IMS is Not That Secure on Your 5G/4G Phones

Jingwen Shi[†]*, Sihan Wang[†]*
Min-Yue Chen[†], Guan-Hua Tu[†], Tian Xie[‡],
Man-Hsin Chen[△], Yiwen Hu[†], Chi-Yu Li[△], Chunyi Peng[◇]
[†]Michigan State University, [‡]Utah State University,
[△]National Yang Ming Chiao Tung University, [◇]Purdue University

ABSTRACT

IMS (IP Multimedia Subsystem) is vital for delivering IP-based multimedia services in mobile networks. Despite constant upgrades by 3GPP over the past two decades to support heterogeneous radio access networks (e.g., 4G LTE, 5G NR, and Wi-Fi) and enhance IMS security, the focus has primarily been on cellular infrastructure. Consequently, IMS security measures on mobile equipment (ME), such as smartphones, lag behind rapid technological advancements. Our study reveals that mandated IMS security measures on ME fail to keep pace, resulting in new vulnerabilities and attack vectors, including denial of service (DoS) across all networks, named SMS source spoofing, and covert communications over Video-over-IMS attacks. All vulnerabilities and proof-of-concept attacks have been experimentally validated in operational 5G/4G networks across various phone models and network operators. Finally, we propose and prototype standard-compliant remedies for these vulnerabilities.

CCS CONCEPTS

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

KEYWORDS

Cellular Networks, IP Multimedia Services (IMS), Security

ACM Reference Format:
Jingwen Shi[†]*, Sihan Wang[†] and Min-Yue Chen[†], Guan-Hua Tu[†], Tian Xie[‡],, Man-Hsin Chen[△], Yiwen Hu[†], Chi-Yu Li[△], Chunyi Peng[◇]. 2024. IMS is Not That Secure on Your 5G/4G Phones. In *The 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24), September 30-October 4, 2024, Washington D.C., DC, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3636534.3649377

*The first two authors contributed equally to this work.

ACM MobiCom '24, November 18-22, 2024, Washington D.C., DC, USA

ACM ISBN 979-8-4007-0489-5/24/09
https://doi.org/10.1145/3636534.3649377

1 INTRODUCTION

The IP Multimedia Subsystem (IMS) delivers IP multimedia services, such as voice/video calling and texting, to mobile users over cellular networks. In the past two decades, IMS services have been augmented to support various access networks, incorporating VoLTE (Voice over LTE), VoNR (Voice over New Radio), and VoWi-Fi (Voice over Wi-Fi). IMS security is also enhanced with a suite of well-examined mechanisms including 5G/4G AKA (Authentication and Key Agreement), cellular-specific multi-layer security, and IMS media security. Specifically, secret keys required for IMS sessions [12] are derived from the AKA mutual authentication, wireless transmission in the air (Layer 2) is encrypted using the derived keys, and IP session (Layer 3) is secured by Internet Protocol Security (IPsec) [14]. Moreover, network operators enforce additional measures such as STIR/SHAKEN [45] required by FCC for caller ID authentication, protecting IMS services from malicious attacks.

However, these security enhancements are primarily centered on cellular network infrastructure. Our security analysis reveals that security measures on the mobile equipment (ME) side have remained relatively unchanged over the years. There are many advances on ME; for example, smartphone vendors have migrated the IMS client from 5G/4G modem chips to application processors and segregated IMS voice and video media processing within modem chips and application processors. Unfortunately, we find that 3GPP-mandated IMS security measures on the ME side fail to keep pace with device-side technological advances, resulting in new security vulnerabilities and unprecedented attacks.

Our security analysis on ME shows neither IMS media sessions nor their control signaling are well protected. Specifically, we discover four new vulnerabilities: (V1) unprotected IMS signaling routing, (V2) unrestricted IMS signaling source, (V3) unprotected video data delivery, and (V4) unrestricted source for IMS video delivery. Details are elaborated in §4 and §5. By exploiting these vulnerabilities on ME, we further develop three proof-of-concept attacks against IMS services: (A1) Denial of Service over All Networks (DoS-ALL),

Category	Vulnerability	Description	Proof-of-concept Attacks	Empirical Validation		
				Carrier	Device	OS
Unprotected ME Routing For IMS Client Signaling (§4)	V1. Unprotected IMS Signaling Routing	ME does not ensure that all outgoing IMS signaling messages are sent to the IMS servers deployed by network operators; Routing to malicious programs at the ME is allowed. (§4.1)	**[A1] DoS-ALL**, a novel DoS attack that prevents IMS clients from using all access networks over Wi-Fi, 4G LTE, 5G NR. (§4.3.1) **[A2] NameSpoofing**, an SMS spoofing attack fabricates the sender's name, which is prohibited by the network. (§4.3.2)	US-I, US-II, US-III, TW-I, TW-II	LG(G3,G7), TCL(40XL), Samsung (S8,S10,S21)	Android 4.4.2, 7, 8, 9, 11, 13
	V2. Unrestricted IMS Signaling Source	ME does not protect IMS client software from receiving IMS signaling messages originated from non-IMS servers (say, local apps). (§4.2)				
Insecure ME Access for IMS Media Sessions (§5)	V3. Unprotected Video Data Delivery	The IMS media transmission between IMS client and cellular network modem is not provided with confidentiality and integrity protection. (§5.1)	**[A3] ViIMS-ANY**, an attack that abuses ViIMS as a covert communication channel between two malicious MEs, bypassing operator policies. (§5.3)	US-I†, US-II†	LG (G3), Samsung (S8,S10), Google (Pixel 1/3/5/7)	Android 4.4.2, 7, 8.1, 10, 13
	V4. Unrestricted Source for IMS Video Delivery	Cellular network modem cannot verify whether IMS video data is transmitted by IMS clients or other non-IMS applications. (§5.2)				

†: ViIMS experiments were conducted in US-I and US-II because US-III supports ViIMS only with very limited phone models; TW-I and TW-II do not support ViIMS yet.

Table 1: Summary of four vulnerabilities and three proof-of-concept attacks in this work.

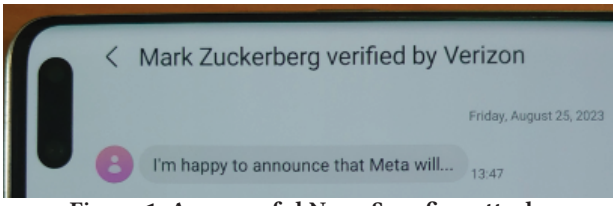


Figure 1: A successful NameSpoofing attack.

(A2) Named SMS Source Spoofing (NameSpoofing), and (A3) Covert Communications over Video-over-IMS (ViIMS-ANY).

The first DoS-ALL attack prevents the victim phones from accessing IMS services over all access networks including Wi-Fi, 4G LTE and 5G NR. It is more threatening than any DoS attacks reported before; it not only denies the IMS service access over Wi-Fi networks but also prevents access to all alternative cellular networks. The second NameSpoofing attack creates a fake short message with a fabricated sender name, which is *prohibited* by cellular infrastructure to mobile users. Figure 1 gives an illustrative example where NameSpoofing is successfully launched on our lab smartphone and the victim receives a message from "*Mark Zuckerberg verified by Verizon*". Unlike the existing SMS spoofing attacks, NameSpoofing is much more threatening because it fabricates the sender name instead of the phone number. Note that network operators do not allow SMS users to fabricate the sender's name (here, Mark Zuckerberg) even though the phone number is spoofed; more importantly, "*verified by Verizon*" cannot be added into the sender's name unless Verizon authenticates that the sender number is not spoofed and truly used by Mark Zuckerberg. It is much harder for the victims to know whether they suffer from the smishing/phishing attacks, particularly when the names are "*verified*" by network operators. In comparison to fake Amber/Wireless alert attacks [23, 31] that primarily target emergency attack scenarios, this attack is applicable to a broader range of attack scenarios.

The third ViIMS-ANY attack abuses ViIMS, which is designated for delivering video calls over IMS. ViIMS is used by two adversary MEs for any data communications, which obtains guaranteed bit rates and a higher service priority that normal data services should not have. As such, ViIMS-ANY bypasses data service policies enforced by operators.

Table 1 summarizes new vulnerabilities and attacks, which are experimentally validated using commodity phones with three top-tier U.S. carriers and two major operators in Taiwan. We further propose countermeasures to address identified vulnerabilities and evaluate their effectiveness (§6).

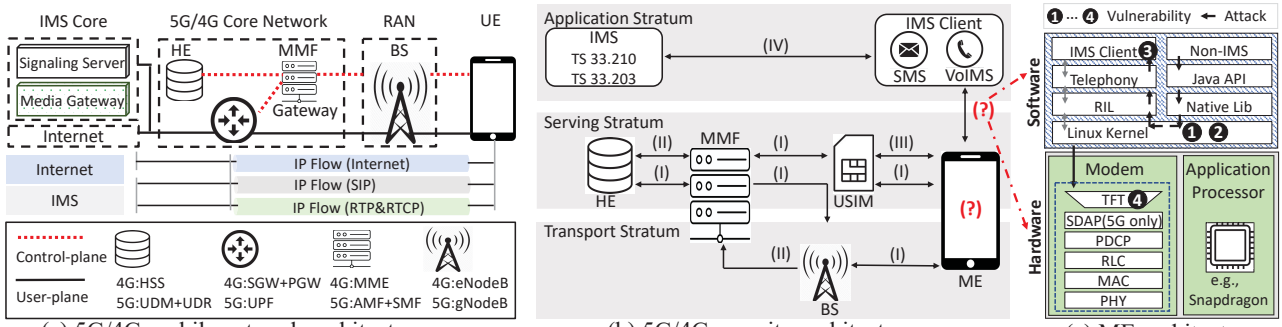
2 BACKGROUND

In this section, we first present the necessary background on 5G/4G network architecture and its security measures. We then introduce the architecture and network stack on Mobile Equipment (ME), and finally present the IMS service flow. For the sake of simplicity, we use one general term for distinct entities with equivalent functions in 4G and 5G.

5G/4G mobile network architecture. Figure 2(a) shows 5G/4G network architecture and its operations in both control-plane and user-plane. From right to left, user traffic traverses the UE (User Equipment), RAN (Radio Access Network), 5G/4G core network and Internet (mobile broadband) or IMS (voice/text). UE is the ME equipped with a valid USIM (UMTS Subscriber Identity Module); RAN uses 5G gNodeB or 4G eNodeB as the BS (Base Station) to provide radio access to the UE. In the control plane, MMF (Mobility Management Function) administrates registration, authentication, IP connectivity, and mobility,whereas HE (Home Environment) stores user data. In the user plane, GWs (Gateways) are used to forward traffic and manage IP connectivity.

To offer guaranteed network performance for each UE, multiple IP flows are created and assigned with distinct QoS levels. Specifically, one flow is established for mobile broadband service to the Internet, whereas two flows are created to support multimedia services (e.g., voice and video calls) offered by the IMS: one for signaling and the other for media traffic; they are managed by IMS signaling servers and media gateways, respectively. The IMS signaling uses Session Initiation Protocol (SIP) [19] and the media traffic is transported over Real-Time Transport Protocol (RTP) [10, 17].

5G/4G security architecture. Figure 2(b) shows that 5G/4G uses a multi-layer security architecture with three stratums: application, service, and transport. The security functions are divided into four domains [15, 16]: (I) network access



(a) 5G/4G mobile network architecture (b) 5G/4G security architecture (c) ME architecture

Figure 2: 5G/4G mobile network architecture and its security; the architecture and potential security vulnerabilities of ME.

domain, which ensures mutual authentication between the core network and the ME, as well as secure service access; (II) network domain, which guarantees secure communication among network entities; (III) user domain, which secures communication between the ME and the USIM; (IV) application domain, which protects message exchanges between ME applications and network servers (e.g., IMS). Such security architecture shows that the access between the applications and the ME is not explicitly protected.

ME architecture. Figure 2(c) illustrates the ME architecture, which includes both software and hardware components, with Android Phones serving as examples. The ME software includes OS, applications, and the user interface. The applications can be classified into IMS and non-IMS types with different protocol stacks on top of the Linux kernel, and specifically, each IMS application serving as an IMS client runs on the Telephony Framework and the RIL (Radio Interface Layer) for IMS functionalities. The ME hardware contains two major components. One is an application processor supporting the ME software, whereas the other is the cellular modem offering cellular connectivity and cellular-related services. The modem mainly contains cellular L1/L2 protocols, including PHY, MAC, RLC (Radio Link Control), and PDCP (Packet Data Convergence Protocol) for both 5G and 4G networks, as well as SDAP (Service Data Adaptation Protocol) for the 5G network only. Moreover, it contains a function, TFT (Traffic Flow Template), for associating packets with each specified IP flow based on the 5-tuple (source/destination IP addresses, source/destination port numbers, protocol ID) information so that the corresponding routing and QoS policy can be applied [6].

IMS service flows. Figure 3 depicts IMS service flows for text, voice, and video services. To access an IMS service, the UE needs to perform three actions. First, *IP Connectivity Establishment* [17] is performed to obtain IP connectivity for communicating with the IMS server. Second, *IMS Service Registration* [17] is made for service registration from the

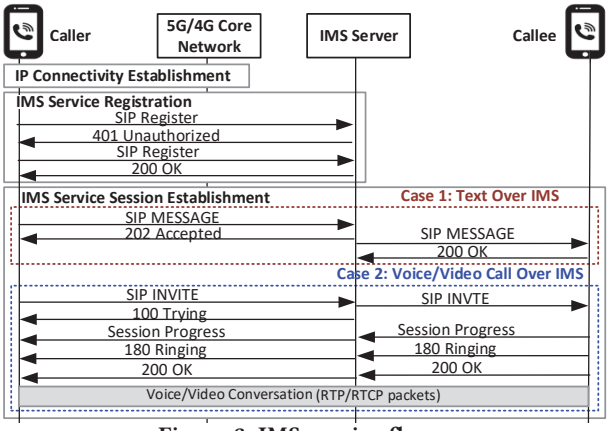


Figure 3: IMS service flows.

UE to the IMS server, but also for mutual authentication between them. It uses the SIP Registration procedure with the IMS-AKA (Authentication and Key Agreement) [13]. When the IMS signaling security is enabled, IPsec SAs (Security Associations) between the IMS server and the UE are established during the registration. Third, the UE carries out *IMS Service Session Establishment* to establish an IMS service session with another UE [17, 28] using SIP. The IMS text and call services have different establishment procedures, which are initialized with initial messages, SIP MESSAGE and SIP INVITE, respectively. In particular, to ensure carrier-grade IMS service quality, the IMS signaling with SIP messages and the IMS media traffic with RTP/RTCP packets are both prioritized over the traffic of mobile data services. Specifically, the QoS levels assigned to mobile data services are the best-effort transmission, with priority indexes ranging from 8 to 9 [9]. In contrast, those assigned to IMS signaling and media traffic are the best effort with a priority of 1 (smaller values indicate higher priority) and the guaranteed bit rate transmission, respectively.

3 VULNERABLE IMS CLIENT ON ME

To support multimedia services, the IMS client is designed differently from the traditional one offering circuit-switched

call and text services. It contains control-plane and data-plane operations. For the control plane, the IMS client is expected to support various multimedia services with a flexibility demand of being updated dynamically, so on most phone models, it uses a software-based design to function as a mobile application. Compared with the traditional one implemented in the phone modem, it has a larger attack surface and may be thus more vulnerable. It may suffer from the hijacking of the IMS signaling session due to the acquisition of root privilege [30, 33] or the delivery of spoofed signaling messages (i.e., SMS) given unprotected packets routing. In this work, we focus on the latter security threat, which has not been explored, in §4.

For the data plane, there are currently two major multimedia services: Voice over IMS (VoIMS) and ViIMS. These two services are supported in different ways according to their different processing resource requirements. The voice data of VoIMS are processed by the phone modem; thus, the corresponding voice packets cannot be captured in the mobile OS. They are inherently protected by the hardware security of the modem. Although the modem can still be compromised by some specialized tools (e.g., QXDM [37]), thereby causing its voice session to be hijacked [30], the security threat is limited since the assumption for attackers in the threat model is too strong to be practical. However, with a demand of large processing resource, a multi-core application processor is deployed to process the video data of ViIMS. This new component might lack the conventional hardware security protection from the modem, thus broadening the attack surface. This motivates us to investigate the security of the IMS data-plane framework in §5.

We next present threat model and experimental methodology, ethical consideration, as well as responsible disclosure.

Threat model. For the control-plane security threats of IMS services in §4, victims are mobile users with the subscription of operational IMS services, whereas the adversary develops a malware application and installs it on the victims' MEs; notably, there have been many ways for the malware propagation [36] and it is not our focus. The malware application does not require any root privileges. Specifically, the DoS-All attack (A1) encompasses two attack scenarios with distinct requirements. First, the adversary compromises the Wi-Fi router which the victim ME connects to and assigns the ME a Wi-Fi configuration during the initial Wi-Fi association. Such malicious router can be deployed in some public areas (e.g., cafes, airports, and restaurants). In this case, the malware requires only the INTERNET permission at most. More threateningly, if the victim ME supports IPv6, the malware is not needed. Second, in case the victim ME is not trapped in the malicious Wi-Fi network, the malware is a must and needs not only the INTERNET permission but also the BIND_VPN_SERVICE one. As for the NameSpoofing attack (A2), the malware necessitates the INTERNET permission, and the BIND_VPN_SERVICE one is also needed for the victim MEs running Android 9 or higher.

For the data-plane security threats in §5, victims are mobile operators and adversaries are mobile users abusing IMS video channels. For the ViIMS-ANY attack (A3), it is assumed that the adversary can install a malware application with root privileges on their own ViIMS-supported MEs. This assumption is practical since the compromised MEs are attack devices held by the adversary for attacking the infrastructure. Notably, only the ME software is compromised, but the others, including the ME hardware, are not.

Methodology. To validate the presented security threats, we conduct experiments in the networks of three top-tier U.S. carriers and two Taiwan carriers, which are denoted as US-I, US-II, US-III, TW-I and TW-II, due to a privacy concern. We mainly focus on 4G networks and 5G NSA (Non-Standalone) networks, since the 5G SA (Standalone) network has not been widely deployed yet. However, it is expected that the 5G SA will encounter the same issues. The reasons are twofold. First, all the discovered vulnerabilities are located at the ME side, rather than the infrastructure side. Second, the 5G SA will share the same IMS infrastructure with the 4G and 5G NSA networks [11].

We totally test 10 carrier-certified COTS (Commercial Off-The-Shelf) phone models, including LG G3/G7, Samsung S8/S10/S21, Google Pixel 1/3/5/7, and TCL 40XL, from four major brands. Their Android versions range from 4.4.2 to 13. The reason why the chosen phone models are mainly those with Android OS is that the Android takes the largest share with 71.8% [1] of the worldwide mobile OS market.

Ethical consideration. We bear in mind that some feasibility tests and attack evaluations may harm mobile users or carriers, so we conduct all the experimental studies in two responsible manners. First, we use only our own phones as the victim UEs. Second, we purchase unlimited plans for the text, call, and data services on all the tested phones. Notably, we do not seek to cause any unnecessary damage but rather to make disclosure about potential security threats in operational mobile networks.

Responsible disclosure. We have reported all the identified vulnerabilities to the parties involved, including mobile OS vendors, phone manufacturers, and carriers. The proposed remedies have also been provided to them.

4 UNPROTECTED ME ROUTING FOR IMS CLIENT SIGNALING

The ME routing requirement for the IMS client signaling seems to be simple and easily fulfilled, but it may not be restricted or protected from a security aspect. Specifically, the requirement needs to cover the delivery of both incoming and outgoing IMS signaling messages; the incoming ones

shall be originated from the IMS server and delivered to the IMS client, whereas the outgoing ones shall be sent in an opposite direction. The other routing rules shall be prohibited; otherwise, some security threats may occur, e.g., the IMS server or the IMS client is spammed/spoofed by a third-party entity, and the IMS signaling session is hijacked by a man-in-the-middle (MiTM) attack.

However, we discover that the routing rules for the IMS client signaling on the ME are not enforced to be exclusively restricted for the routing requirement; that is, the IMS signaling messages may be received from non-IMS parties or be maliciously routed to them. In the following, we present the corresponding two security vulnerabilities, namely (1) unprotected IMS signaling routing and (2) unrestricted IMS signaling source, and introduce two proof-of-concept attacks to show the real-world impact.

4.1 V1. Unprotected IMS Signaling Routing

In the mobile OS, a network interface is created for the exclusive use of the IMS service, designated as IMS interface, and is associated with a set of routing rules to route the IMS signaling. According to our investigation on Android OS with versions from 4 to 13, the IMS signaling routing is implemented by two components: RPDB (Routing Policy Database) and iptables. The RPDB defines the priority of routing policies, as shown in Figure 4, and each routing policy specifies a rule matching to a routing table managed by the iptables. For example, the highest priority is the rule at the topmost line, "`from all lookup local`", which means looking up the "`local`" routing table for the packets from "`all`" sources. For each packet, the first matched rule from the priority list is applied.

To support the IMS signaling over the IMS interface (e.g., "`rmnet_data0`"), there are two approaches observed. First, with the older Android versions, the IMS server address is explicitly specified in a routing policy of the RPDB and the policy is to look up the routing table of the IMS interface. Second, with the newer Android versions, the packets generated by the IMS client are identified by a framework mark [4] that is assigned to the client and the mark is associated with the routing table in a routing policy. For example, the policy set to route IMS signaling packets in the RPDB shown in Figure 4 is the bottommost one. It means that all the packets with the framework mark (i.e., "`fwmark`"), "`0x10fa4/0x1ffff`", are routed by looking up the "`rmnet_data0`" routing table.

Seemingly, the routing of the IMS signaling is secure, since the non-IMS applications without root privilege are not allowed to modify the RPDB and routing tables. However, we discover that adding a routing rule to match and route the IMS signaling packets before they are matched with the IMS routing policy is still possible based on some specific operations supported for normal applications without root

```
0:      from all lookup local        → Highest Priority Routing Policy
10000:  from all fwmark 0xc0000/0xd0000 lookup legacy_system
10500:  from all iif lo oif dummy0 uidrange 0-0 lookup dummy0
10500:  from all iif lo oif rmnet_data0 uidrange 0-0 lookup rmnet_data0
10500:  from all iif lo oif rmnet_data1 uidrange 0-0 lookup rmnet_data1
10500:  from all iif lo oif swlan0 uidrange 0-0 lookup local_network
13000:  from all fwmark 0x10063/0x1ffff iif lo lookup local_network
13000:  from all fwmark 0x10fa4/0x1ffff iif lo lookup rmnet_data0
```

Figure 4: Routing Policy Database (RPDB).

```
local ::1 dev lo proto kernel metric 0 pref medium
local 2600:1007:110b:9b03:c923:69ce:83f:d04d dev rmnet_data0
proto kernel metric 0 pref medium
```

(a) Before activating the malicious VPN service.

```
local ::1 dev lo proto kernel metric 0 pref medium
local 2001:4888:2:fe40:a0:104:0:232 dev tun0 proto kernel metric 0
pref medium                    VPN is using the IMS server IP
local 2600:1007:110b:9b03:c923:69ce:83f:d04d dev rmnet_data0
proto kernel metric 0 pref medium
```

(b) After activating the malicious VPN service.

Figure 5: The routing rules of the 'local' routing table.

No.	Source	Destination	Protocol	Length	Info
406	2607:fc20:…	fd00:976a:…	GSM SMS	1208	Request: MESSAGE
452	2607:fc20:…	fd00:976a:…	GSM SMS	1208	Request: MESSAGE
533	2607:fc20:…	fd00:976a:…	GSM SMS	1208	Request: MESSAGE

Retransmission until timeout since the IMS client receives no response.

Figure 6: Failing to send SMS messages, there are no responses received from the IMS server.

privilege. For example, activating the VPN service on an Android phone is allowed to create a virtual interface (e.g., "`tun0`") and assign the interface an address; connecting the phone to a Wi-Fi network is allowed to assign an IP address to the Wi-Fi interface, which is usually given by the DHCP service of the Wi-Fi network. Once the adversary can compromise a VPN application or a Wi-Fi network, the assigned IP address can be set to the IMS server's IP address and the IMS signaling can be thus routed to a compromised network interface, instead of the IMS interface.

Experimental validation. We validate this vulnerability by developing a VPN application that assigns the IMS server's IP address to the virtual interface. The experiment is conducted across three U.S. carriers and two Taiwan carriers. For each tested phone, we activate the VPN service using the developed application, and then send one SMS message to another phone number using the GUI of the SMS service.

For all the tested phones, we observe that the VPN application can successfully assign the IMS server's IP address to its established virtual network interface, and its routing information is updated in the "`local`" routing table, as shown in Figure 5. Moreover, the outgoing IMS signaling messages that carry SMS ones are routed to the VPN interface (i.e., "`tun0`") instead of the IMS interface. It causes the delivery of the SMS messages to fail and no responses from the IMS server are received, as shown in Figure 6, for all the phones except Google Pixel ones.

The reason why this vulnerability does not work for the Google Pixel phones is that they use the IMS client supported in the phone modem to access IMS services, instead of that

```
2024 Jan 31 05:21:36.774 [00] 0x1FEB Extended Debug Message
sipClientConnectio 1965 H Sub-ID:1 Misc-ID:0
SipClientConnection::qpSipInitRequest() METHOD INVITE
```

Figure 7: Pixel phone modem's extended debug messages collected via the QXDM [37].

software-based IMS client in the Android OS, which is employed by the other tested phones. The modem-based IMS client can be accessed by Android applications via QMI (Qualcomm MSM Interface)[48], which is a proprietary interface for interacting with Qualcomm baseband processors. For example, to initiate a VoIMS call, the call application of a Pixel phone sends its modem a QMI_VOICE_DIAL_CALL_REQ message with a specified calling number and the call type (e.g., emergency and auto-selected). The modem then starts the call setup procedure by transmitting a SIP INVITE message to the cellular infrastructure without involving the Android OS, as shown in Figure 7. Thus, this supported modem operation makes the Pixel phones be immune to V1.

Notably, common VPN applications are not allowed to intercept the IMS signaling without V1. Although they handle most data transmissions on the phones and the malicious ones may cause severe attacks on them, the VPN data transmissions do not cover IMS signallings being transmitted over 4G/5G networks. The importance of V1 lies in its ability to allow malware to intercept IMS signaling messages across all radio access networks, creating a new attack surface.

Root cause and lesson learned. This vulnerability arises from a conventional function (i.e., packet routing) on phones, but the root cause is still a design issue from the IMS standard; that is, there is a lack of security protection over the IMS signaling routing on the phones. The mobile OS has fulfilled the requirement of routing all the packets generated from the IMS client to the IMS server, and this routing policy cannot be modified without root privilege. Without an explicit security manner over the IMS signaling routing from the IMS standard, the mobile OS should not take the blame. To address this vulnerability, a new security mechanism is needed to prevent any potential IMS-related policies or rules from nullifying the actual IMS routing policy.

4.2 V2. Unrestricted IMS Signaling Source

When IMS signaling security is enabled, the IPsec SAs between the IMS client and the IMS server will be established during the IMS registration procedure [19]. The number of the established IPsec SAs can be up to four, as shown in Table 2, since the SIP messages are transmitted in two directions, i.e., outgoing and incoming, and can be sent over UDP and TCP. According to the IMS standard [19], all the packets belonging to these four IPsec SAs shall be offered encryption and integrity protection.

Nevertheless, the IMS standard [19] does not expressly specify that the packets which are sent to the IMS client or

Security Associations	Protocol	IMS Client		Direction	IMS Server	
		IP	Port		IP	Port
1	TCP	IP_A	Server Port A	↔	IP_B	Client Port B
2		IP_A	Client Port A	↔	IP_B	Server Port B
3	UDP	IP_A	Server Port A	↔	IP_B	Client Port B
4		IP_A	Client Port A	↔	IP_B	Server Port B

Table 2: Four IPsec SAs needed for IMS services.

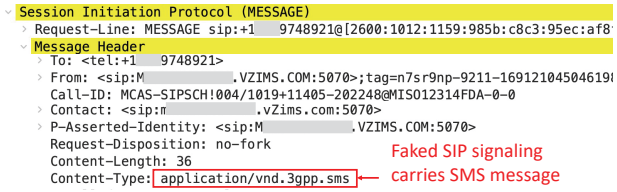


Figure 8: A forged SIP message containing an SMS message.

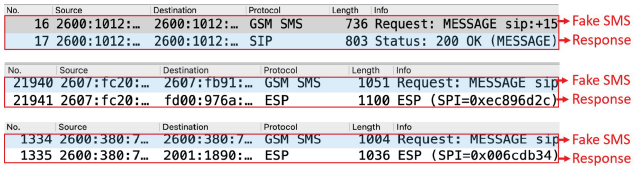


Figure 9: Packet traces collected from US-I (top, w/o IPsec), US-II (middle, with IPSec), and US-III (bottom, with IPSec).

the server but do not belong to those four IPsec SAs shall be discarded, so the ME may still route them based on its own policy, e.g., a pass-through policy. Especially when the source of IMS signaling packets is not restricted, a malware application on the victim UE may be allowed to send fabricated IMS signaling packets to the IMS client locally on the same UE. Given that SMS messages are delivered based on the IMS signaling, this vulnerability can be exploited to launch SMS spoofing attacks when the IMS client accepts and processes the fabricated packets.

Experimental validation. We validate this vulnerability by developing an Android application, designated as *FakeIMSSingaling*, with only the INTERNET permission. It can fabricate a type of IMS signaling messages, SIP MESSAGE [20], which is designed to carry SMS messages. Given the IMS client's IP and port, *FakeIMSSingaling* sends fabricated signaling messages to the IMS client using a local UE IP address as the source IP address, which is different from the IMS server's.

The experiment is conducted in the networks of three U.S. carriers and two Taiwan carriers. For each tested phone, the application sends a plain-text IMS signaling message, as illustrated in Figure 8, to the IMS client, while the tcpdump program captures routed packets. The message is assigned a UDP port number which is not used by those four established IPsec SAs, if the IPsec is adopted for the IMS signaling security (US-II, US-III, and TW-I); otherwise, the UDP port number is randomly selected (US-I and TW-II).

Our experimental results show that the fabricated IMS signaling message can be locally routed to the IMS client for all the tested carriers no matter whether the IPsec is used, as shown in Figure 9. However, this vulnerability does not work for all the tested phones. The Google Pixel phones are

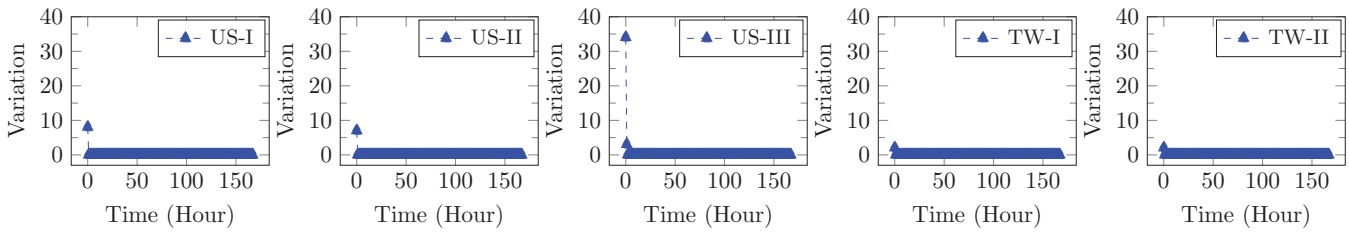


Figure 10: The number of the observed new IMS server IP addresses over time for three U.S. carriers and two Taiwan carriers.

an exception. As explained in §4.1, they employ the modem-based IMS client, instead of the IMS client in the Android OS, so that no IMS signaling messages can be observed in the OS and sent to the IMS client successfully.

Root cause and lesson learned. Seemingly, phone vendors should take most of the blame. After the second thought, it may not be the case. The reason is twofold. First, the common socket communication allows interprocess communication within a system so that the malware can use it to easily send fabricated SIP packets to the IMS client within the same system. However, the IMS standard does not explicitly prohibit it. Second, the phone vendors indeed fulfill the IPsec requirement for the IMS signaling, but the IMS standard does not stipulate how to deal with the packets which do not belong to the IPsec SAs but are sent to the IMS client. In view of these root causes, the vulnerability arises from a design issue of the IMS standard that the IMS client is not protected from receiving messages originating from non-IMS servers. It thus calls for a new security mechanism to ensure the source of the IMS signaling for the IMS client.

4.3 Proof-of-concept Attacks

We devise two novel attacks using the vulnerabilities V1 and V2, respectively: (1) *Denial of Service over All Networks*, designated as DoS-All; and (2) *Named SMS Source Spoofing*, designated as NameSpoofing. DoS-All launches denial of IMS services by exploiting the Wi-Fi association of the victim UE or installing a malware application without root privilege on the UE. It causes the IMS services not only to suffer over Wi-Fi [52] but also to be blocked over cellular radios (e.g., 5G NR). NameSpoofing allows the malware to send spoofed SMS messages with the sender nicknames arbitrarily assigned (e.g., "Daddy"), to the IMS client locally on the victim UE; they can be successfully received by the SMS application and shown to the victim. Notably, these two attacks do not require root privilege and have been successfully validated with Android versions ranging from 4.4.2 to 13.

4.3.1 DoS-All Attack. This attack can cause denial of IMS services for the victim UE by exploiting V1, even though the IMS service continuity between different access networks is supported (e.g., when an IMS service is not available over Wi-Fi, its offering can be handed over to another access network, 4G LTE or 5G NR [51, 52]). It can be very challenging, since IMS services are blocked from being offered through all the access networks, particularly for cellular access networks.

By exploiting vulnerability V1, the adversary can configure one of the victim UE's network interfaces (i.e., Wi-Fi or VPN) to be with the IMS server's IP address assigned to the victim UE, causing all IMS signaling messages to be transmitted to the local network interface. It prevents the IMS client from communicating with the IMS server, no matter which access network is used. There are two attack cases. First, the victim UE associates with a compromised Wi-Fi network, and the Wi-Fi interface is maliciously assigned the IMS IP address. In this case, no malware application is needed on the victim UE. Second, the victim UE installs a compromised VPN application, so the VPN interface is abused to be assigned the IMS IP address.

However, it is not trivial to get the IMS server's IP address assigned to the victim UE in practice. With Android version 10 or lower, this information can be easily obtained when the permission of "READ_PHONE_STATE" is granted. The Android OSes with the later versions constrain the access of this IMS information based on a privileged permission, "READ_PRIVILEGED_PHONE_STATE".

To avoid the requirement of privileged permissions, we propose that the adversary collects a list of IMS server IP addresses in advance within the proximity of each victim UE and assigns them to it during the attack. This mechanism is motivated by the observation that multiple UEs at nearby locations are likely assigned the IMS servers from the same pool; it is also reasonable that serving the UEs in a given range requires only a few IMS servers to be deployed. We conduct an experiment to validate the effectiveness of this mechanism for all the tested carrier networks. In this experiment, we disable and enable the airplane mode periodically on tested phones to trigger a new assignment of the IMS server IP while moving to different locations over time. The ranges of different locations are up to 400 KM and 181 KM for the experiment in U.S. and Taiwan, respectively.

As shown in Figure 10, where the number of the observed new IP addresses over time varies at different locations, there are two main findings. First, the number of the IMS IP addresses assigned to the UEs within nearby areas is limited; specifically, there are 16, 7, 40, 2, and 2 different IP addresses for carriers, US-I, US-II, US-III, TW-I, and TW-II, respectively.

Moreover, all the IP addresses can be collected within a short time, since no more new IP addresses appear after the first two hours in each experiment. Second, the collected IP addresses from two different areas for each carrier have a large overlap in percentages: 57.1% (US-II), 92.5% (US-III), 100% (TW-I), and 100% (TW-II), except for US-I (0%). Moreover, in all tested carriers, the overlap percentage can always reach 100% for any two locations with a distance no larger than 5 KM, which is much larger than the Wi-Fi network range. This experimental result shows that the proposed mechanism can allow the adversary to collect a set of potential IMS IP addresses for each target victim UE.

Note that although the IMS IP address assigned to a victim UE cannot be accurately identified, the Wi-Fi and VPN interfaces are both allowed to be assigned multiple IP addresses so that the set of potential IMS IP addresses can be directly used for the attack. An experiment has been conducted to validate that there are up to 50 IP addresses successfully assigned to any of the Wi-Fi and VPN interfaces; this number of assigned IP addresses is greater than that of the IMS IP addresses observed for each carrier in the experiment.

Attack implementation and evaluation. We implement the DoS-ALL attack by considering two available manners, namely compromised Wi-Fi network and VPN malware.

◇ *Compromised Wi-Fi network*: We develop a customized DHCP server on a widely-used Wi-Fi router, GL.iNet GL-AX1800, with OpenWrt 21.02. It assigns a set of prepared IMS IP addresses to selected Wi-Fi clients (e.g., only smartphones) based on the device model name specified in each DHCP request message. The DHCP server supports the assignment of both IPv4 and IPv6 addresses, where the tested three U.S. carriers all use IPv6, whereas the tested two Taiwan carriers use IPv4. The IPv6 interface can be assigned multiple IP addresses with a mandatory multi-address feature [35], but only a single IP address is accepted by the IPv4 one. To launch the attack against the carriers using the IPv4 address type, a malware program with the INTERNET permission needs to be deployed on the victim UE; notably, the malware is not needed when IPv6 networks are supported by devices and carriers. It detects whether an assigned IP address is correct by listening to the port numbers used by the IMS session (e.g., 5060); when it is correct, the malware can receive IMS signaling messages. Once the assigned IP address is incorrect, the malware disconnects the UE and assigns it with another IP address via the DHCP server. Notably, it is observed that those two Taiwan carriers using the IPv4 each has only two IMS IP addresses, so the IP address can be correctly assigned with at most two assignments.

◇ *VPN malware*: We develop a VPN malware application on Android phones and deploy a VPN server on the Internet. The VPN malware creates and manages the VPN interface based on the *VpnService* class. When connecting to the VPN server, it gets a set of potential IMS IP addresses and assigns them to the VPN interface using the function of `VpnService.Builder.addAddress`.

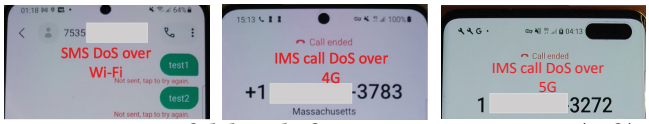


Figure 11: Successful denial of IMS services over Wi-Fi (Left), 4G (Middle), and 5G (Right) networks.

We further launch the two attack approaches against all the tested phone models except for Pixels and cellular network operators. The result shows that victim UEs always suffer from denial of IMS services and are prevented from using IMS-based call or text services, even though the cellular signal quality or the Wi-Fi one is good in all the experiments. Figure 11 shows examples of successful attacks in three different access networks. Notably, we notice that some carriers deployed additional security mechanisms. Specifically, the phones tested in the networks of US-II and TW-II downgrade their access networks to the legacy ones (e.g., 2G and 3G networks) so that they can still have the legacy call and text services, but those tested for the other carriers suffer from denial of all the call and text services.

Attack variance. The DoS-ALL attack can be extended to launch various MiTM attacks. The above malware, lacking root privileges, can intercept all outgoing IMS client messages, and then forward messages to a remote server or interact directly with the client through fabricated replies when IPsec is absent.

4.3.2 NameSpoofing Attack. This attack exploits V2 to send victims spoofed SMS messages in which the sender names can be arbitrarily specified. It differs from conventional SMS spoofing attacks, which deliver spoofed SMS messages through the core network based on spoofed phone numbers, with two advantages. First, the attack does not require SMS messages to be sent through the network, so it cannot be impeded by any security mechanisms deployed in the network. On the other hand, the conventional ones have become much more challenging, since the FCC in the U.S. has mandated carriers to deploy STIR/SHAKEN [45] in the core network to defend against the spoofing attacks; specifically, STIR incorporates digital certificates into the IMS signaling to validate the identity of the SMS sender or the caller. Second, the attack can arbitrarily show the spoofed sender's name without investigating any phone numbers trusted by the victim or stored in the contact list, but the conventional ones can show only inconvincible spoofed numbers on the victim phone if they are not in the contact list.

To show spoofed names, we fabricate SMS messages in the format stipulated by the 3GPP2 standard [21], as shown in Figure 12, instead of the 3GPP standard format [8]. The

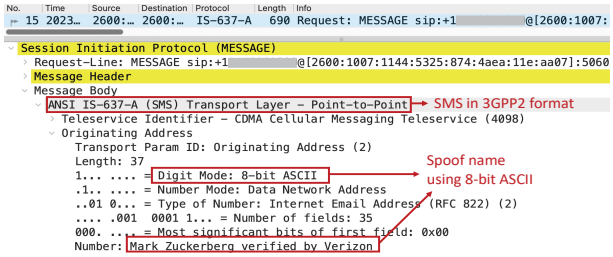


Figure 12: A fabricated SMS message with a spoofed sender name in the 3GPP2 format.

former offers the capability of presenting the message's originating address using ASCII characters (up to 128 characters), whereas the latter allows it to be only in the E.164 format (i.e., the format of phone numbers). Since the spoofed SMS messages do not need to pass through the core network, they can be successfully delivered to the IMS client based on V2 whenever the UE supports the 3GPP2 format. Surprisingly, most phone vendors, such as Samsung, LG, and TCL, still support both 3GPP and 3GPP2 standards for international roaming services. As a result, the present attack can successfully show spoofed names on victim UEs by specifying them in the field of the originating address.

Attack implementation and evaluation. We develop the malware based on the *FakeIMSSingaling* application, which has only the INTERNET permission, by adding two new features for the attack: (1) identifying the IMS client's IP address and UDP port number, which were manually configured during the validation of vulnerability V2; and (2) fabricating SMS messages in the 3GPP2 format. We conduct an experiment that uses the malware to send a spoofed SMS message with the sender name, "Mark Zuckerberg verified by Verizon" to the IMS client locally, for all the tested phones except Google Pixel phones, and all the tested carriers; notably, the carrier name "Verizon" is used solely for the testing purpose. The result shows that all the tested phones can successfully display the spoofed SMS message no matter which carrier network is connected. Figure 1 illustrates the spoofed SMS message displayed on a phone using US-I. Although carriers US-II and US-III conform to the 3GPP standard, the attack can still succeed since the fabricated SMS messages do not pass through the core network. This also validates that many mobile phones support both 3GPP and 3GPP2 standards.

Moreover, note that the malware needs to use the IMS server IP address as the source address of the forged SIP packet carrying the spoofed SMS message if the victim phone is running Android 9 or higher; otherwise, the source address can be assigned any IP address. The IMS clients with Android 9 or higher validate the source address and discard SIP packets not from IMS servers. Given the malware with only the Internet permission, setting the IMS server IP to be the source IP address of fabricated SIP packets can be achieved by binding a UDP socket to the local network interface which is assigned the IMS server IP. For assigning the IP to a network interface, there are two approaches as presented in § 4.3.1: Wi-Fi-based and VPN-based. When the VPN-based approach is adopted, an additional permission, BIND_VPN_SERVICE, is required for the malware.

5 INSECURE ME ACCESS FOR IMS MEDIA SESSIONS

We further explore the insecurity of the IMS media session built on the ME. According to the IMS standard [7], the IMS media session should be encrypted and integrity-protected based on SRTP (Secure RTP); the SRTP keys are derived from the IMS call setup procedure. Since this security mechanism offers the end-to-end protection between the IMS client and the IMS server, without compromising the IMS client, it is almost impossible to forge valid media packets or hijack the media session on the ME if the security protection exists.

However, the SRTP is not a mandatory feature, so it may be absent on the ME, and then allow the adversary to fabricate valid media packets, which are just in plaintext and in the RTP format. Moreover, when the phone modem does not verify the originator of the received IMS media packets, they may be allowed to be dispatched to the radio bearer dedicated for the IMS media session. Thus, the IMS media bearer may be abused. Notably, the IMS voice packets are generated by the phone modem itself, so they do not have this security issue. In the following, we focus on the IMS video session.

Unfortunately, the above potential security threat is discovered on COTS MEs. We identify two vulnerabilities that facilitate the potential abuse of IMS video sessions. The first vulnerability (V3) reveals that video data delivery is not protected by SRTP. All ViIMS packets are transmitted without confidentiality and integrity protection. Consequently, the adversary can easily use ViIMS packets to carry non-video data. The second vulnerability (V4) confirms that the phone modem does not impose any restrictions on the source of ViIMS packets, allowing the adversary to bypass the authentic IMS client and transmit non-video data to the cellular infrastructure over the IMS media bearer.

We next elaborate on these two vulnerabilities and devise a proof-of-concept attack. Note that experiments are mainly conducted in US-I and US-II networks, as ViIMS is not yet supported by TW-I and TW-II, and US-III supports it on only a few phone models.

5.1 V3. Unprotected Video Data Delivery

It has been reported that no SRTP protection is provided over the IMS voice session [30, 32], where voice packets originate from the phone modem. For the video session, though the video data are processed by a different component, the application processor, there exists a high probability that the

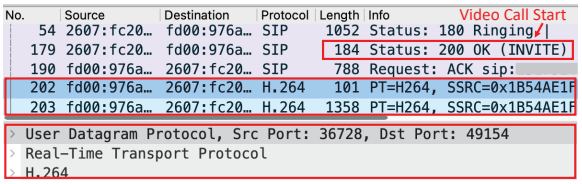


Figure 13: Unprotected IMS video packets in plaintext.

SRTP is still missing due to the common practice. This practice can leave the video data for delivery unprotected on the UE. Once the UE is compromised with root access, the video packets can be captured and learned for the preparation of forging valid IMS video packets.

Experimental validation. We validate this vulnerability on LG G3, Google Pixel 1/3/5/7, and Samsung S8/S10 with Android versions ranging from 4.4.2 to 13. On each tested phone connecting to a carrier network, we use Wireshark to capture packets while dialing a video call to another phone. It is observed that all the IMS video packets are in plaintext without any security protection on all the tested phones. Figure 13 shows one test result as an example.

Root cause and lesson learned. The absence of the SRTP protection does not come without any reasons. The IMS video data delivery has been protected by the user-plane security built between the UE and the base station; it is performed at the PDCP layer with ciphering and integrity protection. Therefore, phone vendors and carriers may consider that such security mechanism has defended the video session against all the potential threats. However, it cannot safeguard the IMS video data on a compromised UE before they are sent to the air. This vulnerability is rooted in that the end-to-end security between the IMS client and the IMS server is not fulfilled; especially, the ME security is not considered. Note that the SRTP protection can be applied, along with SELinux, to safeguard IMS voice sessions. This prevents video calls from being tampered with or extracted, even by adversaries with root privileges. The reason is that SELinux, integrated into Android, employs MAC (Mandatory Access Control) [26] to restrict user access, including root users.

5.2 V4. Unrestricted Source for IMS Video Delivery

The phone modem employs the TFT filter to identify IMS video packets and then dispatches them to the IMS video bearer [18], which offers guaranteed performance for the IMS video session. The TFT filter rule set for each bearer is based on the 5-tuple information (i.e., source/destination IP addresses, source/destination port numbers, and protocol ID); it can be easily obtained by the adversary from normal video packets or control-plane SIP messages. Once the forged video packets are given the correct 5-tuple information and the phone modem does not deploy any security mechanism to verify their delivery source, they could be forwarded to the IMS video bearer by the modem. Moreover, unlike the IMS voice data processed by the modem directly, the IMS video data are sent from the Android OS to the modem, so the forged video packets can be possibly delivered by a malware application in the same way.

Experimental validation. We develop a malware application with root privilege to validate this vulnerability with US-I and US-II. Given a ViIMS call, the malware at the caller generates RTP packets with various payload sizes and sends them to the callee; notably, these packets are assigned a unique RTP SSRC (Synchronization Source) ID, 1234567890 (0x499602D2), and contain random data in the payload. Based on the collected trace at the callee, it is observed that all the RTP packets ranging from 100 to 1346 can be successfully delivered from the caller to the callee in US-I, whereas US-II only allows 10 particular sizes: 37, 169, 393, 489, 537, 585, 729, 1129, 1237, and 1294.

Root cause and lesson learned. The root cause is that the phone modem does not verify the source of the IMS video data delivery, but depends on only the default TFT filter for the dispatching of video packets. Although the IMS server has a chance of inspecting the payload content of video packets to identify the forged ones, it is not allowed except for the approval from at least one party involved in the video call or the court, due to legal provisions for carriers[2]. Thus, addressing this vulnerability has to be at the ME.

5.3 ViIMS-ANY: Covert Communications over Video-over-IMS

We next present a proof-of-concept attack in which two UEs communicate covertly with each other over the ViIMS data-plane channel by exploiting vulnerabilities V3 and V4. Different from previous attacks, the victims in this attack are carriers (e.g., US-I), not individual users. Specifically, adversaries are individuals seeking to exploit the carriers' high-priority resources reserved for the ViIMS service to establish their covert communication channels, with full control over their own phones.

The impact of this attack is expected to grow rapidly as the ViIMS service becomes more popular, even though ViIMS is still in the early stages of deployment. The three major U.S. operators — AT&T, Verizon, and T-Mobile — have introduced the ViIMS service, and some of them support inter-operator ViIMS calls. According to a report [38] by Juniper Research, the number of subscribed users is projected to reach 4.5 billion by 2025, representing 50% of global mobile subscribers. Superior to other video call services such as Skype, ViIMS guarantees performance with minimal overhead, relying on the IMS application for widely deployed VoIMS service; no additional applications are needed.

Attack implementation and evaluation. We develop an attack library called *ViIMSSocket* using C and the raw socket

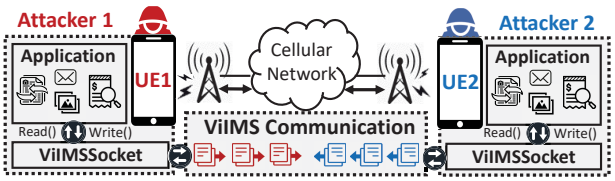


Figure 14: Illustration of ViIMS-ANY attack.

APIs provided by the Linux kernel. It is given root privilege[1] and provides upper-layer applications with a UDP-like packet transmission method for executing covert ViIMS communication, as shown in Figure 14. It contains three major APIs: (1) ViIMSSocket(Callee's Number), which establishes a covert communication channel with the callee over ViIMS and returns a socket ID; (2) ViIMSSocketWriteData(socketID, data), which transmits data to the callee; and (3) ViIMSSocketReadData(socketID, buffer), which receives data from the callee. Notably, *ViIMSSocket* prevents the actual IMS video packets from being transmitted to the IMS server, to maximize the communication capacity.

We evaluate the throughput performance of the covert communication in the networks of US-I and US-II by sending a 10 MB file from one UE to another UE. The experiment runs ten times for each carrier. It is observed that the file is always delivered successfully. The average throughput measured on US-I and US-II is 545.7 Kbps and 581.4 Kbps, respectively. The achieved throughput values are much greater than the one (e.g., up to 38 Kbps) measured from the data transmission over the IMS voice data-plane channel [30]. By abusing IMS video sessions, the covert communication channel is given the guaranteed bit rate resource so that the throughput is guaranteed even in congested scenarios. Furthermore, it is observed that the covert communication can be sustained for at least 100 minutes during a ViIMS call.

Attack variance. With the developed *ViIMSSocket*, potential attacks extend beyond covert communication. Adversaries can hijack video calls to launch video spoofing attacks, allowing for mobile deepfake video calls, encrypted stealthy communication channels, and video frame steganography attacks [49], evading carrier detection of non-video data transmission. Importantly, the video spoofing attack doesn't require a malware application on the victim UE receiving the spoofed video call.

6 SOLUTION

In this section, we propose two remedies to address these four vulnerabilities and evaluate their effectiveness.

6.1 Restricted IMS Routing

We propose a restricted IMS routing mechanism which contains two methods to address vulnerabilities V1 and V2,

[1]For adversaries with an engineering background, obtaining root privileges is technically feasible using tools like Magisk [50] and One-Click Root [5] on Android.

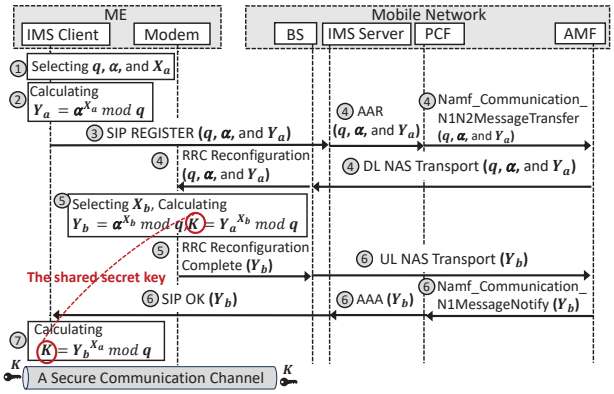


Figure 15: The DHKE procedure integrated into the 3GPP cross-layer communication framework.

respectively. First, the mobile OS shall prohibit any local network interface from being assigned the IMS server's IP address so that the IMS signaling packets cannot be routed locally but to the IMS server. Second, the mobile OS shall be prevented from sending the IMS client any packets originated from local applications, so all the routing policies and tables shall prohibit the local IMS traffic routing. Take the routing table in Figure 5a as an example; the routing rule, "local 2600:...:83f:d04d dev rmnet_data0", with the IMS client's IP address shall be removed.

6.2 Protected IMS Media Sessions

Applying the SRTP protection to safeguard IMS voice sessions can prevent video call tampering (vulnerability V3), but it does not forbid transmitting non-video data. The SRTP protection is built between the IMS client and the IMS server, so the modem is not allowed to verify the authenticity of the IMS video source (vulnerability V4). To this end, a secure communication channel between the IMS client and the modem has to be built. We adopt DHKE (Diffie-Hellman Key Exchange), which is effective in deriving shared secret keys, to establish the secure communication channel. This solution leverages the cellular infrastructure as a trusted intermediary in the DHKE procedure, preventing its common threat, MiTM attacks. It can avoid the use of asymmetric cryptography, which is commonly used to address the MiTM attacks but may not be supported on all MEs. This proposed DHKE procedure exchanges DHKE parameters between the IMS client and the modem during the SIP registration, while doing mutual authentication based on the 3GPP symmetric cryptography. Unless it is compromised, the established secure communication channel remains secure. Notably, this proposed solution does not require any modifications from cellular network protocols or add new signaling messages.

Figure 15 illustrates the proposed DHKE procedure with seven steps: ① in the initiation of the IMS registration procedure [19], the IMS client selects a large prime number, q, a primitive root of q, α, and a private key, X_a; ② the IMS client

calculates its public key Y_a as $Y_a = \alpha^{X_a} \bmod q$; ③ the IMS client transmits the SIP REGISTER message carrying q, α, and Y_a to the IMS server; ④ the IMS server coordinates PCF, AMF, and the serving base station to transmit q, α, and Y_a to the RRC (Radio Resource Control) layer on the phone modem using the RRC Reconfiguration message; ⑤ the RRC layer on the phone modem selects a private key, X_b, calculates the corresponding public key, $Y_b = \alpha^{X_b} \bmod q$, calculates the shared secret key, $\mathbb{K} = Y_a^{X_b} \bmod q$, provides $\mathbb{K}$ for the PDCP layer, and then transmits Y_b to the base station using the RRC Reconfiguration Complete message; ⑥ the phone modem's Y_b is delivered to the IMS client through the SIP OK in response to the SIP REGISTER message; and ⑦ with the received public key, Y_b, the IMS client calculates the shared secret key and finally shares it with the phone modem.

Note that the DHKE has ensured that two communicating parties can derive a shared secret key over an insecure channel. Even with an eavesdropper inside or outside the ME (e.g., eavesdropping on RRC messages) during the DHKE procedure, the shared secret key cannot be inferred or leaked. After the secret key is derived, the mobile OS must ensure that no applications, even those with root privileges, can access the IMS client's memory where the key is stored.

Against legacy and compromised UEs. Adversaries may use legacy UEs or compromised UEs built based on SDR (Software-Defined Radio) platforms (e.g., srsUE [41]) to launch the ViIMS-ANY attack, since they do not allow the proposed solution to be deployed. To address this issue, carriers can reduce attack incentives by preventing them from making high-bandwidth video calls, thereby limiting bandwidth of their video sessions, whenever the deployment of the proposed solution is not detected. Moreover, the infrastructure can also detect them by monitoring their IMS media usage [24].

6.3 Prototype and Evaluation

Restricted IMS routing. We develop an Android system application, designated as IMSProtector, with root privilege. It mainly monitors three pieces of information: (1) the RPDB, (2) routing tables, and (3) network interfaces. It not only removes any routing rule allowing local IMS traffic routing but also deactivates the interface assigned the IMS server's IP address, if there is any to be detected.

To assess the effectiveness of IMSProtector, we launch the attacks of the ineluctable denial of IMS services and the named SMS source spoofing. As shown in Figure 16, IMSProtector can successfully defend against these two attacks. Specifically, it deactivates the local Wi-Fi network interface (i.e., tun0) assigned the IMS server's IP address. It is also observed that the attack application, SMSNameSpoofer, is not allowed to transmit any SMS messages with named sources to the IMS client due to a lack of the local IMS routing.

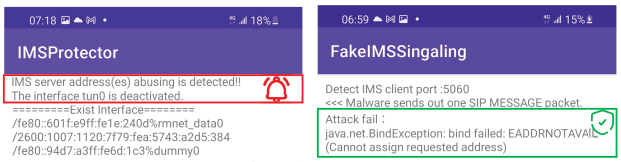
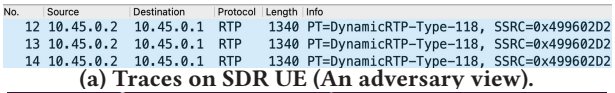
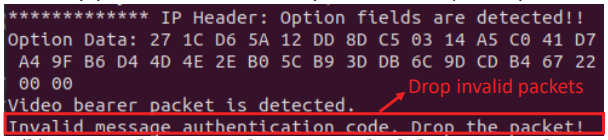


Figure 16: IMSProtector: (Left) disabling an interface assigned the IMS IP address; (Right) local IMS routing is forbidden.



(a) Traces on SDR UE (An adversary view).



(b) Errors shown on the terminal of the SDR modem.

Figure 17: Evaluation of enabling secure communications between the IMS client and the phone modem.

Protected IMS media sessions. We implement and evaluate this solution on an SDR platform, using srsUE (v23.04) for emulating a 5G UE, srsRAN (v23.04) for emulating a 5G gNB, and open5GS (v2.4.11) for emulating a 5G core network; ZeroMQ [3] is used to implement the radio link between the gNB and the UE. Moreover, we develop an IMS client and an IMS server in Python, and deploy them on the srsUE and the open5GS, respectively. The PCF and the AMF in the core network, as well as the gNB, are modified to support the proposed DHKE procedure. This platform is built on a Dell XPS 13 laptop running Ubuntu 22.04, equipped with an i7-1185G7 CPU and 16GB of RAM. In the prototype, we use the shared secret key, $\mathbb{K}$, to provide integrity and data origin authentication for IP packets exchanged between the IMS client and the phone modem. In particular, we add an option using an unassigned option type of 150 [39] to IP headers for the MAC (Message Authentication Code) verification.

To examine the effectiveness of this solution, we launch the ViIMS-ANY attack after a secure communication between the IMS client and the phone modem is established. As shown in Figure 17, it is observed that the fabricated IMS video packets are detected and then dropped.

7 RELATED WORK

Many studies have explored the security issues of IMS services from two aspects: network infrastructure and ME.

Network infrastructure. Several works focus on the insecurity of the IMS server deployed in the cellular network infrastructure. They can be classified into two categories. First, two studies [34, 42] investigate potential flooding and DoS attacks against the IMS server. Specifically, one [34] is to show that the adversary can flood SIP registration messages to the IMS server, yielding the server's extra CPU processing power. The other [42] presents that abrupt changes in

the content of SIP session requests, as well as the SIP message sequence, can be used as detection features of the IMS flooding. Second, three research works [22, 40, 44] attack the IMS session authentication and privacy against the IMS server. They observe that differentiated call response times can be used to identify cellular IoT devices, introduce an attack that eavesdrops on the victim's VoLTE call based on an implementation flaw of reusing the network key stream, and uncover that the weak requirement of network certification in the standard may cause the leakage of the IMSI/APN information for a UE involved in a VoWiFi call, respectively.

ME. The IMS security of the ME has attracted much attention recently. The related studies can be classified into two directions, namely IMS service abuse and DoS attacks. In the first direction, [43] studies the insecurity of the IMS-based SMS and then uncover the corresponding SMS abuse and spoofing attacks. [30] compromises the phone modem to abuse the IMS voice session to transmit malicious data. [25] defends against the caller-ID spoofing by verifying the caller's call state based on a callback.

The other direction focuses on DoS attacks against IMS services. Specifically, [33] hijacks the VoWiFi signaling session to launch stealthy IMS call DoS attacks based on an insecure design of the call state machine. [32] spams the voice bearer to launch a DoS attack by muting an ongoing VoLTE call. [29] presents several vulnerabilities, including an improper cross-layer security binding, for the IMS service, thereby causing DoS attacks on the cellular emergency service against anonymous UEs. [53] introduces side-channel inference techniques to identify specific IMS call signaling messages and launch DoS on the IMS service over Wi-Fi.

The present study differs from prior ones in four aspects. First, it identifies four new vulnerabilities not discovered for the IMS service on the ME. Second, the DoS-ALL attack can deny IMS services across all access networks to the victim UE using a malware application or a malicious Wi-Fi AP, while prior attacks only impact partial IMS services remotely or launch DoS against IMS over Wi-Fi only. Third, the NameSpoofing attack allows the adversary to specify the sender name shown on the SMS application, in contrast to existing attacks that can only spoof the sender's phone number. Fourth, the ViIMS-ANY attack is launched over the ViIMS data plane without compromising the phone modem, while others develop similar attacks over the VoLTE signaling plane, which is protected, or its data plane with a compromised phone modem.

8 DISCUSSION

Is modem-based IMS client better? Google Pixel phones employ the modem-based IMS client, ensuring that no IMS signaling packets are routed in the Android OS, thus making them immune to vulnerabilities V1 and V2. This hardware-based approach appears more secure than software-based methods on other phones but has its limitations. First, the Pixel phones remain vulnerable to the DoS-ALL attack, when extended to prevent IMS media from being sent to the IMS media server. This extension can be achieved by assigning the IMS media server address to a local network interface of the victim UE. Consequently, IMS media data generated by the application processor's domain (e.g., video), rather than the phone modem's, can be sent to the local interface instead of the IMS media server. Second, the modem-based IMS client lacks flexibility in updating IMS-related services, e.g., enabling any of rich communication services (RCS) [27], since updating the phone modem requires collaboration from modem vendors like Qualcomm. It is less convenient and more time-consuming compared to software update.

Are iPhones secure? We conduct experiments on iPhones with four iOS versions (15/15.5/16.5/17) to validate the four discovered vulnerabilities. iPhones are immune to vulnerabilities V1 and V2 due to different network policies applied in iOS, which is built on a Unix-like OS (Darwin) [46]. Specifically, the iOS employs an interface-oriented approach, which restricts the routing of the IMS signaling to only the cellular interface, so V1 does not exist. It drops the IMS packets which do not belong to the established IPsec SAs, thereby avoiding V2. Since most recent iPhones do not support ViIMS [47], the validation of V3 and V4 is left for future investigation.

9 CONCLUSION

Carriers have deployed the IMS system since launching VoLTE. Although 3GPP kept improving its security designs over the last two decades, most enhancements have been focused on the cellular infrastructure. This caused the ME security in the IMS standard to lag behind the infrastructure security, posing security risks to cellular users and carriers. We conducted a comprehensive security study regarding the IMS signaling and media delivery on the ME; four vulnerabilities were identified, and the corresponding three attacks were exposed. These security threats have been validated using ten phone models and five carriers across two countries. Although we have proposed remedies to address them, completely solving them requires collaboration among carriers, phone vendors, and the cellular standard community.

Acknowledgments. We greatly appreciate our shepherd, and all anonymous reviewers, for their intellectual insights and constructive feedback. The work has been partially supported by NSF grants CNS-2246050, CNS-2246051, and CNS-2321416, as well as by NSTC grants 110-2221-E-A49-031-MY3, 112-2628-E-A49-016-MY3, 112-2218-E-A49-021, 112-2634-F-A49-001-MBK, and 112-2218-E-A49-023.

REFERENCES

[1] 20 android statistics in 2024 (market share and users). https://www.demandsage.com/android-statistics/, 2023.

[2] Federal Phone Call Recording Law. https://www.justice.gov/archives/jm/criminal-resource-manual-1050-scope-18-usc-2511-prohibitions, 2020.

[3] srsran 4g with zmq virtual radios. https://docs.srsran.com/projects/4g/en/latest/app_notes/source/zeromq/source/index.html#zeromq-appnote, 2023.

[4] tc-fw(8) — Linux manual page. https://man7.org/linux/man-pages/man8/tc-fw.8.html,.

[5] Kingroot. https://kingrootapp.net/, Jan 2024.

[6] 3GPP. TS 23.125: Overall high level functionality and architecture impacts of flow based charging; Stage 2 (Release 7), Jun. 2007. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=790.

[7] 3GPP. TS33.328: IP Multimedia Subsystem (IMS) media plane security, Nov. 2018. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2295.

[8] 3GPP. TS24.011: Point-to-Point (PP) Short Message Service (SMS) support on mobile radio interface, Nov. 2019. https://www.etsi.org/deliver/etsi_ts/124000_124099/124011/15.03.00_60/ts_124011v150300p.pdf.

[9] 3GPP. TS 23.203: Policy and charging control architecture , Mar. 2021. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=810.

[10] 3GPP. TS 26.139: Real-time Transport Protocol (RTP) / RTP Control Protocol (RTCP) verification procedures (Release 17), Apr. 2022. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3709.

[11] 3GPP. TS 29.228: IP Multimedia (IM) Subsystem Cx and Dx Interfaces; Signalling flows and message contents , Mar. 2022. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1681.

[12] 3GPP. TS 33.102: 3G security; Security architecture, March 2022. V17.0.0.

[13] 3GPP. TS 33.203: 3G security; Access security for IP-based services (Release 17) , Mar. 2022. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1055.

[14] 3GPP. TS 33.210: Network Domain Security (NDS); IP network layer security, Sep. 2022. V17.1.0.

[15] 3GPP. TS 33.401: 3GPP System Architecture Evolution (SAE); Security architecture (Release 17), Sep. 2022. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2296.

[16] 3GPP. TS 33.501: Security architecture and procedures for 5G System (Release 18) , Mar. 2022. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3169.

[17] 3GPP. TS 23.228: IP Multimedia Subsystem (IMS); Stage 2 (Release 18), Mar. 2023. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3100.

[18] 3GPP. TS 24.008: Mobile radio interface Layer 3 specification; Core network protocols; Stage 3 (Release 18), Apr. 2023. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1015.

[19] 3GPP. TS 24.229: IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3 (Release 18) , Apr. 2023. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1055.

[20] 3GPP. TS 24.341: Support of SMS over IP networks; Stage 3 (Release 18) , Jan. 2023. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1085.

[21] 3GPP2. 3GPP2 C.S0015-A: Short Message Service (SMS) for Wideband Spread Spectrum Systems Release A, Sep. 2004. https://www.3gpp2.org/Public_html/Specs/C.S0015-A_v2.0_051006.pdf.

[22] Jaejong Baek, Sukwha Kyung, Haehyun Cho, Ziming Zhao, Yan Shoshitaishvili, Adam Doupé, and Gail-Joon Ahn. Wi not calling: Practical privacy and availability attacks in wi-fi calling. In *Proceedings of the 34th Annual Computer Security Applications Conference*, ACSAC '18, page 278–288, New York, NY, USA, 2018. Association for Computing Machinery.

[23] Evangelos Bitsikas and Christina Pöpper. You have been warned: Abusing 5g's warning and emergency systems. In *Proceedings of the 38th Annual Computer Security Applications Conference*, ACSAC '22, page 561–575, New York, NY, USA, 2022. Association for Computing Machinery.

[24] Fabio Cecchinato, Lorenzo Vangelista, Giulio Biondo, and Mauro Franchin. Anomaly detection using lstm neural networks: an application to voip traffic. In *2021 IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE)*, pages 1–7, 2021.

[25] Haotian Deng, Weicheng Wang, and Chunyi Peng. Ceive: Combating caller id spoofing on 4g mobile phones via callee-only inference and verification. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, MobiCom '18, page 369–384, New York, NY, USA, 2018. Association for Computing Machinery.

[26] Google. Android security paper 2023. https://blog.google/products/android-enterprise/android-security-paper-2023/, Jan 2023.

[27] GSMA. RCS Universal Profile Service Definition Document , Oct. 2019. https://www.gsma.com/futurenetworks/wp-content/uploads/2019/10/RCC.71-v2.4.pdf.

[28] GSMA. IMS Profile for Voice and SMS. https://www.gsma.com/newsroom/wp-content/uploads/IR.92-v15.0-4.pdf, 2020.

[29] Yiwen Hu, Min-Yue Chen, Guan-Hua Tu, Chi-Yu Li, Sihan Wang, Jingwen Shi, Tian Xie, Li Xiao, Chunyi Peng, Zhaowei Tan, and Songwu Lu. Uncovering insecure designs of cellular emergency services (911). In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, MobiCom '22, page 703–715, New York, NY, USA, 2022. Association for Computing Machinery.

[30] Hongil Kim, Dongkwan Kim, Minhee Kwon, Hyungseok Han, Yeongjin Jang, Dongsu Han, Taesoo Kim, and Yongdae Kim. Breaking and fixing volte: Exploiting hidden data channels and mis-implementations. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 328–339, New York, NY, USA, 2015. Association for Computing Machinery.

[31] Gyuhong Lee, Jihoon Lee, Jinsung Lee, Youngbin Im, Max Hollingsworth, Eric Wustrow, Dirk Grunwald, and Sangtae Ha. 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*, MobiSys '19, page 404–416, New York, NY, USA, 2019. Association for Computing Machinery.

[32] Chi-Yu Li, Guan-Hua Tu, Chunyi Peng, Zengwen Yuan, Yuanjie Li, Songwu Lu, and Xinbing Wang. Insecurity of voice solution volte in lte mobile networks. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 316–327, New York, NY, USA, 2015. Association for Computing Machinery.

[33] Yu-Han Lu, Chi-Yu Li, Yao-Yu Li, Sandy Hsin-Yu Hsiao, Tian Xie, Guan-Hua Tu, and Wei-Xun Chen. 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*, MobiCom '20, New York, NY, USA, 2020. Association for Computing Machinery.

[34] Jamila Manan, Atiq Ahmed, Ihsan Ullah, Leïla Merghem-Boulahia, and Dominique Gaïti. Distributed intrusion detection scheme for next generation networks. *Journal of Network and Computer Applications*,

147:102422, 2019.

[35] T. Mrugalski, M. Siodelski, and et al. Dynamic Host Configuration Protocol for IPv6 (DHCPv6), 2018. https://datatracker.ietf.org/doc/html/rfc8415.

[36] Sancheng Peng, Shui Yu, and Aimin Yang. Smartphone malware and its propagation modeling: A survey. *IEEE Communications Surveys & Tutorials*, 16(2):925–941, 2014.

[37] Qualcomm. Qxdm professional tool quick start. https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/80-n9471-1_d_qxdm_professional_tool_quick_start.pdf, Jan 2024.

[38] Juniper Research. Video calling demand booms during pandemic. https://pipelinepub.com/news/12307, Jan 2024.

[39] RFC. Internet Protocol, 1981. https://datatracker.ietf.org/doc/html/rfc791.

[40] David Rupprecht, Katharina Kohls, Thorsten Holz, and Christina Pöpper. Call me maybe: eavesdropping encrypted lte calls with revolte. In *Proceedings of the 29th USENIX Conference on Security Symposium*, SEC'20, USA, 2020. USENIX Association.

[41] srsRAN. srsue. https://docs.srsran.com/projects/4g/en/latest/usermanuals/source/srsue/source/1_ue_intro.html, Jan 2023.

[42] Qibo Sun, Shangguang Wang, Ning Lu, Kok-Seng Wong, and Myung Ho Kim. Sfads: A sip flooding attack detection scheme with the internal and external detection features in ims networks. *Journal of Internet Technology*, 17(7):1327–1338, 2016.

[43] Guan-Hua Tu, Chi-Yu Li, Chunyi Peng, Yuanjie Li, and Songwu 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*, CCS '16, page 1118–1130, New York, NY, USA, 2016. Association for Computing Machinery.

[44] Sihan Wang, Guan-Hua Tu, Xinyu Lei, Tian Xie, Chi-Yu Li, Po-Yi Chou, Fucheng Hsieh, Yiwen Hu, Li Xiao, and Chunyi Peng. Insecurity of operational cellular iot service: new vulnerabilities, attacks, and countermeasures. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, MobiCom '21, page 437–450, New York, NY, USA, 2021. Association for Computing Machinery.

[45] Wikipedia. STIR/SHAKEN. https://en.wikipedia.org/wiki/STIR/SHAKEN,.

[46] Wikipedia. Darwin (operating system). https://en.wikipedia.org/wiki/Darwin_(operating_system), Feb 2024.

[47] Wikipedia. ios. https://www.apple.com/iphone-15/specs/, Jan 2024.

[48] Wikipedia. Qualcomm msm interface. https://en.wikipedia.org/wiki/Qualcomm_MSM_Interface, Jan 2024.

[49] Wikipedia. Steganography. https://en.wikipedia.org/wiki/Steganography, Jan 2024.

[50] John Wu. Magisk. https://github.com/topjohnwu/Magisk, Jan 2024.

[51] T. Xie, G. Tu, C. 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)*, pages 1–1, May 2018.

[52] Tian Xie, Guan-Hua Tu, Bangjie Yin, Chi-Yu Li, Chunyi Peng, Mi Zhang, Hui Liu, and Xiaoming Liu. The untold secrets of wifi-calling services: Vulnerabilities, attacks, and countermeasures. *IEEE Transactions on Mobile Computing*, 20(11):3131–3147, 2021.

[53] Tian Xie, Sihan Wang, Xinyu Lei, Jingwen Shi, Guan-Hua Tu, and Chi-Yu Li. Mpkix: Towards more accountable and secure internet application services via mobile networked systems. *IEEE Transactions on Mobile Computing*, 22(6):3489–3507, 2023.