Insecurity of Operational Cellular IoT Service: New Vulnerabilities, Attacks, and Countermeasures

Sihan Wang*, Guan-Hua Tu*, Xinyu Lei*†, Tian Xie*,
Chi-Yu Li‡, Po-Yi Chou‡, Fucheng Hsieh‡, Yiwen Hu*, Li Xiao*, Chunyi Peng△

*Michigan State University, †Michigan Technological University, ‡National Yang Ming Chiao Tung University,
△Purdue University

ABSTRACT

More than 150 cellular networks worldwide have rolled out 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 five 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 and launch data/text spamming attacks against them. We experimentally validate these vulnerabilities and attacks with three major U.S. IoT carriers. 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 and increase an IoT text bill at a rate of $5 per second; moreover, cellular IoT devices may suffer from denial of IoT services. We finally propose, prototype, and evaluate recommended solutions.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**.

KEYWORDS

Cellular IoT, Security, Service Charging

1 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) [45] and NB-IoT (Narrow Band IoT) [46]. They can extend the battery life of cellular IoT devices up to 10 years while reducing modem complexity by 70%~90%[39]. Different from other IoT technologies with data service only, cellular IoT supports not only data service but also voice and text services.

In practice, most massive cellular IoT users demand only small data volumes, 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.

We are thus motivated to study whether those new IoT-specific charging policies, together with new cellular IoT features (e.g., PSM (Power Saving Mode) [30]), may create new security issues. Although there have been many security studies of the cellular network charging [41, 51, 54, 55, 60–62, 79, 81], 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 [54]) 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., WiFi IoT devices which connect to WiFi-to-Cellular home gateways, so profiling IoT traffic may not be able to clearly differentiate cellular IoT devices from the other IoT ones. In addition to the data spamming, cellular IoT devices may also suffer the text spamming that can cause overbilling since they are charged for receiving text messages. The prerequisite of the text spamming 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 five 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

Category	Vulnerability	Type	Affected Protocols	Description	Victims
Data service	V1: Cellular IoT IP addresses can be identified remotely.	Implementation flaw	TCP [16]	Cellular IoT devices do not terminate ongoing TCP connections with Internet servers before sleeping. The TCP connections allow an adversary to probe if an IP address used by a cellular IoT device (§4.1).	Cellular IoT devices using the PSM feature.
	V2: Cellular IoT PSM-unaware charging.	Design defect	EMM [30]	The management-plane functions in the core network are unaware of the cellular IoT PSM. When an IoT device is sleeping, the data accounting and charging functions are not suspended for it (§4.2).	
Text service	V3: Leakage of phone-number device type from VoLTE signaling.	Design defect	SIP [35]	Cellular IoT inherits the function of phone number from conventional non-IoT services. When VoLTE calls are made to the IoT devices without voice service, the call response times from the VoLTE server are clearly different from those of the calls made to non-IoT devices. (§5.1).	Cellular IoT devices subscribing to text service.
	V4: Leakage of phone-number status from SMS signaling.	Operational issue	SMRP [17, 29]	The SMS signaling shows an error cause that may leak too much information about the recipient's phone number. It can be used to further infer phone numbers assigned to cellular IoT devices. (§5.2).	
	V5: Insecure pushed text service.	Operational issue	SMRP [17, 29]	Some carriers charge cellular IoT users for both incoming and outgoing text messages, but most of them do not provide users with necessary security mechanisms against incoming text spam (§5.3).	

Table 1: Summarizing the identified security vulnerabilities of operational cellular IoT services.

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 from the 3GPP standards. For the service-integrated aspect, we investigate the security of the services used by cellular IoT and then uncover three vulnerabilities: leakage of phone-number device type from VoLTE (Voice over LTE [43]) signaling (V3), leakage of phone-number status from SMS (Short Message Service) signaling (V4), and insecure pushed text service (V5). V3 is a design defect from the 3GPP standards, whereas V4 and V5 are operational issues and operator-dependent. These vulnerabilities are summarized in Table 1.

We further devise two proof-of-concept attacks, data and text spamming, 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, and increase an IoT text bill at up to a rate of $5 per second. Moreover, when the auto-renewal service is disabled, cellular IoT users would suffer from denial of IoT service after an initial service quota is exhausted. Note that the attack cost of sending data and text spam is not high, since many Internet service providers (e.g., Xfinity [23]) offer unlimited Internet data plans, and most carriers provide inexpensive unlimited text services. We finally propose a suite of solutions to address the discovered vulnerabilities and confirm their effectiveness based on a prototype and its evaluation.

This paper makes three key contributions: (1) we identify five vulnerabilities of the cellular IoT from standard design defects, network operation slips, and device implementation flaws. We validate them experimentally and analyze root causes; (2) we devise two proof-of-concept attacks by exploiting the identified vulnerabilities and assess their real-world impact on three major U.S. IoT carriers; (3) 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 while providing new insights for upcoming 5G IoT services.

2 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 [30, 34, 45, 47]. 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
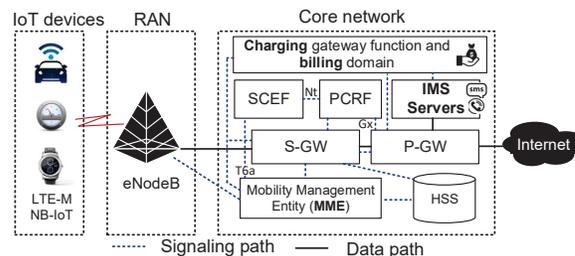
Figure 1: Cellular IoT network architecture.

are mainly supported by LTE-M and NB-IoT. We next introduce the network architecture supporting cellular IoT devices and present cellular IoT-specific functions, services, and charging policies.

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 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 [43], and text service [44]. 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. Note that LTE-M supports the data, voice, and text services, whereas NB-IoT has the data service only.

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 [45, 46], 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 [30, 34, 45, 47], which can increase the battery life of massive IoT devices. It allows an IoT device to enter a sleep mode to save

2

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), 15GB($50)	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($15), 5GB($25), 10GB($40)	Stop services	$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 an 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.

Table 2: Comparison of Non-IoT and IoT service plans for three major U.S. carriers (studied in Feb. 2021).

power; it needs to inform the MME of its desirable sleep and active time periods. By cellular IoT standards[45, 46], 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 respond to the infrastructure's `Paging` requests [34] for connection reestablishment.

Operational cellular IoT services and 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 2. 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 13~1,111 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 $0.0027~$0.0075 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.

3 THREAT MODEL, METHODOLOGY, AND ETHICAL CONSIDERATION

Threat Model. In this work, victims are cellular IoT users attacked remotely by adversaries who are organizations or people. We assume that the adversaries compromise neither cellular IoT networks nor devices. For the two presented attacks, there are different assumptions on the adversaries' capabilities.

For the *data spamming attack* presented in Section 4.3, adversaries launch a MiTM (Man in The Middle) attack [66, 67, 73, 75, 85] against a victim by sitting between the victim's cellular IoT devices and their 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
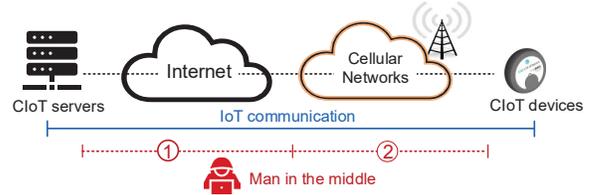
Figure 2: MiTM attacks in the threat model.

the cellular network and the IoT servers, whereas Segment 2 indicates the inside of the cellular network. In this attack, the adversary is assumed to sit somewhere on public communication channels from Segment 1, so (s)he can intercept and modify messages exchanged between IoT devices and servers, and inject messages into their communication; however, the adversary adheres to all cryptographic assumptions, e.g., encrypted messages cannot be decrypted without decryption keys. There have been several techniques exposed to achieve such a MiTM attack. For example, the adversary can leverage the DNS spoofing [24] or the ARP spoofing [21], or compromise a host at the IXP (Internet Exchange Point) [13, 14].

For the second attack presented in Section 5.4, *text spamming attack*, the adversary has full control of a rooted smartphone that has the VoLTE and text services enabled.

Experimental Methodology. We validate the vulnerabilities and attacks of cellular IoT in the networks of three major U.S. carriers, which are denoted as US-I, US-II, and US-III; they together take more than 80% of market share. We test three kinds of devices: (1) various carrier-certified cellular IoT devices, such as Wio CIoT Tracker [15], Pycom FiPy [7], and mangOH Yellow [4]; (2) non-cellular IoT devices including 2 WiFi-connected smart sockets, Geekbes YM-WS-5 and TECKIN SP10; (3) cellular non-IoT devices with four smartphones, Google Pixel 5, Apple iPhone XS MAX, and Samsung S5/S10. We connect them to operational cellular networks in the experiments. Note that the names of those three carriers were not revealed since the discovered vulnerabilities had not been fully addressed[1] while the camera-ready version was prepared.

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

---

[1]We had reported the vulnerabilities to the relevant cellular IoT operators and provided them with recommended solutions. US-II had confirmed the vulnerabilities and been working on the development of remedies, whereas US-I had acknowledged the vulnerabilities and been investigating them; US-III had not yet responded.

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.

4 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 2). 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 WiFi IoT devices that connect to the cellular network through WiFi-to-Cellular home gateways. Such mixed usage scenario including both cellular and WiFi 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 PSM feature, 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 of the vulnerabilities with experimental validation and then present the spamming attack.

4.1 V1: Cellular IoT IP Addresses can be Identified Remotely

We can identify the IP addresses used by massive cellular IoT devices by probing whether they have the PSM (TS24.301 [30], CLP.28 [47], TS36.331 [34]) 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 a MiTM attack, the adversary can observe the traffic coming from an IP address and interact with its device 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 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 from the device, the probability can be small.

To this end, we develop a probing mechanism based on the cellular IoT PSM, designated as CIoT-Prober. The major idea is that it sends multiple probes to a given IP address at different times and makes sure that at least one probe proceeds while the device is sleeping if it is a PSM-enabled IoT device. When one failed probe (i.e., no responses are received) can be observed for each PSM-enabled IoT device due to its sleep and successful probes are always obtained from the other devices, the PSM-enabled IoT device can be successfully identified. Note that a probe may contain multiple probing messages to cover packet loss cases. The reason why multiple probes are used is that it is unknown whether the probed device can go to sleep and when it is sleeping if it can.

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 and determine if a probed device has the PSM based on a failed probe. According to our observation on all the cellular IoT devices with us, 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 [5] and HTTP [40]) in practice; a recent study [19] 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 active 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 [16]. 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 [45, 46], 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 2). 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.

4

Predicted class / Actual 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					Cellular IoT without PSM		Non-cellular IoT Devices		Non-IoT Devices			
Arduino MKR	Botletics SIM7000	RAKWireless RAK2011	Sixfab CIoT HAT	Pycom FiPy	Wio CIoT Tracker	MangoH Yellow	Geekbes YM-WS-5	TECKIN SP10	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

Table 3: Confusion matrix of the classification of PSM-enabled cellular IoT devices based on CIoT-Prober.
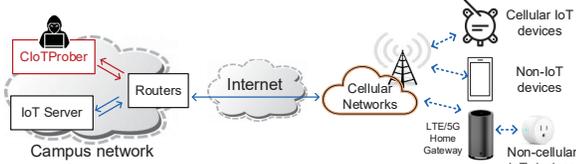
Figure 3: CIoT-Prober validation experiment settings.

In practice, carriers may set their specific constraints on the minimum active time; the value of 16 seconds is observed from AT&T, Verizon, and T-Mobile. Also, most device vendors restrict active times for longer battery life; specifically, 80% of massive cellular IoT devices [69] are with average active times less than 5 minutes. 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 16 seconds and 5 minutes.

4.1.1 Validation. We experimentally validate the effectiveness of CIoT-Prober by examining whether it can successfully identify IP addresses used by cellular IoT devices. We conduct the experiment using 13 test devices in our campus network: 7 cellular IoT devices including Arduino MKR NB 1500, Botletics SIM7000, RAKWireless RAK2011, Sixfab CIoT HAT, Pycom Fipy, Wio CIoT Tracker, and MangoH Yellow; 2 WiFi-connected smart sockets including Geekbes YM-WS-5 and TECKIN SP10; 4 smartphones (i.e., non-IoT devices) including Samsung S5/S10, Google Pixel 5, and Apple iPhone XS. 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.

Figure 3 shows the network topology of the validation experiment. We deploy CIoT-Prober 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. Once observing a new TCP connection coming from cellular networks, CIoT-Prober sends 6 probing messages with intervals, 15s, 30s, 60s, 180s, and 300s to the connection's source IP, and does IP spoofing in the probing messages. To cover packet loss cases, each probing message is retransmitted once if no response is observed within 5s after its initial transmission. When no responses are received for the probing message, the probed IP address is identified to be used by a PSM-enabled cellular IoT device.

Experimental result. Table 3 summarizes the experimental results, and we make two observations. First, CIoT-Prober can accurately 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., taking 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 a probing message once the probability of the false positive cases can be greatly reduced. Moreover, the impact of the false positive cases is very lightweight on attack cost. The reason is that launching a spamming attack against each IP address identified as being used by a cellular IoT device 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, since the probing of one device stops whenever the device is identified; the PSM-enabled cellular IoT devices take much shorter probing times than the other devices do. Specifically, 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 former devices can be identified once a probe occurs while they are sleeping, but probing the latter devices 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.

4.1.2 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.

4.1.3 Rate-based screening: reducing probing cost. We further 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, its device can be inferred as a non-IoT device.

We conduct an experiment to examine whether the rate-based screening can work. We test the peak downlink rates of two non-IoT devices including Samsung S5/S10 in various scenarios: two network types (3G/4G), four signal conditions (near deadzone and

(a) Near deadzone: 3G (-110 < RSSI ≤ -105), 4G (-140 < RSRP ≤ -119)

(b) Poor signal: 3G (-105 < RSSI ≤ -99), 4G (-119 < RSRP ≤ -110)

(c) Fair signal: 3G (-98 < RSSI ≤ -88), 4G (-110 < RSRP ≤ -98)

(d) Good signal: 3G (-88 ≤ RSSI), 4G (-98 ≤ RSRP)
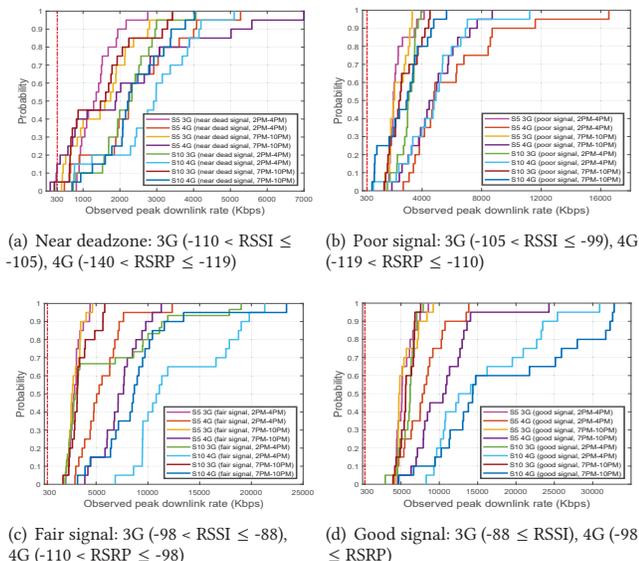
Figure 4: The CDF of the peak downlink rates observed on non-IoT devices with various signal conditions (RSSI and RSRP in dBm), networks, and times.

poor/fair/good signals), and two rush hours (2pm-4pm and 7pm-10pm [78]). We use IPerf to generate downlink traffic to the devices. The experiment lasts for 5 days with a total of 640 measurement data sets, as shown in Figure 4. It is observed that only 5 measurement data sets, which take only 0.78% of the total, have peak downlink rates lower than 300 Kbps; they are collected in the near-deadzone case. It shows that non-IoT devices can indeed be identified in most cases based on the rate-based screening of peak downlink rates.

CIoT-Prober with Rate-based Screening. We next integrate the rate-based screening into CIoT-Prober to reduce the cost of probing non-IoT devices. CIoT-Prober will record the downlink traffic statistics for each IP address whenever any downlink traffic is observed, and keep calculating downlink rates over time. Specifically, CIoT-Prober logs all the times when it observes that each packet is sent and its ACK is received. For each packet, we can calculate its downlink rate by dividing its size by the time interval between its leaving time and its ACK's arrival time. For time points in the past when there were no unacknowledged packets, we can accumulate downlink rates of all the packets which were traveling; the highest accumulated rate over time is used as the peak downlink rate.

We redo the experiment in Section 4.1.1 to examine the effectiveness of the rate-based screening. We enable the test application of IoT and non-IoT devices to upload an amount of 58 KB data and download a small amount of 9 KB data after connecting to the IoT server. For the IoT devices, the data transfer actions are used to emulate the IoT device initialization with its server. For the non-IoT devices, we use those actions to emulate a use scenario that the IoT application on a smartphone communicates with the IoT server and further accesses an IoT device. Note that the used data amounts are obtained from a public IoT traffic dataset [59]; the data amounts generated by the non-IoT devices are actually larger than 58 KB and 9 KB, but the effectiveness of the rate-based screening can be shown even for those smaller data amounts.

The result shows that with 4 different phone models and 30 test cases each, CIoT-Prober can successfully identify non-IoT devices

(a) Established connections/bearers for an active UE.

(b) Alive connections/bearers for an inactive UE.

(c) Data service reactivation procedure for an inactive UE.
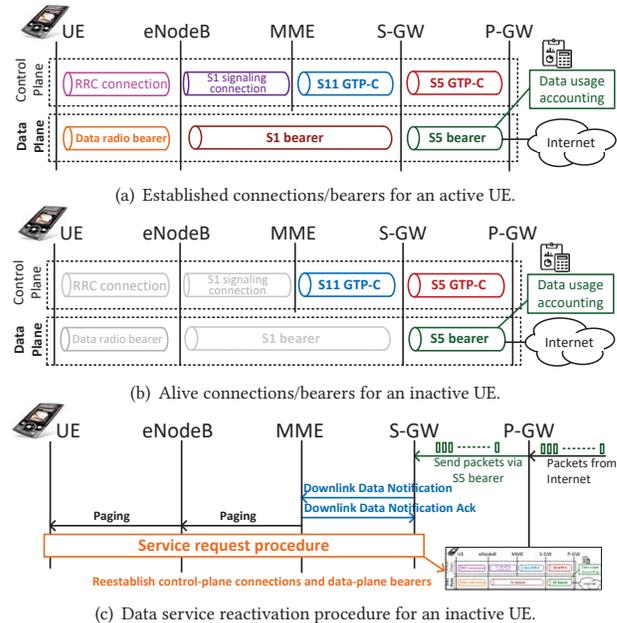
Figure 5: Established control-plane connections and data-plane bearers for active and inactive UEs, and the data service reactivation procedure of the inactive UE.

with accuracy higher than 98.33% (118/120) using the rate-based screening; there are 2 false negative cases but no false positive cases. Moreover, the rate-based screening takes only 1 minute, which is currently the recording unit of traffic statistics, for all the devices. It not only reduces the time of identifying non-IoT devices by more than 89.74% saving, but also excludes a large number of non-IoT devices from the probing pool for CIoT-Prober. Note that although there are false negative cases due to bad channel quality, their non-IoT devices can still be identified by the PSM probing mechanism.

4.2 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 *sleep* mode of cellular IoT devices (see details in Section 2). 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), i.e., cellular device, has several established control-plane connections and data-plane bearers, as shown in Figure 5(a); it can become inactive due to no signaling or data traffic for a while, and then some control-plane connections and data-plane bearers are temporarily released, as shown in Figure 5(b). When any data traffic sent to the UE reaches the P-GW/S-GW as shown in Figure 5(c), the MME is notified and then sends a `Paging` message [34] to notify the UE; afterwards, the UE performs the service request procedure [28] to reestablish the released connections and bearers.

Once the core network treats a sleeping IoT device as the same 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 S5 bearer and forward the data usage to the charging gateway function [31], as shown in Figure 5(b). This operation does not have any issues with inactive non-IoT devices, which can be notified to receive the data, but it can cause charging issues to sleeping IoT devices; that is, the sleeping IoT devices are charged for the incoming downlink data yet without receiving them. Moreover, if the incoming downlink packets are spam, the sleeping IoT devices cannot take any immediate defense manner against the spam. Note that the cellular IoT standards [26, 31] do not stipulate that the P-GW shall suspend the charging function for sleeping IoT devices.

4.2.1 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 three carriers including US-I, US-II, and US-III with TCP traffic. The experiment consists of four steps for each carrier. First, we keep a test IoT device power off for three days and then obtain its latest data usage amount from its subscribed carrier. Second, we power on the IoT device, connect it to the carrier network, and enable its PSM. We configure the lengths of the PSM active and sleep time periods to the minimum values allowed by the carrier, e.g., they are 16 seconds and 3 hours, respectively, for both US-I and US-II. Third, the IoT server is configured with the PSM time values and sends 100 KB data to the IoT device while it is sleeping. After the device wakes up, we keep it on for 30 minutes and then power it off. Lastly, we wait for three days and then check the test device's latest data usage.

Experimental result. We have two observations: (1) the IoT devices tested in the networks of those three carriers do not receive any packets; and (2) all the test devices are charged for the 100 KB data. The result confirms that the charging function is unaware of the cellular IoT PSM, and is not suspended for it.

4.2.2 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 [30]) 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 [26, 31–33] 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 traffic 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.

4.3 Proof-of-concept Attack

We devise a spamming attack against cellular IoT devices using vulnerabilities V1 and V2, and then evaluate its damage. To launch the attack, the adversary uses a MiTM attack to sit between cellular IoT devices and their IoT servers, as the threat model described in Section 3. Although there is a large amount of IP addresses which the adversary can see from the eavesdropping, (s)he does not probe all the IP addresses but only the ones 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
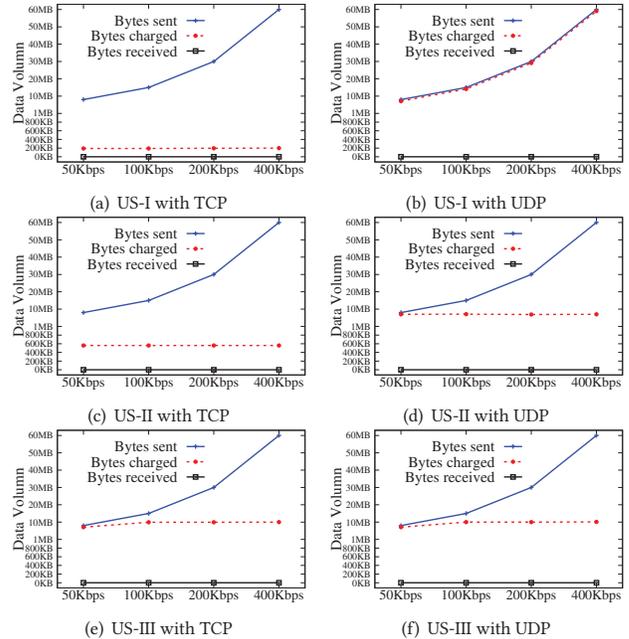
Figure 6: Under an IoT spamming attack, the spam traffic volume is sent, charged, and received for three carriers with TCP and UDP traffic cases.

some free online databases [11], then probe those IP addresses only 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 three carriers including US-I, US-II, and US-III 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 including Samsung Galaxy S5/S10, Pixel 5, and iPhone XS. We deploy an IoT server and a laptop with the CIoT-Prober module in our campus network. The CIoT-Prober launches an ARP spoofing attack to intercept all the traffic of the IoT server. To start the experiment for each carrier, the test 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 if there is any. Each spamming attack lasts for 20 minutes.

Experimental result. Figure 6 shows the spam traffic volume sent, charged, and received for each carrier with TCP and UDP traffic cases. We have four findings. First, CIoT-Prober can successfully identify those two cellular IoT devices and no false positive cases are observed. Second, for all the cases as shown in Figure 6, 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 the device is sleeping; then, when the spamming attack is launched right after the identification result, the sleeping device cannot receive any spam traffic. Third, for the TCP results shown in Figures 6(a), 6(c), and 6(e), US-I and US-II impose charging volume caps, 200 KB and 540 KB, respectively, but US-III has a higher cap with 9.8 MB. Fourth, for the UDP spam results shown in Figures 6(b), 6(d), and 6(f), US-I does not impose any charging volume

INVOICE SUMMARY
Jun 06, 2020 - Invoice #I34832 PDF XLS

T125452	Data - 5 MB/month for $6.00/month $6 Monthly Per Active Line (You had 1 active line(s))	BILL	$ 6.00
T125453	"Data - 5 MB/month for $6.00/month" $2 Per MB Over 5MB based on total line count of 1 lines with usage of 118 MB (actual usage volume was 117.07949161529541 MB)	BILL	$ 226.00
		INVOICE TOTAL	**$ 232.00**

Figure 7: Excess bills caused by the IoT spamming attack.

caps, which can achieve up to 60 MB; US-II and US-III have charging volume caps about 6~7 MB and 9~10 MB, respectively.

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 service for their IoT devices; this service helps users to automatically purchase more data quota when it is exhausted. Figure 7 shows that 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.

Seemingly, the data spamming attack can be easily defended by an upper threshold of charging volumes (e.g., 200 KB) at the P-GW for sleeping IoT devices. However, the sleeping IoT devices can still suffer, though the charging amount of data spam is small or grows slowly under multiple attacks. Moreover, without notifying source ISPs of unwanted traffic, carriers still need to pay them the Inter-AS packet routing fees [18] for data spam.

5 INSECURE CELLULAR IOT TEXT SERVICE

As non-IoT devices, the text service is one essential service for cellular IoT devices; an IoT device can also 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 [20] (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 whether 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 (Short Message Service) signaling (e.g., SM-RP-ERROR and SM-RP-DATA [17]) 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
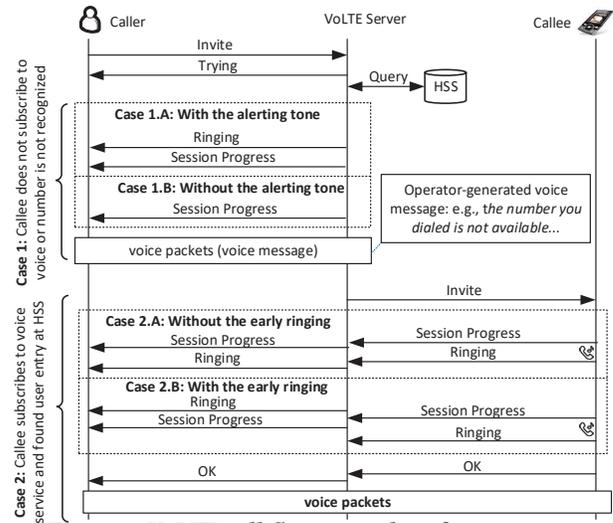
Figure 8: VoLTE call flow procedure for two cases.

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.

5.1 V3: Leakage of Phone-Number Device Type from VoLTE Signaling

Most IoT numbers have only the text service but do not subscribe to the voice service, which non-IoT numbers always have; 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 the 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 the voice service.

Calling the numbers with and without the voice service can lead to two different cases of call initialization procedure. Figure 8 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 the 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, where the callee has a non-IoT number, the HSS can discover the callee's subscription data. The VoLTE server then forwards the SIP INVITE to the callee and two cases of call procedure
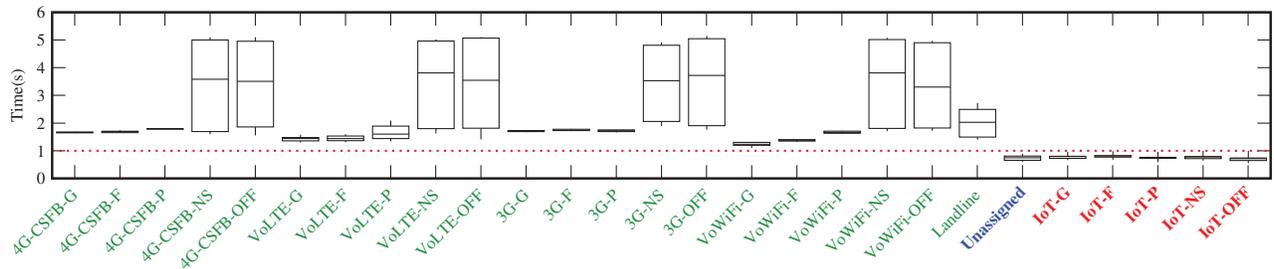
Figure 9: The Inv-R/S RTT values in quartiles, median, maximum, and minimum are observed at the VoLTE caller for each callee type in various scenarios: IoT (Red), non-IoT (Green), and unassigned numbers (Blue); signal strength cases: good (G), fair (F), poor (P), and no-signal (NS); device statuses: power-on by default and power-off (OFF).

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 the voice service or an unassigned number, much sooner than 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 the same carrier US-II in the experiment since US-II is the only carrier supporting the text service for cellular IoT in our area. 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. We use a rooted smartphone, Samsung Galaxy S10, as the caller to dial VoLTE calls to those 9 numbers. To examine possible variance of the vulnerability, we consider the callees in various scenarios with different voice technologies including 3G (Circuit-Switched) [49], 4G CSFB [27], 4G VoLTE, and 4G VoW-iFi [48], different signal strengths, and power on/off statuses. The experiment of making a VoLTE call in each scenario for each number is run for 20 times. In the experiment, we use the Tcpdump software [10] to collect all the signaling messages, and develop a Python program with the Scapy [8] library to analyze the messages. From each call trace, we can collect the `Inv-R/S` RTT (Round Trip Time), which is the time period between the leaving `INVITE` and the arrival `RINGING` or `SESSION PROGRESS`, to be used as the call response time.

Figure 9 plots the `Inv-R/S` RTT values in quartiles, median, minimum, and maximum for each device type in various scenarios. We make two observations. First, the `Inv-R/S` 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 `Inv-R/S` 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, the `Inv-R/S`

RTT values 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, since this vulnerability roots in the core network functions; thus, it can be generally applied to all the IoT devices.

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 [25, 44], 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 by adding some randomness to the delivery of its responses.

5.2 V4: Leakage of Phone-Number Status from SMS Signaling

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 [17]) 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 of the text messages sent to them. SMRP is a protocol used to transmit text messages to the SMSC through the IMS servers; all its messages are encapsulated by the SIP.

Validation. We validate this vulnerability by sending a text message to an IoT number and an unassigned number through the IMS using a rooted smartphone, and then analyzing each message's delivery status. Figure 10 shows the SMRP signaling responses which the smartphone receives from the IMS server after sending the text message to those two numbers. The `SM-RP-DATA` message with a delivery report shows "delivered" for the IoT number, whereas the `SM-RP-ERROR` indicates an error [29] with a cause "Requested facility not implemented (69)" for the unassigned number.

Root cause and lesson. The SMS standard [29] 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,

9

No.	Time	Source	Destination	Protocol	Length	Info
6	6.381539	2001:4888:...	2600:1007:...	GSM SMS	757	Request: Message sip... RP-DATA...

...... Operator Server Cellular Device
∨ Session Initiation Protocol (Message) → Message Type: DATA
......
> GSM A-I/F RP - RP - DATA (Network to MS)
∨ GSM SMS TPDU (GSM 03.40) SMS-STATUS REPORT
  ......
> TP-User-Data
    SMS text: Message to 949312**** delivered. ← Delivered!

(a) The text recipient with an IoT number.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.356131	2001:4888:...	2600:1007:...	SIP	771	Request: Message sip... RP-ERROR...

...... Operator Server Cellular Device
∨ Session Initiation Protocol (Message) → Message Type: ERROR
......
∨ GSM A-I/F RP - RP - ERROR (Network to MS)
  Message Type RP-ERROR (Network to MS)
  ......
> RP-Cause – (69) Requested facility not implemented
    Length: 1
    0... .... = Extension: No extension
    .100 0101 = Cause: Requested facility not implemented (69) ← Fail to send!

(b) The text recipient with an unassigned number.

Figure 10: The SMRP signaling responses received by the text sender vary with different recipients.

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.

5.3 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 2). 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 provided by 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.

5.4 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 [2]. For each phone number, the attack first checks whether it is an IoT number based on V3 and V4; if yes, many spam text messages are sent to the number. We develop two programs for this attack on Android phones: (1) `IoTNumProber`, which checks if a given phone number is an IoT

number; (2) `TextSpamSender`, which generates a large number of spam text messages to the given IoT number within a short time interval by exploiting the reported SMS vulnerabilities [77].

We evaluate the attack by using the same list of phone numbers as the validation experiment presented in Section 5.1, 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 the IoT number is successfully identified, all spam text messages are received by the IoT device victim, 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 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 large-scale number examination yet; several approaches can be adopted to further improve the performance (e.g., dialing multiple probing calls simultaneously [57]).

Note that the insecure cellular IoT text service can be more threatening and far-reaching than the vulnerable data service of cellular IoT. The IoT numbers used for the text service remain unchanged, but the IP addresses of the data service usually change over time. Once an IoT number is identified, the text spamming attack against the number can last for a long time.

6 SOLUTION

In this section, we propose a suite of solution approaches to address the above five 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 coordination mechanism that vertically crosses the transport/application layers and the underlying non-access stratum layer (e.g., EMM and ESM [30]) 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 a horizontal security manner to address vulnerability V2, cellular IoT PSM-unaware charging. It is a collaboration mechanism that horizontally spans network elements (e.g., MME and P-GW) and IoT devices. It 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 [30] 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

(a) Preventing incoming packets for sleeping cellular IoT devices.

(b) Spam packets received by P-GW over time while they are blocked.
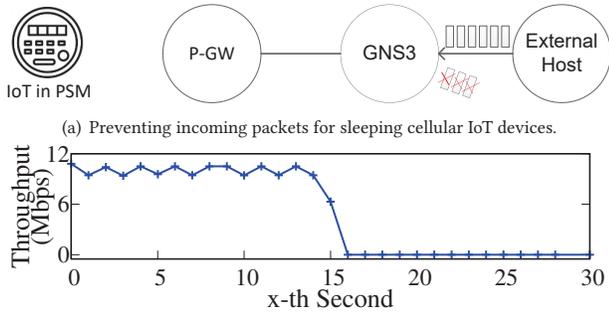
Figure 11: PSM-aware charging method.

	Call starting time	Cellular Device	openIMS Server	No delay responding time		
No.	Time	Source	Destination	Protocol	Length	Info
61	4.008433	233	237	SIP	858	Request: INVITE sip: +13231112222@...
62	4.009423	237	233	SIP	597	Status: 100 trying
63	4.017729	237	233	SIP	641	Status: 180 Ringing

(a) No additional delay.

	Call starting time	Cellular Device	openIMS Server	With delay responding time		
No.	Time	Source	Destination	Protocol	Length	Info
292	9.183832	233	237	SIP	858	Request: INVITE sip: +13231112222@...
293	9.184276	237	233	SIP	597	Status: 100 trying
359	11.186148	237	233	SIP	641	Status: 180 Ringing

(b) A delay of 2 seconds.

Figure 12: Generating a delay to `Inv-R/S` RTT.

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 [22]) packet routing fees [18].

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 (i.e., VoLTE), we propose to add a small random delay to the message responses (e.g., `Ringing` and `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) [29]) 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 restrictions: one is to allow only pre-approved numbers to send text messages to a certain IoT number, whereas the other is to restrict 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 [50] to address the disclosed vulnerabilities of SS7 or upgrade the SS7-involved text service to the IMS-based SMS [44].

6.1 Prototype and Evaluation

We prototype and evaluate two major solution approaches, which can already mitigate the data/text spamming attacks: the PSM-aware charging from the horizontally integrated IoT security and the privacy-aware voice service. To emulate the cellular IoT network architecture, we use srsLTE [42], Open IMS Core [6], and Twinkle 1.10.2 [12] to serve as the 4G LTE infrastructure, the IMS core with a VoLTE server, and the VoLTE client app, respectively. We next elaborate on these two solution approaches.

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
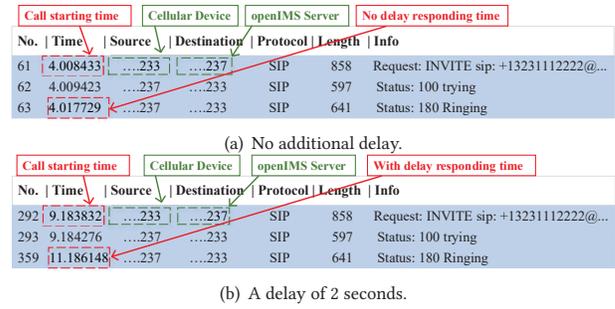
PCRF via the interfaces including T6a, Nt, and Gx (see Figure 1). Right after a cellular IoT device enters the sleep mode, the data spamming attack against the device cannot be prevented until the P-GW receives the notification and takes action. The damage can depend on the PSM status update interval, 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 [3] simulator, of the spam as unsolicited traffic through BGP (Border Gateway Protocol [63]), as shown in Figure 11(a). We send spam traffic to a cellular device through the GNS3 router (i.e., the source router of the P-GW) 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(b), 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 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 as much as possible.

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 12 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 5.1 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

11

secure, the delay value can be given dynamically so that no specific distribution can be observed for one device type.

7 DISCUSSION

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 [9], 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 by the proposed attacks). Third, the adversary may seek to attack against cellular IoT devices with some common trait, e.g., the devices within the same geographic proximity [82].

8 RELATED WORK

Cellular Network Security. The research studies about cellular network security can be classified into three categories. First, some research works examine security issues of the cellular control-plane, such as a signaling injection attack [84], persistent handoff loops [56], rogue base stations [71], and spoofed urgent alerts [52]. Second, several works study the insecurity of essential cellular services. Li *et al.* [54], Xie *et al.* [80], and Tu *et al.* [77] study the insecurity of VoLTE, VoWiFi, and SMS services, respectively. Third, many of them investigate security threats related to service charging models/policies. Kim *et al.* [51] point out that the user who controls application processor can potentially exploit the call setup process to cause DoS and over-billing attacks. Peng *et al.* [60–62] show that both carriers and users may suffer from charging-based attacks; users can take the advantage of carriers to receive free data services and suffer from various spamming attacks. Different from them, the present study targets the insecurity of newly deployed cellular IoT service, which has not been fully explored yet.

Cellular IoT Security. The cellular IoT security is getting more attention recently. Some works [79, 81] study the vulnerabilities of IoT charging, which can cause IoT users to be overcharged and lead to loss of profit for operators. These two studies mainly focus on CAT-1/CAT-4 cellular IoT devices (i.e., critical IoT devices), which can still support high transmission rates with 10~150 Mbps downlink and 5~50 Mbps uplink speeds, 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. The present study mainly 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 remotely identify their IP addresses and numbers, respectively. These vulnerabilities do not exist in most of critical IoT devices; thus, they are not exposed by those two prior studies. Yang *et al.* [83] develop a hardware NB-IoT diagnostic tool to examine undisclosed cellular IoT operations and assess their power consumption impact on NB-IoT devices.

Non-Cellular IoT Security. There are several works focusing on the security issues of IoT devices, such as user authentication [38, 64, 72], privacy leakage [37, 53, 76], and secure access control [36, 65]. Besides, several papers [58, 59, 68, 70, 74] study the recognition of IoT devices from network traffic analysis (e.g., small TCP window size). However, these studies cannot be simply applied to identifying cellular IoT devices, since they share similar traffic patterns with other IoT devices (e.g., Wi-Fi, Bluetooth, and LoRa IoT) that connect to cellular networks through a gateway. We then use the essential PSM feature of cellular IoT to remotely distinguish cellular IoT devices from other non-cellular IoT devices.

9 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 shown that the security threats are not limited to particular carriers or devices. Although we have proposed quick remedies and shown their effectiveness on mitigation of the spamming attacks, the ultimate solution requires a concerted effort from the standard community, carriers, and IoT device vendors.

The discovered security issues are not short-living, though they come from the cellular IoT integration in the 4G network. Such IoT integration will happen again in the 5G network, since the GSMA and 3GPP standard communities have confirmed that NB-IoT and LTE-M will coexist with 5G components in the upcoming 5G network. Without lessons learned for the need of prudent integration from both system and service points of view, the similar security issues will threaten the 5G cellular ecosystem in the near future.

10 ACKNOWLEDGMENTS

12

REFERENCES

[1] Cellular iot market. https://www.marketdataforecast.com/market-reports/cellular-iot-market.

[2] Free carrier lookup service. https://www.freecarrierlookup.com/.

[3] Graphical network simulator-3. https://www.gns3.com/.

[4] mangoh yellow testbed. https://mangoh.io/mangoh-yellow.

[5] Mqtt (message queuing telemetry transport). https://mqtt.org/.

[6] Open ims core: an open source implementation of ims call session control functions. http://openimscore.sourceforge.net/.

[7] Pycom fipy testbed. https://pycom.io/product/fipy/.

[8] Scapy. https://github.com/secdev/scapy/.

[9] Shodan search engine. https://www.shodan.io/.

[10] Tcpdump. https://www.tcpdump.org/.

[11] The trusted source for ip address data. https://ipinfo.io/.

[12] Twinkle: a softphone for voip and instant messaging communications using the sip protocol. https://mfnboer.home.xs4all.nl/twinkle/.

[13] Understanding physical internet infrastructure vulnerabilities. https://cip.gmu.edu/2016/10/26/understanding-physical-internet-infrastructure-vulnerabilities/.

[14] What is an internet exchange point (ixp)? https://blog.stackpath.com/internet-exchange-point/.

[15] Wio lte cat m1/nb-iot tracker. https://wiki.seeedstudio.com/Wio_LTE_Cat_M1_NB-IoT_Tracker/.

[16] Transmission Control Protocol. RFC 793, Sept. 1981.

[17] Digital cellular telecommunication system (phase 2); point-to-point (pp) short message service (sms) support on mobile radio interface (gsm 04.11). https://www.etsi.org/deliver/etsi_gts/04/0411/05.01.00_60/gsmts_0411v050100p.pdf, 1996.

[18] The value of peering. https://www.menog.org/presentations/menog-6-7-8-9/Value-Of-Peering.pdf, 2010.

[19] 2020 iot developer survey key findings. https://iot.eclipse.org/community/resources/iot-surveys/assets/iot-developer-survey-2020.pdf, 2020.

[20] E.164: The international public telecommunication numbering plan. https://www.itu.int/rec/T-REC-E.164/, 2020.

[21] Arp spoofing. https://www.veracode.com/security/arp-spoofing, 2021.

[22] Autonomous system. https://en.wikipedia.org/wiki/Autonomous_system_(Internet), 2021.

[23] Can i get unlimited data? https://www.xfinity.com/support/articles/exp-unlimited-data, 2021.

[24] Dns spoofing tool. https://www.charlesproxy.com/documentation/tools/dns-spoofing/, 2021.

[25] 3GPP. IP Multimedia Subsystem. Technical Specification (TS) 23.228, 3rd Generation Partnership Project (3GPP), 04 2014. Version 12.4.0.

[26] 3GPP. TS 23.203: Technical Specification Group Services and System Aspects; Policy and charging control architecture, 2020.

[27] 3GPP. TS 23.272: Universal Mobile Telecommunications System (UMTS); LTE; Circuit Switched (CS) fallback in Evolved Packet System (EPS); Stage 2, 2020.

[28] 3GPP. TS 23.401: General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access, 2020.

[29] 3GPP. TS 24.011: Technical Specification Group Core Network and Terminals; Point-to-Point (PP) Short Message Service (SMS) support on mobile radio interface, 2020.

[30] 3GPP. TS 24.301: Technical Specification Group Core Network and Terminals; Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 2, 2020.

[31] 3GPP. TS 32.240: Technical Specification Group Services and System Aspects; Telecommunication management; Charging management; Charging architecture and principles, 2020.

[32] 3GPP. TS 32.260: Technical Specification Group Services and System Aspects; Telecommunication management; Charging management; IP Multimedia Subsystem (IMS) charging, 2020.

[33] 3GPP. TS 32.298: Telecommunication management; Charging management; Charging Data Record (CDR) parameter description, 2020.

[34] 3GPP. TS 36.331: Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification, 2020.

[35] 3GPP. TS 24.229: IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3, 2021.

[36] Bagchi, S., Abdelzaher, T. F., Govindan, R., Shenoy, P., Atrey, A., Ghosh, P., and Xu, R. New frontiers in iot: Networking, systems, reliability, and security challenges. *IEEE Internet of Things Journal 7*, 12 (2020), 11330–11346.

[37] Celik, Z. B., Fernandes, E., Pauley, E., Tan, G., and McDaniel, P. Program analysis of commodity iot applications for security and privacy: Challenges and opportunities. *ACM Computing Surveys (CSUR) 52*, 4 (2019), 1–30.

[38] Das, R., Gadre, A., Zhang, S., Kumar, S., and Moura, J. M. A deep learning approach to iot authentication. In *2018 IEEE International Conference on Communications (ICC)* (2018), IEEE, pp. 1–6.

[39] Ericsson. Cellular iot alphabet soup. https://www.ericsson.com/en/blog/2016/2/cellular-iot-alphabet-soup, 2016.

[40] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T. Rfc 2616, hypertext transfer protocol – http/1.1, 1999.

[41] Go, Y., Jeong, E., Won, J., Kim, Y., Kune, D. F., and Park, K. Gaining control of cellular traffic accounting by spurious TCP retransmission. In *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014* (2014), The Internet Society.

[42] Gomez-Miguelez, I., Garcia-Saavedra, A., Sutton, P. D., Serrano, P., Cano, C., and Leith, D. J. srslte: An open-source platform for lte evolution and experimentation. In *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization* (2016), pp. 25–32.

[43] GSMA. Volte service description and implementation guidelines. https://www.gsma.com/futurenetworks/wp-content/uploads/2014/05/FCM.01-v1.1.pdf, 2014.

[44] GSMA. Ims profile for voice and sms. version 13.0. https://www.gsma.com/newsroom/wp-content/uploads//IR.92-v13.0-2-1.pdf, 2019.

[45] GSMA. Lte-m deployment guide to basic feature set requirements. https://www.gsma.com/iot/wp-content/uploads/2019/08/201906-GSMA-LTE-M-Deployment-Guide-v3.pdf, 2019.

[46] GSMA. Nb-iot deployment guide to basic feature set requirements. https://www.gsma.com/iot/wp-content/uploads/2019/07/201906-GSMA-NB-IoT-Deployment-Guide-v3.pdf, 2019.

[47] GSMA. Nb-iot deployment guide to basic feature set requirements. https://www.gsma.com/newsroom/wp-content/uploads/CLP.28v1.0.pdf, 2019.

[48] GSMA. Ims profile for voice, video and sms over untrusted wi-fi access; version 8.0. https://www.gsma.com/newsroom/wp-content/uploads//IR.51-v8.0.pdf, 2020.

[49] Holma, H., and Toskala, A. *WCDMA for UMTS: HSPA Evolution and LTE.* John Wiley & Sons, Inc., USA, 2007.

[50] ITU. Technical report on ss7 vulnerabilities and mitigation measures for digital financial services transactions. https://www.itu.int/en/ITU-T/extcoop/figisymposium/Documents/ITU_SIT_WG_Technical%20report%20on%20the%20SS7%20vulnerabilities%20and%20their%20impact%20on%20DFS%20transactions_f.pdf, 2017.

[51] Kim, H., Kim, D., Kwon, M., Han, H., Jang, Y., Han, D., Kim, T., and Kim, Y. Breaking and fixing volte: Exploiting hidden data channels and mis-implementations. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (2015), pp. 328–339.

[52] Lee, G., Lee, J., Lee, J., Im, Y., Hollingsworth, M., Wustrow, E., Grunwald, D., and Ha, S. This is your president speaking: Spoofing alerts in 4g lte networks. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services* (New York, NY, USA, 2019), MobiSys '19, Association for Computing Machinery, p. 404–416.

[53] Leu, P., Puddu, I., Ranganathan, A., and Čapkun, S. I send, therefore i leak: Information leakage in low-power wide area networks. In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks* (2018), pp. 23–33.

[54] Li, C.-Y., Tu, G.-H., Peng, C., Yuan, Z., Li, Y., Lu, S., and Wang, X. Insecurity of voice solution volte in lte mobile networks. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (New York, NY, USA, 2015), CCS '15, Association for Computing Machinery, p. 316–327.

[55] Li, Y., Kim, K.-H., Vlachou, C., and Xie, J. Bridging the data charging gap in the cellular edge. In *Proceedings of the ACM Special Interest Group on Data Communication* (New York, NY, USA, 2019), SIGCOMM '19, Association for Computing Machinery, p. 15–28.

[56] Li, Y., Xu, J., Peng, C., and Lu, S. A first look at unstable mobility management in cellular networks. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications* (2016), pp. 15–20.

[57] Lu, Y.-H., Li, C.-Y., Li, Y.-Y., Hsiao, S. H.-Y., Xie, T., Tu, G.-H., and Chen, W.-X. Ghost calls from operational 4g call systems: Ims vulnerability, call dos attack, and countermeasure. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking* (New York, NY, USA, 2020), MobiCom '20, Association for Computing Machinery.

[58] Meidan, Y., Bohadana, M., Shabtai, A., Guarnizo, J. D., Ochoa, M., Tippenhauer, N. O., and Elovici, Y. Profiliot: a machine learning approach for iot device identification based on network traffic analysis. In *Proceedings of the symposium on applied computing* (2017), pp. 506–509.

[59] Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A.-R., and Tarkoma, S. Iot sentinel: Automated device-type identification for security enforcement in iot. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)* (2017), IEEE, pp. 2177–2184.

[60] Peng, C., Li, C.-y., Tu, G.-H., Lu, S., and Zhang, L. Mobile data charging: New attacks and countermeasures. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security* (New York, NY, USA, 2012), CCS '12, Association for Computing Machinery, p. 195–204.

[61] Peng, C., Li, C.-Y., Wang, H., Tu, G.-H., and Lu, S. Real threats to your data bills: Security loopholes and defenses in mobile data charging. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (New

13

York, NY, USA, 2014), CCS '14, Association for Computing Machinery, p. 727–738.

[62] Peng, C., Tu, G.-h., Li, C.-y., and Lu, S. Can we pay for what we get in 3g data access? In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking* (New York, NY, USA, 2012), Mobicom '12, Association for Computing Machinery, p. 113–124.

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

[64] Restuccia, F., D'Oro, S., and Melodia, T. Securing the internet of things in the age of machine learning and software-defined networking. *IEEE Internet of Things Journal 5*, 6 (2018), 4829–4842.

[65] Roy, N., Shen, S., Hassanieh, H., and Choudhury, R. R. Inaudible voice commands: The long-range attack and defense. In *15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18)* (2018), pp. 547–560.

[66] Rupprecht, D., Kohls, K., Holz, T., and Pöpper, C. Breaking lte on layer two. In *2019 IEEE Symposium on Security and Privacy (SP)* (2019), IEEE, pp. 1121–1136.

[67] Rupprecht, D., Kohls, K., Holz, T., and Pöpper, C. Imp4gt: Impersonation attacks in 4g networks. In *NDSS* (2020).

[68] Saidi, S. J., Mandalari, A. M., Kolcun, R., Haddadi, H., Dubois, D. J., Choffnes, D., Smaragdakis, G., and Feldmann, A. A haystack full of needles: Scalable detection of iot devices in the wild. In *Proceedings of the ACM Internet Measurement Conference* (2020), pp. 87–100.

[69] Shafiq, M. Z., Ji, L., Liu, A. X., Pang, J., and Wang, J. Large-scale measurement and characterization of cellular machine-to-machine traffic. *IEEE/ACM Transactions on Networking 21*, 6 (2013), 1960–1973.

[70] Shahid, M. R., Blanc, G., Zhang, Z., and Debar, H. Iot devices recognition through network traffic analysis. In *2018 IEEE International Conference on Big Data (Big Data)* (2018), IEEE, pp. 5187–5192.

[71] Shaik, A., Borgaonkar, R., Park, S., and Seifert, J.-P. On the impact of rogue base stations in 4g/lte self organizing networks. In *Proceedings of the 11th ACM Conference on Security and Privacy in Wireless and Mobile Networks* (New York, NY, USA, 2018), WiSec '18, Association for Computing Machinery, p. 75–86.

[72] Shi, C., Liu, J., Liu, H., and Chen, Y. Smart user authentication through actuation of daily activities leveraging wifi-enabled iot. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing* (2017), pp. 1–10.

[73] Shirvanian, M., and Saxena, N. Wiretapping via mimicry: Short voice imitation man-in-the-middle attacks on crypto phones. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (2014), pp. 868–879.

[74] Sivanathan, A., Gharakheili, H. H., Loi, F., Radford, A., Wijenayake, C., Vishwanath, A., and Sivaraman, V. Classifying iot devices in smart environments using network traffic characteristics. *IEEE Transactions on Mobile Computing 18*, 8 (2018), 1745–1759.

[75] Stute, M., Heinrich, A., Lorenz, J., and Hollick, M. 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 *30th {USENIX} Security Symposium ({USENIX} Security 21)* (2021).

[76] Sun, K., Chen, C., and Zhang, X. " alexa, stop spying on me!" speech privacy protection against voice assistants. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems* (2020), pp. 298–311.

[77] Tu, G.-H., Li, C.-Y., Peng, C., Li, Y., and Lu, S. New security threats caused by ims-based sms service in 4g lte networks. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (New York, NY, USA, 2016), CCS '16, ACM, pp. 1118–1130.

[78] Wang, H., Xu, F., Li, Y., Zhang, P., and Jin, D. Understanding mobile traffic patterns of large scale cellular towers in urban environment. In *Proceedings of the 2015 Internet Measurement Conference* (New York, NY, USA, 2015), IMC '15, Association for Computing Machinery, p. 225–238.

[79] Xie, T., Li, C.-Y., Tang, J., and Tu, G.-H. How voice service threatens cellular-connected iot devices in the operational 4g lte networks. In *2018 Ieee International Conference on Communications (icc)* (2018), IEEE, pp. 1–6.

[80] Xie, T., Tu, G., Yin, B., Li, C., Peng, C., Zhang, M., Liu, H., and Liu, X. The untold secrets of wifi-calling services: Vulnerabilities, attacks, and countermeasures. *IEEE Transactions on Mobile Computing* (2020), 1–1.

[81] Xie, T., Tu, G.-H., Li, C.-Y., and Peng, C. How can iot services pose new security threats in operational cellular networks? *IEEE Transactions on Mobile Computing* (2020).

[82] Xu, Q., Huang, J., Wang, Z., Qian, F., Gerber, A., and Mao, Z. M. Cellular data network infrastructure characterization and implication on mobile content placement. In *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems* (New York, NY, USA, 2011), SIGMETRICS '11, Association for Computing Machinery, p. 317–328.

[83] Yang, D., Zhang, X., Huang, X., Shen, L., Huang, J., Chang, X., and Xing, G. Understanding power consumption of nb-iot in the wild: tool and large-scale measurement. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking* (2020), pp. 1–13.

[84] Yang, H., Bae, S., Son, M., Kim, H., Kim, S. M., and Kim, Y. Hiding in plain signal: Physical signal overshadowing attack on {LTE}. In *28th {USENIX} Security Symposium ({USENIX} Security 19)* (2019), pp. 55–72.

[85] Zhang, R., Wang, X., Farley, R., Yang, X., and Jiang, X. On the feasibility of launching the man-in-the-middle attacks on voip from remote attackers. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security* (2009), pp. 61–69.

14