When Good Turns Evil: Encrypted 5G/4G Voice Calls Can Leak Your Identities

Jingwen Shi*, Tian Xie†, Guan-Hua Tu*, Chunyi Peng‡, Chi-Yu Li§, Andrew Hou*,
Sihan Wang*, Yiwen Hu*, Xinyu Lei¶, Min-Yue Chen*, Li Xiao*, Xiaoming Liu*
*Department of Computer Science and Engineering, Michigan State University,
†Department of Computer Science, Utah State University, ‡Department of Computer Science, Purdue University,
§Department of Computer Science, National Chiao Tung University,
¶Department of Computer Science, Michigan Technological University,
Email: *{shijingw,ghtu,houandr1,wangsih3,huyiwen3,chenmi33,lxiao,liuxm}@msu.edu,
†tian.xie@usu.edu, ‡chunyi@purdue.edu, §chiyuli@cs.nctu.edu.tw, ¶xinyulei@mtu.edu

***Abstract*—5G/4G voice calls are always encrypted for security and privacy. However, in this work, we unveil several vulnerabilities which can unintentionally leak 5G/4G call state information, despite encryption protection. They stem from recent call optimization techniques standardized in the 3GPP specifications and adopted by mobile network operators. While these techniques are effective to enhance 5G/4G call quality and efficiency, they unfortunately expose extra call information, which can be exploited to *precisely* infer call states and launch side-channel attacks. By leveraging precise call states, we devise a Cross-domain Identity Linkage attack, CrossIL, which aims to infer mobile users' user identities and cellular identities, thereby enabling powerful cyberattacks or privacy inferences against high-value victims. We have experimentally validated these vulnerabilities and assessed the attack damages with three major U.S. carriers. Our experimental result shows that the success rate on the identity inference ranges from 89% to 98%. Finally, we propose and evaluate a cellular-friendly solution.**

I. Introduction

Mobile voice has been a killer communication service since its origin. It is still prevalent, despite the emerging popularity of rich communication services over mobile broadband. Many users make phone calls on a daily basis [1]. Mobile voice advances to a Voice-over-IP (VoIP) solution as 5G/4G networks are fully over IP. This voice solution is called as Voice over IP Multimedia Subsystem (VoIMS) [2], which is referred to as Voice-over-New-Radio (VoNR) for 5G and Voice-over-LTE (VoLTE) for 4G. Until now, more than 235 operators in 105 countries have rolled out VoIMS service [3], which is projected to serve 5 billion devices by 2025 [3].

Unsurprisingly, 5G/4G voice calls are encrypted for the purpose of confidentiality, secrecy and privacy. They are protected by well-examined security measures including 5G/4G authentication and key agreement (AKA) and multi-layer network security (in Layers 2 and 3). Specifically, they leverage secret keys protected in a physical SIM card or an eSIM module to bootstrap mutual authentication between the device and 5G/4G network and generate derived keys for subsequent voice/data sessions [4]. All voice packets including voice traffic and signaling messages are delivered over IP (Layer 3), which is protected by IPSec (Internet Protocol Security) with confidentiality and integrity protection [5]. To protect transmissions over the air, they are ciphered at Layer 2 (via Packet Data Convergence Protocol (PDCP)) [6].

Unfortunately, we find that vulnerabilities stem from several optimization techniques designated to enhance VoIMS quality and efficiency. All these optimization techniques have been standardized in 3GPP specifications and adopted by commercial 5G/4G networks (at least all three U.S. operators in our study). Specifically, VoIMS uses guaranteed-bit-rate radio bearers, different from best-effort radio bearers for mobile broadband [2]; it compresses packet headers to save bandwidth via RObust Header Compression (ROHC) [7]; it handles various radio conditions with Adaptive Multi-Rate (AMR) audio codecs which are optimized for speech compression and coding [8]; and it handles silence (i.e., when no one talks) which likely results in terminating a call through a new technique called comfort noise (injecting artificial noises during the silence at a call conversation) [9]. Each optimization technique is good to enhance call quality and efficiency.

However, the good turns evil when putting them together. Those new techniques together pose unexpected yet intriguing implications that can turn security threats into VoIMS calls over 5G/4G. VoIMS traffic can be distinguished from all other traffic when ROHC and comfort noise are used together to generate tiny packets (< 16 bytes). Notably, the packets for all other traffic are more than 30 bytes (Table I). By this means, a VoIMS call can be detected by checking the presence of tiny packets. Moreover, voice call states can be further inferred by checking IMS voice packets with/without comfort noises.

We further devise a proof-of-concept attack over side-channel inference: **Cross**-domain **I**dentity **L**inkage (CrossIL), which is designed to discover the confidential identities of multiple VoIMS users in public; it links the victim's cellular identity to a specific user by correlating the call state information inferred from cellular radio domain and visual domain. We validate and assess this attack using various phone models and cellular operators in the U.S. The results show that CrossIL has a success rate ranging from 89% to 98% in the controlled and wild settings. We finally propose a standard-compliant

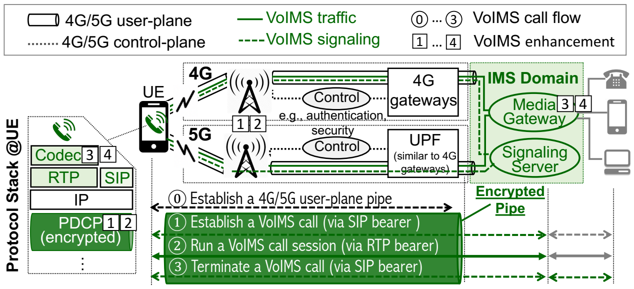
4G/5G user-plane
4G/5G control-plane
VoIMS traffic
VoIMS signaling
⓪ ... ③ VoIMS call flow
1 ... 4 VoIMS enhancement
Protocol Stack @UE
Codec 3 4
RTP
SIP
IP
PDCP 1 2
(encrypted)
UE
4G
5G
1 2
Control
e.g., authentication, security
Control
4G gateways
UPF (similar to 4G gateways)
IMS Domain
Media Gateway 3 4
Signaling Server
Encrypted Pipe
⓪ Establish a 4G/5G user-plane pipe
① Establish a VoIMS call (via SIP bearer)
② Run a VoIMS call session (via RTP bearer)
③ Terminate a VoIMS call (via SIP bearer)

Fig. 1: Network architecture, main protocols and an operation flow for 5G/4G voice over VoIMS.

solution to address the discovered vulnerabilities and evaluate its effectiveness.

II. 5G/4G Voice Primer

VoIMS is an essential VoIP-based voice solution for 5G/4G networks [2]. Figure 1 depicts its network architecture, main protocols, and a basic work flow.

5G/4G network architecture supporting VoIMS. It comprises two parts: the 5G/4G network infrastructure and the IMS domain. The former provides User Equipment (UE, e.g., mobile phones) with active mobile connections (user-plane data pipes) to deliver user traffic over IP. User traffic packets in turn traverse the UE, the base station and the gateways in the core cellular network to reach the external Internet or the IMS domain (for 5G/4G voice), or vice versa. The IMS domain comprises two key components: media gateway and signaling server. The former delivers IP multimedia data (e.g., voice packets) to IMS clients (e.g., UEs); the latter processes all signaling messages to establish and manage call sessions.

Main protocols for VoIMS. The main protocols above IP are Session Initiation Protocol (SIP) and Real-time Transport Protocol (RTP). SIP is used for voice signaling to initiate, maintain, modify, and terminate voice calls over IP. RTP transmits a live multimedia stream over IP. VoIMS takes the same choices used by VoIP. Below IP, the main protocol is PDCP [6]. It performs three main functions within 5G/4G networks. First, it compresses the IP headers of data packets to improve transmission efficiency over the air. Second, it supports ciphering and integrity protection to protect the upper-layer data, namely, IP packets. The session keys are generated through 5G/4G security functions [10], [11]. Third, it dispatches the upper-layer data to their corresponding radio bearers: Dedicated Radio Bearer (DRB) and Signaling Radio Bearer (SRB). DRB is used to carry traffic in the user plane, and SRB is for 5G/4G signaling in the control plane. PDCP is the *only* Layer-2 protocol studied in this work because PDCP wraps other lower L2/L1 protocols to offer a user-plane pipe for IP packet delivery. Conceptually, there is no difference between 5G and 4G except that 5G supports varying QoS settings for distinct IP data flows [12].

VoIMS call flow. A VoIMS call typically takes three steps: establishment (①), call conversation (②), and termination (③), if a 5G/4G user-plane pipe below IP is available. Otherwise, it first establishes this pipe (⓪). Actually, this pipe is encrypted using the keys derived from the mutual authentication between the UE and the network. A VoIMS call session is established by SIP signaling; it starts when someone dials a phone number to generate a call request and ends when the call request is accepted by the other call party (①). A call conversation is then carried over this established call session (②). The voice call application uses a speech audio codec to convert voice traffic into a digital format, which is later delivered by RTP. To end the call, SIP is used again to terminate the VoIMS call session (③). Both RTP and SIP packets are encapsulated into IP packets for delivery. Specifically, they are forwarded to the IMS through the user-plane (PDCP) pipe provided and encrypted by 5G/4G networks.

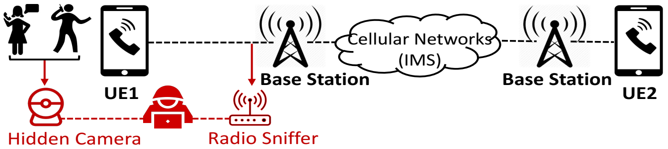

Fig. 2: MiTM attacks in the threat model.

Voice enhancement techniques. Four techniques have been introduced by 3GPP to enhance quality and efficiency of VoIMS services, as illustrated in Figure 1. From bottom up, they include special radio bearers ($\boxed{1}$), ROHC ($\boxed{2}$), comfort noise (CN) ($\boxed{3}$), and AMR speech codecs ($\boxed{4}$). VoIMS uses a special DRB with a guaranteed-bit-rate to ensure sufficient radio bandwidth for voice [2]; ROHC compresses the headers of VoIMS packets to reduce transmission overhead [6]; CN injects some background noise to prevent an unexpected call termination caused by a period of silence [9]; AMR speech codecs [9] offer adaptive rates for voice, and VoIMS uses a lower coding rate for unvoiced packets that carry background noise. These techniques indeed enhance 5G/4G voice quality and efficiency. However, the good turns evil as they together bring unanticipated side effects to leak confidential call information despite encryption (§IV).

III. Threat Model and Methodology

Threat Model. Adversaries are organizations or people who attempt to monitor/attack mobile users through 5G/4G radio channels (via MiTM attacks). As shown in Figure 2, they can eavesdrop on all messages over public communication channels, but they cannot decrypt the encrypted messages without knowing the decryption keys. Specifically, they can deploy their equipment near the victim UEs to capture all the packets over the air, but they cannot compromise any victim's smartphone or the 5G/4G networks. Moreover, adversaries can deploy hidden cameras at public locations to stealthily record videos of victims.

Responsible methodology and ethics. We conduct this study in a responsible manner. We run real experiments with all three top-tier U.S. operators (denoted as OP-I, OP-II, and OP-III) to validate the identified vulnerabilities and assess attack damages. We understand that some feasibility tests and attack evaluations might be detrimental to network operators and their mobile users. As a result, unless specified, we run all the experiments in a *fully controlled* environment,

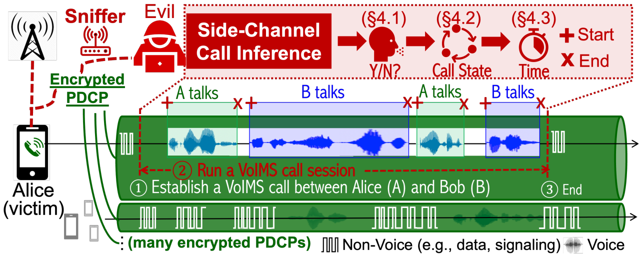

Fig. 3: Overview of side-channel call inference.

where we deploy an attacker (a 5G/4G sniffer over software-defined radio (SDR)) to infer call information of participating smartphones which are all owned by our lab. To prevent inadvertent attacks on non-participating smartphones, we take two measures: (1) we run all experiments in a private laboratory during off-peak times and ensure that no passersby are present. In this setup, one smartphone serves as the victim, and several other smartphones act as users nearby to simulate normal 5G/4G traffic over the air; and (2) when passersby are present, we utilize phone-side cellular trace collectors, such as MobileInsight [13], to collect the victim's cellular radio traffic exclusively, thus ensuring that no cellular radio traffic is collected from non-participating smartphones. A few attack experiments are launched in semi-controlled environments or at public places. More experimental settings are given in §V.

IV. Side-Channel Call Inference

In this section, we present call inference techniques to obtain confidential call information over encrypted packets.Figure 3 gives an overview of side-channel inference with three tasks. First, it detects the presence of a VoIMS call out of all the packets received in the air (§IV-A). Note that most packets are not for voice (say, for mobile data and 5G/4G signaling). Second, it infers call states for the detected VoIMS call, particularly who is talking (§IV-B). It means that the adversary `Evil` is capable of knowing more about how this voice call is going on by dividing a call conversation into fine-grained segments (e.g., Alice talks most time or rarely talks). Last, it infers the start and end time for each conversation segment (marked as "+" and "×" in Figure 3). Such precise call state information makes it possible to launch attacks to infer more confidential information (say, user identities) or selectively manipulate the target call at specific time.

A. *Detecting VoIMS Calls*

At first glimpse, detecting the presence of an ongoing VoIMS call is not hard. This is because the radio bearers used for VoIMS (voice traffic and signaling) are different from those for mobile Internet data. A prior study [14] has observed the use of distinct DRBs (say, DRB1: mobile Internet data, DRB2: VoIMS signaling, DRB3: VoIMS voice packets). Even though traffic is ciphered at PDCP, it is not hard to detect VoIMS calls by analyzing the use of all DRBs.

However, the reality is more complex and challenging. First, the mapping between a DRB number and its supported traffic type is never fixed or explicitly defined by any VoIMS standard. As a result, it varies with network operators and changes over time. Second, there are many packets transmitted over DRBs, and all of them are encrypted. It is not scalable or even technically feasible (unless with a super-powerful sniffer) to inspect all the encrypted packets of multiple mobile devices for a long while and identify the DRBs reserved for VoIMS traffic. Here, we need a reliable, scalable and lightweight solution to effectively detect the presence of VoIMS calls by concurrently screening all DRBs used by nearby mobile users.

Our call detection approach exploits two voice quality optimization techniques: ROHC [6] and CN [9]. It is fueled by two facts: (1) both ROHC and CN are mandatory features for VoIMS services regulated by 3GPP standards [9]; and (2) they together produce special voice packets whose sizes are much smaller than the non-VoIMS ones. As a matter of fact, the use of both techniques is observed in all VoIMS call experiments with three US operators. Notably, VoIMS standard specifies that the CN generator on a smartphone should cooperate with a Voice Activity Recognizer (VAR) module that detects the activity of human speech (i.e., talking or not talking) [9]. There are speech pauses (silence) in a call conversation, where synthetic background noises are injected to generate CN packets. The noise payload is small; with ROHC, the packet is compressed into tiny packets which are never used by non-VoIMS traffic.

Specifically, a CN voice packet contains 35∼48 bits (4.375∼6 bytes) for noise information [8]; it is then encapsulated into an RTP packet using the payload type of 13 [15]. With ROHC, it is further compressed into a PDCP packet with the length of **8**∼**13** bytes. We find that such tiny PDCP packets appear *only* during VoIMS calls. Consequently, the presence of tiny PDCP packets is considered as an indicator to VoIMS calls. Note that the minimum IP packet size is 20 bytes (IPv4) or 40 bytes (IPv6) as ROHC is not activated for non-VoIMS traffic. Once a tiny packet is detected, we further obtain its DRB number (which is not encrypted) and learn which DRB is used for VoIMS. By this means, it is able to detect multiple VoIMS calls over distinct DRBs. Notably, the vulnerability lies in that both ROHC and CN are used only for VoIMS.

Empirical validation. We have conducted experiments with three U.S. operators to validate that only VoIMS calls generate tiny PDCP packets but non-VoIMS applications do not. We test four phone models: Google Pixel 5, Pixel 3, Samsung S5, and LG G3, all supporting VoLTE and Pixel 5 supporting VoNR. We run three VoIMS-based applications (VoLTE, VoNR, and Google Voice) and 57 non-VoIMS applications selected from the top-100 mobile Internet applications. These test applications are roughly classified into three other categories: (1) Non-VoIMS VoIP (e.g., Skype), (2) Non-VoIP streaming (e.g., Netflix, Youtube), and (3) Non-streaming (e.g., Amazon, Twitter, Reddit). For all VoIP applications (including VoIMS and non-VoIMS VoIP), each test is a 30-second voice call with 10s for ringing and 20s for call conversation. For non-VoIP streaming applications, each test runs video streaming for about 5 minutes. For non-streaming applications, we continuously access their Internet services (e.g., refreshing online

	No.	App Name	Len		No.	App Name	Len
VoIMS	1	**VoLTE (4G)**	**11 – 13**	Non-Streaming	31	Amazon	42
	2	**VoNR (5G)**	**12 – 13**		32	Reddit	42
	3	**Google Voice**	**13**		33	McDonald's	42
Non-VoIMS VoIP	4	WhatsApp	32		34	DuckDuckGo	42
	5	WhatsApp Business	32		35	DoorDash	42
	6	Skype	37		36	WeatherPort	42
	7	Telegram	42		37	Waze	42
	8	Discord	42		38	Instagram	42
	9	TextNow	42		39	TikTok	42
	10	Google Hangouts	50		40	Walmart	42
	11	Snapchat	54		41	Airbnb	42
Non-VoIP Streaming	12	Twitch	42		42	AAA	42
	13	Spotify	42		43	Snapfish	42
	14	Netflix	42		44	Apex News	42
	15	SoundCloud	42		45	Expedia	42
	16	Disney+	42		46	Chrome	42
	17	Amazon Prime Video	42		47	Brave Browser	54
	18	Xbox	42		48	Google Earth	61
	19	NewsBreak	42		49	SpeedVPN	30
	20	Bigo Live	42		50	Thunder VPN	37
	21	Shazam!	54		51	Google Translate	42
	22	YouTube Kids	59		52	Microsoft Authenticator	42
Non-Streaming	23	The Weather Channel	42		53	Acrobat Reader	42
	24	Microsoft Edge	42		54	Google Docs	42
	25	Twitter	42		55	DNS Speed Test	42
	26	Photomath	42		56	FTP Server	42
	27	Microsoft Teams	54		57	Outlook	42
	28	Zillow	42		58	Canva	54
	29	Booking	42		59	Google Authenticator	54
	30	Duolingo	42		60	Pinterest	54

TABLE I: *Tiny* packets are only observed in VoIMS.

content and searching for products) for 1 minute.

Table I shows the PDCP packet lengths for three VoIMS applications and the minimal length per each non-VoIMS application observed in our study. We have four findings. First, all three U.S. operators have adopted both ROHC and CN techniques for VoIMS calls. Second, no *tiny* PDCP packets are observed from any non-VoIMS application, but many of them are seen during VoIMS calls. This finding is consistent across four phone models and three operators. Third, it is effective in detecting multiple ongoing calls (up to four tested in our study). Last, the number of tiny packets observed during a VoIMS call changes with network operators and mobile device models. We later show that it is due to varying speech coding rates (AMR), which impacts call state inference (§IV-B).

B. *Inferring Call States*

We next describe how to infer call states when a VoIMS call is detected. Consider an adversary usually eavesdrops on one call party (say, Alice), since two call parties (say, Alice and Bob) are unlikely in close proximity and do not use the same radio channel sniffed by the adversary. Consider the adversary `Evil` deploys a sniffer near Alice, not Bob. From Alice's viewpoint, there are two call states: *talking* and *not-talking (listening)* when the call conversation is on. Here, *talking* means that Alice is talking, and *listening* means Alice is not talking, but listening to Bob.

An intuitive approach is to check the presence of **CN** and non-Comfort-Noise (**nonCN**) voice packets in both directions – downlink (DL) and uplink (UL) – to infer whether Alice is talking. When Alice is talking, UL-nonCN packets will be sent out; when Alice is not talking (silence), UL-CN packets will be sent out. Similarly, DL packets are observed depending on whether Bob is talking. Figure 4 presents this approach by zooming into the start of a call conversation where Alice begins to talk. Note that the target DRB has been identified while detecting a VoIMS call (§IV-A). It is thus easy to recognize the packets transmitted over this target DRB, out of many packets from concurrent DRBs.

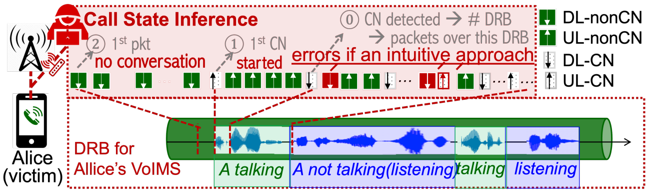

Fig. 4: Call state inference for one detected call (§IV-B).

Call State	Intermediate Criteria			
	#UL-CN	**#UL-nonCN**	**#DL-CN**	**#DL-nonCN**
Talking	$*$	>0	>0	$*$
Listening	>0	$*$	$*$	>0
No conversation	$=0$	$*$	$=0$	>0

TABLE II: Intermediate criteria for 3 call states ($*$: wildcard).

However, there are three practical issues to address, as illustrated in Figure 4. First, we find that the call conversation might not start yet even when some packets are transmitted over the target DRB. It happens with some premium voice services. For instance, Early Media [16] allows the callee to deliver a preferred alerting music (e.g., a song) to the caller. As a result, we introduce a new call state of *no conversation*: the DRB is active but the call is not established yet. We observe that the CN packets are never used before the call starts. Hence, we use the first CN packet over this DRB as the start of this call conversation.

Second, CN packets are tiny, much smaller than non-VoIMS packets. However, it is unclear whether CN and nonCN voice packets can be distinguished by their packet lengths. The good news is that we do find that nonCN voice packets are always larger than the CN ones. The minimum PDCP payload length of nonCN voice packets is 15.08 $(4,750bits/8/50+3.2=15.08)$ bytes, which is rounded up to 16 bytes. It is observed when the lowest VoIMS codec bit-rate for nonCN voice packets is used (4.75 Kbps via AMR); the inter-arrival time is 20 ms on average (50 packets per second), and ROHC reduces the size of RTP headers to $3.2\sim6.5$ bytes; on contrast, the maximum length of CN packets is 13 bytes. In this work, we choose a threshold θ =16 (14, 15, and 16 all work) and the packet with the payload length < 16 is a CN packet.

We further collect packet statistics every second: the counts of CN and nonCN packets in UL and DL, denoted as *#UL-CN*, *#UL-nonCN*, *#DL-CN* and *#DL-nonCN*. Table II lists the criteria (actually, intermediate results) to infer three call states simply based on the presence of these four packet types, namely ***No conversation***: the conversation does not start when *#UL-CN*$=0$ and *#DL-CN*$=0$; ***Talking***: When *#UL-nonCN*>0 and *#DL-CN*>0, the user sends voice packets to the remote call party and receives comfort noise packets from them, which means that the user is talking to the remote party; and ***Listening***: When *#DL-nonCN*>0 and *#UL-CN*>0, the user does not send voice packets to the remote call party but sends CN packets to them, which means that the user is listening.

Third, we find that short-term call state inference based on the above criteria may not be sufficient to infer accurate call time due to some noises. In particular, we observe tiny CN packets in both directions when Alice is talking or listening (more explained in §IV-C). We will next present our final

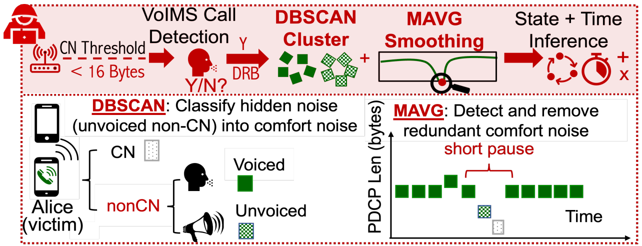

Fig. 5: Side-channel inference uses DBSCAN and MAVG to prevent unnecessary talking-listening state switches (§IV-C).

approach along with inferring call time.

C. Inferring Call Time

The intuitive approach in §IV-B infers a call state per second and then accumulates time periods during the talking and listening states. However, we find that it does not work well in practice. It is because there exist other noise packets. In particular, there are two types of other noise packets: *hidden* noise and *redundant* comfort noise. The *hidden* noise packets are unvoiced nonCN packets, which are generated based on uncancelled environment noises when the call party is listening; the *redundant* comfort noise is produced over human's short speech pause during talking. Thus, there are frequent switches between talking and listening states. A short speech pause (with redundant comfort noises) results in inaccurate inference on the talking state; i.e., talking stops earlier and switches to the listening state but soon returns back to the talking state, as Alice is still talking. Similarly, the hidden noise packets impact the inference on when listening stops.

We thus devise two approaches: (1) density-based spatial clustering of applications with noise (DBSCAN) and (2) the moving average of voiced packet ratio (MAVG) to reduce the unnecessary listening and talking state switches. They yield a more accurate inference on the start and end time of each call state, as illustrated in Figure 5.

○ *DBSCAN is used to handle hidden noises and prevent the unwanted listening→talking state transitions*. We analyze nonCN packets and classify them into two categories: (1) voiced nonCN and (2) unvoiced nonCN. The hidden noise packets belong to the unvoiced nonCN. The unvoiced nonCN packets do not carry any user voice but carry uncanceled environment noise. Although the coding bit rate of unvoiced nonCN packets is specified to be smaller than that of voiced nonCN according to VoIMS standards [17], differentiating those two categories is still challenging due to the AMR audio codec used by VoIMS, in which voice coding rates may vary with time. To this end, we develop a classifier based on the well-established DBSCAN algorithm [18]. DBSCAN is employed for the classification with the input of a desired number of clusters or categories and a proper value of ϵ, which represents the maximum distance range between two data points belonging to the same cluster. In our prototype, ϵ is set to 10, which achieves comparable performance on all audio coding rates stipulated by VoIMS standards.

Metrics		Cross-Carrier Exp.			Cross-Phone Exp.		
		OP-I	OP-II	OP-III	OP-III	OP-III	OP-III
		S5	S5	S5	G3	Pixel 3	Pixel 5†
C	Inference accuracy	100%	100%	100%	100%	100%	100%
	Time errors	0.51s	0.3s	1.8s	0.53s	0.57s	0.39s
T	Inference accuracy	100%	100%	100%	100%	100%	100%
	Time errors	0.36s	0.6s	0.99s	0.14s	0.89s	0.22s
L	Inference accuracy	100%	100%	100%	100%	100%	100%
	Time errors	0.48s	0.4s	1.22s	0.45s	0.76s	0.31s

TABLE III: Accuracy of VoIMS call detection, state and time inference (†:VoNR). C: Conversation, T: Talking, L: Listening.

○ *MAVG is used to deal with redundant comfort noises and prevent the unwanted talking→listening state switches.* To address this issue, the talking state should not be transferred to the listening state based on only little comfort noise. We develop a moving average algorithm to prevent unnecessary state switches. Notably, since the short speaking pauses may generate not only comfort noise packets but also unvoiced nonCN packets, both of them are considered in the algorithm. The devised algorithm works as follows. First, given a pre-defined time window (e.g., 2∼4 seconds) over time, wnd, we collect statistics on the numbers of uplink comfort noise packets, unvoiced nonCN packets, and voiced nonCN packets, denoted as *#CN*, *#Unvoiced-nonCN*, and *#Voiced-nonCN*, respectively. Second, we calculate the percentage of voiced packets, VP, within each wnd, by $\frac{\#Voiced-nonCN}{\#CN+\#Unvoiced-nonCN+\#Voiced-nonCN} * 100\%$. Third, if the observed VP is larger than 50%, the talking state is inferred; otherwise, the state is inferred as listening.

Finally, we infer the time for this call conversation as follows. The conversation starts when the *talking* or *listening* state is identified at the first time, which is more robust than detecting the first PDCP packet over this target DRB. The conversation ends when the last PDCP packets is sent/received over this target DRB. Note that time inference may be slightly inaccurate because the call is officially established or terminated by SIP, which is delivered over SRB, not DRB.

D. Evaluation on 5G/4G Call Inference

We have conducted extensive experiments with those three US operators to evaluate the accuracy of the proposed side-channel inference techniques, in terms of VoIMS call detection, and call state and time inference. We have tested with four 4G/5G smartphones, including Samsung Galaxy S5, LG G3, Google Pixel 3, and Google Pixel 5 and two mainstream VoIMS services, VoLTE and VoNR (Google Pixel 5 only). There are 20 runs for each experiment setting (operator, phone model, VoIMS service). In each run, a victim VoIMS call takes 30 seconds with 10s for alerting time, 10s for talking, and 10s for listening. In the meanwhile, other participating smartphones (not the victim) run various accompanying traffic including non-VoIMS Internet applications and VoIMS calls. Here, we use fixed 10s intervals in the evaluation experiments as they are sufficient for us to examine all different call state transitions. The call inference with varying intervals and multiple pauses will be evaluated in §V.

Table III shows that side-channel call inference works well in both cross-carrier and cross-phone cases. Due to space limits, we present the results with all operators using Samsung S5 and the results with all phone models using OP-III. We have three findings. First, all the VoIMS calls and states are successfully detected, while non-VoIMS traffic is never mistakenly recognized as VoIMS calls. Second, all the average time errors are below 9%, except the inference using S5 over OP-III. Specifically, they ranged in 0.3-0.57s (1.5%∼2.85%), 0.14-0.89s (1.4%∼8.9%), and 0.31-0.76s (3.1%∼7.6%) for conversation (20s), talking (10s), and listening (10s), respectively. Third, in the experiments using S5 with OP-III, CN packets are transmitted at a low rate (i.e., $\leq$7 pkt/s) to the cellular infrastructure. Such a low rate is much lower than the rate stipulated by the standard (i.e., 50 pkt/s) [8], thereby resulting in a longer call state inference and higher error rates.

V. Cross-domain Identity Linkage Attack

Motivation. Identity leakage is a big privacy concern for mobile users; two kinds of identities are mainly considered, namely *user identity* and *cellular identity*. The user identity, e.g., name, phone number, can be used to identify an individual uniquely, whereas the cellular identity, e.g., International Mobile Subscriber Identity (IMSI), and Radio Network Temporary Identity (RNTI), can identify a mobile user in cellular networks. However, if a mobile user's user and cellular identities can be inferred and linked successfully, many powerful cellular-identity-based cyberattacks (e.g., IMSI-based DoS attacks [19]) can be launched against high-value victims rather than randomly selected ones.

The existing works and their limitations. Many studies (e.g., [20], [21]) have shown that the adversary can easily steal the user identities, including both name and phone number, of a mobile user through online payment services (e.g., PeopleLooker [22]), social network platforms (e.g., [20]), and data breaches from online service providers [23]. For the cellular identities, several methods [24]–[26] have been also proposed to infer them or link them to each other (e.g., forcing a device to immediately transmit its IMSI, linking RNTIs to an IMSI [26]). However, there are currently no studies which can stealthily link the user identities to the cellular ones for a mobile user. Even though some works have shown that such linkage is feasible, their schemes are not stealthy. For example, for the linkage, Hussain *et al.* [25] need to dial multiple calls to the victim with knowing his/her number in advance. Note that compromising carriers' infrastructure is not considered in the threat model in §III.

Proposed Approach. We thus develop a novel **Cross**-domain **I**dentity **L**inkage attack, *CrossIL*, which exploits precise call state inference and correlates inferred call states with related motions extracted from the visualization domain (say, video recordings). Figure 6 shows the basic attack idea.

Besides the success of VoIMS call state inference in the cellular radio domain, the proposed CrossIL attack is motivated by two rationales in the visualization domain: (1) the user postures of using the VoIMS services (e.g., holding a phone next to an ear) are usually different from those of the other mobile services (e.g., Internet surfing and texting), which provides a special feature distinguishing call activities from others. We conducted an online questionnaire studying the phone call access behavior; most participants are college students. From 83 collected responses, 53 participants prefer to place phones to their ears when making phone calls in public locations, whereas 30 participants do not (e.g., using earphones); and (2) face recognition techniques are getting increasingly mature; several studies (e.g., [27], [28]) have demonstrated that adversaries can successfully recognize people's faces with a high accuracy (>90%) in video frames even when faces are small/tiny. With recognized face images, adversaries can first discover their owners' names by reverse image search engines (e.g., PimEyes [29]) and then obtain their phone numbers by paid online services (e.g., PeopleLooker and Spokeo).

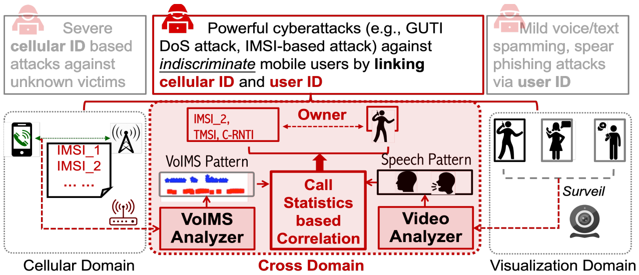
Severe cellular ID based attacks against unknown victims
Powerful cyberattacks (e.g., GUTI DoS attack, IMSI-based attack) against indiscriminate mobile users by linking cellular ID and user ID
Mild voice/text spamming, spear phishing attacks via user ID
IMSI_1
IMSI_2
... ...
IMSI_2, TMSI, C-RNTI
Owner
VoIMS Pattern
Speech Pattern
Call Statistics based Correlation
VoIMS Analyzer
Video Analyzer
Surveil
Cellular Domain
Cross Domain
Visualization Domain

Fig. 6: Overview of *CrossIL* attack.

Practical Attack? Critics may argue that the proposed CrossIL faces three limitations: (1) adversaries have to install cellular radio sniffers and surveillance cameras in public areas; (2) multiple users may have VoIMS calls at similar times; and (3) there may be no VoIMS users with ongoing calls during surveillance. However, we contend that these issues can be addressed without significant technical challenges. Specifically, modern hidden cameras [30] are unobtrusive with extended battery life and ample storage capacity, while portable sniffers can support more than 1 km coverage [24], thereby allowing adversaries to operate covertly. Our precise call state inference mechanism can distinguish between multiple VoIMS users with concurrent calls. Moreover, CrossIL specifically targets VoIMS users, but not all mobile users without accessing VoIMS services. Adversaries can strategically deploy the sniffers and cameras at selected public locations, such as airports and hotel lobbies, where phone calls are frequent. With many people throughout the day, it is likely to observe individuals making calls within the surveillance coverage.

Attack design. The high-level attack idea is to obtain cellular identities from radio traces (via *VoIMSAnalyzer*) and user identity from the visualization domain (via *VideoAnalyzer*), and then link them by correlating the victim's call states and related motions. The biggest challenge of launching this attack is to accurately detect when a voice conversation starts from recorded videos. Specifically, it is hard to distinguish the following two scenarios: (1) the user dials an outgoing call and waits for the called party to answer; and (2) the user answers an incoming call and listens to the caller without speaking at all. In both scenarios, the user has the same behaviors: the user moves a phone to his/her ear and has no lip motions for a while. This issue results in non-negligible inference errors in the visualization domain, thereby significantly downgrading

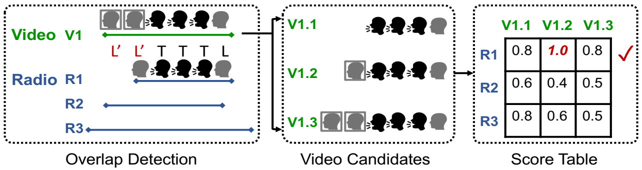

Fig. 7: Three steps for cross-domain identity linkage.

the attack effectiveness. We thus develop a novel approach, *cross-domain indeterministic call state correlation* to address the issue by adding an indeterministic state L′ to present these two scenarios. We next present its three key components.

○ *VoIMSAnalyzer*. The new function added for this attack is to extract cellular radio identity of each VoIMS call (i.e., RNTI, IMSIs, TMSIs, GUTIs), if there are any. That is to say, *VoIMSAnalyzer* not only discovers cellular identities but also infers each identity's corresponding VoIMS call states (e.g., talking and listening times) over ciphered radio traces.

○ *VideoAnalyzer*. It leverages increasingly mature face recognition techniques which can successfully recognize people's faces with a high accuracy (>90%) in video frames even when faces are small/tiny [27], [28]. It also exploits public image reserve searching engines that search people by face images (e.g., PimEyes.com [29]). *VideoAnalyzer* extracts each user identity's call-related motions and then generates estimated call statistics from video recording. *VideoAnalyzer* contains three sub-modules: (1) *call motion detector*, which detects two voice-call-specific human activities (i.e., moving a phone close to/away from an ear) to identify call start and end times, respectively; (2) *lip motion detector*, which identifies the start and end times of each talking or listening interval by analyzing human lip motions using the recurrent neural network (RNN) model [31]; and (3) *face detector and recognizer*, which locates each user's face and discovers the corresponding user identity (e.g., name) using the Dual Shot Face Detector (DSFD) [32] and ResNet50 [33]. For each identified user identity, *VideoAnalyzer* outputs the start and end times of each call conversation, and the call's talking and listening time intervals, which may be interleaved.

○ *Cross-domain identity linkage.* This component associates a cellular radio identity with a user identity by correlating their corresponding call event sequences generated by *VoIMSAnalyzer* and *VideoAnalyzer*. Figure 7 shows three steps for cross-domain indeterministic call state correlation.

Step 1. Given a video-induced call record produced by *VideoAnalyzer*, the correlator searches through all radio-induced call records and checks whether any of them overlaps with it based on their call start and end times.

Step 2. Due to the indeterminacy of L′, which indicates listening to the other call party or waiting for the called party to answer, the video-induced call event sequence, designated as $CESeq_{video}$, is not a deterministic form. We thus expand $CESeq_{video}$ to multiple deterministic call event sequences by exploring all possible states of L′ in practice. For example, a video-induced call event sequence, "S, L′, L′, T, T, L, E", can be expanded into three sequences: (1) "S, L, L, T, T, L, E"; (2)"S, L, T, T, L, E"; and (3) "S, T, T, L, E", which outputs all possible call event sequences.

Step 3. We calculate matching scores between each of the radio-induced call event sequences and all the sequences expanded from the given video-induced call event sequence; the correlation with the highest matching score is chosen. We calculate the Edit distance (i.e., Levenshtein distance), which quantifies the similarity between two strings, between two selected call event sequences and obtain their matching score as $1 - \frac{\text{Edit distance}}{|\text{Longest call event sequence}|}$. For example, the Edit distance between "S, T, T, L, E" and "S, L, T, T, L, E" is 1, and their matching score is 0.83 ($1 - \frac{1}{6}$).

Attack implementation. We implement *VoIMSAnalyzer* using Python3 on a PowerSpec computer with Ubuntu 16.0, 8 Intel i7-9900k CPUs, and 15GB RAM. We implement *VideoAnalyzer* using Python3 on HPCC servers with the following libraries: Keras (Mask R-CNN and RNN lip movement model), Pytorch (DSFD), keras_vggface (ResNet50), scipy (Cosine Similarity), and cv2 libraries. Notably, we did not need to collect a large-scale dataset, since all used models were pretrained [34]. Correlator was implemented in Python3 using timestamps recorded in radio traces and videos.

Attack evaluation. We run experiments at public places covered by three U.S. major carriers in a responsible manner. All recording experiments conform to the United States Recording Laws [35]; we record videos only with our participants. We record videos without capturing their conversations when several participants are speaking. More specifically, the attack evaluation is performed in both controlled (without passersby) and wild (with passersby) environments. The controlled experiment was conducted in a classroom, where only experiment participants were on campus during holidays, whereas the wild experiment was carried out in the lobby of a dormitory with passersby. There were 7 participants, and each of them was required to freely dial/receive 15 VoIMS calls within two hours under the surveillance of two cameras (iPhone 12). The participants can make phone calls simultaneously. In the experiment, we gauged the inference accuracy in terms of call start time, call end time, talking/listening times, and the association between the cellular and user identities.

Table IV summarizes the experimental results. We have four observations. First, the success rates of linking cellular identities to user identities are 59/60 (98%) and 40/45 (89%) in controlled and wild environments. Such high success rates are achieved even when *VideoAnalyzer* has up to 17.3% error in estimating talking and listening times. Second, most estimation errors from *VideoAnalyzer* in the wild environment are obviously larger than those in the controlled environment. There are two reasons: (1) cameras were occasionally blocked by passersby (1/45 phone calls), and (2) the brightness of natural light is not always stable; specifically, 14/45 phone calls experienced short-time (a few seconds) underexposure/overexposure issues. Third, *VideoAnalyzer* can precisely recognize the faces of participants for all the VoIMS calls and then discover their names from our database. Fourth, *VoIMSAnalyzer* in the controlled experiment has similar errors

Module	Performance Metrics		Controlled Settings				Wild Settings		
			User1	User2	User3	User4	User5	User6	User7
RadioAnalyzer	Call Events Time Estimation	call start error	0.92s	0.32s	0.85s	0.85s	1.20s	1.43s	1.60s
		call end error	0.18s	0.27s	0.32s	0.37s	0.32s	0.15s	0.28s
		talking & listening time error	1.8s (4.6%)	1.4s (2.8%)	2.3s (4.9%)	1.6s (3.9%)	1.5s (2.8%)	2.88s (6.5%)	2.65s (6.3%)
VideoAnalyzer	Call Events Time Estimation	call start error	1.87s	2.53s	1.99s	2.0s	6.42s	3.09s	2.85s
		call end error	2.15s	3.74s	3.97s	2.14s	3.01s	4.95s	1.18s
		talking & listening time error	3.2s (8.2%)	4.12s (8.3%)	2.15s (4.6%)	2.23s (5.4%)	9.25s (17.3%)	6.67s (15.1%)	4.37s (10.4%)
	Face Detection and Recognition	Accuracy	100%	100%	100%	100%	100%	100%	100%
Correlator	**Cellular ID and User Linkage**	**Accuracy**	**100%** **(15/15)**	**93.5%** **(14/15)**	**100%** **(15/15)**	**100%** **(15/15)**	**86.7%** **(13/15)**	**86.7%** **(13/15)**	**93.5 %** **(14/15)**

TABLE IV: Summary of cross-domain identity linkage attack performance.

in estimating call start and end times as in the wild experiment, whereas its estimation on talking and listening times in the wild setting has higher errors (2.8%∼6.5%) than that in the controlled setting (2.8%∼4.9%). The reason is that the background noise of the wild environment is larger than that of the controlled one.

Current prototype limitations. Although the proposed attack is effective in our experiments, the current prototype has several limitations: (1) video recordings at 1080P or higher are required; (2) skewed, crooked, and blurred faces cannot be well recognized; (3) video call and earphone users are immune to this attack; and (4) it only considers the locations with good cellular signals. Some techniques can be used to improve the prototype. For example, [36] and [37] can be applied to recognizing faces from low-resolution video recordings and dealing with the skewed/crooked face recognition, respectively. We leave these potential improvements to our future work.

VI. Solution

We propose to add paddings to compressed CN packets so that there are no tiny packets. Specifically, we would like to control paddings below IP (at the PDCP layer), rather than above IP. Adding paddings above IP results in two practical issues. First, users need to pay for additional paddings, because they are charged based on the volume of data usage. Second, IMS media gateways suffer with increasing loads while handling extra paddings in VoIMS packets. Specifically, we propose to develop a singular rectifier (SR) at the PDCP layer, which is implemented on the UE and the base stations, without reaching the core network. By this means, users do not need to pay for extra paddings and handling extra paddings is distributed among front-end base stations, which is acceptable.

Prototype. We prototype the proposed solution over an open-sourced cellular network. We use srsUE, srsLTE, Open IMS Core [38], and UCT IMS Client 1.0.14 [39] to serve as the 4G UE, the 4G infrastructure, the IMS core with a VoLTE server, and the VoLTE client app, respectively. We implement SR@PDCP by modifying the PDCP layer at both the UE and the eNB to handle necessary paddings of the VoIMS packets. In particular, we add paddings to all the PDCP packets whose payload lengths are smaller than 20 bytes and increase the size to 42 bytes, which can be observed from many non-VoIMS applications (Table I). The inserted paddings are removed at the PDCP layer before the corresponding packets are forwarded to the upper layer or next network element (e.g., the 4G gateway and 5G UPF).

Evaluation. We evaluate its effectiveness and overhead. We re-run VoIMS call experiments where a user dials ten VoIMS calls from the srsUE and the callee answers each incoming call immediately. Each call lasts for 25 seconds with 10s for talking and 15s for listening. The result shows that *VoIMSAnalyzer* fails to detect any of the calls. Notably, the 4G core network does not receive any VoIMS packets with additional paddings, so no additional charges are made.

We evaluate the solution overhead in terms of CPU usage, memory usage, and processing time. In practice, VoIMS clients do not keep sending out compressed CN packets during voice calls. Here, we assess the overhead in an extreme case where the VoIMS client keeps transmitting 11-byte comfort noise VoIMS packets to the OpenIMS server. Our experimental results show that the average CPU and RAM usages slightly increase by 0.72% and 0.02% when the proposed solution is enabled. The average processing time at the PDCP layer increases by 1.46 μs per packet (from 15.24 μs to 16.70 μs).

VII. Discussion

Limited to 4G? People may argue that 5G users should be immune to the proposed attacks, since mobile devices in the 5G network do not transmit the permanent cellular identity (e.g., IMSI) in plaintext but in ciphertext (i.e. Subscription Concealed Identifier (SUCI)), and the cellular identity cannot be learned. However, it may not be the case since some researchers have shown the network downgrade attacks [40] can downgrade 5G mobile devices to legacy 4G networks.

Having more attacks? We mainly use the above proof-of-concept attack to evaluate the effectiveness and responsiveness of call inference and applications in real world using wild experimental settings. Its real-world impact is not limited to our proof-of-concept attack only. For example, many studies in sociology and linguistics have leveraged call state information to infer user profiles (e.g., residents, visitors [41]), personality (e.g., extroversion, agreeableness [42]), and social interactions [43].

VIII. Related Work

4G/5G voice security has attracted increasing attention in the recent years [14], [44]–[47]. Prior studies focus on other security problems, such as VoLTE call reliability [44], free data access by abusing VoLTE signaling [45], DoS attacks [46], deciphering VoLTE calls [14], and 911 call security [47]. Note that although [14] claims that the ciphered packets of VoLTE calls can be decrypted, it leverages a vulnerability that an encryption key is reused for different VoIMS calls of the same mobile user. However, it should rarely happen in practice since the key reuse is explicitly forbidden by 3GPP.

To the best of our knowledge, there are no similar security studies to infer 5G/4G call information without knowing the decryption keys and then launching attacks. The most relevant work is [48], which infers confidential or hidden information in VoWi-Fi. It infers three call states (ringing, conversation, and non-conversation) of a VoWi-Fi call by analyzing the intercepted IPSec packets. However, their inference approach is not applicable to VoIMS calls. They cannot detect VoIMS calls in the presence of non-VoIMS traffic; they cannot infer fine-grained call states and time.

IX. Conclusion

5G/4G voice calls are advancing with new optimization techniques over time. However, the gains are not without a cost. In this work, we present side-channel call inference over several 5G/4G call optimization techniques and the resulting threats against 5G/4G calls. An adversary can infer confidential call information accurately (whether a call is ongoing, who is talking and when) without decrypting the encrypted packets in the air. Such call information can be further exploited to launch real attacks against 5G/4G call users. We believe that new techniques never intend to sacrifice security for call enhancements. However, their security implications are subtle, resulting in unanticipated side effects. Handling new security issues raised by evolving technologies in mobile networks is a big challenge and an endless task. It warrants concerted efforts from all the stakeholders, including standard makers, carriers, vendors, and mobile users.

Acknowledgment

We appreciate the insightful and constructive comments from the anonymous reviewers. This work is supported in part by the National Science Foundation (NSF) under Grants No. CNS-1750953, CNS-1815636, CNS-1814551, CNS-2112471, CNS-2153393, CNS-2226888, CNS-2246050, CNS-2246051, and CCF-2007159, and by the National Science and Technology Council (NSTC) under Grants No. 109-2628-E-009-001-MY3, 110-2221-E-A49-031-MY3, 112-2628-E-A49-016-MY3, 112-2218-E-A49-021 and 112-2634-F-A49-001-MBK. 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 NSF and NSTC.

References

[1] "'good call study' finds phone calls are more relevant than ever," https://www.weboost.com/blog/good-call-study-finds-phone-calls-are-more-relevant-than-ever, 2020.

[2] GSMA, "Ir 92 ims profile for voice and sms," 2020.

[3] ——, "Volte market update," https://www.gsma.com/services/wp-content/uploads/2022/04/GSMAi-VoLTE-Market-Update-final.pdf.

[4] 3GPP, "TS 33.102: 3G security; Security architecture," 2022.

[5] ——, "TS 33.210: 3G security; IP network layer security," 2022.

[6] ——, "TS 36.323, Packet Data Convergence Protocol (PDCP)," 2022.

[7] IETF, "RFC 3095 RObust Header Compression (ROHC)," 2001.

[8] 3GPP, "TS 26.071: AMR speech Codec," 2020.

[9] ——, "TS 26.092: Adaptive Multi-Rate (AMR) speech codec," 2020.

[10] ——, "TS 33.501: Security architecture and procedures for 5G," 2022.

[11] ——, "TS 33.401: System Architecture Evolution (SAE)," 2022.

[12] ——, "TS 37.324: Service Data Adaptation Protocol (SDAP)," 2020.

[13] MobileInsight, "Mobileinsight," http://www.mobileinsight.net/, 2021.

[14] D. Rupprecht, K. Kohls, T. Holz, and C. Pöpper, "Call me maybe: Eavesdropping encrypted lte calls with revolte," in *USENIX Security'20*.

[15] IETF, "RFC 3389: Real-time Transport Protocol (RTP) Payload for Comfort Noise (CN)," 2002.

[16] ——, "RFC 3960 Early Media and Ringing Tone Generation in the Session Initiation Protocol (SIP)," 2005.

[17] 3GPP, "TS 26.445: Codec for Enhanced Voice Services (EVS)," 2020.

[18] M. Ester *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *ACM SIGKDD'96*.

[19] H. Yang, S. Bae, M. Son, H. Kim *et al.*, "Hiding in plain signal: Physical signal overshadowing attack on {LTE}," in *USENIX Security'19*.

[20] X. Chen, X. Song, G. Peng, S. Feng, and L. Nie, "Adversarial-enhanced hybrid graph network for user identity linkage," in *ACM SIGIR'21*.

[21] S. Liu, S. Wang, F. Zhu *et al.*, "Hydra: Large-scale social identity linkage via heterogeneous behavior modeling," in *ACM SIGMOD'14*.

[22] "Peoplelooker," https://www.peoplelooker.com, 2020.

[23] "Yahoo! data breache," https://en.wikipedia.org/wiki/Yahoo!_data_breaches, 2020.

[24] M. Kotuliak, S. Erni, P. Leu, M. Roeschlin, and S. Capkun, "LTrack: Stealthy Tracking of Mobile Phones in LTE," in *USENIX Security'22*.

[25] S. R. Hussain *et al.*, "Privacy attacks to the 4g and 5g cellular paging protocols using side channel information," in *NDSS'19*.

[26] S. Kumar, E. Hamed, D. Katabi, and L. Erran Li, "Lte radio analytics made easy and accessible," in *ACM SIGCOMM'14*.

[27] Z. Li, X. Tang, J. Han, J. Liu *et al.*, "Pyramidbox++: High performance detector for finding tiny face," *arXiv preprint arXiv:1904.00386*, 2019.

[28] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *IEEE CVPR'15*.

[29] "Pimeyes-ceo: The user is the stalker, not the search engine." https://netzpolitik.org/2022/pimeyes-ceo-the-user-is-the-stalker-not-the-search-engine/, 2022.

[30] "Mini spy camera 1080p," https://www.amazon.com/Spy-Camera-Charger-Hidden-Surveillance/dp/B07GCKZKX8/, 2021.

[31] Sachinsdate, "Speaker detection by watching lip movements," https://github.com/sachinsdate/lip-movement-net, 2016.

[32] J. Li, Y. Wang, C. Wang, Y. Tai, J. Qian, J. Yang, C. Wang, J. Li, and F. Huang, "Dsfd: dual shot face detector," in *IEEE CVPR'19*.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE CVPR'16*.

[34] K. He, G. Gkioxari, P. Dollár *et al.*, "Mask r-cnn," in *IEEE ICCV'17*.

[35] "United states recording laws," https://recordinglaw.com/united-states-recording-laws/, 2022.

[36] D. Kamenetsky, S. Y. Yiu, and M. Hole, "Image enhancement for face recognition in adverse environments," in *IEEE DICTA'16*.

[37] X. Wang, K. C. Chan, K. Yu, C. Dong *et al.*, "Edvr: Video restoration with enhanced deformable convolutional networks," in *IEEE CVPR'19*.

[38] "Openimscore," http://openimscore.sourceforge.net/, 2008.

[39] D. Waiting *et al.*, "the uct ims client," in *IEEE TRIDENTCOM*.

[40] M. Khan, P. Ginzboorg, K. Järvinen, and V. Niemi, "Defeating the downgrade attack on identity privacy in 5g," in *SSR'18*.

[41] B. Furletti, L. Gabrielli, C. Renso, and S. Rinzivillo, "Identifying users profiles from mobile calls habits," in *ACM SIGKDD'12*.

[42] C. Stachl, Q. Au, R. Schoedel *et al.*, "Predicting personality from patterns of behavior collected with smartphones," *PNAS'20*.

[43] D. Wyatt, T. Choudhury, J. Bilmes, and J. A. Kitts, "Inferring colocation and conversation networks from privacy-sensitive audio with implications for computational social science," *ACM TIST'11*.

[44] Y. J. Jia, Q. A. Chen, Z. M. Mao, J. Hui, K. Sontinei, A. Yoon, S. Kwong, and K. Lau, "Performance characterization and call reliability diagnosis support for voice over lte," in *ACM MobiCom'15*.

[45] C.-Y. Li, G.-H. Tu, C. Peng, Z. Yuan, Y. Li, S. Lu *et al.*, "Insecurity of voice solution volte in lte mobile networks," in *ACM CCS'15*.

[46] Y.-H. Lu, C.-Y. Li, Y.-Y. Li, Hsiao *et al.*, "Ghost calls from operational 4g call systems: Ims vulnerability, call dos attack, and countermeasure," in *ACM MobiCom'20*.

[47] Y. Hu, M.-Y. Chen *et al.*, "Uncovering insecure designs of cellular emergency services (911)," in *ACM Mobicom'22*.

[48] T. Xie, G.-H. Tu, B. Yin *et al.*, "The untold secrets of wifi-calling services: Vulnerabilities, attacks, and countermeasures," *IEEE TMC'22*.