 Wi-Fi-connected smart sockets, and 4 smartphones (i.e., non-IoT devices). The PSM mechanism is enabled on all the cellular IoT devices except for Wio CIoT Tracker and MangoH Yellow; the lengths of their active times are randomly set to the available values between 16s and 300s. To emulate TCP connections of the test devices, a test application is deployed at each of them to build a TCP connection with our deployed IoT server. The TCP connection is created 3 times per day (i.e., once at each of the morning, afternoon, and evening times). There are 13 participants, and each of them carries one test device; the experiment lasts for 10 days.

The pseudocode of the probing algorithm is shown in Algorithm 1. Once observing a packet of a new TCP connection coming from the cellular network, CIoT-Prober sends 6 probing messages with intervals, 15s, 30s, 60s, 180s, and 300s to the packet's source IP while spoofing the source IP in the probing messages using the packet's destination IP. To cover packet loss cases, each probing message is retransmitted if no response is observed within 5s after its initial transmission. When no response is received for the retransmission, the probed IP address is identified to be used by a PSM-enabled cellular IoT device.

Experimental result. Table III summarizes the experimental results, and we make two observations. First, CIoT-Prober can identify 5 PSM-enabled cellular IoT devices with 100% accuracy. There are 150 positive cases from the 10-day experiment where a TCP connection is built 3 times per day by each device. We believe that some false positive cases may happen in practice, but they can be rare. For example, a non-IoT device may skip responses of the probing messages when encountering temporary out-of-service (e.g., handover) or power-off; it can mislead CIoT-Prober to identify them as cellular IoT devices. Since CIoT-Prober has employed a dual-probing mechanism, which retransmits probing messages, so the probability of the false positive cases can be greatly reduced. Moreover, the impact of those cases is very lightweight on attack cost. The reason is that launching a spamming attack against each identified IP address needs only a small amount of spam traffic (e.g., several MBs) to cause an excess bill or service termination.

Second, the probing cost varies with different devices; specifically, the PSM-enabled cellular IoT devices take much shorter probing times than the other devices do. The probing times of the PSM-enabled cellular IoT devices range from 1m46s to 3m18s, whereas those for the other devices range from 9m46s to 9m50s. The reason is that the PSM-enabled devices can be identified once a probe occurs while they are sleeping, but probing the others cannot stop until all the probing messages are sent. Note that the latter probing time takes around the sum of all the probing intervals (i.e., $15s + 30s + 60s + 180s + 300s = 9m45s$) and transmission times.

2) Root Cause and Lesson: This vulnerability can be attributed to a common implementation flaw that when the software is deployed on IoT devices, its functions or protocols are not reviewed with the underlying PSM mechanism of cellular IoT from a security aspect. This imprudent deployment leads to the inconsistent state between the transport-layer communication (i.e., TCP) and the PSM. It can be observed on all the tested cellular IoT devices. To secure them, it calls for a review of vertically integrated security from new cellular IoT features at low layers to conventional upper-layer functions/protocols, thereby making appropriate updates.

B. V2: Cellular IoT PSM-Unaware Charging

Conventional cellular non-IoT devices do not have the PSM mechanism, so the core network functions need to be updated to support the cellular IoT PSM. Although the non-IoT devices have an *inactive* mode, it is different from the IoT *sleep* mode (see details in Section II). An inactive non-IoT device can be notified to become active whenever it has any downlink traffic reaching the core network, whereas a sleeping IoT device cannot be notified until it leaves the sleep mode. Specifically, an active user equipment (UE) has several established control-plane connections and data-plane bearers with the infrastructure; it can become inactive due to no signaling or data traffic for a while, and then the control-plane connections and data-plane bearers are temporarily released. When any data traffic sent to the UE reaches the P-GW/S-GW, the MME is notified and then sends a Paging message [16] to notify the UE; afterwards, the UE performs the service request procedure [36] to reestablish the released connections and bearers.

Once the core network treats a sleeping IoT device as an inactive UE, current network operations may be directly applied to the cellular IoT PSM without any modification; it can cause the vulnerability of cellular IoT PSM-unaware charging. For inactive UEs, the P-GW can still account for the downlink data usage of the alive data-plane bearer and forward the data usage to the CGF. This operation does not have any issues with inactive non-IoT devices, which can be notified to receive the data, but it can cause sleeping IoT devices to be charged for the incoming downlink data yet without receiving them. Moreover, they cannot take any immediate defense manner against the incoming data, when they are spam. Note that the cellular IoT standards [37], [38] do not stipulate that the P-GW shall suspend the charging function for sleeping IoT devices.

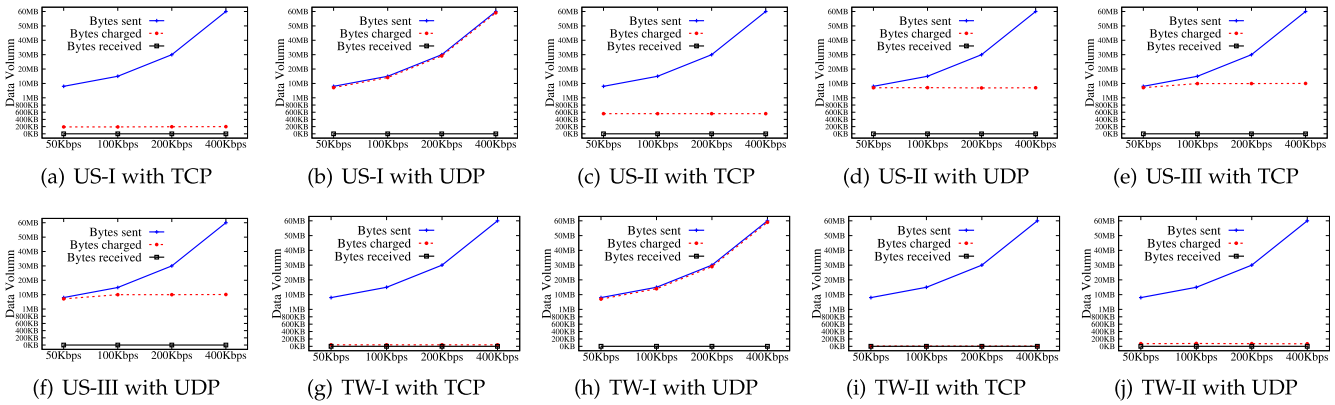
1) Validation: We conduct an experiment to validate this vulnerability by sending traffic to sleeping cellular IoT devices and then checking whether the devices are charged for the traffic or not. We test those aforementioned five carriers with TCP traffic. The experiment consists of four steps: (**??**) keeping a tested IoT device power-off for three days and then obtaining its latest data usage amount from its subscribed carrier; (2) powering on the IoT device, and connecting it to the carrier network with PSM enabled and a configuration of the PSM active and sleep times set to the minimum values allowed by the carrier (e.g., they are 16s and 3hrs, respectively, for both US-I and US-II); (3) letting the IoT server send 100 KB data to the IoT device while it is sleeping, and keeping the IoT device on for 30mins after it wakes up and then powering it off; and (4) waiting for three days and then checking the tested IoT device's latest data usage.

Experimental result. We have two observations: (1) the IoT devices tested in the networks of those five carriers do not receive any packets; and (2) all the tested devices in US-I, US-II, and US-III are charged for 100 KB data, whereas those in TW-I and TW-II are charged for 20 KB and 3 KB data, respectively. The result confirms that the charging function is unaware of the cellular IoT PSM, and is not suspended for it; however, the charging volume cap varies with carriers.

2) Root Cause and Lesson: When the PSM mechanism is introduced as a new cellular IoT feature, the management-plane functions including accounting and charging shall be adapted for its operation. The MME in the control plane can know when each attached IoT device is sleeping through the PSM active and sleep times specified in the EMM (EPS Mobility Management [5]) protocol messages (e.g., Attach Request), which are exchanged between cellular IoT devices and the MME, and the P-GW in the data plane can also know the information from the MME. However, the 3GPP charging standards [37], [38], [39], [40] do not stipulate that the charging function at the P-GW shall deal with sleeping IoT devices. Such design defect causes the cellular IoT to bear the potential security threat of data spamming. To secure the ecosystem of cellular IoT, a prudent design review of the horizontally integrated security between device and network ends is a must.

C. Proof-of-Concept Attack

We devise a spamming attack against cellular IoT devices using V1 and V2 and then evaluate its damage. To launch the attack, the adversary uses a MiTM attack to sit between



(a) US-I with TCP (b) US-I with UDP (c) US-II with TCP (d) US-II with UDP (e) US-III with TCP

(f) US-III with UDP (g) TW-I with TCP (h) TW-I with UDP (i) TW-II with TCP (j) TW-II with UDP

Fig. 3. Under an IoT spamming attack, the spam traffic volume is sent, charged, and received for those five carriers with TCP and UDP traffic cases.

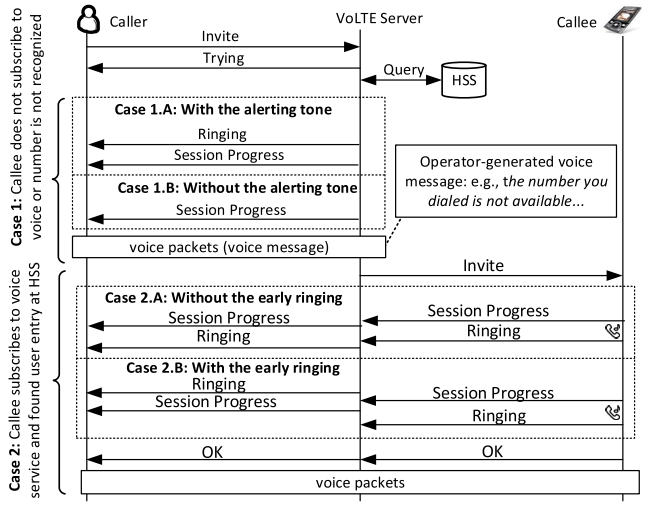


Fig. 4. VoLTE call flow procedure for two cases.

cellular IoT devices and their IoT servers. Although there are many IP addresses which the adversary can see from the eavesdropping, (s)he probes only the IP addresses belonging to her/his target carriers, which support cellular IoT services. For each target carrier, the adversary can obtain a list of IP addresses owned by it using some free online databases [41], then probe those IP addresses to identify the ones used by cellular IoT devices using CIoT-Prober, and finally send spam traffic to the identified IP addresses.

We conduct an experiment to evaluate the spamming attack. We test those 5 carriers with TCP and UDP traffic using 8 different devices: 2 PSM-enabled cellular IoT devices including RAKWireless RAK2011 and Sixfab CIoT Hat, 2 non-cellular IoT devices including Geekbes YM-WS-5 and TECKIN SP10, and 4 smartphones. We deploy an IoT server and a laptop with the CIoT-Prober in the campus network. The CIoT-Prober launches an ARP spoofing attack to intercept all traffic of the IoT server. To start the experiment for each carrier, the tested application on each of those devices connects to the IoT server. Afterwards, CIoT-Prober starts to identify the IP addresses used by cellular IoT devices and send spam traffic to each identified IP address at various source rates. Each spamming attack lasts for 20 minutes.

Experimental result. Our results show that CIoT-Prober can successfully identify those two cellular IoT devices without any false positive cases. Figure 3 plots the spam traffic volume charged for each carrier with TCP and UDP traffic when spamming attacks are launched against the identified victim devices. We have three findings. First, for all the cases, the IoT devices do not receive any spam traffic but are charged for it. The reason is that an IoT device's IP address can be identified only when it is sleeping; then, when the spamming attack is launched right after the identification result, the sleeping device cannot receive any spam traffic. Second, for the UDP results, US-I and TW-I do not impose any charging volume caps, and their charging volumes can achieve up to 60 MB; US-II, US-III and TW-II have charging volume caps about 6∼7 MB, 9∼10 MB and 30∼60 KB, respectively. Third, for the TCP results, US-I, US-II, TW-I and TW-II impose charging volume caps, 200 KB, 540 KB, 20 KB and 3 KB, respectively, but US-III has a higher cap with 9.8 MB.

The IoT spamming attack can lead to two kinds of damage on IoT users: excess bills and denial of IoT service. The excess bills can be made when the users enable the auto-renewal IoT service; this service helps users to automatically purchase more data quota when it is exhausted. In one experiment for an IoT device, an increase of $226 in a monthly bill can be made by the spamming attack with only less than 120 MB spam traffic. On the other hand, when the auto-renewal service is not enabled, the users can suffer from the denial of IoT service after an available data quota is exhausted. Note that since the cost of this spamming attack is not high (e.g., several MBs for a device), the adversary may launch a large-scale attack against many cellular IoT devices to cause significant damage.

V. Insecure Cellular IoT Text Service

An IoT device can get an assigned phone number, denoted as IoT number thereafter, for its text service. However, the unit price of text messages for IoT users (e.g., $0.05 per message) is much higher than that for non-IoT users (e.g., unlimited messages with a subscribed data service). It can give an adversary the incentive to launch a text spamming attack against cellular IoT devices using non-IoT devices, thereby causing the IoT users to suffer from excess text fees. The prerequisite of this attack is to identify the IoT numbers which belong to the cellular IoT users with subscribed text services. Identifying the IoT numbers can be challenging, since the numbers assigned to cellular IoT and non-IoT users are formed in the same format as E.164 [42] (e.g., +1-800-342-6626). Moreover, carriers do not adopt any different assignment policies for the IoT and non-IoT phone numbers.

We then study if IoT numbers can be identified based on a side-channel attack from the cellular services depending

on them. It leads us to discover two vulnerabilities from operational voice and text services. The first vulnerability is that the signaling messages of VoLTE can leak two types of phone numbers: non-IoT numbers, and the others including IoT and unassigned numbers (V3). The second one is that the SMS signaling (e.g., `SM-RP-DATA` [43]) can be exploited to differentiate IoT numbers from unassigned ones (V4).

Given that IoT numbers can be identified from the above two vulnerabilities, we further discover that the text services offered by carriers are not protected against spam text messages (V5). Thus, the text spamming attack can be successfully launched against cellular IoT devices; moreover, the attack cost can be lightweight when a smartphone with an unlimited plan of the text service is used. In the following, we first elaborate on the three vulnerabilities and then present the text spamming attack.

A. V3: Leakage of Phone-Number Device Type From VoLTE Signaling

Most IoT numbers have only text service but do not subscribe to voice service; it may cause different call responses on the VoLTE signaling from calling IoT and non-IoT numbers, and then be exploited to leak the device type of a phone number. This practice is observed from our two studies. First, we study all the cellular networks supporting LTE-M and NB-IoT; there are 12 cellular IoT networks that support E.164 numbers for cellular IoT devices. 10 of those 12 IoT networks, which include US-II, restrict IoT numbers to text service only, whereas the other 2 networks support both voice and text services for the cellular IoT. Second, we confirm with four major U.S. carriers including Verizon, AT&T, T-Mobile, and Sprint that their non-IoT numbers are always offered voice service.

Calling the numbers with and without voice service can lead to two different cases of call initialization procedure. Figure 4 shows the VoLTE call procedure in those two cases. At the beginning, the VoLTE user sends a SIP `INVITE` message to the VoLTE server, and then the server attempts to obtain the subscription data of the callee by querying the HSS. In Case 1, where the callee has an IoT number without voice service, the HSS cannot find the subscription data associated with the callee. The VoLTE server then sends the SIP `RINGING` and `SESSION PROGRESS` (Case 1.A), or `SESSION PROGRESS` (Case 1.B), to the caller. In Cases 1.A and 1.B, the caller can hear an alerting tone before an operator-generated voice error message, and the voice error message directly, respectively. The call procedure of this case is similar to that of calling an unassigned number.

In Case 2, the callee has a non-IoT number, and the HSS can discover its subscription data. The VoLTE server then forwards the SIP `INVITE` to the callee and two cases of call procedure may happen. In Case 2.A without the early ringing, the VoLTE server waits for the callee's response and forwards its `SESSION PROGRESS` and `RINGING` back to the caller. In Case 2.B with the early ringing, the VoLTE server sends `RINGING` back to the caller directly without the callee's response after waiting for a pre-defined time period (e.g., 5 seconds), thereby avoiding a long silence.

Thus, when a VoLTE user makes a call to a phone number, the user can receive the `SESSION PROGRESS` or `RINGING` message from the VoLTE server in Case 1, where the callee number is an IoT one without voice service or an unassigned one, much sooner than he does in Case 2, where the number is a non-IoT one. The different call response times from non-IoT and IoT/unassigned numbers can result in the leakage of phone-number device types.

Validation. We validate this vulnerability by considering 3 IoT numbers, 4 non-IoT ones, and 2 unassigned ones from US-II in the experiment; US-II is the only carrier supporting text service for cellular IoT in our tested areas. The IoT numbers are used by two US-II-certified IoT devices, Sixfab Cellular IoT HAT and Pycom FiPy, whereas the non-IoT numbers are from two smartphones, Apple iPhone 12 and Samsung Galaxy S8, and two campus landline phones. To examine possible vulnerability variance, we consider the callees with different voice technologies and conditions, including 3G, 4G CSFB [44], 4G VoLTE/VoWiFi, different signal strengths, and power on/off statuses.

The experiment is conducted as follows: (1) using a rooted smartphone, Samsung Galaxy S10, to dial VoLTE calls to each of those 9 numbers in each case with 20 runs; and (2) collecting signaling messages using Tcpdump [45] and then analyzing them by developing a Python program with the Scapy [46] library. From each call trace, the `Inv-R/S` RTT (Round Trip Time), which is the time period between the leaving `INVITE` and the arrival `RINGING` or `SESSION PROGRESS`, is measured to be used as the call response time.

Figure 5 plots the `Inv-R/S` RTT values for each device type in various cases. We make two observations. First, the RTT values obtained from calling the non-IoT and IoT numbers can be clearly differentiated. Specifically, the minimum values from the non-IoT numbers are still 0.1∼0.67s higher than the maximum values from the IoT numbers. Second, the RTT values from the IoT numbers are comparable to those from the unassigned numbers with the median values, 0.76s and 0.74s, respectively. Thus, they can be exploited to differentiate non-IoT numbers from IoT and unassigned numbers. Notably, it is observed that the IoT devices in all the tests do not receive any VoLTE signaling messages.

Root cause and lesson. For easy deployment, cellular IoT inherits the function of phone numbers from conventional non-IoT services, but it is not carefully reviewed to examine if there are any new security vulnerabilities. The phone numbers assigned to IoT devices allow the VoLTE caller to make calls to them, but they do not subscribe to the VoLTE service. When the VoLTE server responds to these IoT calls based on its normal operations defined by standards [18], [47], the clear difference between the call response times from non-IoT and IoT numbers can be used for the side-channel attack. To prevent the timing from being leaked, the VoLTE server may disturb the actual response times.

B. V4: Leakage of Phone-Number Status From SMS

We further discover that the SMS signaling gives different responses to the text messages sent to IoT and unassigned numbers, since the results of their text message deliveries shall be successful and failed, respectively. The delivery results can be obtained from the SMRP (Short Message Relay Protocol [43]) signaling messages generated for each text message by the SMSC (SMS Center). Thus, the IoT numbers can be differentiated from the unassigned ones based on the delivery results. Note that SMRP is a protocol used to transmit text

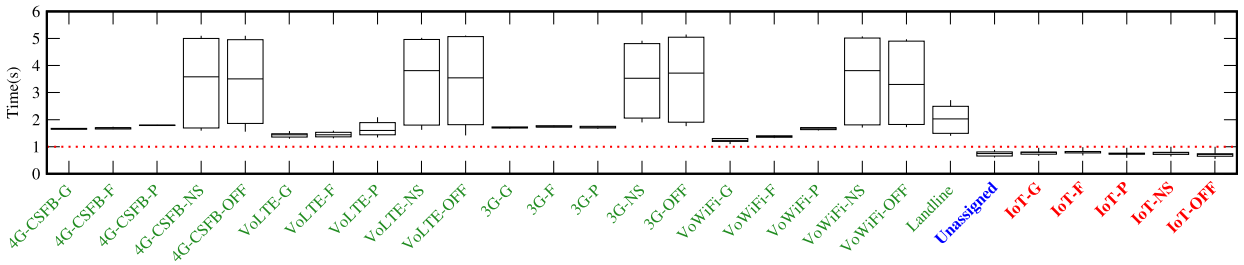
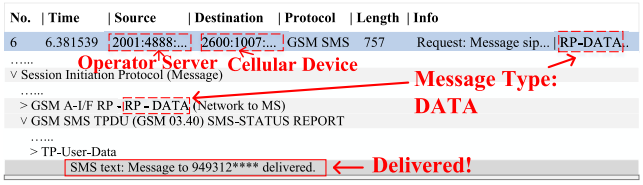
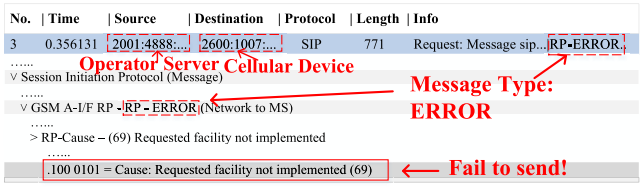


Fig. 5. The `Inv-R/S` RTT values in quartiles, median, maximum, and minimum are observed at the VoLTE caller for each callee type in various scenarios: IoT (Red), non-IoT (Green), and unassigned numbers (Blue); signal strength cases: good (G), fair (F), poor (P), and no-signal (NS); device statuses: power-on by default and power-off (OFF).



(a) The text recipient with an IoT number.



(b) The text recipient with an unassigned number.

Fig. 6. The SMRP signaling responses received by the text sender vary with different recipients.

messages to the SMSC through IMS servers; all its messages are encapsulated by the SIP.

Validation. The validation experiment is conducted as follows: (1) using a smartphone to send text messages that request delivery reports to an IoT number and an unassigned one through the IMS; and (2) analyzing each message's delivery status. Figure 6 shows the SMRP signaling responses which the smartphone receives from the IMS server. The `SM-RP-DATA` message with a delivery report shows "delivered" for the IoT number, whereas the `SM-RP-ERROR` indicates an error [48] with a cause "Requested facility not implemented (69)" for the unassigned number.

Root cause and lesson. The SMS standard [48] specifies that the SMSC shall show an error cause in the `SM-RP-ERROR` or delivery report message for the failed text message delivery, but the error cause can leak too much information. To eliminate the vulnerability, carriers may need to either hide that information together with other information useful for the status inference, or restrict the request of the delivery report in a certain way.

C. V5: Insecure Pushed Text Service

We find that some carriers (e.g., Bell, Tellus, Deutsche Telekom, and Vodafone) charge cellular IoT users for both outgoing and incoming text messages. Such text charging policy is different from the conventional non-IoT text service, which charges for only outgoing text messages or charges the fee of a service plan including the text service (see Table II). However, the incoming text messages can be pushed from an outsider to the IoT device without the device's permission. When the carriers do not deploy any security mechanisms against malicious pushed text messages, their IoT users may receive text spam, thereby suffering excess text fees.

Validation. We validate this vulnerability by sending 10 consecutive text messages from a smartphone to one cellular IoT device in the US-II network. We confirm that the IoT device receives all the messages and is charged for all of them. But, we do not find any mechanisms from US-II to block a specific phone number that generates text spam.

Root cause and lesson. It is not surprising that some carriers charge IoT users for incoming pushed text messages, since the resources allocated to IoT devices are considered to be small for supporting a large number of IoT devices. Once the incoming text service is free for IoT devices, the IoT users may take advantage of this policy by sending commands to the devices with text messages. However, when the pushed text service is not free of charge, carriers shall provide defense mechanisms against incoming text spam.

D. Proof-of-Concept Attack

We next devise an IoT text spamming attack based on vulnerabilities V3, V4, and V5. Before launching this attack, we need to collect a list of phone numbers belonging to the target carrier with V5; it can be done by using some online databases [49]. We develop two programs for this attack on Android phones to check each given phone number: (1) `IoTNumProber`, which checks if the phone number is an IoT number based on V3 and V4; (2) `TextSpamSender`, which generates many spam text messages to the given IoT number within a short time interval by exploiting the reported SMS vulnerabilities [50].

We evaluate the attack by using the same list of phone numbers as the validation experiment, but reduce the number of IoT numbers to one. The `TextSpamSender` is configured to send text spam to the identified IoT number at different source rates from 20 test messages per second (msg/s) to 100 msg/s. Our result shows that the IoT number is successfully identified, all spam text messages are received by the IoT device, and the carrier charges for these spam messages. With a $0.05 charge of a text message in the carrier network, the IoT victim which enables the auto-CIoT-service-renewal feature can suffer from excess text fees at up to a rate of $5 per second with the spam rate of 100 msg/s.

Moreover, the `IoTNumProber` can accurately identify the IoT numbers without any false positive/negative cases while spending 1.6 seconds averagely on examining a phone number. Note that the current implementation of the `TextSpamSender` has not been optimized for the

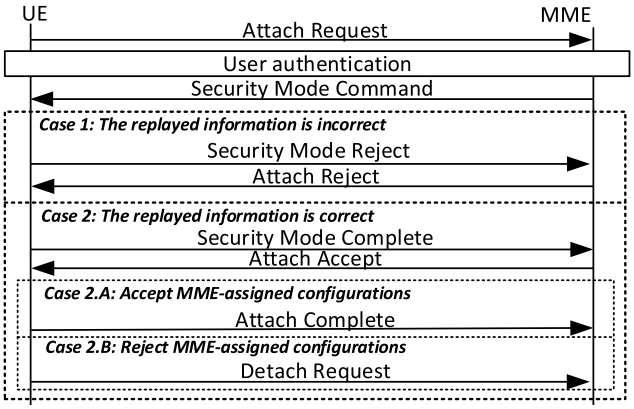


Fig. 7. The confirmation mechanism in 4G EMM protocol [5].

large-scale number examination yet; several approaches can be adopted to further improve the performance (e.g., dialing multiple probing calls simultaneously [51]).

VI. Manipulable Power Saving Operation

To enable the PSM with the core network, cellular IoT devices need to transmit desirable PSM configuration, including sleep and active times, to the MME through the procedures of EMM Attach and Tracking Area Update [5]. Specifically, the PSM configuration is carried in the messages of Attach Request and Tracking Area Update Request, respectively. Seemingly, the Attach Request message may be tampered by the adversary to launch various MiTM attacks, since the message is sent before the completion of user authentication and secure communication establishment procedures [5], which is not confidentially and integrally protected yet. However, this type of the MiTM attacks could be prevented by the confirmation mechanism stipulated in the 3GPP standard [5]; that is, the infrastructure confirms the configuration information (e.g., UE security capability) with each attaching UE via the Security Mode Command message, which has integrity protection, after its successful user authentication, as shown in Figure 7. If there exists any inconsistent information, which could be caused by the message tampering, is observed by the attaching UE, it can abort the attach procedure by sending the Security Mode Reject message (Case 1). Moreover, if the attaching UE does not agree the infrastructure-assigned configurations carried in Attach Accept, it can detach from the connected network by sending the Detach Request message (Case 2.B).

However, we discover that this confirmation mechanism may not be applied to the PSM configuration (V6), so its tampering may not be detected by the UE. Even though the PSM configuration returned for the confirmation is different from the requested one, the IoT UE may still accept it without careful examination from a security perspective (V7). Given these two vulnerabilities, the adversary can manipulate the PSM operation of IoT devices to result in two attacks, battery draining and device hibernation. In the following, we first elaborate on the two vulnerabilities and then present those two proof-of-concept attacks.

A. V6: No Secure Confirmation of PSM Configuration

From the cellular IoT standard, we discover that the confirmation mechanism of the PSM configuration in the Security Mode Command message is not stipulated. So, carriers may not apply the confirmation to the PSM configuration for IoT devices in the secure communication. It may cause the tampering of the Attach Request message from a MiTM attack to be undetected, and then the IoT device and the infrastructure have different views of the PSM configuration, thereby causing the PSM operation to be anomalous.

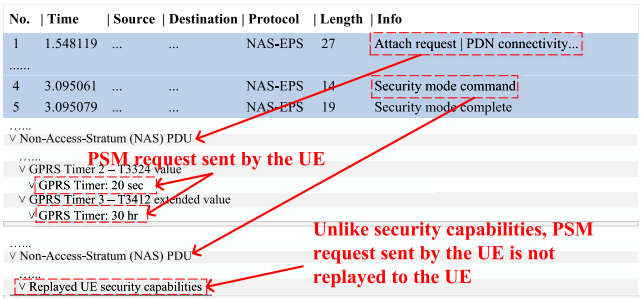


Fig. 8. The PSM configuration requested by the IoT device is not included in the Security Mode Command message for the confirmation.

Validation. We conduct an experiment to validate the vulnerability by connecting an emulated IoT device to operational cellular networks with specified PSM configuration and then observing whether the PSM configuration is included in the Security Mode Command message received by the device. The validation experiment comprises three steps: (1) upgrading the srsUE platform to emulate an IoT device with the PSM operation; (2) testing all the three U.S. carriers by using the srsUE platform to perform the Attach procedure with each carrier; and (3) keeping each successful connection with the infrastructure for one hour.

For all the carriers, it is observed that the PSM configuration sent by the IoT device is not included in the Security Mode Command message, as shown in Figure 8. After the connection is established, the PSM configuration is never re-assigned or confirmed from the infrastructure. As a result, the confirmation of the PSM configuration is not employed by those three U.S. carriers; it may cause IoT devices to suffer from the tempering of the PSM configuration.

Root cause and lessons. The confirmation mechanism of the UE information has been stipulated in the standard to prevent the tampering of the Attach Request message, but it is not carefully reviewed for the new IoT PSM operation.

B. V7: No Anomaly Detection on PSM Configuration

Once the PSM configuration in the Attach Request message is tampered, the infrastructure will rely to the IoT device with the tampered PSM configuration in the Attach Accept message. When no anomaly detection mechanism is deployed at the IoT device, the device may accept the tampered PSM configuration, thereby possibly hurting the IoT service or battery saving. Even though the infrastructure can impose some constraints on the device-requested PSM configuration, they may be loose and still allow the adversary to maliciously manipulate the PSM operation.

Validation. We experimentally validate this vulnerability from two aspects: (1) whether any constraints are imposed by the infrastructure and (2) whether IoT devices accept the assigned PSM configuration different from their requested ones. For the infrastructure aspect, we use two IoT devices, Sixfab CIoT HAT and Arduino MKR NB 1500, to test

TABLE IV
THE ALLOWED MIN/MAX ACTIVE AND SLEEP TIMES FOR DIFFERENT CARRIERS

	Active Time (Min)	Active Time (Max)	Sleep Time (Min)	Sleep Time (Max)
US-I	16 secs	186 mins	190 mins	310 hrs
US-II	16 secs	186 mins	180 mins	310 hrs
US-III	0 sec	186 mins	4 secs	413 days
TW-I	0 sec	186 mins	2 mins	413 days
TW-II	0 sec	186 mins	4 secs	310 hrs

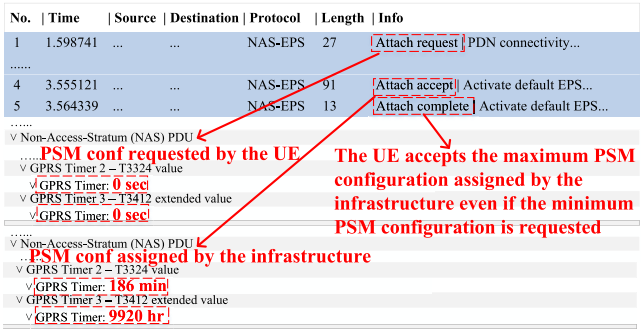


Fig. 9. An IoT device accepts the sleep and active times assigned from the infrastructure while replying with the Attach Complete message.

those five operational cellular networks regarding the allowed min/max sleep and active times. Table IV summarizes the results. There are three observations. First, US-III and TW-I keep the maximum sleep time stipulated by the standard, 413 days, without any more stringent constraint, whereas the other three carriers reduce it to 310hrs. Second, the carriers allow different minimum sleep times with a range from 4s to 190mins, but the GSMA suggests 240mins [2]. Third, all the carriers keep the maximum active time stipulated by the standard, 186mins; for the minimum active time, US-I/US-II and US-III/TW-I/TW-II take 16s and 0s, respectively. Given these constraints, there is still a large range of available active/sleep times for the adversary to exploit.

For the device aspect, four representative cellular IoT devices are tested: Sixfab CIoT HAT, Arduino MKR NB, Waveshare CIoT kit, and Telit Charlie Evaluation Kit; they together take more than 40% of the global market share in 2021 [52]. The validation experiment is conducted as follows: (1) connecting all the four cellular IoT devices to an open-source SDR cellular IoT infrastructure testbed, Sonica [53], which has been upgraded to support the PSM operation; (2) testing an extreme case that the IoT devices are configured to send the minimum supported sleep/active times to the infrastructure via the Attach Request message, whereas the infrastructure assigns the maximum sleep/active times back to them via the Attach Accept message. The result shows that all the IoT devices accept the assigned sleep/active times while replying with the Attach Complete message, as shown in Figure 9.

Root cause and lessons. To satisfy various power saving demands in practice, the cellular IoT standard stipulates a wide range of the PSM sleep/active times. However, the standard does not develop any anomaly detection mechanisms for carriers or IoT devices. It may cause the infrastructure to forward the tampered PSM configuration to devices and further make the devices accept it.

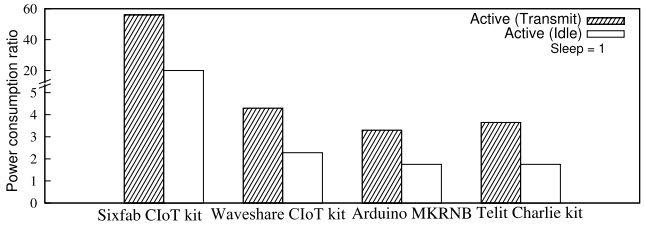


Fig. 10. Power consumption ratios of the transmission and idle states to that of the sleep state for four cellular IoT devices.

C. Proof-of-Concept Attacks

We devise two attacks, battery draining and device hibernation, by exploiting V6 and V7. In the first attack, IoT devices are prevented from entering the PSM sleep mode as long as possible so that they keep being active to consume energy, thereby draining their batteries soon. In the second attack, the adversary gets them stuck in the PSM sleep mode to hinder their normal operations. To launch these two attacks, the adversary needs to modify the PSM active and sleep times in the Attach Request messages sent by IoT devices. Notably, several techniques (e.g., LTE overshadowing attack [23] and LTE relay attack [19]) have been proposed to carry out the modification of the messages sent by mobile devices.

Battery Draining Attack. We validate this attack by tampering the active time in the Attach Request message to be the maximum allowed value (i.e., 186mins for all the tested 5 carriers). The attack is launched against those four cellular IoT devices used in the validation. It is observed that all the tested devices accept the 186mins active time, and none of them stay active for less than 186mins. We further measure their power consumption at three states, namely active transmission, active idle, and sleep. As shown in Figure 10, for all the four devices, the idle and transmission states consume more power than the sleep state by 1.75∼20 and 3∼56 times, respectively. It shows that given negligible transmission time for IoT devices, this attack can indeed drain their batteries sooner by up to 20 times.

Device Hibernation Attack. On the other hand, the adversary can tamper the Attach Request message to set the sleep time as long as possible. In the experiment, we first connect those four IoT devices to the emulated IoT infrastructure built based on Sonica, and then change their active and sleep times to 16s and 310hrs, respectively, in the intercepted Attach Request messages. To test whether they keep sleeping, we have them each establish a TCP connection with an IoT server, which transmits 100-byte data to each connected IoT device every minute. It is observed that under the attack, each IoT device does not respond for more than three hours after entering the sleep mode.

VII. SOLUTIONS

In this section, we propose a suite of solution approaches to address the identified vulnerabilities and evaluate them.

Vertically integrated IoT security. We introduce a vertical security manner for cellular IoT to address vulnerability V1, where cellular IoT IP addresses can be identified remotely. It is a cross-layer PSM coordination mechanism that vertically crosses the transport/application layers and the underlying non-access stratum layer (e.g., EMM and ESM [5]) on the device side. It makes the transport/application layers be aware

of the IoT PSM status and then adapt accordingly. The adaptation is to terminate all ongoing transport/application sessions before the device enters the sleep mode.

Horizontally integrated IoT security. We propose horizontal security manners to address vulnerabilities V2, cellular IoT PSM-unaware charging, and V6/V7, insecure PSM configuration. They require concerted efforts that horizontally span network elements (e.g., MME) and IoT devices.

The remedy for V2 consists of two parts: device-initiated defense and PSM-aware charging. For the first part, the IoT device can block spam packets upon detection in the active status and stop all the incoming traffic before sleeping. It relies on modifying packet filters of the Traffic Flow Template (TFT) associated with the device's EPS bearer. The packet filters with a 5-tuple filter are used to inform the serving P-GW which packets are allowed to be forwarded from the Internet to the device and then charged for. This approach can be done by simply using EPS bearer context modification procedure [5] without any modification to the cellular network standards. For the PSM-aware charging, P-GW with the PSM information shall prevent incoming packets for sleeping cellular IoT devices. It shall not only discard all the packets without any charge, but also notify the source ISP of unwanted traffic to prevent possible Inter-AS (autonomous systems) packet routing fees.

To address V6/V7, the solution includes the confirmation of the PSM configuration from the infrastructure side and the anomaly detection at the device side. For the confirmation, the MME needs to put the PSM configuration included in the Attach Request message sent by the device into the Security Mode Command message. For the anomaly detection, there are two mechanisms. First, the IoT device shall verify the PSM configuration in the Security Mode Command message; if it is different from the requested one, the device shall abort the attach procedure by sending the Security Mode Reject message to the MME. Second, the device needs to check the PSM configuration from the Attach Accept message; if the difference between its values and the requested ones is larger than a pre-defined threshold (e.g., 20%), the device shall abort the attach procedure by sending the Detach Request message to the infrastructure while providing the device owner with a warning message.

Privacy-aware voice and text services. We devise two solution methods to make voice and text services be privacy-aware to address V3, leakage of phone-number device type from VoLTE signaling, and V4, leakage of phone-number status from SMS signaling, respectively. For the voice service, we propose to add a small random delay to the message responses (e.g., Session Progress) from the VoLTE server to the caller, when the intended call recipient is an IoT number without voice service subscription or an unassigned number. The added random delay that contributes to Inv-R/S RTT can thus prevent non-IoT numbers from being easily differentiated from the others. For the text service, the SMSC shall provide a general error cause (e.g., temporary failure (41) [48]) that discloses less information.

Spamming-resistant cellular IoT text service. To address the insecure pushed text service (V5), carriers shall impose some restrictions on the pushed text service. There are two possible manners: (1) allowing only pre-approved numbers to send text messages to a certain IoT number; and (2) restricting the number of inbound text messages to be below a specified threshold; once the threshold is reached for an IoT number, an alert is sent to its owner. However, text spoofing may bypass the mechanism with pre-approved numbers, so carriers shall defend against it by either deploying the ITU-recommended countermeasures [54] to address the disclosed vulnerabilities of SS7 or upgrading the SS7-involved text service to the IMS-based SMS [18].

A. Security Overhead and Analysis

We next examine the cost of computation and communication, and present security analysis for each proposed solution with operation involving multiple network components; the examined solutions include the cross-layer PSM coordination, PSM-aware charging, and privacy-aware voice/text services. Notably, compared with the Internet, the cellular infrastructure is a relatively closed system and its security has been broadly examined by prior art [6], [7], [8]. Therefore, in the security analysis of each solution, we mainly analyze the maximum potential attack damages that may be caused by distributed environments (e.g., the detection of a sleeping IoT device and the termination of IoT data charging are performed on different network elements).

Assumptions. Four assumptions are made in our analysis: (1) the maximum downlink and uplink throughputs of cellular networks and IoT devices follow the specification in 3GPP and GSMA standards; (2) the time complexity of sending/receiving a protocol message (e.g., a TCP segment and an EMM Attach Request) is $O(1)$; (3) the time complexity of a search or insertion operation is $O(\log n)$, where n is the number of records in the database; this assumption is made based on the B+ tree, which is a common data structure used by database systems, such as MySQL, at cellular networks and IoT devices; (4) the spamming attack is launched against cellular IoT devices given that they have been identified.

Notations. We denote the cellular network infrastructure as $\mathbb{C}$, cellular IoT devices as $\mathbb{D}$, the maximum uplink and downlink throughputs of cellular IoT services as $\mathcal{R}_{ul}$ and $\mathcal{R}_{dl}$, and the maximum transmission rate of spamming traffic as $\mathcal{R}_{attacker}$.

1) Cross-Layer PSM Coordination: This device-side remedy enables cellular IoT devices to terminate all the ongoing TCP connections before entering the sleep mode; it can prevent the adversaries from successfully probing the PSM-based cellular IoT devices by using the must-response TCP probe messages. We model this approach as follows. Step 1: $\mathbb{D}$'s EMM layer notifies the TCP layer that the device will enter the sleep mode. Step 2: $\mathbb{D}$ discovers all active TCP connections, the number of which is α, by querying its local database. Step 3: $\mathbb{D}$ terminates all the active TCP connections by transmitting TCP Reset packets to the peers.

$\diamond$ *Computation and Communication Cost:* the computation overhead includes the discovery of active TCP connections and the processing of sending α TCP Reset packets. The computation costs are $O(\log n)$ and $O(\alpha)$, respectively, and thus the total cost is $O(\log n)$ + $O(\alpha)$. The communication cost is $\alpha \times 40$ bytes; the TCP Reset packet size is 40 bytes.

$\diamond$ *Security Analysis:* The prerequisite of the data spamming attack against cellular IoT devices is to exploit the vulnerability V1 that identifies the devices by probing their non-closed TCP connections. The proposed remedy ensures that all active TCP connections will be terminated before the cellular IoT

device enters the sleep mode, thereby addressing the V1 and preventing the data spamming attack.

2) PSM-Aware Charging: The PSM-aware charging mechanism is modeled as follows. Step 1: after finding that $\mathbb{D}$ enters the sleep mode based on the expiration of the $\mathbb{D}$'s active timer, $\mathbb{C}_{MME}$ prepares a notification message, $\mathbb{M}_1$, carrying the information (e.g., IP address) about $\mathbb{D}$ and sends it to $\mathbb{D}$'s serving P-GW, $\mathbb{C}_{P-GW}$; the creation and delivery of $\mathbb{M}_1$ together take the time $T_{notifyPGW}$. Step 2: $\mathbb{C}_{P-GW}$ takes the time $T_{stopCharging}$ to process $\mathbb{M}_1$ and stop the charging of all the incoming data traffic destined to $\mathbb{D}$. Step 3: $\mathbb{C}_{P-GW}$ prepares a notification message, $\mathbb{M}_2$, carrying the IP address previously assigned to $\mathbb{D}$, and sends it to the source ISP for preventing possible Inter-AS routing fees caused by unwanted or spamming traffic for $\mathbb{D}$; the creation and delivery of $\mathbb{M}_2$ together take the time $T_{notifyISP}$. Step 4: the source ISP takes the time $T_{stopRouting}$ to process $\mathbb{M}_2$ and stop routing packets to $\mathbb{D}$.

◇ *Computation and Communication Cost:* We assume that the number of mobile devices served by $\mathbb{C}_{MME}$ or $\mathbb{C}_{P-GW}$ is n. Since the time complexity of sending/receiving a protocol message is $O(1)$, the computation cost of the proposed solution can be thus modeled as $O(\log n)$ + $O(\log n)$, which is equivalent to $O(\log n)$; The first $O(\log n)$ is the time taken by $\mathbb{C}_{MME}$ to discover the $\mathbb{D}$ information from its database, whereas the latter one is the time required by $\mathbb{C}_{P-GW}$ to search for the $\mathbb{D}$ information from its local database. For the communication, the cost for the cellular infrastructure is $O(\mathbb{M}_1 + \mathbb{M}_2)$; notably, both $\mathbb{M}_1$ and $\mathbb{M}_2$ messages are smaller than 100 bytes in our prototype.

◇ *Security Analysis:* We assume that the adversary requires additional time $\mathbb{X}$ to discover a victim device after the device enters the sleep mode. The proposed remedy needs the time, $T_{attack} = T_{notifyPGW} + T_{stopCharging} + T_{notifyISP} + T_{stopRouting}$, to disable the charging at P-GW and notify the source ISP router. Thus, the largest attack time window is calculated as $T_{attack} - X$.

Note that in our experiment, the minimum value of $\mathbb{X}$ is about 2mins, but the time needed by the remedy is only a few seconds (e.g., 1.1s). Thus, the remedy can timely take actions after each IoT device sleeps and then effectively prevent the data spamming attack. More details about our experimental results will be elaborated in Section VII-B.

3) Privacy-Aware Voice and Text Services: We next analyze only the privacy-aware voice service, since the text service one, where only error causes in the SMRP signaling responses need to be modified, does not introduce any additional computation/communication cost or new security loopholes. The voice service can be modeled as follows: When an incoming call is routed to an IoT number or an unassigned number, $\mathbb{C}_{IMSServer}$ needs to wait for an additional random delay, $\mathbb{T}$, before responding to the caller's call attempt.

◇ *Computation and Communications Cost:* The time complexity of newly introduced random delay generation is $O(1)$ [55]. No additional signaling message exchanges are required, so there is no additional communication cost for this remedy.

◇ *Security Analysis:* According to the findings from the vulnerability V3, adversaries can differentiate IoT or unassigned numbers from the others based on a clear gap between their Inv-R/S RTT times. The probability $\mathbb{P}$ of identifying them can be calculated as follows:

$$\mathbb{P} = 1 - \frac{\max(0, \min(R_{\text{N,u}}, R_{\text{I,u}} + \mathbb{L}) - \max(R_{\text{N,l}}, R_{\text{I,l}} + \mathbb{T}))}{R_{\text{I,u}} - R_{\text{I,l}}},$$

where $R_{\text{I,u}}$ and $R_{\text{I,l}}$ are the upper and lower bounds, respectively, of the Inv-R/S RTT times observed from the calls towards IoT or unassigned numbers, whereas $R_{\text{N,u}}$ and $R_{\text{N,l}}$ are those of the times observed from the calls towards the other numbers. Therefore, to completely prohibit the IoT number inference (i.e., $\mathbb{P} = 0$), the lower bound of $\mathbb{T}$ must be not smaller than $R_{\text{N,l}}$ - $R_{\text{I,l}}$. For example, from the experiment results in Section V-A, $R_{\text{N,l}}$ and $R_{\text{I,l}}$ are 1.1s and 0.6s, respectively; the lower bound of $\mathbb{T}$ can be set at least 0.5s to ensure that $\mathbb{P}$ is 0.

B. Prototype and Evaluation

We prototype and evaluate three major solution approaches, which can already mitigate the data/text spamming and the PSM manipulation attacks: the PSM-aware charging and anti-manipulation PSM from the horizontally integrated IoT security, and the privacy-aware voice service. To emulate the cellular IoT network architecture, we use srsLTE [56], Open IMS Core [57], and Twinkle 1.10.2 [58] to serve as the 4G LTE infrastructure, the IMS core with a VoLTE server, and the VoLTE client app, respectively.

PSM-aware charging. There are two major mechanisms. First, we enable the P-GW to stop packet forwarding and charging for sleeping cellular IoT devices. To achieve it, we modify the MME to send the P-GW a notification message regarding the event that an IoT device has a PSM status change as soon as the event is detected. The notification message needs to be sent through the SCEF and PCRF via the interfaces including T6a, Nt, and Gx (see Figure 1). Right after a cellular IoT device enters the sleep mode, the data spamming attack against the device cannot be prevented until the P-GW receives the notification and takes action. The damage can depend on the PSM status update interval, which is from the time of the PSM status change to the time that the P-GW takes action, so we measure it on our testbed. With 10 runs, the interval ranges from 0.9s to 1.1s. So, if the adversary cannot immediately launch the attack within 1.1s after the IoT device victim enters the sleep mode, the victim will not get any damage. Second, we modify the P-GW to notify its source router, which is built with a GNS3 [59] simulator, of the spam as unsolicited traffic through BGP (Border Gateway Protocol [60]). We send spam traffic to a cellular device through the GNS3 router and the P-GW. The traffic is generated at a rate of 10 Mbps for 30s. At the 14th second, the P-GW starts to deny the spam traffic by notifying the GNS3 router. As shown in Figure 11, all the spam packets arriving after the 14th second are discarded by the P-GW. After the 15.5th second, the P-GW does not receive any spam traffic; it means that the P-GW needs around 1.5s to notify the GNS3 router of the spam.

The above two mechanisms are deployed to protect IoT devices and carriers, respectively. They restrict the data spamming attack to be effective for them only within 1.1s and 2.6s (i.e., 1.1+1.5), respectively, right after the device victim enters the sleep mode. However, the proposed probing mechanism needs to take at least 10 seconds, which are spent on waiting for the failure of two consecutive probing messages, to identify

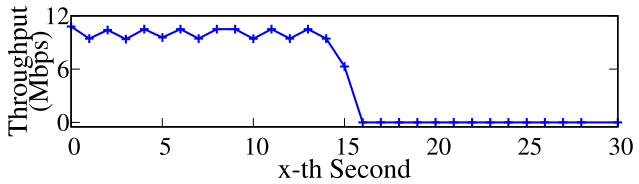
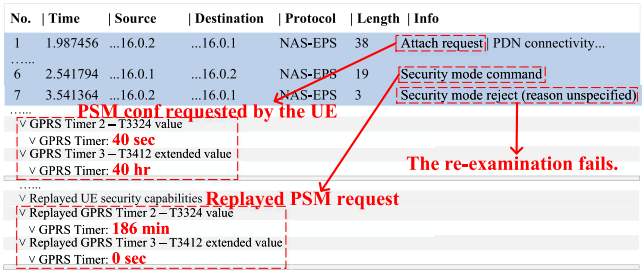
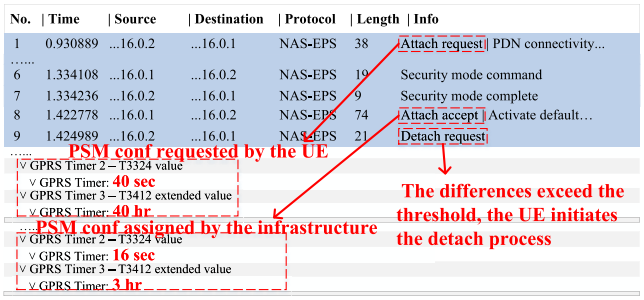


Fig. 11. Spam packets received by P-GW over time.



(a) Reject security mode command when the received PSM configuration differs from the requested one.



(b) Detach from the infrastructure when the difference between the requested and assigned PSM configurations is larger than 20%.

Fig. 12. Anti-manipulation PSM evaluation.

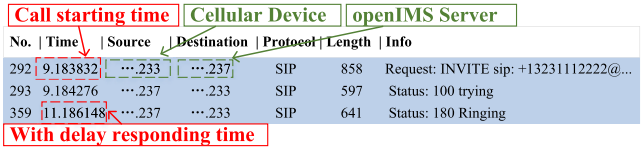


Fig. 13. Generating an additional delay of 2 seconds to Inv-R/S RTT.

an IoT IP address. It shows that the attack can be completely prevented by the proposed PSM-aware charging. *Note that* the notification delays may vary with carriers, but they shall be minimized to void the attack.

Anti-manipulation PSM. We implement this solution on the SDR-based cellular network infrastructure and device platforms, which are Sonica and srsUE, respectively; it mainly includes the confirmation of the PSM configuration from the infrastructure side and the anomaly detection at the device side. The pre-defined threshold used by the anomaly detection is set to 20%. Figure 12 confirms the effectiveness of this solution prototype.

Privacy-aware voice service. We modify the VoLTE server to add an additional delay (here, 2 seconds) to the Inv-R/S RTT for IoT numbers. Figure 13 shows that the Inv-R/S RTT for an IoT number can be successfully increased by 2 seconds. To verify whether the additional delay can eliminate V3, we run a test by considering the Inv-R/S RTT values collected from the validation experiment in Section V-A and increasing all the RTT values of IoT devices by 2 seconds. The test result shows that IoT numbers cannot be distinguished from non-IoT numbers, since the RTTs of IoT devices are overlapped with those of the other devices. Note that the additional delay X may vary with carriers due to diversified infrastructure and operations, so each carrier needs to set a proper value based on its empirical result. To be more secure, the delay value can be given dynamically so that no specific distribution can be observed for one device type.

VIII. Discussion

Reducing probing cost in CIoT-Prober: We can adopt a rate-based screening mechanism to reduce the cost of probing non-IoT devices for CIoT-Prober, where each non-IoT device needs to be probed and has at least 9m45s probing time. The mechanism lies in the existence of a clear gap between maximum downlink rates of cellular IoT and non-IoT devices. Specifically, the maximum downlink rates of the LTE-M/NB-IoT IoT devices are limited to 300/26 Kbps, whereas those of non-IoT devices with 3G UMTS and 4G LTE Advanced are 2 Mbps and 1 Gbps, respectively. When any peak downlink rate is observed for an IP address to be higher than 300 Kbps, the device owning this IP can be inferred as a non-IoT device.

Why not using current IoT search engines? Current IoT search engines may not successfully identify cellular IoT devices. Take one of the most popular engines, Shodan [61], as an example. Given a target IP address, Shodan sends various pre-defined probing messages to different TCP/UDP port numbers; it can discover which network services are available on the device and then collect information returned by each service. Based on the collected service information, Shodan identifies IoT devices based on whether any IoT device names are included or not. However, there are two major issues with this method. First, IoT device names may not be embedded in the service information; e.g., no results can be obtained by searching for three cellular IoT devices including Arduino MKR, RAK2011, and Sixfab at Shodan. Second, the service information returned by non-IoT devices or servers may also contain some IoT device names, e.g., a web server with the retail of IoT products.

Attack incentives? There are three kinds of incentives to attack cellular IoT devices. First, if the adversary's business (e.g., non-cellular IoT services) is a competitor to cellular IoT services, (s)he can launch the proposed attacks to discourage users from using them. Second, the adversary can benefit from the price drop of the carrier stock by shorting the stock in advance (before any financial losses or customer lawsuits are caused). Third, the adversary may seek to attack against cellular IoT devices with some common trait, e.g., the devices within the same geographic proximity [62].

Scalability of experimental methodology. Our experimental methodology mainly leverages commercial off-the-shelf (COTS) IoT devices and operational cellular networks to validate the discovered vulnerabilities and the devised attacks. This approach may not be very scalable due to limited access to the devices and networks. For example, not all cellular IoT devices are allowed to configure all standard-stipulated PSM timers. However, this issue can be mitigated by leveraging some SDR-based cellular IoT platforms, such as Sonica with Power Saving [63], which is an SDR-based cellular IoT infrastructure and can be used to explore the vulnerabilities of COTS IoT devices, and srsRAN UE [64], which can be revised to serve as a cellular IoT device and used to explore

the security loopholes of the CIoT infrastructure. These SDR platforms allow us to scale the experimental methodology to validate vulnerabilities in various scenarios.

IX. Related Work

Cellular IoT Security. The cellular IoT security is getting more attention recently. Tian et al. [10], [12] study the vulnerabilities of IoT charging, which can cause IoT users to be overcharged or allow adversaries to obtain inexpensive data services by spoofing IoT devices using their mobile phones. They mainly focus on critical IoT devices (i.e., CAT-1/CAT-4 IoT technology), whereas the present study considers massive IoT devices with LTE-M and NB-IoT technologies, which are used for low-cost IoT devices with only 300 Kbps and 26 Kbps maximum downlink speeds, respectively. Moreover, this work exploits the new PSM feature of massive IoT devices and the practice where the massive IoT devices are assigned phone numbers but with only text service, to launch data/text spamming attacks. These vulnerabilities do not exist in most of critical IoT devices; thus, they are not exposed by those two prior studies. Tan et al. [65] study the vulnerabilities of Layers 1 and 2 of cellular network protocols used by IoT devices and base stations, and then devise various attacks, including location privacy breach, packet delivery loop, prolonged data delivery, and compromising IoT users' transmission and privacy, whereas this present study explores the vulnerabilities of the IoT devices in terms of data, text, and PSM services. Shaik et al. [66] study the vulnerabilities of cellular IoT power saving service and propose a battery-draining attack against cellular IoT devices by removing the PSM configuration from Attach Request messages sent by the victim devices. Although the damage of this attack is similar to our battery draining attack, the prior attack, i.e., disabling power saving services, will be easier to be detected than ours since according to the IoT standard [2], [3], the PSM service is mandatory and must be supported by all cellular IoT carriers.

Non-Cellular IoT Security. There have been several works focusing on the security issues of non-cellular IoT devices, such as user authentication [67], [68], privacy leakage [69], [70], secure access control [71], [72], and companion applications [73], [74], [75], [76]. Besides, several papers [77], [78] study the recognition of IoT devices based on network traffic analysis (e.g., small TCP window size). However, these solutions cannot identify cellular IoT devices, since their network traffic patterns do not obviously differ from non-cellular IoT devices' in the cellular network. Moreover, the present study mainly focuses on design issues of IoT-related standards and the discovered issues can be applied to all the cellular IoT devices, so the device vulnerability inference based on companion applications from some studies is not needed.

X. Conclusion

Cellular IoT technologies including LTE-M and NB-IoT have been deployed worldwide to support massive IoT services. We uncover that the integration of the cellular IoT in the existing cellular network can lead to security vulnerabilities from both system-integrated and service-integrated aspects. The root cause is that the operation features of the cellular IoT differ from those of conventional non-IoT devices, but the existing functions and services which support non-IoT devices are not carefully reviewed or adapted for the cellular IoT from a security aspect.

We have validated the identified vulnerabilities and attacks with three major U.S. IoT carriers and two Taiwan IoT carriers, and shown that the security threats are not limited to particular carriers or devices. Although we have proposed quick remedies and shown their effectiveness, it still calls for an ultimate solution as a concerted effort from the standard community, carriers, and IoT device vendors. Moreover, the GSMA and 3GPP standard communities have confirmed that NB-IoT and LTE-M will coexist with 5G components in the upcoming 5G network, so the lessons learned from this work can facilitate the security of the 5G cellular ecosystem with massive IoT support.

References

[1] (2020). *Cellular IoT Market*. [Online]. Available: https://www.marketdataforecast.com/market-reports/cellular-iot-market

[2] *Clp.29: LTE-M Deployment Guide to Basic Feature Set Requirements*, GSMA, London, U.K., 2019.

[3] *Clp.28: Nb-IoT Deployment Guide to Basic Feature Set Requirements*, GSMA, London, U.K., 2019.

[4] Ericsson. (2016). *Cellular IoT Alphabet Soup*. [Online]. Available: https://www.ericsson.com/en/blog/2016/2/cellular-iot-alphabet-soup

[5] *TS 24.301: Technical Specification Group Core Network and Terminals; Non-Access-Stratum (NAS) Protocol for Evolved Packet System (EPS); Stage 2*, 3GPP, Sophia Antipolis, France, 2020.

[6] C. Peng, C.-Y. Li, H. Wang, G.-H. Tu, and S. Lu, "Real threats to your data bills: Security loopholes and defenses in mobile data charging," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2014, pp. 1–12.

[7] Y. Go, J. Won, D. F. Kune, E. Jeong, Y. Kim, and K. Park, "Gaining control of cellular traffic accounting by spurious TCP retransmission," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2014, pp. 1–15.

[8] C.-Y. Li et al., "Insecurity of voice solution VoLTE in LTE mobile networks," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 1–12.

[9] H. Kim et al., "Breaking and fixing VoLTE: Exploiting hidden data channels and mis-implementations," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 1–12.

[10] T. Xie, C.-Y. Li, J. Tang, and G.-H. Tu, "How voice service threatens cellular-connected IoT devices in the operational 4G LTE networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[11] Y. Li, K.-H. Kim, C. Vlachou, and J. Xie, "Bridging the data charging gap in the cellular edge," in *Proc. ACM SIGCOMM*, 2019, pp. 15–28.

[12] T. Xie, G.-H. Tu, C.-Y. Li, and C. Peng, "How can IoT services pose new security threats in operational cellular networks?" *IEEE Trans. Mobile Comput.*, vol. 20, no. 8, pp. 2592–2606, Aug. 2021.

[13] *Fcm.01: Volte Service Description and Implementation Guidelines V1.1*, GSMA, London, U.K., 2014.

[14] (2021). *Can I Get Unlimited Data?* [Online]. Available: https://www.xfinity.com/support/articles/exp-unlimited-data

[15] (2018). *Mobile IoT in the 5G Future—Nb-IoT and LTE-M in the Context of 5G*. [Online]. Available: https://www.gsma.com/iot/wp-content/uploads/2018/05/GSMA-5G-Mobile-IoT.pdf

[16] *Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC) Protocol Specification*, document TS 36.331, 3GPP, 2020.

[17] *Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall Description; Stage 2*, document TS 36.300, 2016.

[18] *Ir.92: IMS Profile for Voice and SMS. Version 13.0*, GSMA, London, U.K., 2019.

[19] D. Rupprecht, K. Kohls, T. Holz, and C. Pöpper, "Breaking LTE on layer two," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 1121–1136.

[20] D. Rupprecht, K. Kohls, T. Holz, and C. Poepper, "IMP4GT: IMPersonation attacks in 4G networks," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2020, pp. 893–907.

[21] M. Stute, A. Heinrich, J. Lorenz, and M. Hollick, "Disrupting Continuity of Apple's wireless ecosystem security: New tracking, DoS, and MitM attacks on iOS and macOS through Bluetooth low energy, AWDL, and Wi-Fi," in *Proc. USENIX Security*, 2021, pp. 1–19.

[22] (Oct. 26, 2016). *Understanding Physical Internet Infrastructure Vulnerabilities*. [Online]. Available: https://cip.gmu.edu/2016/10/26/understanding-physical-internet-infrastructure-vulnerabilities/

[23] H. Yang, S. Bae, and M. Son, "Hiding in plain signal: Physical signal overshadowing attack on LTE," in *Proc. USENIX Security*, 2023, pp. 1–12.

[24] (2023). *WIO LTE Cat M1/Nb-IoT Tracker*. [Online]. Available: https://wiki.seeedstudio.com/Wio_LTE_Cat_M1_NB-IoT_Tracker/

[25] (Nov. 22, 2016). *Pycom Fipy Testbed*. [Online]. Available: https://pycom.io/product/fipy/

[26] (Oct. 23, 2019). *Mangoh Yellow Testbed*. [Online]. Available: https://mangoh.io/mangoh-yellow

[27] (2023). *Sixfab CIOT Hat*. [Online]. Available: https://sixfab.com/product/arduino-lte-m-nb-iot-egprs-cellular-shield/

[28] (2021). *Arduino MKR Nb 1500 Testbed*. [Online]. Available: https://store.arduino.cc/usa/arduino-mkr-nb-1500

[29] (2023). *Telit Charlie Evaluation Kit*. [Online]. Available: https://www.telit.com/support-tools/development-evaluation-kits/charlie-evaluation-kit-for-cellular-lpwa/

[30] (2023). *Waveshare CIOT Kit*. [Online]. Available: https://www.waveshare.com/wiki/SIM7080G_Cat-M/NB-IoT_HAT

[31] (2020). *2020 IoT Developer Survey Key Findings*. [Online]. Available: https://iot.eclipse.org/community/resources/iot-surveys/assets/iot-developer-survey-2020.pdf

[32] J. Postel, *Transmission Control Protocol*, document RFC 793, Sep. 1981.

[33] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang, "Large-scale measurement and characterization of cellular machine-to-machine traffic," *IEEE/ACM Trans. Netw.*, vol. 21, no. 6, pp. 1960–1973, Dec. 2013.

[34] I Cisco Systems. (2023). *Dynamic ARP Inspection*. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/dynarp.html

[35] J. Arkko, J. Kempf, and B. Zill. (2005). *Secure Neighbor Discovery (Send)*. [Online]. Available: https://tools.ietf.org/html/rfc3971

[36] *TS 23.401: General Packet Radio Service (GPRS) Enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Access*, 3GPP, Sophia Antipolis, France, 2020.

[37] *TS 32.240: Technical Specification Group Services and System Aspects; Telecommunication Management; Charging Management; Charging Architecture and Principles*, 3GPP, Sophia Antipolis, France, 2020.

[38] *Technical Specification Group Services and System Aspects; Policy and Charging Control Architecture*, document TS 23.203, 2020.

[39] *Technical Specification Group Services and System Aspects; Telecommunication Management; Charging Management; IP Multimedia Subsystem (IMS) Charging*, document TS 32.260, 2020.

[40] *Charging Management; Charging Data Record (CDR) Parameter Description*, document TS 32.298, 2020.

[41] (2023). *The Trusted Source for IP Address Data*. [Online]. Available: https://ipinfo.io/

[42] (2020). *E.164: The International Public Telecommunication Numbering Plan*. [Online]. Available: https://www.itu.int/rec/T-REC-E.164/

[43] *Digital Cellular Telecommunication System (Phase 2); Point-to-Point (PP) Short Message Service (SMS) Support on Mobile Radio Interface (GSM 04.11)*, ETSI, Sophia Antipolis, France, 1996.

[44] *Universal Mobile Telecommunications System (UMTS); LTE; Circuit Switched (CS) Fallback in Evolved Packet System (EPS); Stage 2*, document TS 23.272, 3GPP, 2020.

[45] (2023). *Tcpdump*. [Online]. Available: https://www.tcpdump.org/

[46] *Scapy*. Accessed: 2021. [Online]. Available: https://github.com/secdev/scapy/

[47] *IP Multimedia Subsystem*, 3GPP, document TS 23.228, 2014.

[48] *Technical Specification Group Core Network and Terminals; Point-to-Point (PP) Short Message Service (SMS) Support on Mobile Radio Interface*, document TS 24.011, 2020.

[49] (2023). *Free Carrier Lookup*. [Online]. Available: https://www.freecarrierlookup.com/

[50] G.-H. Tu et al., "New security threats caused by IMS-based SMS service in 4G LTE networks," in *Proc. ACM CCS*, 2016, pp. 1–13.

[51] Y.-H. Lu et al., "Ghost calls from operational 4G call systems: IMS vulnerability, call DoS attack, and countermeasure," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw.*, Apr. 2020, pp. 1–14.

[52] (2021). *Global Cellular IoT Module Shipments Q4 2021*. [Online]. Available: https://www.counterpointresearch.com/global-cellular-iot-module-shipments-q4-2021/

[53] B. Ding, J. Zhao, Z. Tan, and S. Lu, "Sonica: An open-source NB-IoT prototyping platform," in *Proc. 27th Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2021, pp. 868–870.

[54] *Technical Report on SS7 Vulnerabilities and Mitigation Measures for Digital Financial Services Transactions*, ITU, Geneva, Switzerland, 2017.

[55] D. Carlo. (2012). *Random Number Generation: Types and Techniques*. [Online]. Available: https://digitalcommons.liberty.edu/honors/308

[56] I. Gomez-Miguelez, A. Garcia-Saavedra, P. D. Sutton, P. Serrano, C. Cano, and D. J. Leith, "SrsLTE: An open-source platform for LTE evolution and experimentation," in *Proc. 10th ACM Int. Workshop Wireless Netw. Testbeds, Experim. Eval., Characterization*, Oct. 2016, pp. 1–9.

[57] *Open IMS Core: An Open Source Implementation of IMS Call Session Control Functions*. Accessed: 2022. [Online]. Available: http://openimscore.sourceforge.net/

[58] (Feb. 25, 2009). *Twinkle*. [Online]. Available: https://mfnboer.home.xs4all.nl/twinkle/

[59] (2023). *Graphical Network Simulator-3*. [Online]. Available: https://www.gns3.com/

[60] Y. Rekhter, S. Hares, and T. Li, *A Border Gateway Protocol 4 (BGP-4)*, document RFC 4271, Jan. 2006.

[61] (2023). *Shodan Search Engine*. [Online]. Available: https://www.shodan.io/

[62] Q. Xu, J. Huang, Z. Wang, F. Qian, A. Gerber, and Z. M. Mao, "Cellular data network infrastructure characterization and implication on mobile content placement," in *Proc. ACM SIGMETRICS*, 2011, pp. 1–12.

[63] (2023). *Sonica With PSM*. [Online]. Available: http://www.cse.msu.edu/~ghtu/nets-ciot/index.html

[64] *Srsue*. Accessed: 2022. [Online]. Available: https://github.com/srsran/srsRAN_4G

[65] Z. Tan, B. Ding, J. Zhao, Y. Guo, and S. Lu, "Data-plane signaling in cellular IoT: Attacks and defense," in *Proc. ACM MOBICOM*, 2021, pp. 465–477.

[66] A. Shaik, R. Borgaonkar, S. Park, and J.-P. Seifert, "New vulnerabilities in 4G and 5G cellular access network protocols: Exposing device capabilities," in *Proc. 12th Conf. Secur. Privacy Wireless Mobile Netw.*, May 2019, pp. 221–231.

[67] R. Das, A. Gadre, S. Zhang, S. Kumar, and J. M. F. Moura, "A deep learning approach to IoT authentication," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[68] F. Restuccia, S. D'Oro, and T. Melodia, "Securing the Internet of Things in the age of machine learning and software-defined networking," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4829–4842, Dec. 2018.

[69] Z. B. Celik, E. Fernandes, E. Pauley, G. Tan, and P. McDaniel, "Program analysis of commodity IoT applications for security and privacy: Challenges and opportunities," *ACM Comput. Surveys*, vol. 52, no. 4, pp. 1–30, Jul. 2020.

[70] K. Sun, C. Chen, and X. Zhang, "Alexa, stop spying on me!': Speech privacy protection against voice assistants," in *Proc. 18th Conf. Embedded Networked Sensor Syst.*, Nov. 2020, pp. 298–311.

[71] S. Bagchi et al., "New frontiers in IoT: Networking, systems, reliability, and security challenges," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11330–11346, Dec. 2020.

[72] N. Roy et al., "Inaudible voice commands: The long-range attack and defense," in *Proc. USENIX NSDI*, 2018, pp. 1–15.

[73] E. Chatzoglou, G. Kambourakis, and C. Smiliotopoulos, "Let the cat out of the bag: Popular Android IoT apps under security scrutiny," *Sensors*, vol. 22, no. 2, p. 513, Jan. 2022.

[74] S. Neupane et al., "On the data privacy, security, and risk postures of IoT mobile companion Apps," in *Proc. DBSec*, 2022, pp. 162–182.

[75] X. Wang, Y. Sun, S. Nanda, and X. Wang, "Looking from the mirror: Evaluating IoT device security through mobile companion APPs," in *Proc. USENIX Security*, 2019, pp. 1151–1167.

[76] F. Tazi et al., "Accessibility evaluation of IoT Android mobile companion APPs," in *Proc. ACM CHI EA*, 2023, pp. 1–7.

[77] A. Sivanathan et al., "Classifying IoT devices in smart environments using network traffic characteristics," *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1745–1759, Aug. 2019.

[78] S. J. Saidi et al., "A haystack full of needles: Scalable detection of IoT devices in the wild," in *Proc. ACM IMC*, 2020, pp. 87–100.