How Can IoT Services Pose New Security Threats In Operational Cellular Networks?

Tian Xie, Guan-Hua Tu, Chi-Yu Li, Chunyi Peng

Abstract—Carriers are rolling out Internet of Things (IoT) services including various IoT devices and use scenarios. Compared with conventional non-IoT devices such as smartphones and tablets, IoT devices have limited network capabilities (e.g., low rates) and specific use scenarios (e.g., inside vehicles only). These specialized use scenarios lead to carries often offering cheaper device access fees for IoT devices. However, the aforementioned disparity of service charging between IoT and non-IoT devices may lead to security issues. In this work, we conduct the first empirical security study on cellular IoT service charging over two major US carriers and make three major contributions. First, we discover four security vulnerabilities and analyze their root causes, which help us identify two significant security threats, IoT masquerading and IoT use scenario abuse. Second, we devise three proof-of-concept attacks and assess their real-world impact. We determine that they can be exploited to allow adversaries to pay 43.75%-80.00% less for cellular data services. Third, we analyze the challenges in addressing these vulnerabilities and develop an anti-abuse solution to mitigate attack incentives. The solution is standard-compliant and can be used immediately in practice. Our prototype and evaluation confirm its effectiveness.

Index Terms—Cellular network, IoT, security, and charging.

1 Introduction

The Internet of Things (IoT) is becoming more and more pervasive. Through well-connected things, such as wearables, vehicles, robots, and smart meters, the IoT improves the ways we interact with and control cyber-physical systems and empowers smart IoT applications in a multitude of vertical markets, including climate and environment control, agriculture, healthcare, smart cities, smart home, industry, *etc.* To interconnect various IoT devices, the cellular network, which is the only large-scale wireless network infrastructure on a par with the Internet, plays a critical role. Compared with other emerging non-cellular wireless IoT solutions like LoRa (Long Range) and other LPWA (Low Power Wide Area) technologies [1], the cellular network is ready to roll out ubiquitous IoT services to the massive IoT market. The cellular IoT market is forecasted to reach 15 billion devices in 2021, representing a staggering four-fold increase from 4 billion devices in 2015 [2].

However, to the best of our knowledge, cellular IoT security has not been fully explored by academia and industry. When faced with cellular IoT rollout, one of the key issues is to secure its service charging. Many studies [3]–[7] have shown that cellular service charging security is critical for operators due to its negative impacts of an insecure system on both their profits and their user's rights. IoT devices have different traffic patterns and limited use scenarios compared to conventional cellular devices. The differences include much lower data rates, smaller traffic volumes, and limited use scenarios (e.g., inside vehicles only, small screens). Carriers usually provide IoT users with cheaper and more competitive data plans. In the Verizon and AT&T networks, the device access fee of an IoT device is much cheaper than that of a smartphone. For example, a user pays $10 for an IoT device and $20 for a smartphone in AT&T's limited data plans. In practice, users can receive IoT and non-IoT SIM cards for their IoT (e.g., smartwatches) and non-IoT devices (e.g., smartphones). One natural question arises: *will the new IoT service charging expose the current cellular network ecosystem to emerging attack vectors?*

Unfortunately, the answer to the above question is yes. We have identified four security vulnerabilities from two major US carriers, denoted as OP-I and OP-II for privacy concerns. First, an IoT SIM card can be used for a non-IoT device (V1). Second, the network infrastructure is unable to correctly identify IoT and non-IoT devices (V2). Third, the infrastructure does not impose any restrictions on IoT data services (V3). Fourth, the infrastructure is unable to confine IoT devices to their pre-defined use scenarios (V4). These vulnerabilities result in two major security threats: (1) adversaries can disguise non-IoT devices as IoT devices to pay less without service downgrades; (2) adversaries can use IoT devices in unanticipated use scenarios. Each of the vulnerabilities can be attributed to design defects of the cellular IoT standards, operational slips of the network infrastructure, and/or device implementation issues. Table 1 summarizes these vulnerabilities.

We exploit these vulnerabilities to devise three proof-of-concept attacks: IoT masquerading, wearable IoT abuse, and car-connected IoT abuse. The IoT masquerading allows the adversary to gain cheaper smartphone services by disguising a smartphone as an IoT device. The other two attacks abuse IoT devices in unanticipated use scenarios to gain

- *T. Xie and G.-H. Tu are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, 48825. E-mail:{xietian1, ghtu}@msu.edu.*
- *C.-Y. Li is with the Department of Computer Science, National Chiao Tung University. E-mail:chiyuli@cs.nctu.edu.tw*
- *C. Peng is with the Department of Computer Science, Purdue University, West Lafayette, IN, 47907. E-mail:chunyi@purdue.edu*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.2984192, IEEE Transactions on Mobile Computing

Category	Vulnerability	Type	Root Cause
Device	V1: an IoT SIM card can be used for a non-IoT device.	Design Defect	No mutual authentication between the SIM card and the device is stipulated in cellular IoT standards (Section 4.1.1).
Infrastructure	V2: the infrastructure is unable to correctly identify IoT and non-IoT devices.	Design Defect	No device authentication mechanism is stipulated in cellular IoT standards. (Section 4.1.1).
	V3: the infrastructure does not impose any restrictions on IoT data services.	Operational Slip	Operators merely rely on the hardware constraints of IoT devices instead of imposing restrictions from the infrastructure (Section 4.1.2).
	V4: the infrastructure is unable to confine IoT devices to their pre-defined use scenarios.	Operational Slip/ Implementation Issue	Operators restrict the IoT use scenarios by device-based security mechanisms and constraints. However, they are not bullet-proof. (Section 4.2).

TABLE 1: Summary of security vulnerabilities and root causes.

cheaper mobile hotspot services. Our study shows that the adversary can pay 43.75% to 80.00% less for cellular services while using two top-tier US carriers.

At first glance, carriers can address the vulnerabilities by binding IoT services to IoT SIM cards and limiting their maximum rates based on the profiling of their normal use scenarios. Even though the adversary can disguise a smartphone as an IoT device using an IoT SIM card, its maximum rate is limited by the IoT service associated with the SIM card. This can prevent the adversary from gaining cheaper services. However, this solution is not practical. With the expected proliferation of cellular IoT devices in the near future, there are more and more unprecedented IoT devices and use scenarios. It is not only challenging but also non-scalable for carriers to determine appropriate maximum rates for various IoT services/devices (e.g., car-connected mobile hotspots and critical traffic control devices). We thus propose a service-oriented charging solution, anti-abuse service model, which provides differential service quality for each cellular device based on its cellular technology category and device access fee. With only minimal support from the infrastructure, it is compatible with current cellular network standards and practices. Our model can eliminate the adversary's incentives to launch the IoT masquerading and abuse attacks.

Contributions: This paper makes three key contributions.

- We conduct the first empirical security study on cellular IoT charging over three mainstream cellular IoT technologies, including CAT-4 (Category 4), CAT-1 (Category 1), and CAT-M1 (Category M1) [8]–[10], which provide users with different transmission rates and battery life for the support of critical and massive IoT applications (see details in Table 2). We then identify four vulnerabilities and analyze root causes.
- We devise three proof-of-concept attacks by exploiting the identified vulnerabilities and assess their real-world impact on two major US carriers.
- We examine why the solution can be challenging for carriers; then, we propose a standard-compliant solution, as well as prototype and evaluate it. The lessons learned can secure and facilitate the global deployment of cellular IoT services, as well as provide new insights for upcoming 5G networks.

Paper Organization: The rest of this paper is structured as follows. §2 introduces the background of the cellular IoT service charging. We analyze its security in §3, as well as uncover vulnerabilities, and devise three proof-of-concept attacks in §4. §5 models and showcases attack incentives. We present challenges of securing cellular IoT service charging in §6 and propose a solution in §7. §8, §9, and §10 present discussion, related work, and conclusion, respectively.

		CAT-4 (R8)	CAT-1 (R8)	CAT-M1 (R13)	NB-IoT (R13)
KPI	IoT types	Critical	Critical/Massive	Massive	Massive
	DL peak rate	150 Mbps	10 Mbps	1 Mbps	0.2 Mbps
	UL peak rate	50 Mbps	5 Mbps	1 Mbps	0.2 Mbps
	bandwidth	20 Mhz	20 Mhz	1.4 MHz	180 KHz
	battery life	day(s)	year(s)	>10 years	>10 years
Roll-out	Consumer IoT	●	●	◐ (Few)	◐ (Few)
	product carrier	●	●	◐ (Partial)	◐ (Partial)

TABLE 2: Summary of cellular IoT technologies in operational LTE networks from US carriers [8]–[11].

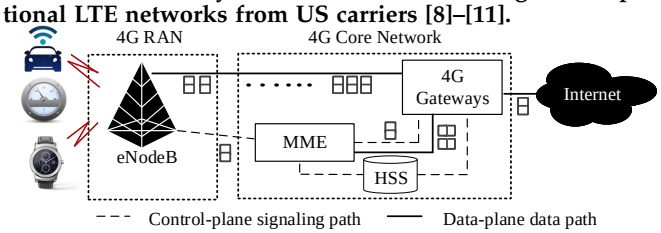


Fig. 1: 4G LTE network architecture with IoT support.

2 How to Charge for Cellular IoT Services?

Cellular IoT technologies. Cellular IoT is a newly emerging solution for IoT devices connected over cellular networks. They share network infrastructure with non-IoT devices (*e.g.*, smartphones), but require special support, such as long sleep time and the delivery of small data over the control plane. Several technologies have been proposed to meet their diverse demands: CAT-4, CAT-1, CAT-M1, and NB-IoT (Narrowband IoT) [8]–[10], which are summarized in Table 2. These cellular IoT technologies support two major types of IoT applications: critical (*e.g.*, traffic/safety control and mobile health) and massive (*e.g.*, smart agriculture) applications. The critical IoT applications require ultra reliability, low latency, and high availability, whereas the massive IoT applications focus on low cost, low energy, and small data volumes. In the market, CAT-4 and CAT-1 have been widely deployed by US carriers, but other technologies have not (e.g., Verizon and AT&T support only CAT-M1 whereas T-Mobile supports only NB-IoT). Most consumer IoT devices, such as wearable devices, car-connected mobile hotspots, and tracking sensors, belong to CAT-4, CAT-1, and CAT-M1.

Figure 1 shows the 4G LTE network architecture with IoT support. The network architecture consists of two major components: Radio Access Network (RAN) and core

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.2984192, IEEE Transactions on Mobile Computing

Carriers	Data plan	Monthly Charge fees	Non-IoT devices		IoT devices		
			Smartphone	Portable Mobile Hotspot	CAT-4		CAT-1/CAT-M1
					Wearable	Car-connected Mobile Hotspot	
OP-I	Limited data plan	Device access fee	$20	$20	$10	$10	$0
		Service access fee	$50 (3GB)	$0*	$0*	$0*	$0.99(0.5MB),$14(0.1GB),$22(1GB),$35(5GB)
	Unlimited data plan	Device access fee	$35	No unlimited data plans	$10	$20	No unlimited data plans
		Service access fee	$75		$0*	$0*	
OP-II	Limited data plan	Device access fee	$20	$10	$5, $10 (varying with models)	$10	$0
		Service access fee	$35 (2GB)	$0*	$0*	$0*	$2(0.2MB),$18(0.15GB),$25(1GB),$50(5GB)
	Unlimited data plan	Device access fee	$0(1),$65(2),$75(3),$85(4)	$20	$5, $10 (varying with models)	$20	No unlimited data plans
		Service access fee	$75	$0*	$0*	$0*	

TABLE 3: Data plans for IoT and non-IoT devices in two US carriers (studied in Dec. 2018). The price and volume cap are shown by per month unless explicitly specified. Many variants may not be included, for example, $60 for 10GB per 30 days for OP-I IoT sim cards [12]. ($0*: Shared the fee with phones)

network. The RAN allows IoT devices to transmit IoT data to cellular network infrastructure using the aforementioned cellular IoT technologies. The core network includes three main network elements: Mobility Management Entities (MMEs), 4G gateways, and the Home Subscriber Server (HSS). The MMEs are responsible for user mobility, user authentication, and resource reservation. Additionally, the MMEs are responsible for new IoT functions [13], such as power saving mode and extended discontinuous reception [14]. The HSS stores user subscription data and user information profiles. The 4G gateways forward data between the RAN and the Internet, as well as collect device data usage.

Cellular IoT service charge. We investigate the service charges of IoT devices from two top-tier US carriers denoted as OP-I and OP-II and compare them with those of non-IoT service charges. Table 3 summarizes the comparison. The SIM card used for each device is associated with the owner's non-IoT or IoT data plan. For both device types, a device's charge includes two kinds of fees, device and service access fees. Its bill can be formulated as $B(u) = \alpha + u \otimes \beta$, where α is the device access fee and $u \otimes \beta$ represents the service access fee determined by actual data usage volume u and unit price β. In most cases, unlimited voice and text services are offered, so the formula does not include them. The service charges vary not only with device types and models but also with limited and unlimited data plans.

The unlimited data plans often have higher device access fees than those of the limited data plans. For instance, the device access fees are $20 and $35 for a smartphone line in OP-I's limited and unlimited plans, respectively. The limited plans usually have service access fees increasing with capped data usage volumes, in contrast to fixed service fees in the unlimited data plans. For example, OP-II charges $35, $50 and $70 for monthly volumes of 2 GB, 4 GB, and 8 GB, respectively. Note that the increase is not proportional for most carriers except Google Project Fi [15], which charges $15 for each 1 GB, is one of few exceptions.

In terms of the service charging policies, IoT devices differ from non-IoT devices in two aspects. First, IoT device access fees are cheaper, since IoT devices require much smaller data usage volumes than non-IoT devices. The IoT device access fees may also vary with device models. For example, OP-II charges $5 for an LG Watch Urbane2 and $10 for an Apple Watch. Second, IoT service access fees are usually tied to non-IoT data plans, but there are still some IoT-specific data plans. The IoT-specific data plans offer lower service fees per data unit. For example, OP-I offers 5 GB [12] to IoT users at only $35, but offers the same amount of data to smartphone users at $50.

3 What May Go Wrong?

We aim to explore the dark side of the emerging IoT service charging scheme and its technical support from a security perspective. Any of its vulnerabilities may cause cellular users and/or carriers to suffer monetary losses. We start from an observation that IoT devices have cheaper data plans than non-IoT devices, which can be attributed to their distinct use scenarios. For example, smartwatches are designed for simple voice/data services, and car-connected hotspots are used only inside vehicles. It appears to be reasonable, but one question arises: *are the underlying technologies sufficient to secure this differential charge?* We answer it by starting with the following questions.

Q1. Given different charges for the same data service of an IoT device and a smartphone, can the smartphone masquerade as the IoT device to pay less?

Q2. If yes, can the smartphone retain its data service quality (*e.g.*, no speed downgrade)?

Q3. Can adversaries abuse IoT devices in unanticipated use scenarios so as to take advantage of operators?

Unfortunately, we disclose that the IoT charging, as well as the technical support behind it, is not bullet-proof. The answers to the above three questions are all yes. Specifically, we uncover four vulnerabilities from design, implementation, and network operation aspects. The cellular network standards, network operators/vendors, and device manufacturers all share the blame for these vulnerabilities. The fundamental problems are rooted not in how to charge IoT and non-IoT devices, but in how to provision and safeguard IoT services.

Threat model. In this work, the adversary is a mobile user who uses only commodity devices: smartphones and IoT devices available on the market. To launch attacks, (s)he needs to either know how to install tools on smartphones and modify their settings, or rely on some one-click software/hardware package, the development of which is not our main focus. In all cases, (s)he has no access to the cellular network infrastructure or other devices. Moreover, the network infrastructure and the device hardware are not compromised. Given this model, the identified security loopholes can be translated into realistic attacks against carriers.

Methodology. We validate vulnerabilities and attacks in two top-tier US carriers, OP-I and OP-II. They, together, take more than 65% of market share [16] in the U.S. We conduct experiments using IoT devices including two popular

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.2984192, IEEE Transactions on Mobile Computing

smartwatch models and two car-connected hotspots, as well as non-IoT devices including four Android phone models, with the two carriers' SIM cards. The two smartwatch models are LG Watch Urbane 2nd edition with Android 6.1.1 and Samsung Gear S3 frontier with Tizen OS 2.3.2. The four phone models include Samsung Galaxy S5/S6, LG G3 and Google Nexus 6P, which run Android 4.4.4, 5.0.2/6.0.1, 4.4.2 and 7.1.1, respectively. Note that all the results can be applied to both carriers, unless explicitly stated otherwise.

Responsible experiments. We understand that some feasibility tests and attack evaluations might be harmful to carriers, so we proceed with this study in a responsible manner. We run experiments in fully controlled environments. We purchase plans with sufficient data/voice/text quotas, so the carriers do not get hurt. We seek to disclose new security vulnerabilities and effective attacks on cellular IoT services, but not to aggravate the damages caused by them.

4 How does cellular IoT charging go wrong?

In this section, we answer the three aforementioned questions by considering two potential threats, IoT masquerading and IoT use scenario abuse. We validate vulnerabilities and devise proof-of-concept attacks for each threat, as well as evaluate a long-term IoT attack to show real-world impact.

4.1 IoT Masquerading

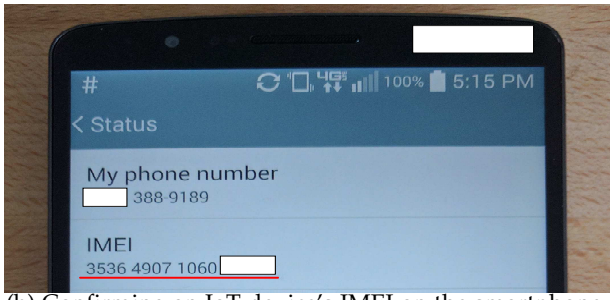
We first introduce three vulnerabilities and then devise an IoT masquerading attack.

4.1.1 Can Non-IoT Devices Masquerade as IoT Ones?

The answer is yes, due to two vulnerabilities discovered within the 3GPP security design. Each of vulnerabilities corresponds to a lack of mutual authentication between two parties. One is between IoT SIM cards and mobile devices, so an IoT SIM card can be used for a non-IoT device (V1). The other is between mobile devices and the infrastructure, so the latter is unable to correctly identify IoT and non-IoT devices (V2). These two vulnerabilities allow non-IoT devices to masquerade as IoT devices without being detected by SIM cards or the infrastructure.

The cellular authentication solely relies on the Authentication and Key Agreement (AKA) procedure [13], where users and the infrastructure are mutually authenticated. Each user is identified by his/her International Mobile Subscriber Identity (IMSI) and authenticated based on a secret key. Both the IMSI and the secret key are stored in the SIM card. However, neither the SIM card nor the infrastructure authenticates the used device; the former does not differentiate types of mobile devices in its operation, whereas the latter identifies a connected device purely based on its reported information, which may be fake without the device authentication and thus lead to wrong identification. By current design, the non-IoT/IoT data plan to which each user subscribes is bound to the IMSI or the SIM card, so the used device is not restricted by the subscribed plan. That is, an IoT SIM card, which is associated with an IoT data plan, can also work on non-IoT devices. This allows

(a) Replacing the smartphone's IMEI with an IoT device's using the EFS tool [17].



(b) Confirming an IoT device's IMEI on the smartphone.

(c) A snapshot of the OP-II's web page shows that the smartphone is recognized as an IoT device, a smartwatch.

Fig. 2: Making IMEI spoofing on a smartphone (LG G3) to masquerade as an IoT device (LG Watch Urbane 2nd).

the IoT masquerade to be possible. Moreover, differential non-IoT/IoT charges, where the IoT plans are cheaper, can be a strong incentive for the masquerade.

Validation. We first validate V1 by showing that IoT SIM cards work for non-IoT devices. We purchase IoT SIM cards used for CAT-4, CAT-1, and CAT-M1 IoT devices. We insert each of them into our test smartphones, properly configure their networking settings, and then restart the phones. Our experimental results, collected from OP-I and OP-II, show that all the smartphones successfully obtain IP addresses from the cellular networks and access the Internet without any issues.

We next validate V2 by examining whether the infrastructure can be fooled into thinking that a smartphone is an IoT device. Initially, we discover that the OP-I and OP-II networks can correctly identify connected non-IoT or IoT devices, and show the device information on their web pages. We then analyze the control-plane protocol traces by using cellular diagnosis tools (e.g., MobileInsight [18]). It is observed that the infrastructure identifies a connected device based on the IMEI (International Mobile Equipment Identity) carried in its `IDENTITY RESPONSE` message [13]. When connecting to the network, the device reports its IMEI in the response to the message of `IDENTITY REQUEST`. This implies that if the device reports a fake IMEI, the spoofing can happen.

We then investigate how to make a mobile device report

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.2984192, IEEE Transactions on Mobile Computing

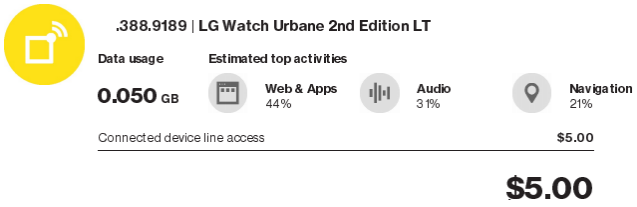


Fig. 3: When the IMEI spoofing on the smartphone lasts for one month, OP-II still recognizes it as an IoT device, the LG smartwatch, with a $5 charge as the IoT device access fee.

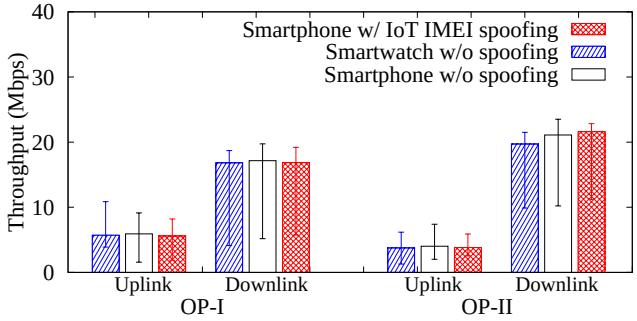


Fig. 4: The uplink and downlink TCP throughput at the 10^{th}, 50^{th}, and 90^{th} percentiles for an IoT device (*i.e.*, LG Watch Urbane 2nd), a smartphone (*i.e.*, Samsung S5), and the smartphone with the spoofing of the IoT device's IMEI in the OP-I and OP-II networks.

a fake IMEI. The IMEI is stored in the non-volatile memory of the device modem, and the memory can be modified by some tools (e.g., EFS Professional [17]). We here show that the IMEI of a smartphone, LG G3, can be spoofed as that of an IoT device, LG Watch Urbane 2nd, in the OP-II network; the same result is also observed in OP-I. The validation consists of four steps. First, we connect to the smartphone's modem via the EFS tool [17] and replace its IMEI with the IoT device's IMEI (i.e., `353649071060XXX`), as shown in Figure 2a. Second, we confirm the IMEI replacement on the smartphone as shown in Figure 2b. Third, we reboot the smartphone to let it report the spoofed IMEI to the network. We then confirm its IMEI change on OP-II's web page, as shown in Figure 2c. It shows that the smartphone has been recognized as the IoT device. Last, we keep the IMEI spoofing on the smartphone for a monthly billing cycle and discover that OP-II does not detect this abuse but still charges the IoT device's access fee (*i.e.*, $5), as shown in Figure 3. From an extended experiment with eight months (the results are elaborated on in Section 4.3), we find that the operator cannot detect the spoofing, even though several hundred megabytes of mobile data are consumed on the smartphone spoofing the IMEI of the IoT device.

Security implications. As new cellular IoT service charging demands arise, current security mechanisms for cellular IoT support in the 3GPP standards are not sufficient to secure carriers. We believe that addressing V1 and V2 requires revisiting these security mechanisms.

4.1.2 Any Limitations Imposed on IoT Data Services?

The answer is expected to be yes when the infrastructure offers differential data services to IoT and non-IoT devices. However, this is not the case for the tested carriers. We discover that a non-IoT device masquerading as an IoT device can still retain the same data service quality while paying less (V3). This allows adversaries to take advantage of the carriers by purchasing cheaper IoT device access for their non-IoT devices.

Validation. We validate this vulnerability by using `iPerf` to examine TCP throughput performance on three devices: (1) an IoT device (*i.e.*, LG Watch Urbane 2nd) equipped with an IoT SIM card, (2) a smartphone (*i.e.*, Samsung S5) with a non-IoT SIM card, and (3) the smartphone spoofing the IoT devices IMEI with an IoT SIM card. We consider both uplink and downlink cases and test each case for 10 runs. Figure 4 shows the 10^{th}, 50^{th}, and 90^{th} percentiles of the throughput results of those three devices in the OP-I and OP-II networks. We observe that all the three devices have comparable performance on the uplink and downlink throughput in each operator's network. For example, in the OP-I network, the median uplink/downlink throughout speeds for the IoT device, the smartphone, and the smartphone masquerading as an IoT device are 5.73/16.82 Mbps, 5.91/17.15 Mbps, 5.59/16.85 Mbps, respectively. This shows that the networks do not enforce any noticeable restrictions on IoT devices in terms of data transmission rates. Besides, we do not observe that any restrictions are imposed on IoT data usage volumes.

Security implications. Seemingly, carriers just make a simple operational mistake, but this may not be the case. This vulnerability can be attributed to two possible reasons. First, carriers may not have incentives to restrict IoT data services for IoT devices due to its limited benefits. For example, for limited IoT data plan users, the more data that IoT users use, the more profit that carriers can make. Second, carriers may impose service restrictions based on the theoretical maximum uplink and downlink rates of IoT device categories (e.g., CAT-4: 50Mbps/150Mbps, CAT-1: 5Mbps/10Mbps), but they do not take any effect. This is because wireless resources are shared by multiple devices and the theoretical maximum rates are usually much higher than the actual rates available to the networks.

4.1.3 A Proof-of-concept Attack

We devise an IoT masquerading attack based on the vulnerabilities V1, V2, and V3. We consider that an adversary has subscribed to a cellular network service with a limited or unlimited data plan. (S)he adds a smartwatch to his/her account and obtains its IoT SIM card from the carrier. Afterwards, (s)he can start to launch the attack by letting his/her smartphone masquerade as the smartwatch based on the IMEI spoofing. We test three main cellular network services on the smartphone: data, voice and text. The results are summarized in Table 4. With the attack smartphone, the adversary can make voice calls, send/receive short messages and access the Internet at 10 different locations, but only pay the IoT device access fee. The adversary can save 50% and 75% of the smartphone device access charges in the OP-I and OP-II networks, respectively. Note that OP-II does not assign a dedicated phone number to the smartwatch for voice and text services; the user has to use

Operator	Device	Data Service	Voice Service	Text Service	Dedicated number?	Charge (per month)
OP-I	Smartphone w/ spoofing	√	√	√	√	$10
	Smartwatch	√	√	√	√	$10
	Smartphone	√	√	√	√	$20
OP-II	Smartphone w/ spoofing	√	√	√	√	$5
	Smartwatch	√	√	√	×	$5
	Smartphone	√	√	√	√	$20

TABLE 4: Offered services and charges vary with the devices with or without the IMEI spoofing in the OP-I and OP-II networks based on limited data plans.

the phone number belonging to the paired smartphone's SIM card. However, the attack smartphone can obtain a dedicated phone number. It may be because OP-II prevents the IoT SIM card from registering the VoLTE system on the smartwatch, but it is not prohibited on the smartphone.

4.2 IoT Use Scenario Abuse

We next investigate whether IoT devices can work in unanticipated use scenarios. Current carriers offer cheaper device access fees to some IoT devices due to their limited use scenarios. However, we discover the fourth vulnerability (V4) that those IoT devices may not be restricted to their anticipated use scenarios. We validate this vulnerability on two different types of popular IoT devices: car-connected mobile hotspots and smartwatches.

4.2.1 Car-connected Hotspots: Not Limited to only Vehicles

Car-connected hotspots are, by default, designed for using only inside vehicles. However, when they are fully controlled by adversaries, some malicious manipulations can be performed to bypass the usage restriction. We discover that the adversary may turn these car-connected hotpots into common mobile hotspots, which offer mobile data services.

We observe that two hardware features of car-connected hotspots restrict their usage to only inside operating vehicles. First, its power supply is from the diagnostic connector of OBD-II (On-Board Diagnostics II), which is a system for the status report of various vehicle subsystems. The OBD-II connector is not used for other non-vehicle systems, so the car-connected hotspot is hardly powered on outside vehicles.

Second, the hotspot automatically enters a sleep mode after the vehicle has been turned off for a period of time. The hotspot detects whether the vehicle is operating based on voltage changes of the OBD-II connector. According to the hotspot's specification, it operates normally when the voltage of the OBD-II connector is higher than 11.7 V. The voltage of the OBD-II connector can increase up to 15.5 V at the moment that the vehicle engine is ignited. The device disables its hotspot function and enters the sleep mode, when the voltage of the OBD-II connector drops to 11.7 V and 9 V, respectively. Once the adversary makes a power supply with the OBD-II connector interface and then sets its voltage to be higher than 11.7 V, the device can be turned into a common portable hotspot.

Validation We validate this vulnerability by testing whether a car-connected hotspot can continue to be used outside vehicles with our customized power supply. To keep the device's hotspot function active, the power supply is made to output 12 V from a power bank through a voltage regulator. We then connect the power bank's ground and power pins to the fourth and sixteenth pins of the OBD-II connector, respectively, via the regulator. After powering on the hotspot, we connect a Wi-Fi client to the hotspot and use the client to keep generating traffic to/from the Internet using `ping`. We run the test for a whole day, and the traffic is not interrupted.

4.2.2 Smartwatches: Not Constrained by Hardware or Software

Smartwatches with hardware constraints (*e.g.*, small screen) are mainly developed to assist mobile users in getting voice/text services, simple data services (*e.g.*, voice assistants), and notifications from their paired smartphones. Therefore, by design, there is only a small number of smartwatch applications, and their functions are more limited than smartphone applications. For example, Google wearable devices are not allowed to install standalone Gmail (*i.e.*, working without paired smartphones), Chrome browser, and Youtube. However, these hardware/software constraints are not sufficient to restrict the real-world usage of the smartwatch. Specifically, the smartwatch can be turned into a mobile data gateway, which forwards data packets between a Wi-Fi device and the Internet, to provide Internet access over Wi-Fi. Note that the Wi-Fi device connects to the smartwatch via Wi-Fi and the smartwatch connects to the Internet via the cellular network.

Validation. We validate this vulnerability by examining whether the network forbids a smartphone's data packets which are forwarded by the smartwatch. We develop a data forwarding application on the smartwatch. It first receives the smartphone's data packets from the Wi-Fi interface and sends them to our external UDP server on the Internet through the cellular network interface. In our test, the smartphone transmits about 500 MB traffic to the forwarding smartwatch. Our experiment shows that all the transmitted packets are received by our Internet server. No restrictions from the OP-I and OP-II networks are observed for this usage.

Security implications. This vulnerability can be attributed to two potential issues. First, there are various cellular IoT use scenarios, so it is challenging for the infrastructure to identify all possible use scenarios. Second, even though carriers deploy some constraints or security

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.2984192, IEEE Transactions on Mobile Computing

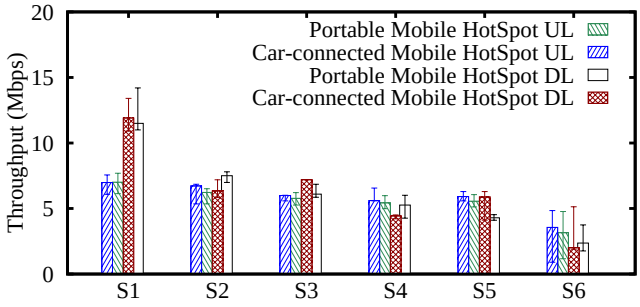


Fig. 5: The uplink/downlink TCP throughput results at the 10^{th}, 50^{th}, and 90^{th} percentiles are plotted for an IoT-masqueraded hotspot (*i.e.*, Mobley) and a normal mobile hotspot (*i.e.*, Velocity) in the OP-I network, given a laptop client placed at six indoor locations in our campus.

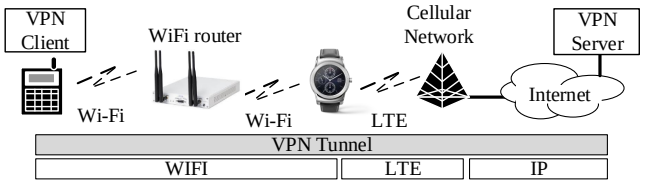


Fig. 6: The network architecture that turns a smartwatch to a mobile data gateway based on a VPN approach.

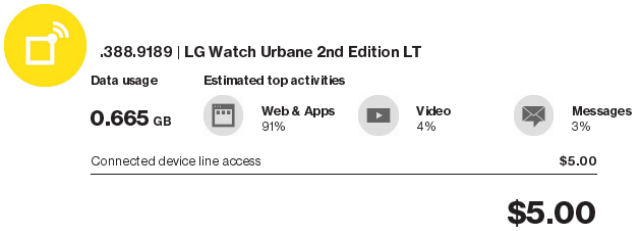


Fig. 7: The data usage of the smartwatch that masquerades as a mobile gateway. The 91% traffic volume of the total 665 MB, which is consumed by the web and applications, is mostly used by the gateway application.

mechanisms on the IoT devices, they can be easily bypassed at low cost. For example, car-connected devices have to be powered on via the OBD-II interface, and smartwatch users are not allowed to install the applications that smartphone users can install.

4.2.3 *Two Proof-of-concept Attacks*

We devise two proof-of-concept attacks to assess the real-world damages of the vulnerability V4.

Car-connected IoT abuse: portable mobile hotspot In this attack, we turn a car-connected IoT hotspot (i.e., Mobley) into a mobile hotspot and then compare its performance with an ordinary mobile hotspot (i.e., Velocity). We here present the results obtained in the OP-I network, but skip that of OP-II because of similar phenomena. We connect a laptop with an 802.11ac Wi-Fi card to each of those two hotspots, Mobley and Velocity, and gauge its uplink/downlink performance. In the test, the hotspots are located at the same location, and the laptop is placed at six different locations, which are spaced at 2-meter intervals, for a total range of 10 meters (i.e., S1-S6). S1 is the closest to the hotspot location, whereas S6 is the farthest from the hotspots.

We test uplink and downlink throughput for 10 runs in each case and plot 10^{th}, 50^{th}, and 90^{th} percentiles of the throughput results in Figure 5. We observe that the two hotspots have comparable performance for both uplink and downlink throughput at each location. Specifically, the differences between their median throughput results are within only 5.62% and 2.03% for all the cases in the OP-I and OP-II networks respectively. Neither of the hotspots always outperforms the other. Take the OP-I's limited data plans as an example for the gain estimation of this attack. $10 and $20 device access fees are charged for the car-connected and normal mobile hotspots. With this attack, the adversary can gain a hotspot service for 50% cheaper. The gains can vary with different carriers and data plans, as shown in Table 3.

Wearable IoT abuse: mobile data gateway We devise an attack that abuses a wearable IoT, smartwatch, to be a mobile data gateway, which can provide a local area network with Internet access through the mobile data service. This IoT-masqueraded gateway can cooperate with a Wi-Fi AP to supply Internet access to Wi-Fi devices. We enable it to work for all the applications on Wi-Fi devices by taking a VPN approach.

Figure 6 shows the network architecture that turns a smartwatch to a mobile data gateway. It consists of four components: (1) a VPN server deployed on the Internet, (2) an IoT device supporting both Wi-Fi and cellular networks (*e.g.*, LG Watch Urbane and Samsung Gear S3), (3) a Wi-Fi AP, and (4) a VPN client installed on the Wi-Fi device (here, a smartphone). Both the smartphone and the smartwatch connect to the AP. The VPN client on the smartphone establishes a VPN tunnel with the VPN server, and the smartwatch forwards data between the VPN client and the VPN server through its Wi-Fi and LTE interfaces.

Our experimental results show that the smartphone's applications can access the Internet and work as usual without any changes. Figure 7 shows the smartwatch's data usage. The 91% traffic volume consumed by the web and applications is mostly used by the application that forwards data between the Wi-Fi and LTE networks. We further examine the smartwatch's forwarding bandwidth based on TCP traffic using `iPerf`. It is observed that the median of the TCP throughput over 10 runs can achieve 4.1 Mbps. Note that this attack can work without the Wi-Fi AP in two cases. First, the IoT device supports the Wi-Fi direct technology, which enables Wi-Fi devices to connect to each other directly. Second, the Wi-Fi device can run the VPN and Wi-Fi AP functions simultaneously. As a result, this attack allows the adversary to pay 50% and 75% less in the OP-I and OP-II networks, respectively. Both operators are not capable of detecting or preventing this attack.

4.3 Long-term IoT Attack Evaluation

We conduct a long-term attack evaluation on the IoT masquerading for eight months, in order to examine whether carriers deploy any anomaly detection mechanism for IoT attacks. In the experiment, we subscribe to a 2 GB data plan and then add a smartphone (i.e., Samsung J7) and an IoT

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.2984192, IEEE Transactions on Mobile Computing

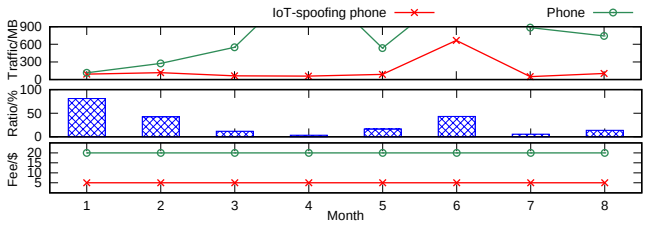


Fig. 8: An 8-month evaluation of the IoT masquerading attack: a smartphone and an IoT device which another smartphone masquerades as (i.e., IoT-spoofing phone) subscribe to the same 2 GB mobile data plan. Top: monthly data usage volumes; middle: the ratio of the IoT-spoofing phone's data usage to the normal phone's; bottom: monthly device access fees from OP-II.

device (i.e., LG Watch Urbane 2nd) to this plan. Their device access fees are $20 and $5, respectively. We use another smartphone (i.e., Samsung S5) to masquerade as the IoT device (i.e., LG Watch Urbane 2nd) with IMEI spoofing. During the 8-month duration, the IoT-spoofing phone is scheduled to access the Internet at least once every day.

Figure 8 shows monthly data usage volumes for both the smartphone and the IoT-spoofing phone (top), monthly usage ratios (the ratio of the data usage of the IoT-spoofing phone to that of the normal smartphone) (middle), and device access fees charged by carriers (bottom). We make three observations. First, the data usage volumes of the IoT-spoofing phone range from 50 MB to 650 MB, whereas those of the normal smartphone are from 115 MB to more than 900 MB. Second, the ratio of the data usage of the IoT-spoofing phone to the normal smartphone ranges from 3.36% to 80.87%. Third, the tested carrier keeps treating the IoT-spoofing phone as an IoT device according to its persistent IoT device access fee of $5. This result shows that current anomaly detection mechanisms are not able to detect the attack, even though the IoT-spoofing phone's monthly usage volume can be as high as 650 MB or the ratio of its usage to that of the normal smartphone is 80.87%.

5 Attack Incentive Modeling

In this section, we model mobile user bills and analyze the adversary's maximum gain, as well as give three attack instances to showcase real-world impact.

5.1 Mobile User Bills Modeling

Suppose that there are s different monthly service plans from an operator, and a mobile user has a subscribed service plan j, monthly data usage u, and n_t devices from each device type t. Given i different device types, the number of devices owned by the user can be represented by $\Sigma_{t=1}^{i} n_t$. The user's monthly bill can thus be modeled as follows:

$$Bill_j(u, n_1, \cdots, n_i) = \Sigma_{t=1}^{i} n_t \cdot \alpha_{j,t} + max\{\beta_{j,1}, \beta_{j,2} \cdot u \cdot \mathrm{I}(u \leq cap_j), \beta_{j,2} \cdot cap_j \cdot \mathrm{I}(u > cap_j)\} + \beta_{j,3} \cdot (u - cap_j) \cdot I(u > cap_j),$$

where $\alpha_{j,t}$ is the device access fee of device type t in plan j, $\beta_{j,1}$ is the minimal data service fee in plan j (*e.g.*, $35 in the OP-II's 2GB plan), $\beta_{j,2}$ is the unit price when u is lower than cap_j, which is the maximum data usage for the unit price $\beta_{j,2}$, $\beta_{j,3}$ is the unit price after u exceeds cap_j, and $I(x > y)$ is a boolean value (0 or 1) indicating if x is larger than y.

Maximal attack gain. Suppose that the adversary uses a service plan j before launching an attack. To maximize the attack gain, the adversary can choose the best service plan for his overall usage and the best device type to masquerade as for each device. The gain can be represented as follows:

$$Bill_j(u, n_1, \cdots, n_i) - min\{Bill_k(u, n'_1, \cdots, n'_i)\}$$

where $\Sigma_{t=1}^{i} n_t = \Sigma_{t=1}^{i} n'_t$ and $k = 1, \cdots, s$. By considering all the possible service plans and the charges of all the device types, the adversary can identify an attack policy that maximizes the gain.

5.2 Three Attack Instances

Example I: light usage (Saving:$70→$14). Bob usually has free Wi-Fi access and thus requires only small volume of mobile data service on his smartphone. Assume that the required volume is less than 1 GB per year. According to OP-I's monthly data plans, he needs to subscribe to at least a 3 GB data plan with a monthly service fee of $50 and adds his smartphone to the plan with a monthly device access fee of $20. For a one-year time period, he should pay $840 ($70×12). Based on the analysis of maximum gain, the best attack policy is to purchase a monthly 100 MB IoT CAT-1/CAT-M1 plan, which has a *monthly* service fee $14, and then launch the IoT masquerading attack on his smartphone. The attack can reduce his annual bill from $840 to $168 , offering an 80% saving.

Example II: moderate usage (Saving:$70→$22). Bob usually uses around 3 GB mobile data per month. The OP-I's 3 GB monthly data plan can be a perfect match for him. The monthly fee is $70 including $50 service access and $20 device access fees. The best attack policy for him is to purchase a 3 GB monthly IoT data plan, which only charges $22, and then launch the IoT masquerading attack on his smartphone. His monthly bill can have a 68.5% reduction, from $70 to $22.

Example III: heavy usage (Saving:$160→$90). Bob and his three family members together use more than 8 GB mobile data per month. The OP-II's unlimited data plan is a good match for them. With four smartphone lines, a monthly fee $160 is charged for the unlimited data plan. By launching the IoT masquerading attack, the cost can be reduced to $90, where $75 comes from one smartphone line in the unlimited plan and $15 ($5×3) comes from three smartwatch lines that can be used to masquerade for their three smartphones. This results in a 43.75% saving for Bob's family; on the other hand, there is a 43.75% revenue loss for OP-II on this account.

6 Why is it Hard to Secure Cellular IoT Service Charging?

To secure cellular IoT service charging, the network infrastructure needs to accurately identify IoT devices and use scenarios. However, this can be challenging in practice. We next analyze several potential and existing solutions.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.2984192, IEEE Transactions on Mobile Computing

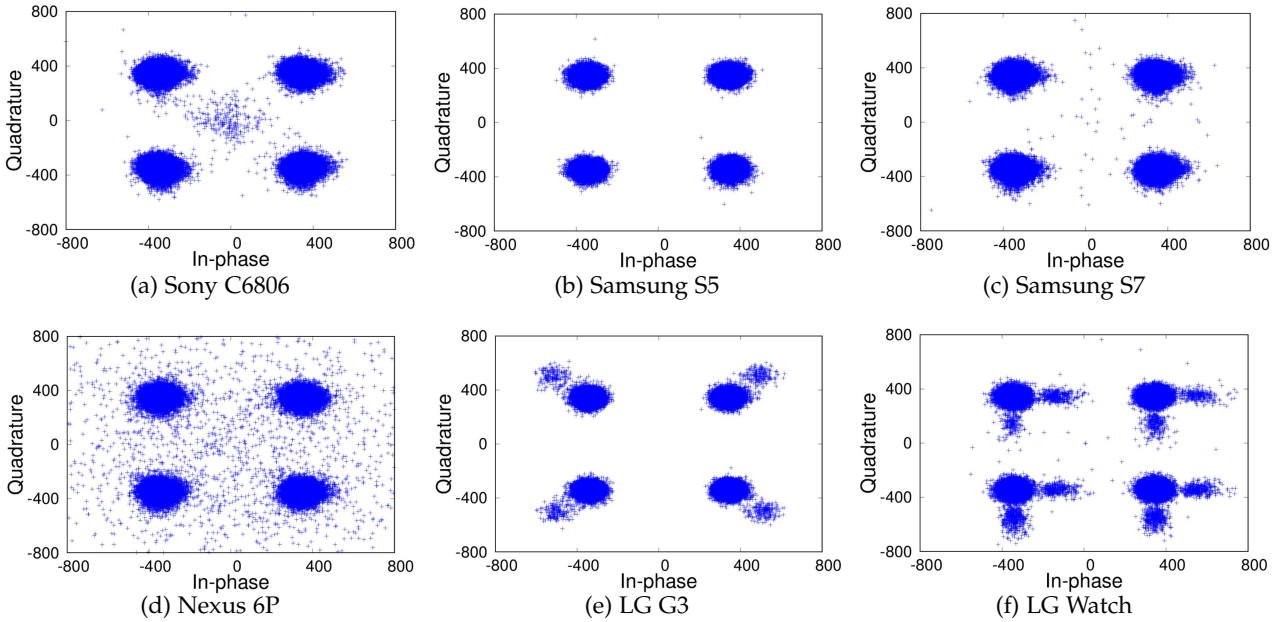


Fig. 9: QPSK constellation diagrams collected on six mobile devices.

6.1 Identifying Devices is Challenging

Current cellular networks identify a device based on the IMEI reported by the device itself. When the adversary has full control over the device, it is challenging to prevent its IMEI from being altered. We next introduce four possible remedies for the device identification and discuss their drawbacks.

Profiling-based device identification. Cellular IoT devices usually have limited software/hardware capabilities, so the usage volume of their mobile data services can be expected to be low. For example, due to the smartwatchs small display, its Android OS does not support standalone browser and Youtube applications. This can prevent IoT devices, such as smartwatches, from consuming as much data traffic as smartphones. The infrastructure may thus be able to identify the IoT devices based on such low-traffic profiles.

However, this approach has two potential technical issues. First, data usage patterns can vary with users. Given that an IoT device's daily usage volume exceeds a specified threshold, which may be determined based on some statistical usage results, the carrier is still unable to ensure whether the IoT masquerading attack is indeed happening. Second, various IoT devices can have different data usage patterns. Profiling all IoT device types can lead to non-negligible overhead for carriers since there will be more and more new IoT devices in the near future.

Hardware-based device identification. Potential hardware-based solutions include the ARM TrustZone and the hardware-based public-key cryptography. The ARM TrustZone has been supported by many popular Cortex-A class processors, crypto chips and secure elements with tamper-proof blocks. Carriers can leverage it to protect the IMEI from being modified by the adversary. However, not all the user devices support this feature. The adversary can easily bypass the protection by using the mobile devices developed on top of the SDR (Software Defined Radio) platforms, which lack the ARM TrustZone support.

With public-key cryptography, each mobile device needs to be assigned a key pair of private and public keys, and an X.509 certificate which is signed by a CA (Certification Authority). The infrastructure can identify each device based on its response to a challenge. Nevertheless, this approach has two major issues. First, not all of IoT devices can support public-key cryptography due to resource constraints (e.g., there is no enough storage space to install security libraries). Second, enabling the public-key cryptography support for the device identification requires modifications to current cellular network standards.

RF fingerprint-based device identification. Another possible solution is to identify devices based on their different RF fingerprints. The differences come from device types and the imperfections of device hardware. This has been proposed to address some security issues such as intrusion detection [19], access control [20], wormhole detection [21], and to improve inter-cellular security [22], to name a few. To assess the effectiveness of this approach, we conduct experiments using the OpenAirInterface (OAI) platform, a software-defined 4G LTE infrastructure [23]. We collect the RF signals transmitted by various mobile devices that connect to the OAI eNodeB. The experiment starts after the tested device is powered on and stops after the RRC (radio resource control) connection between the device and the OAI eNodeB is established. Note that we take two measures to prevent the experiment from affecting other normal mobile users. First, we configure the eNodeB to use the LTE band 7, which is not used by carriers in North America. Second, we put the OAI platform and the mobile devices in a RF shielded enclosure box. Figure 9 shows the QPSK (Quadrature Phase-Shift Keying) constellation diagrams of six mobile devices including IoT and non-IoT devices. Seemingly, we can identify these devices by

analyzing their constellation diagrams, especially for the LG G3, LG Watch Urbane 2nd and Sony C6806. However, this approach is not scalable as it requires the eNodeB to collect all the IoT devices' RF fingerprints.

Tethering-detection-based device identification. Tethering detection has been deployed by operators to detect if users provide their PCs with Internet access by enabling Wi-Fi or USB tethering on their smartphones. However, it still requires significant modifications since it is designed for smartphones rather than IoT devices, and some studies have reported that they can be bypassed (e.g., faking OS signatures).

6.2 Identifying Use Scenarios is Challenging

The network is capable of identifying abnormal use scenarios of an IoT device to some extent. Take car-connected hotspots as an example. The network has cell-level mobility information of each hotspot, and can keep track of its mobility patterns. However, it is still difficult to identify whether the hotspot is being used inside a vehicle or not. Even if the hotspot keeps staying within a cell for a long time period, it is not necessarily outside the vehicle. It may be due to a serious traffic jam. With the proliferation of cellular IoT devices in the near future, there may be more unprecedented IoT use scenarios. It can be very challenging for the network infrastructure to identify the use scenario of each device.

7 Solution: Mitigating Attack Incentives

We seek for a standard-compliant solution that can rapidly mitigate the IoT attacks. We thus consider eliminating V3, and it can also mitigates the attack incentives on the other three vulnerabilities. We leave the solutions for V1, V2, and V4, which require time-consuming standard modifications and cannot be done shortly, to the future design. Specifically, two new mutual authentication mechanisms are required to address V1 and V2: one is between an IoT SIM card and an IoT device, as well as the other is between the device and the infrastructure. The mutual authentication based on the public-key cryptography can be a potential solution option, but it requires modifications to 3GPP standards, which is time-consuming and cannot be done in a short time. To address V4, a new security mechanism shall be introduced to confine IoT devices to their specific use scenarios. It not only requires standard support but also is challenging for carriers.

To this end, we propose an anti-abuse service model to address V3. This can also largely mitigate the attack incentives on other vulnerabilities. Specifically, our approach ensures that no IoT users can get better service quality than non-IoT users when the IoT users pay less, which does not require any modifications to SIM cards, mobile devices, and cellular network standards but minimal support from the infrastructure. Moreover, our model is scalable to support various IoT devices and use scenarios and achieves both data service fairness and spectrum utilization efficiency. We finally implement and evaluate it using the OAI platform.

7.1 Anti-Abuse Service Model

The major idea of this service model is to serve each cellular-connected device with service quality based on its cellular IoT technology category and the device access fee paid by its owner. This can prevent different charges on the same quality of services that the adversary can abuse. Our model consists of two components: operational IoT service consistency and charge-aware service access control. They together ensure that no IoT users can get better service quality than non-IoT users when the IoT users pay less. Note that this assurance cannot be achieved by simple IoT service throttle mechanisms (e.g., limiting data rates to 1 Mbps), since the available data rates of all the devices can be smaller than the IoT rate limits in practice.

7.1.1 Operational IoT Service Consistency

With distinct cellular IoT technologies, IoT devices have different capabilities in terms of theoretical maximum uplink/downlink speed. For example, for an IoT device supporting CAT-M1, the theoretical maximum uplink/downlink speed is 1 Mbps/1 Mbps, whereas for an IoT device supporting CAT-1, the theoretical maximum uplink/downlink speed is 5 Mbps/10 Mbps. However, in practice, different entities including IoT devices, SIM cards, and the network infrastructure do not operate in consistency with the cellular IoT profiles. That is, the network may not restrict the performance of the IoT SIM cards based on their profiles. This leads to the gains which the adversary can get by the IoT masquerading. We thus propose that all the parties in the cellular ecosystem shall be consistent with the support of the IoT profiles. For example, when an IoT user subscribes to an IoT sim card for his/her CAT-1 IoT device, the maximum uplink/downlink speed of the CAT-1 IoT SIM card shall be limited to 5 Mbps/10 Mbps by the network no matter what device is used for the SIM card. Therefore, even if the adversary performs the IoT masquerading on a non-IoT device using the IoT SIM card, the device can get only 10 Mbps as its maximum speed.

This service consistency mechanism contains two major tasks in the core network operation. First, the network infrastructure should maintain maximum uplink/downlink speed information for each IoT service subscription based on its subscribed cellular IoT technology category. Second, it should apply the maximum speed to the EPS bearer context activation procedure [13], which is initiated when an IoT device accesses the IoT service with which the SIM card is associated.

7.1.2 Charge-aware Service Access Control

Due to fewer resources needed for IoT services, carriers inevitably provide them with cheaper charge plans than conventional non-IoT plans. However, they do not restrict the IoT services from the network but only rely on the inherent constraints of IoT devices. This is why the adversary can abuse the IoT devices to have non-IoT services with cheaper charges. We argue that these differential charges shall be reflected in the service quality which includes traffic priority and maximum transmission rate. This causes the gaps between IoT and non-IoT services to correlate with their charges, thereby reducing attack incentives. We next

elaborate on how to correlate the charges with the priority and the maximum rate.

In the LTE network, there are 9 priority levels, which are assigned to different types of traffic [24]. The level number decreases with the increase of priority. For example, the signaling and voice traffic flows of VoLTE (Voice over LTE) respectively have levels 5 and 1, whereas the flows of mobile data services on non-IoT devices are usually given the level 9, which is the lowest priority. Since IoT services are cheaper, their traffic flows should have lower priority than level 9. We then propose to use the level ranging from 9 to 10 to set priority for IoT services and correlate it to their differential charges.

The priority value for an device X can be formulated as:

$$Priority_X = 10.0 - \frac{Charge_X}{Charge_{Highest}}, \quad (1)$$

where $Charge_X$ is the device access fee of device X and $Charge_{Highest}$ is the highest device access fee among the devices in the same type of data plan (e.g., limited or unlimited data plan) in the same network. For example, in a 2GB limited data plan, \$20 and \$5 are charged for a smartphone and an LG smartwatch, respectively. Their priority levels should be set to 9 and 9.75 (i.e., $10.0-\$5/\$20$). The cheaper a device's access fee a user pays, the lower the priority of device traffic flows (s)he can receive. Note that we elaborate on how to set various priority levels in Section 7.2.

We next restrict maximum uplink/downlink transmission rates for IoT devices. We determine the maximum transmission rate of each device by considering both its priority value and the maximum rate given by the operational IoT service consistency. Assume that the maximum rate for non-IoT devices in the same type of data plan is $NonIoTMaxRate$ and the maximum IoT rate from the service consistency is $InitialMaxRate_X$. Then the maximum rate for the IoT device X is formulated as:

$$MaxRate_X = Min(NonIoTMaxRate \times (10 - Priority_X), InitialMaxRate_X). \quad (2)$$

Take the LG smartwatch as an example. Since its device access fee is \$5 and the smartphone's is \$20 in the OP-II network, its service priority and maximum rate are respectively 9.75 and 25% of the maximum rate that the smartphone can receive when $NonIoTMaxRate\times$ 0.25 is smaller than $InitialMaxRate_{watch}$.

7.1.3 Computational Complexity Analysis

We next analyze computational complexity of the operational IoT service consistency and the charge-aware service control. We consider the time complexity of associating a new IoT service subscription with its transmission capability. After the association, the network can easily apply the transmission capability to an IoT device based on its SIM profile when it attaches to the network. In the analysis, we assume that (1) the GSMA's and operators' IMEI and SIM card databases are maintained based on the B+ tree [25] (B+ tree is a common data structure used by database systems, such as mySQL), (2) the time complexity of performing an arithmetic operation, such as subtraction, multiplication, division, is $\mathcal{O}(1)$, and (3) the time complexity of reading/writing an item in GSMA or operators' databases is $\mathcal{O}(1)$.

Operational IoT service consistency: Making the service consistent consists of three main steps. First, the network obtains the information of cellular IoT technologies that the device can support based on its IMEI, which is collected from the device owner. It can be queried from the GSMA's global central IMEI database (https://imeidb.gsma.com), which stores all the IMEIs with device profiles, such as manufacturers and software/hardware capabilities. The time complexity of the search operation on a B+ tree database is $\mathcal{O}(\log n)$ [25], where n is the number of global mobile devices stored in GSMA's IMEI database. Second, the network identifies the theoretical maximum uplink/downlink speed of the supported IoT technologies. It takes $\mathcal{O}(\alpha)$, where α is the number of various cellular IoT technologies. Third, the network associates the device's IoT SIM card with the transmission capability, and adds it into the SIM card database. The time complexity of a B+ database insertion is $\mathcal{O}(\log \beta)$, where β is the number of active SIM cards stored in the operator's SIM card database. In summary, the total time complexity is $\mathcal{O}(\log n)+\mathcal{O}(\alpha)+\mathcal{O}(\log \beta)$. Since the time complexity related to the number of global mobile devices can dominate in practice, the time complexity for operational IoT service consistently mechanism can be reduced to $\mathcal{O}(\log n)$.

Charge-aware service access control: This module takes three major steps to add a new IoT service subscription. First, it obtains the highest charge among all the devices in the same type of data plans. The time complexity is $\mathcal{O}(\beta')$, where β' is the number of active SIM cards in the type of data plan to which the IoT user subscribes (e.g., limited data plan). Since, in practice, β' is smaller than β (i.e., the number of all active SIM cards that the operator currently support), we can reduce $\mathcal{O}(\beta')$ to $\mathcal{O}(\beta)$. Second, it obtains the maximum rate of non-IoT devices. It takes only a constant time $\mathcal{O}(1)$, since carriers, including OP-I and OP-II, usually apply the same maximum rate to all non-IoT devices. Third, it calculates the IoT subscription's priority value and then determines the final maximum rate according to Equation 2. The calculation costs only a constant time $\mathcal{O}(1)$. In summary, the total time complexity is $\mathcal{O}(\beta) + \mathcal{O}(1) + \mathcal{O}(1)$ and can be reduced to $\mathcal{O}(\beta)$.

Overall complexity: As a result, the overall time complexity is $\mathcal{O}(\log n) + \mathcal{O}(\beta)$, where n is the number of global mobile devices including cellular IoT devices and β is the number of active SIM cards that the operator currently support. In practice, n is much larger than β.

7.1.4 Merits

We next summarize three major merits of the anti-abuse service model. First, the model does not require any modification to cellular IoT standards or devices, since its two components can be carried out in the standard EPS bearer context activation procedure [13], which is initiated by the infrastructure when an IoT device accesses the IoT services. Second, it can be scalable to support a variety of devices and use scenarios, as it does not require calibration of the IoT service rates for various devices and use scenarios. This is especially relevant with more and more devices being

introduced in the future. Third, it can achieve both data service fairness and spectrum utilization efficiency. For the fairness, it can guarantee that no IoT devices can get better services than non-IoT devices when the IoT owners pay less. For the efficiency, IoT devices still have chances to achieve their maximum speeds when radio resources are sufficient (e.g., no contention comes from non-IoT devices). Note that for limited IoT data plan users, the more data that IoT users use, the more profit that carriers can make.

7.2 Implementation

We implement the anti-abuse model on the OAI platform. It consists of the 4G core network and RAN. The 4G core network runs on a laptop (Acer Aspire E5-575-53EJ). The RAN contains the eNodeB on a PC (Dell Inspiron 3268) and a software-defined radio (USRP B210). We mainly modify three entities: the HSS, the MME, and the eNodeB (see Figure 1).

HSS. We add two types of new information in the user subscription data, which are associated with each SIM card: user equipment profile and charge rate class. The former indicates the highest technology category (*e.g.*, CAT-4) that the SIM card can support. The latter represents the operator-specific charge rate class (*e.g.*, 25% off, 50% off) to which the SIM belongs. These are used by the MME to determine service priority for the SIM. We add the delivery of this information to the normal procedure that the MME has to obtain user authentication information from the HSS. The new information entries are included in an element `UE-Usage-Type` of the response to the request *Authentication Information Retrieval*, which is sent from the MME to the HSS.

MME. The maximum uplink/downlink rates and the service priority are set for each SIM card based on that new information provided by the HSS. We introduce new QoS to the EPS radio access bearer (E-RAB). During the *E-RAB Setup* procedure [26], the MME specifies those two restrictions in the fields, `UE Aggregate Maximum Bit Rate` [26] and `E-RAB Level QoS Parameters` [26], respectively in the *E-RAB Setup Request* message, which is sent to the eNodeB.

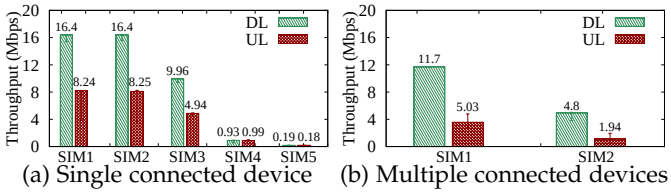
eNodeB. We support new priority levels (*e.g.*, 9.25 and 9.5) by defining new QoS Class Identifier (QCI) values, which are used to represent QoS classes in the LTE network. Each QCI value is an 8-bit unsigned octet. The QCI values, ranging from 128 to 254, are reserved for operator-specific usage, so new QCI values can be added in this range. In our implementation, we define two new priority levels 9.25 and 9.5 by adding new QCI values 129 and 130, respectively. Note that the eNodeB in the current OAI implementation does not support full QCI functions specified by the standards. We thus add a Service Control Entity (SCE), which is a Linux server, between the eNodeB and the 4G core to fulfill the regulation of the maximum rates and the service priority.

7.3 Evaluation

We evaluate our solution based on the OAI-based prototype. We use 5 sysmoUSIM-SJS1 SIM cards, which are standard-compliant, and add their information to the HSS database.

SIM	Highest Theoretical UE DL/UL speed	Priority Value	Mapped Operator Plan
SIM1	CAT-10 (450 Mbps/150 Mbps)	9	Non-IoT, $20
SIM2	CAT-4 (150 Mbps/50 Mbps)	9.5	IoT, $10
SIM3	CAT-1 (10 Mbps/5 Mbps)	9.5	IoT, $10
SIM4	CAT-M1 (1 Mbps/1 Mbps)	9.75	IoT, $5
SIM5	NB-IoT (0.2 Mbps/0.2 Mbps)	9.75	IoT, $5

TABLE 5: The configurations of our test SIM cards.



(a) Single connected device (b) Multiple connected devices

Fig. 10: Maximum, median and minimum uplink/downlink speeds vary with SIM cards.

They are configured to have five different categories (*i.e.*, CAT-10, CAT-4, CAT-1, CAT-M1 and NB-IoT), and classified into three priority classes: 9, 9.5, and 9.75. The device access fees of those three priority classes are respectively 0%, 50% and 75% cheaper than non-IoT devices. These configurations are summarized in Table 5. We use the `iPerf` tool to assess throughput of user devices.

Operational IoT service consistency. We use one device (Nexus 6p) with different SIM cards shown in Table 5 to assess the operational consistency for IoT profiles. We test both uplink and downlink speed performance. The test on each SIM card has 10 runs with 30 seconds each. Figure 10a shows maximum, median and minimum downlink/uplink speed results for the SIM cards. There are two observations. First, the maximum throughput results of SIM1 and SIM2 are similar (*i.e.*, 8 Mbps and 16 Mbps for uplink and downlink, respectively), because they are bound by the OAI platform's maximum throughput, which is smaller than their maximum speeds. Second, for the other three SIM cards, the maximum uplink/downlink speeds are 4.94 Mbps/9.96 Mbps (SIM3), 0.99 Mbps/0.93 Mbps (SIM4) and 0.18 Mbps/0.19 Mbps (SIM5), respectively. They are bound by the regulated maximum speeds of the cellular IoT technologies.

Charge-aware service control. We next examine whether the service priority control can take effect in the prototype. We use two phones, Nexus 6p and Samsung S5, with SIM1 and SIM2, respectively. Both phones have much larger maximum downlink/uplink throughput than the OAI platform's throughput bottleneck. The service priority levels assigned to them are respectively 9 and 9.5 based on the priority classes. We have 10 runs for each test. In each run, we generate traffic to gauge throughput performance on them simultaneously, and examine how they affect each other. Figure 10b plots maximum, median and minimum uplink/downlink results. It is observed that the maximum throughput results for Nexux 6p with SIM1 and Samsung S5 with SIM2 are 5.03 Mbps/11.7 Mbps and 1.94 Mpbs/4.8 Mbps, respectively. It confirms that the service flows of Nexus 6P with priority level 9 have higher priority than those of Samsung S5.

8 Discussion

We next discuss several concerns.

Why not just limit the rates or usage of IoT services? This is not practical in two aspects. First, for the limited IoT data plans, carriers do not have incentives to limit rate/usage, since no limitations of using data can make more profit for the carriers. Second, for the unlimited data plans, though carriers may be willing to limit rate/usage, it is challenging to determine rate caps which can satisfy various IoT users and scenarios. For example, it is not easy for carriers to determine a proper rate/usage cap for a car-connected mobile hotspot since it provides similar functions as a portable mobile hotspot. A lower rate/usage cap may reduce users' willingness to subscribe to IoT services, but a higher rate/usage cap can increase attack incentives.

Simple pricing policy issues? People may think that our findings are carrier-specific pricing policy issues instead of security loopholes. However, this is not the case. The fundamental issues are rooted in 3GPP security design flaws, where there is a lack of mutual authentication between SIM cards and devices, as well as that between devices and the infrastructure. These flaws can be exploited by the adversary to launch a variety of attacks. For example, the adversary can connect an SDR-based mobile device that is not carrier-certificated and flood many control-plane spam messages to the network. Our cellular IoT charging-based attack is just one instance of them.

Short-lived and IoT-technology-specific issues? Our identified issues are independent of IoT technologies, but are rooted in device authentication, access control and IoT charging security. No evidence indicates that they will be addressed in those security components for upcoming IoT technologies (e.g., NB-IoT).

How about 5G IoT? According to the latest 5G security standards [27], the mutual authentication mechanisms between IoT SIM cards, IoT devices, and the network infrastructure are still missing. Thus, 5G carriers will still suffer from the insecure IoT data charging if they apply differential charging schemes to IoT and non-IoT users.

Does the embedded SIM (eSIM) address all issues? The eSIM is still in its early deployment phase, and the removable SIM will not be phased out in the foreseeable future. Moreover, using the eSIM cannot address all identified security issues, *e.g.*, the IoT use scenario abuse.

Carrier-dependent issues? Some vulnerabilities are carrier dependent (*e.g.*, no limits on IoT data speeds), but they are not the critical ones (e.g., device authentication) for the security threats. They can only affect the degree of attack damage.

9 Related Work

Mobile Security. Mobile security has been an active research area in recent years. Researchers mainly study the security vulnerabilities of mobile data service charging, mobile devices, mobile network infrastructure, and mobile applications/services. Some interesting findings are reported, which include the anonymization of the SIP protocol [28], design flaws of mobile operating systems (e.g., Android and iOS) [29]–[31], charging attacks of mobile data services [3]–[5], [7], [32], spam and fraudulence attacks through text and voice services [33], [34], vulnerable usage of Android Internet sockets [35], vulnerabilities of VoWiFi [36] to name a few. Most of the early research works target non-IoT devices (e.g., smartphones), as well as their mobile applications and services. However, our work focuses on cellular IoT devices instead of smartphones, tablets or other non-IoT mobile devices.

IoT Security. Current research studies can be categorized into three dimensions: (1) device software and hardware, (2) network protocols, and (3) security architecture. In the first dimension, a study [37] shows that an IoT botnet based on the Mirai malware [38] is able to launch a 600 Gbps traffic attack. Another work [39] presents a threat that adversaries can compromise smart meters to reduce their utility bills. Liu *et al.* [40] propose an ARM TrustZone based virtual sensing system to enable a safe, isolated environment for IoT devices. Gao *et al.* [41] develop an easy access solution for authenticated users to access the voice-based assistants. Ding *et al.* [42] discover possible physical interactions and generates all potential interaction chains across applications in IoT environment.

For the IoT network protocols, Sastry *et al.* [43] discover several security vulnerabilities and pitfalls in IEEE 802.15.4, which is designed for wireless communication among low-power IoT devices. Soltan *et al.* [44] and Herwig *et al.* [45] study the IoT botnet and analyze its attacks on power grids and peer-to-peer networks.

For the IoT security architecture, some novel security mechanisms have been proposed, e.g., data-origin authentication, integrity verification, privacy preserving, and identity-based encryption. Jia *et al.* [46] propose `ContexIoT`, a context-based permission system for IoT platforms. It provides contextual integrity [47] and implements it on the Samsung SmartThings platform. Das *et al.* [48] propose a deep-learning based classifier for IoT authentication. Harris *et al.* [49] propose to protect user data against leakage by adopting the CryptoCoP-based encryption and a unique MAC address rotation mechanism. Wang *et al.* [50] conduct an analysis of the IFTTT and enumerate the inter-rule vulnerabilities that exist within trigger-action platforms. Haddadi *et al.* [51] introduce the SIOTOME architecture between the network edge and the ISP to defend against attacks from compromised IoT devices. Memos *et al.* [52] study the security challenges of the upcoming IoT network architecture, and media security and privacy in wireless sensor networks (WSNs) and develop an efficient algorithm for media-based surveillance systems in IoT network for smart city framework. Stergiou *et al.* [53] do the security survey of IoT and Cloud Computing and show the security challenges of the integration of IoT and Cloud Computing. Celik *et al.* [54] present a policy-based enforcement system IoTGuard for IoT, which protects users from unsafe and insecure states. Moreover, some researchers focus on improving the efficiency of the systems leveraging the blooming of IoT devices (e.g., media-based IoT devices such as security camera and senors) and cloud computing to secure our society. For example, Psannis *et al.* [55] develop an efficient algorithm for encoding advanced scalable

media-based smart big data on intelligent cloud computing systems, which can efficiently process the smart big data generated by a great number of media-based IoT devices (e.g., security camera). Stergiou *et al.* [56] leverage the blooming of IoT in cloud computing to develop a new type of network for intelligent media-data transfer. Plageras [57] investigates new systems for efficiently collecting and managing sensors data in a smart building by leveraging IoT, big data, cloud computing, and monitoring technologies. Different from them, we here focus on the security of cellular IoT devices and their charging functions in the operational 4G LTE networks.

10 Conclusion

The cellular IoT is thriving and being deployed worldwide. The security of the cellular IoT is playing an important role in its development, but has not been fully explored yet. In this work, we examine the security implications in the service charging scheme of the cellular network. We show that the cellular IoT charging can be exploited to launch attacks against carriers. The adversary can gain 43.75%-80.00% cheaper bills on cellular data services by masquerading non-IoT devices as IoT devices and abusing them in unanticipated use scenarios. The fundamental issue lies in that no sufficient security manners, which include mutual authentications between involved cellular entities, support differential charges between non-IoT and IoT devices. In light of heavy burdens on standard modification, we propose an anti-abuse solution to mitigate attack incentives instead of addressing the vulnerabilities directly. It can be used immediately in practice so as to benefit carriers on securing the cellular IoT ecosystem. We hope that our preliminary study can also stimulate new security designs for the cellular IoT technology in the upcoming standards.

Acknowledgments

We greatly thank Dr. Marwan Krunz, Xinyu Lei, Jiawei Li, James Mariani, and anonymous reviewers for helping us to improve this work. This work is supported in part by the National Science Foundation under Grants No. CNS-1814551, CNS-1815636, and CNS-1749045, the Ministry of Science and Technology in Taiwan under Grants No. 106-2628-E-009-003-MY3 and 108-2218-E-009-045, and the Center for Open Intelligent Connectivity from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education in Taiwan. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors only and do not necessarily reflect those of the National Science Foundation.

References

[1] L. Alliance, "Lora Alliance Technology," 2017, https://www.lora-alliance.org/technology.

[2] Ericsson, "Cellular networks for massive IoT," Jan. 2016, https://www.ericsson.com/res/docs/whitepapers/wp_iot.pdf.

[3] Y. Go, E. Jeong, J. Won, Y. Kim, D. F. Kune, and K. Park, "Gaining control of cellular traffic accounting by spurious TCP retransmission," in *NDSS*. The Internet Society, 2014.

[4] C. Peng, C.-Y. Li, H. Wang, G.-H. Tu, and S. Lu, "Real threats to your data bills: Security loopholes and defenses in mobile data charging," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 727–738.

[5] C. Peng, G.-h. Tu, C.-y. Li, and S. Lu, "Can we pay for what we get in 3g data access?" in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 113–124.

[6] C.-Y. Li, G.-H. Tu, C. Peng, Z. Yuan, Y. Li, S. Lu, and X. Wang, "Insecurity of voice solution volte in lte mobile networks," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15. New York, NY, USA: ACM, 2015, pp. 316–327. [Online]. Available: http://doi.acm.org/10.1145/2810103.2813618

[7] C. Peng, G. Tu, C. Li, and S. Lu, "Can we pay for what we get in 3g data access?" in *ACM Mobicom*, 2012.

[8] T. Tirronen, "Cellular IoT Alphabet Soup," Feb. 2016, https://goo.gl/HmEgN7.

[9] AT&T, "Low cost LTE modules for the Internet of Things," 2016, https://goo.gl/4g8AJW.

[10] Nokia, "LTE evolution for IoT connectivity," 2017, https://onestore.nokia.com/asset/200178.

[11] A. Leckie, "LTE Category-0 & LTE-M low power M2M device roadmaps," May 2015, https://goo.gl/cEGHDE.

[12] AT&T, "At&t iot plan," https://marketplace.att.com/iot-connectivity, 2018.

[13] 3GPP, "TS 24.301: Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3," 2011.

[14] GSMA, "LTE-M Deployment Guide to Basic Feature Set Verison 2.0," April 2018. [Online]. Available: https://gsma.com/newsroom/wp-content/uploads//CLP.29-v2.0.pdf

[15] "Google project fi," https://fi.google.com/about/plan/, 2018.

[16] Statista, "Market share of mobile network carriers in the U.S," 2017, http://www.statista.com/statistics/199359/market-share-of-wireless-carriers-in-the-us-by-subscriptions/.

[17] lyriquidperfection, "EFS Professional," Aug. 2011, https://goo.gl/wQKJ59.

[18] "Mobileinsight," 2019, http://www.mobileinsight.net/.

[19] T. M. Khoshgoftaar, S. V. Nath, S. Zhong, and N. Seliya, "Intrusion detection in wireless networks using clustering techniques with expert analysis," in *Fourth International Conference on Machine Learning and Applications (ICMLA'05)*. IEEE, 2005, pp. 6–pp.

[20] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*. ACM, 2008, pp. 116–127.

[21] K. B. Rasmussen and S. Capkun, "Implications of radio fingerprinting on the security of sensor networks," in *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops-SecureComm 2007*. IEEE, 2007, pp. 331–340.

[22] D. R. Reising, M. A. Temple, and M. J. Mendenhall, "Improving intra-cellular security using air monitoring with rf fingerprints," in *2010 IEEE Wireless Communication and Networking Conference*, April 2010, pp. 1–6.

[23] "Openairinterface," 2018, http://www.openairinterface.org/.

[24] 3GPP, "TS23.203: Policy and charging control architecture," Dec. 2017.

[25] "B+ tree," https://en.wikipedia.org/wiki/B

[26] 3GPP, "TS36.413: Evolved Universal Terrestrial Radio Access Network (E-UTRAN); S1 Application Protocol (S1AP)," 2011.

[27] ——, "TS33.501: Security architecture and procedures for 5G system; (Release 15)," Sep. 2018, https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3169.

[28] I. Leontiadis, C. Delakouridis, L. Kazatzopoulos, and G. F. Marias, "Anosip: anonymizing the sip protocol," in *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*. ACM, 2012, p. 4.

[29] H. Zhang, D. She, and Z. Qian, "Android ION hazard: the curse of customizable memory management system," in *ACM CCS*, 2016.

[30] Y. Shao, Q. A. Chen, Z. M. Mao, J. Ott, and Z. Qian, "Kratos: Discovering inconsistent security policy enforcement in the android framework," in *IEEE NDSS*, 2016.

[31] X. Zhou, Y. Lee, N. Zhang, M. Naveed, and X. Wang, "The peril of fragmentation: Security hazards in android device driver customizations," in *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, ser. SP '14. Washington, DC, USA:

IEEE Computer Society, 2014, pp. 409–423. [Online]. Available: http://dx.doi.org/10.1109/SP.2014.33

[32] C.-Y. Li, G.-H. Tu, C. Peng, Z. Yuan, Y. Li, S. Lu, and X. Wang, "Insecurity of voice solution volte in lte mobile networks," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 316–327.

[33] G.-H. Tu, C.-Y. Li, C. Peng, Y. Li, and S. Lu, "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*. ACM, 2016, pp. 1118–1130.

[34] H. Tu, A. Doupé, Z. Zhao, and G. Ahn, "Sok: Everyone hates robocalls: A survey of techniques against telephone spam."

[35] W. Bu, M. Xue, L. Xu, Y. Zhou, Z. Tang, and T. Xie, "When program analysis meets mobile security: an industrial study of misusing android internet sockets," in *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*. ACM, 2017, pp. 842–847.

[36] T. Xie, G.-H. Tu, C.-Y. Li, C. Peng, J. Li, and M. Zhang, "The dark side of operational wi-fi calling services," in *2018 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2018.

[37] E. Bertino and N. Islam, "Botnets and internet of things security," *Computer*, vol. 50, no. 2, pp. 76–79, Feb 2017.

[38] "Mirai Malware for Botnet," 2017, https://en.wikipedia.org/wiki/Mirai_(malware).

[39] B. Krebs, "FBI: Smart Meter Hacks Likely to Spread," 2012, https://krebsonsecurity.com/2012/04/fbi-smart-meter-hacks-likely-to-spread/.

[40] R. Liu and M. Srivastava, "Virtsense: Virtualize sensing through arm trustzone on internet-of-things," in *Proceedings of the 3rd Workshop on System Software for Trusted Execution*. ACM, 2018, pp. 2–7.

[41] C. Gao, V. Chandrasekaran, K. Fawaz, and S. Banerjee, "Traversing the quagmire that is privacy in your smart home," in *Proceedings of the 2018 Workshop on IoT Security and Privacy*. ACM, 2018, pp. 22–28.

[42] W. Ding and H. Hu, "On the safety of iot device physical interaction control," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 832–846.

[43] N. Sastry and D. Wagner, "Security considerations for ieee 802.15. 4 networks," in *Proceedings of the 3rd ACM workshop on Wireless security*. ACM, 2004, pp. 32–42.

[44] S. Soltan, P. Mittal, and H. V. Poor, "Blackiot: Iot botnet of high wattage devices can disrupt the power grid," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 15–32.

[45] S. Herwig, K. Harvey, G. Hughey, R. Roberts, and D. Levin, "Measurement and analysis of hajime, a peer-to-peer iot botnet." in *NDSS*, 2019.

[46] Y. J. Jia, Q. A. Chen, S. Wang, A. Rahmati, E. Fernandes, Z. M. Mao, and A. Prakash, "ContexIoT: Towards Providing Contextual Integrity to Appified IoT Platforms," in *IEEE NDSS*, 2017.

[47] H. Nissenbaum, "PRIVACY AS CONTEXTUAL INTEGRITY," 2004, https://crypto.stanford.edu/portia/papers/RevnissenbaumDTP31.pdf.

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

[49] A. F. Harris, H. Sundaram, and R. Kravets, "Security and privacy in public iot spaces," in *Computer Communication and Networks (ICCCN), 2016 25th International Conference on*. IEEE, 2016, pp. 1–8.

[50] Q. Wang, P. Datta, W. Yang, S. Liu, A. Bates, and C. A. Gunter, "Charting the attack surface of trigger-action iot platforms," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2019, pp. 1439–1453.

[51] H. Haddadi, V. Christophides, R. Teixeira, K. Cho, S. Suzuki, and A. Perrig, "Siotome: An edge-isp collaborative architecture for iot security," *Proc. IoTSec*, 2018.

[52] V. A. Memos, K. E. Psannis, Y. Ishibashi, B.-G. Kim, and B. B. Gupta, "An efficient algorithm for media-based surveillance system (eamsus) in iot smart city framework," *Future Generation Computer Systems*, vol. 83, pp. 619–628, 2018.

[53] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta, "Secure integration of iot and cloud computing," *Future Generation Computer Systems*, vol. 78, pp. 964–975, 2018.

[54] Z. B. Celik, G. Tan, and P. D. McDaniel, "Iotguard: Dynamic enforcement of security and safety policy in commodity iot." in *NDSS*, 2019.

[55] K. E. Psannis, C. Stergiou, and B. B. Gupta, "Advanced media-based smart big data on intelligent cloud systems," *IEEE Transactions on Sustainable Computing*, vol. 4, no. 1, pp. 77–87, 2018.

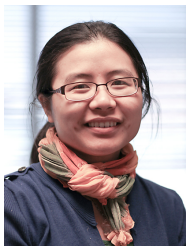
[56] C. Stergiou, K. E. Psannis, A. P. Plageras, Y. Ishibashi, B.-G. Kim *et al.*, "Algorithms for efficient digital media transmission over iot and cloud networking," *Journal of Multimedia Information System*, vol. 5, no. 1, pp. 27–34, 2018.

[57] A. P. Plageras, K. E. Psannis, C. Stergiou, H. Wang, and B. B. Gupta, "Efficient iot-based sensor big data collection–processing and analysis in smart buildings," *Future Generation Computer Systems*, vol. 82, pp. 349–357, 2018.

Tian Xie received the Bachelor degree with honor in Electrical and Computer Engineering from Michigan State University, USA, in 2016. Currently, he is working toward the Ph.D. degree in Computer Science and Engineering at Michigan State University. His research focuses on mobile networks, mobile systems, mobile IoT, and network security.

Guan-Hua Tu received the Ph.D. degree in computer science from the University of California, Los Angeles (UCLA), CA, USA, in 2015. He is currently an Assistant Professor in the Computer Science and Engineering Department at Michigan State University, MI, USA. His research interests cover focus on mobile networks, mobile IoT, wireless networking, and network security.

Chi-Yu Li received the Ph.D. degree in computer science from the University of California, Los Angeles (UCLA), CA, USA, in 2015. He is currently an Assistant Professor in the Department of Computer Science at NCTU. His research interests include wireless networking, mobile networks and systems, and network security.



Chunyi Peng received the Ph.D. degree in computer science from the University of California, Los Angeles (UCLA), CA, USA, in 2013. She is currently an Assistant Professor in the Department of Computer Science at Purdue University. Her current research interests are in the broad areas of mobile networking, system and security, with a recent focus on innovating 5G/4G mobile network architecture/protocol/technologies for demanding apps (performance and reliability), mobile network data analytics and inference,