On Key Reinstallation Attacks over 4G/5G LTE Networks: Feasibility and Negative Impact

Preprint · November 2018

DOI: 10.13140/RG.2.2.27590.32323

Muhammad Taqi Raza
University of California, Los Angeles
24 PUBLICATIONS 161 CITATIONS

Songwu Lu
University of California, Los Angeles
189 PUBLICATIONS 15,289 CITATIONS

On Key Reinstallation Attacks over 4G/5G LTE Networks: Feasibility and Negative Impact

Muhammad Taqi Raza and Songwu Lu – Computer Science Department, UCLA

***Abstract*—This paper studies the feasibility of key reinstallation attacks in the 4G LTE network. It is well known that LTE uses session keys for confidentiality and integrity protection of its control-plane signaling and ciphering of its data-plane packets. However, if the keys are not updated and counters are reset, key reinstallation attacks may arise. In this paper, we show that several design choices on both control and data planes in the current LTE security setup are vulnerable to key reinstallation attacks. Specifically, on the control plane, the LTE security association setup procedures, which establish security between the device and the network, are disconnected. The keys are installed through one procedure, whereas their associated parameters (such as uplink and downlink counters) are reset through another different procedure. The adversary can thus exploit the disjoint security setup procedures, and launch the key stream reuse attacks. He consequently breaks message encryption, when he tricks the victim to use the same pair of keys and counter value to encrypt multiple messages. This control-plane attack hijacks the location update procedure, thus rendering the device to be unreachable from the Internet. Moreover, it may also deregister the victim from the LTE network. On the data plane, vulnerability arises when the device establishes a new data session with the network. The data access setup procedure resets the counter, but the encryption key is never updated. Leveraging this design deficiency, the attacker can reset counters at the victim device by altering the data establishment procedure. The negative impact of this attack includes decrypting voice messages over the LTE calls, as well as threats on the Cellular IoT (the new approach to IoT in 5G) data traffic. We have confirmed our findings with two major US operators, and found that such attacks can be launched with software-defined radio devices that cost about $299. We further propose remedies to defend against such threats.**

I. Introduction

The current fourth-generation (4G) Long Term Evolution (LTE) technology provides billions of users their daily mobile Internet access. Different from the wired Internet, LTE has made security a top design goal, thus deploying several built-in security mechanisms. Together, such procedures provide all key security functions of authentication, encryption, integrity and access control.

In this paper, we examine the LTE security from a new perspective. It is well known that, the encryption and integrity protection components in LTE use mature and well-tested crypto algorithms that have been used for decades. Therefore, it seems that neither exhibits vulnerabilities. Motivated by the recent efforts on key reinstallation threats over wireless [1], [2], [3], [4], we hypothesize that LTE may suffer from similar vulnerabilities. Indeed, our findings confirm the hypothesis. However, the threats are exposed via completely different procedures. Moreover, the threats exhibit for control-plane signaling handshake and data-plane packet forwarding. The impacts are also more damaging.

Specifically, we study LTE security key installation method and counters handling process for a number of LTE procedures (such as device registration, deregistration, location update and others). 4G (like 3G) employs Authentication and Key Agreement (AKA) protocol to install the security keys and enables the integrity protection of its signaling messages. After that it runs Security Mode Command procedure to activate ciphering of messages at LTE subscriber. LTE employs stream ciphers which have been a popular method of encryption for the confidentiality of its signaling and data packets. The ciphering algorithm takes key (installed through AKA procedure), counter value and a couple others as an input and generates keystream block. The keystream block, k is exclusive-ored (xored) with the plaintext message, m, to produce the encrypted message, $k \oplus m = e$. In practice, the keystream is truly random that generates the cipher text known as a one-time pad, proved unbreakable by Shannon[5]. It is an established fact that the security of stream ciphers rests on never reusing the keystream block k [6]. In case k is reused to cipher two different plaintext messages, m and n, then the encrypted texts $k \oplus m$ and $k \oplus n$ can be xored together to recover $m \oplus n$. By using chosen-text attack, one can further break $m \oplus n$, and gets the messages m and n.
The scenario in which LTE ciphering algorithm gives same keystream block over multiple rounds is the one in which the ciphering key remains constant and the counter value (responsible of generating random keystream block) is reset. We call this *"key reinstallation"* vulnerability. In this paper, we look LTE control-plane and data-plane procedures that lead to key reinstallation attacks.

The idea behind our control-plane attacks can be summarized as follows. In the security establishment procedure, the device first installs new key through authentication procedure. Once the key is installed, the network runs security mode command procedure to reset the counter values for encryption. In reality, the signaling message may be lost or dropped. In case, the device response to security mode command request is dropped, the network reinitiates the security mode command procedure. On receiving the replayed security mode command request from the network, the device resets the counter values again before generating the response message. This means two signaling messages sent after two security mode command responses are encrypted with same keystream block at device. We show that an attacker can force count resets by blocking the response to security mode command request. By intentionally forcing count resets, the confidentiality protocol can be attacked, e.g., packets can be replayed, decrypted, and/or forged. The attacker can launch attacks on device location update and de-registration procedures. These attacks render the victim device to be unreachable from the outside world (e.g. it cannot receive voice calls), or even leaves the device without

LTE service (i.e. no service scenario).

On the data plane, when the device migrates from idle to connected state, it installs a new radio key to encrypt its data packets. We find that whenever the device establishes a new path (i.e. bearer) for subscriber traffic (e.g. VoLTE call), the counters are reset. It means all those voice calls which are established while the device remains in the connected state are encrypted with the same keystream block. The attacker takes advantage of this vulnerability and launches *chosen-voice attack* towards the victim. We show that an attacker can easily reset the counters by establishing and terminating the VoLTE call connection. He gets his chosen-voice packets encrypted at victim device (e.g. talking to victim over the phone), resets the encryption counters, and then can decrypt, replay and even forge the victim private call's voice packets. Similarly, he can extend this attack towards Cellular IoT for 5G (standardized in recent 3GPP release).

It is worth noting that our attacks do not violate the security properties proven in formal LTE analysis work, such as LTEInspector [7]. The formal method proofs state that LTE key should not be shared over the air, and all protocols behave as desired by the 3GPP standard. Our attacks do not leak ciphering or integrity keys and strictly follow LTE standards. Further, although the attacker can launch the attacks by reseting the counts, in control-plane he cannot repeatedly do so for more than one signaling message as the integrity protection becomes mandatory thereafter. However, this is sufficient for an attacker to launch as serious an attack as deregistering the victim subscriber from the LTE network.

In our experiments, we have verified all attack steps through two major US LTE operators. We use Software Defined Radio (SDR) to conduct our proof-of-concept studies. The experimental results show that LTE key reinstallation attacks are practical and pose a realistic threat to the LTE users. Last, we propose 3GPP standard-compliant remedies to address the discussed vulnerabilities. We prototype our solution and provide its security analysis. A summary of our findings is shown in Table I.

Ethical Consideration: This work does not raise any ethical and legal concerns. The attacker and victim devices are part of a testbed setup established in our lab. We have especially purchased sim cards from two US operators to conduct our experiments. We did not use any other commercial sim card to launch the attack towards any other LTE subscriber. The purpose of this study is to strengthen LTE security, especially when LTE security mechanisms are considered to be the building blocks for 5G security (e.g. Cellular IoT security).

II. Background on LTE and Key reinstallation

We provide background on LTE infrastructure, integrity and ciphering procedures in LTE, as well as on key reinstallation vulnerabilities.

A. LTE network and its elements

LTE network consists of three main components: device, LTE base station, and the core network, as shown in Figure 1.

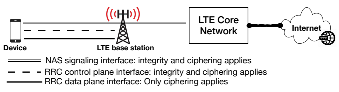
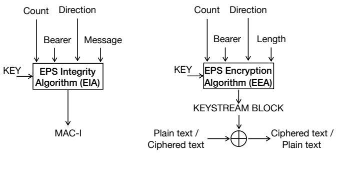

LTE Core Network
Internet
Device
LTE base station
NAS signaling interface: integrity and ciphering applies
RRC control plane interface: integrity and ciphering applies
RRC data plane interface: Only ciphering applies

Fig. 1: LTE background: an overview.


Count
Direction
Bearer
Message
KEY
EPS Integrity Algorithm (EIA)
MAC-I
Count
Direction
Bearer
Length
KEY
EPS Encryption Algorithm (EEA)
KEYSTREAM BLOCK
Plain text / Ciphered text
Ciphered text / Plain text

(a) Integrity protection

(b) Ciphering / unciphering

Fig. 2: Integrity and ciphering procedures.

1) LTE device: It provides LTE service to end user. The network operators assign the subscriber device a permanent identity called International Mobile Subscriber Identity (IMSI), and a number of temporary identities. One of these temporary identities is called Temporary Mobile Subscriber Identity (TMSI), a temporary identification number that is used instead of the IMSI to ensure the privacy of the mobile subscriber. The other temporary identity is known as RNTI (Radio Network Temporary Identifier) that uniquely identifies an LTE subscriber over the radio interface. LTE device applies ciphering to its both control and data planes, whereas the integrity is applied to control-plane only.

2) LTE base station: It acts as a radio interface between LTE subscriber and the core network. It provides radio resource management to its subscribers and encrypts user traffic over the air. Through RNTI, it discerns a particular user traffic from other subscribers over the air. The control-plane radio signaling messages between device and LTE base station are exchanged through Radio Resource Control (RRC) protocol. RRC is responsible of activating radio-plane security (through Security Mode Command procedure) and managing the radio resources (such as establishment, release, and radio configurations).

3) LTE core network: It is also called Evolved Packet Core (EPC) which acts as a central entity and provides authentication, mobility management, and Internet connectivity to LTE subscribers. The control-plane signaling messages between device and LTE core network are exchanged through Non-Access Stratum (NAS) procedures. NAS procedures include device Authentication (that installs security `key`), Security Mode Command (that enables ciphering), Attach (registration), Detach (de-registration), Tracking Area Update (location update), and few others procedures. In this paper, we first exploit vulnerabilities in Authentication and Security Mode Command procedures, and then launch attacks towards Tracking Area Update (TAU) and Detach control-plane procedures. Second, we exploit the vulnerability during establishment of data-plane connection and launch attacks on LTE data-plane.

B. Integrity and confidentiality procedures in LTE

LTE employs integrity and confidentiality procedures which are applied at both device and network side. Figures 2(a) and 2(b) show integrity and ciphering procedures, respectively. LTE uses two separate algorithms for integrity and ciphering of messages. Both algorithms take a number of input parameters and output the Message Authentication Code (MAC), if integrity algorithm is used, or `keystream block`, if ciphering algorithm is used. As shown in Figure 2, the input parameters are 28-bit integrity/ciphering `key` named KEY, a 32-bit `count` named Count, a 5-bit bearer identity, i.e.

TABLE I: Summary of findings

Plane	Vulnerabilities	Loophole	Key reinstallation Attacks	Root Cause	Defense Solution
Control-plane	Count reset on security mode failure; weak integrity protection	Re-authenticating the device on receiving msg failing the integrity chk rather than rejecting the msg	(i) Location hijacking and (ii) Subscriber de-registerationtion from LTE	Failure of security mode procedure does not renew the key	Bounding key installing and count reset procedures, activating integrity protection after authentication
Data-plane	Count reset on establishing *new* PDCP entity	Uplink and downlink counts are reset while keeping same radio key	Decrypting (iii) VoLTE calls, and (iv) Cellular IoT traffic	Not renewing the radio connection on establishing new PDCP entity	Installing fresh key on establishment of new PDCP entity

Bearer, the 1-bit direction of the transmission i.e. Direction (0 for uplink, and 1 for downlink transmission). The integrity algorithm takes the message itself, i.e. Message, as input as well, and outputs MAC; whereas, the ciphering algorithm inputs the length of the keystream required i.e. Length, to generate the output `keystream block`. This *Length* parameter affects only the length of the `keystream block`, not the actual bits in it[8]. The `keystream block` is xored with characters in the plaintext to produce the ciphertext at sender side. Xoring the ciphertext with same `keystream block` produces the plaintext at receiver.

C. Key reinstallation attack in retrospect

Stream ciphers, as discovered by Gilbert Vernam in 1917 [9], have been a popular method of encryption even today. In a stream cipher, the plaintext and the `key` are xored to produce the ciphertext. This cipher is never used again and known as one-time pad for encrypting plaintext message. To ensure that all ciphers do not occur more than once, the ciphering algorithm takes the nonce (we call `count` in this paper) as an input along with the `key`. Counts have the property that each value only occurs once within a given context. As long as the `key` is unchanged, the `count` must not repeat. Otherwise, it introduces the two-time pad problem [10] in which the adversary can get the encrypted plaintext without knowing the `key` and `count` values. The key reinstallation attacks can be defined as the attacks in which the adversary can *willfully* trick the victim to reuse the `count` values while keeping the `key` unchanged for encryption of the plaintext messages. It means stream cipher is reused for encryption, hence gives birth to two-time pads[10].

In wireless networks, David Wagner and his team have first shown stream cipher reuse attacks in WiFi[1], and WSN[2]. Following a number of papers [4], [11], [12] afterwards, last year, Mathy Vanhoef and et al.[3] show that the `key` reuse attacks are still possible in modern WiFi systems. They attribute this problem to design or implementation flaws. In contrast, in this paper, we are first to show that LTE security is vulnerable to key reinstallation attacks. It was challenging because unlike WiFi and WSN, LTE has separate security keys and `counts` for control-plane and data-plane operations. Even within the control-plane, RRC and NAS messages use separate security keys and `counts` for their messages integrity and ciphering. Furthermore, LTE security also splits its `counts` into Uplink (UL) `count` and Downlink (DL) `count` values that make `count` reuse harder. Despite all these efforts by LTE standard to avoid `key` reuse, we have shown the key reinstallation attacks in both LTE control and data planes.

III. System Settings and Threat Model

System settings The attacker controls LTE device (i.e. attacker device) that is associated with the same LTE network operator as that of victim subscriber. Both attacker and victim are located in an area where the network operator supports both 3G and 4G LTE services. The attacker knows the phone number of victim device, and can dial Circuit Switched Fall Back (CSFB) call towards victim device[1]. The victim device can receive the call either through CSFB or Voice over LTE (VoLTE). Lastly, we consider both the attacker and victim devices are static during the attack period. That is, we do not evaluate the mobility scenarios.

Threat Model Similar to threat models as discussed in [13], [3], [7], our attacker has capability to act as a passive and an active attacker. Being a passive attacker, he can sniff radio channel with which the victim has associated. He can do so by sniffing Physical Downlink Shared Channel (PDSCH). PDSCH is used to transport both broadcast system information for all devices and signaling/data payload for particular mobile devices. The attacker identifies different subscribers through their unique radio identity, C-RNTI.

Being an active attacker, he has capability to modify the contents of the messages (after decryption) that he has sniffed over the air. There exists a number of commercial LTE signal messages sniffers, such as WaveJudge[14], ThinkRF[15], and others that the attacker can use to sniff both broadcast and device specific signaling messages. Contrary to attack models discussed in [13], [3], our attack model is more practical in which the attacker does not need to act as Man-in-the-Middle (MitM) to forward modified messages towards the network. To impersonate the victim device, if required, the attacker spoofs victim's C-RNTI and TMSI values when he creates his own RRC and NAS messages, and sends his signaling messages to LTE base station. The spoofing is essential to trick LTE base station to use victim context (not the attacker's context) while forwarding message to core network.

In order to ensure that failure of certain signaling messages result in reseting the `count` values, the attacker has capability to block UL (from device to network) signaling messages. He achieves this by jamming UL signaling. There are a number of techniques[16], [17], [18] to jam the signaling messages. We consider Asynchronous Off-Tone Jamming (AOTJ) approach to jam only UL signaling messages between victim device and the network. The core idea of jamming is to introduce the inter-channel interference (ICI) among orthogonal OFDM subchannels. The interference brings loss of subchannel orthogonality, and as a result the network cannot recover the original OFDM data symbols over its subchannels which are spectrally overlapping. In our AOTJ technique, the jammer is off-tone or not synchronous with the target signal. It transmits asynchronous off-tones which are not perfectly periodic or have an offset at the sampling frequencies that brings ICI at the receiver.

[1] The attacker selects CSFB option (which is voice call option over 3G) in android/iOS phone call settings.

Evaluation of attacks We evaluated our attacks in terms of their feasibility and practicality over real operational LTE networks. We use Google Pixel 2 as an attacker device, and Google Pixel 1 as victim device. We consider two U.S LTE operators, i.e. AT&T (OPI) and T-mobile (OPII) to run our experiments. The attacker and victim devices use AT&T and T-mobile pre-paid sim cards to register with these two network operators. We use LTE signaling messages analyzer, MobileInsight, to capture LTE signaling messages at both attacker and victim devices. We run total of 200 experiments on each network operator to access the practicality of attack for each attack step. We run experiments in a lab setting with LTE radio signal strength range -90 to -100 dBm for both operators.
To evaluate the practicality of the attacks, we use low-cost commodity SDR hardware (HACKRF One) of the value of $299 to jam LTE signaling messages. HACKRF One has capability to block UL and DL LTE signaling messages by generating ICI signals towards LTE frequency band. To calibrate start and stop of jamming with respect to LTE signaling messages, we use QXDM[19] which is a real time LTE signaling messages sniffing/capturing tool from Qualcomm. The victim device is connected to QXDM (running on a PC) through USB.

IV. Overview on attacks and their root causes

We provide a brief overview of our findings on both control and data planes.

A. Control plane

On the control plane, the attacker can launch two types of key reinstallation attacks by exploiting signaling vulnerabilities. In the first type, he hijacks the location update procedure of the victim device. Consequently, the network cannot reach the victim for the incoming voice calls and data packets destined to the device. In the second type, the attacker may incur LTE service outage at the victim device by deregistering the device from the network. In our experiments, we demonstrate the feasibility of these attacks over real LTE carrier networks and their practicality in our testbed. There are two root causes for the attacks. First, the LTE control-plane procedure is vulnerable to key reinstallation attacks. Such attacks arise when the `count` reset procedure (i.e. LTE Security Mode Command procedure) is allowed to re-execute many times after the completion of `key` installation procedure (i.e. LTE NAS Authentication procedure). The other root cause is that certain control-plane messages are partially accepted even though they fail the integrity check. These messages are finally accepted when the network re-authenticates the device. The network does not request the device to re-send the message that has failed the integrity check.

B. Data plane

On the data plane, the adversary attacks the LTE voice service (i.e., VoLTE). The attacker can decrypt the LTE voice packets exchanged between the victim and his calling party. Both incoming and outgoing voice packets can be decrypted, i.e., the adversary can sniff the victim conversation on both incoming and outgoing voices, even if the victim uses ear-phones to confine incoming voice packets from leaking to the surroundings. Later we show that the attacker can exploit the same vulnerability to launch attacks against Cellular IoT. The main root cause is that the LTE data-plane procedure resets `count` values when the device establishes a new data connectivity. The attacker can establish the voice connectivity with the victim device by simply making a phone call. These vulnerabilities thus lead to key reinstallation attacks.

V. Attacking LTE Control Plane

Overview We demonstrate the feasibility and practicality of key reinstallation attacks in LTE control-plane. The adversary adopts the fact into his advantage that on inter-system switch from LTE → 3G→ LTE, location update procedure is triggered that installs the `key` and resets the `count` values. The attacker silently[2] brings an inter-system switch at victim device through CSFB procedure. He lets the device to complete the `key` installation procedure, but strategically blocks the victim device UL signaling messages to bring `count` reset procedure failure. The network re-initiates the failed procedure that resets the `count` values at device again. This results into `keystream block` reuse for those signaling messages that the victim device sends after resetting the `count` values. The attacker stops jamming, encrypts his spoofed message by using victim's `keystream block`[3], and dispatches it to the network without being MitM. The network receives two messages, the one originated from the device and the other from the attacker. The network executes the latest received message, according to 3GPP standard[20], and discards the message received earlier. This makes our attack realistic as the attacker message and the victim messages are not racing with each other. Because the message was modified by the attacker, it fails the integrity check at the network. However, instead of dropping the packet, the network re-authenticates the victim device and accepts the received spoofed message.

Roadmap We first show LTE location hijacking attack due to key reinstallation vulnerabilities. We provide the feasibly analysis from LTE standard followed by the detailed attack procedure. We show step by step attack procedure and access the practicality of each step through experiments. Lastly, we extend this attack and design an other type of attack, i.e. LTE service outage attack.

A. LTE Location Hijacking Attack

1) Feasibly analysis from LTE standard: Following we discuss two vulnerabilities that we exploit in attacking LTE confidentiality and integrity protocols.

1.1 LTE Integrity and confidentiality are enforced through two disjoint procedures LTE security is enforced through two separate procedures. In the first procedure, the LTE core network invokes mutual authentication procedure, i.e. AKA procedure, with the subscriber device. In LTE AKA procedure, as shown in Figure 3 (upper rectangular part), the core network element sends an Authentication Request message to the device. The device authenticates the LTE core network element, installs the `key` and sends the Authentication

[2] Attacker terminates the call before the victim device starts ringing, hence making it silent inter-system switch.

[3] Attacker gets the `keystream block` by *xoring* location update message with the encrypted message.

Response message to network. LTE core network verifies the response message and installs the key at its end. After authentication procedure, the network triggers NAS Security Mode Command (SMC) procedure. The network sends SMC message to device, as shown in Figure 3 (lower rectangular part) that includes NAS security algorithms to derive integrity and ciphering keys[4], as well as NAS-MAC (NAS Message Authentication Code). As the device does not know the selected encryption algorithm yet, this message is integrity protected only but not ciphered. On receiving the SMC message the device resets the pair of count (one for UL and one for DL transmission) values to *zero* after NAS-MAC verification for integrity protection. The LTE security specification (3GPP TS 33.401[8]) states:

> "*Only after EPS AKA the NAS security mode command message shall reset NAS uplink and downlink COUNT values. Both the NAS security mode command and NAS security mode complete messages are protected based on reset COUNT values (zero).*"

Thereafter, the device generates NAS Security Mode Complete message to network which is both ciphered and integrity protected. The network successfully verifies the integrity of the received NAS Security Mode Complete message and resets the counts. Now the NAS security setup procedure is said to be completed.

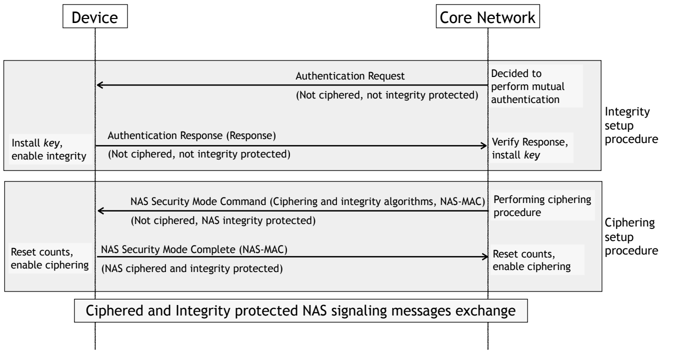

Device
Core Network
Authentication Request
(Not ciphered, not integrity protected)
Decided to perform mutual authentication
Integrity setup procedure
Install *key*, enable integrity
Authentication Response (Response)
(Not ciphered, not integrity protected)
Verify Response, install *key*
NAS Security Mode Command (Ciphering and integrity algorithms, NAS-MAC)
(Not ciphered, NAS integrity protected)
Performing ciphering procedure
Ciphering setup procedure
Reset counts, enable ciphering
NAS Security Mode Complete (NAS-MAC)
(NAS ciphered and integrity protected)
Reset counts, enable ciphering
Ciphered and Integrity protected NAS signaling messages exchange

Fig. 3: Authentication procedure installs security key and enables integrity protection at the device and the network. The NAS Security Mode Command procedure activates ciphering at the device and the network sides after successful completion of the authentication procedure.

Now it is easy to see the vulnerability in which the attacker exploits the fact that the device resets the count values after installing the key. The attacker can block the transmission of NAS Security Mode Complete message and lets the network to re-initiate the SMC procedure; causing the device to reset the counts again. In this way, the signaling messages sent by device between subsequent SMC procedures use same keystream block for their encryption.

Vulnerability 1: Failure of SMC procedure does not renew the security key.

1.2 Network accepts certain NAS messages that fail the integrity check It is understandable that a number of NAS signaling messages can be exchanged between device and the network before the activation of NAS security. These signaling messages include Attach Request, Authentication Request/Response/Failure, Security Mode Reject, Identity Response, and few others. However, there are a number of other messages (that include TAU Request and Detach Request/Accept messages) that are "conditionally" accepted when they fail the integrity check.

LTE NAS specification (3GPP TS 24.301[20]) states:

> "*These messages are processed by the MME even when the MAC that fails the integrity check or cannot be verified, as in certain situations they can be sent by the UE protected with an EPS security context that is no longer available in the network.*"

However, LTE core network re-authenticates the device before finally accepting the message. As stated in LTE NAS specification (3GPP TS 24.301[20]):

> "*If a TRACKING AREA UPDATE REQUEST message fails the integrity check, the MME shall initiate a security mode control procedure to take a new mapped EPS security context into use.*"

Such a 3GPP standard approach is vulnerable in which the network accepts the spoofed message, failing the integrity check, after re-authenticating the device.

Vulnerability 2: The network re-authenticates the device instead of rejecting certain messages failing the integrity check.

2) Detail attack procedure: We describe step by step attack procedure as follow.

Pre-condition Before launching the attack, the attacker needs to know the TMSI of the victim subscriber for identification purpose. The attacker gets the TMSI through broadcast paging message addressing the victim device. He can easily generate the paging message for victim device by simply calling the victim. If the victim phone is in idle state, the core network sends a paging broadcast message that includes victim's temporary identity (TMSI). On receiving the paging message, the victim device switches to connected state and prepares to receive its call. Because the paging is a broadcast message within the tracking area, the attacker device also receives the paging message[7]. By repeating this procedure, the attacker can ensure that the TMSI maps to the victim device (subscriber's phone number).

Experiment results: A clever attacker would hang-up the call before the victim device starts ringing. To access the practicality of hanging-up the call so that the victim device does not start ringing, we run several experiments. We record the signaling messages and the time between call initialization and call ringing events. In our experiments, both caller and callee phones are time synchronized through which we accurately correlate the events between two phones. In total we collected 200 logs with 2 major US operators. We consider the cases when the victim device receives the call through CSFB, and VoLTE. The attacker always makes a CSFB call (by turning off VoLTE option at its phone).

After initiating the call, the attacker must wait for paging message to be delivered to victim device before hanging-up the call. We can see from Table II that it takes around 3.5 seconds and 4.6 seconds (on average) for paging message to be received at victim device for OPI and OPII, respectively. The attacker can terminate the call afterwards where he has the error margin of 3.3 seconds and 5.3 seconds (on average) to hang-up the call so that victim device does not ring for OPI and OPII, respectively. Table II also shows the results when

[4] For simplicity, in this paper we name all types of keys (i.e. integrity, ciphering) as key.

the victim device receives the call through VoLTE instead of CSFB.

TABLE II: Silently getting victim TMSI: The time margin the attacker has to hang-up the call by making sure that (1) paging message is broadcasted towards victim, and (2) victim device does not ring.

Victim receives call through CSFB								
Operator	**Call init to paging msg**				**Paging msg to call ringing event**			
	Min	**Max**	**Avg**	**STD**	**Min**	**Max**	**Avg**	**STD**
OPI	3.2s	6.1s	4.6s	0.5s	2.4s	4.4s	3.3s	0.4s
OPII	2.5s	4.8s	3.5s	0.6s	3.5s	6.6s	5.3s	0.9s

Victim receives call through VoLTE								
Operator	**Call init to paging msg**				**Paging msg to call ringing event**			
	Min	**Max**	**Avg**	**STD**	**Min**	**Max**	**Avg**	**STD**
OPI	2.3s	4.4s	3.3s	0.5s	0.7s	2.0s	1.3s	0.3s
OPII	2.2s	4.6s	3.3s	0.6s	1.6s	2.6s	2.2s	0.3s

There is a possibility that the call from the attacker does not trigger any paging message towards the victim device. This is the case when the victim device is in connected state. From Table II, we can see that the attacker can easily determine whether the victim device is in idle or connected state. He first waits from call init to paging message triggering time. If he does not sniff the broadcast paging message during this period then he assumes that the victim device is in connected state. The attacker then backs-off for tens of seconds (the device's default inactivity timer – time to transition from connected to idle state – is 10s) and retries the call.

We now discuss our attack procedure in 3 main steps as shown in Figure 4.

① Triggering key update through inter-system switch The attacker's goal is to install fresh key and reset count values at victim device. To achieve this, he dials a phone call towards victim to get CSFB call connection established with victim device and then hangs-up the call. The CSFB call forces victim device to perform inter-system switch (from LTE to 3G/2G). Once the attacker hangs-up the call, the victim device moves back to LTE (from 2G/3G) and performs random-access channel (RACH) procedure to synchronize with LTE base station. Through RACH procedure, the device receives a temporary radio identity (C-RNTI) mapped with its TMSI from the base station. The attacker sniffs RACH messages to associate victim subscriber's TMSI with its C-RNTI. After RACH procedure, the device setups its radio connection and sends unciphered TAU Request message as initial NAS message. The device also starts timer T3430 (default value of 15s) to retransmit the TAU Request message if timeout occurs. On receiving the TAU Request message, the network performs the Authentication procedure through which both victim device and the network authenticate each other and install the key.
Experiment results: We run more than 200 experiments to access the practicality of the attack. At first, we access how successfully an attacker can trigger inter-system switch by dialing a phone call. We find that there are two cases: (1) either victim device *or* the network does not supports VoLTE feature; or (2) both the victim device *and* its associated network support VoLTE. In case of (1) the victim device automatically switches to circuit switched network, i.e. 2G or 3G, to receive the call. However, in case of (2) the victim device does not performs automatic inter-system switch, and the attacker needs to enforce it. From our experiments, we find that if the VoLTE call is blocked at device for 5 seconds then the LTE modem chipset (Qualcomm LTE modem) aborts VoLTE call in favor of making the call through CSFB. This feature has also been reported in several other studies[21], [22]. Now, the attacker strategy is to temporarily block (through UL jamming) the signaling messages between victim device and its network. But the question arises (i) when to start jamming after dialing the call?; (ii) how long the attacker can delay in starting jamming because in practice it is hard to start jamming at a precise time?; and (iii) when the attacker should hang-up the call after stopping jamming so that the victim device does not ring? For (i), table III shows error margin with min, max and average values of 2.2s, 4.3s, 3.3s with standard deviation of 0.5s for attacker to start UL jamming. That is, the time he has from initiating the call to sniffing the paging message (voice call indication for victim device in idle state). Once the attacker has decided to start UL jamming, he has an error margin of 0.4s (on average) with standard deviation of 40ms to start jamming as shown in Table III. This is the time in which victim device establishes the VoLTE call connection with the network, answering question point (ii). The jamming lasts for 5s that induces victim device to perform CSFB procedure to establish voice call connection over 3G/2G network instead of LTE. The attacker hangs-up the call before the victim device rings (i.e. within 3.3 seconds – refer to Table II Paging to Call ringing time – after stopping the jamming) which addresses our question point (iii). On hanging-up the call, the device switches back to LTE and performs RACH procedure that facilitates attacker to map TMSI with C-RNTI. The attacker has on average 45ms (10ms of STD) to capture RACH Response and/or RRC Connection Request message to successfully establish mapping, as shown in Figure 5(a).

TABLE III: Forcing victim to establish CSFB call connection instead of VoLTE: The error margin in terms of time the attacker has to start UL jamming so that the victim device fails to establish the VoLTE call connection. As a result, the victim phone receives call through CSFB procedure.

Operator	**Call init to call indication**				**Paging to VoLTE connection**			
	Min	**Max**	**Avg**	**STD**	**Min**	**Max**	**Avg**	**STD**
OPI	2.2s	4.3s	3.3s	0.4s	0.3s	0.6s	0.4s	0.04s
OPII	2.2s	4.6s	3.3s	0.6s	0.4s	0.7s	0.5s	0.04s

② Administrating key reinstallation attack through one-time jamming After the authentication procedure, the core network activates the Security Mode procedure by sending integrity protected SMC message to the device and sets the message retry timer T3460 (default value of 6s). The attacker who is sniffing the radio traffic finds the SMC message matching the victim's C-RNTI and *starts* UL jamming. The attacker has the error margin of 2 messages in starting UL jamming (i.e. either after sniffing Authentication Response message, or Security Mode Command message). On receiving the SMC message from the network, the device verifies message integrity, resets counts (*vulnerability 1* in Section V-A1), and sends Security Mode Complete message to the network. Because this UL message from device is blocked over the air, the network does not receive this response message and its timer T3460 expires. The network re-sends SMC message to victim device by resetting the timer T3460. The victim subscriber resets its UL/DL transmission count values and sends the Security Mode Complete message which is blocked as well by the attacker. Similarly, the third response to network initiated Security Mode procedure is also blocked. Meanwhile, the TAU timer T3430 at victim device times out. At this point, the device has already enabled ciphering (as it has sent out Security Mode Complete messages thrice). The victim subscriber prepares new TAU Request message and applies ciphering and integrity protection. It sends out the TAU request message which the attacker sniffs and stores it at his end. We call this message TAU_1, that is TAU Request message 1 which is encrypted with `keystream block`[5]. Note that the attacker can recover the TAU message as he himself is jamming resilient. This is because he knows his off tone jamming signals and can cancel interference added to jam the signals[23], [24], [25], [18]. However, the TAU message is non-decodable at the network side due to unknown interference. When the Security Mode procedure fails for the fourth time, the attacker *stops* UL jamming. As a result, the Security Mode procedure succeeds on its fifth try where the network resets `counts` and enables ciphering its end. From this point onward, the network only accepts messages which are both integrity protected and ciphered.

Experiment results: In order to make the attack practical, the attacker has to ensure that he (i) identifies the victim over the radio before starting UL jamming, and (ii) starts UL jamming before Security Mode Command complete receives at the network. For (i), he has an error margin of 380ms on average (with STD of 20ms) to identify the victim device through PDSCH. This is the time between RRC Connection request and Security Mode Command messages, as shown in Figure 5(b). For (ii), the attacker has on average 48ms (with 5ms STD) to start UL jamming (after Authentication Response message but before Security Mode Complete message), as shown in Figure 5(c).

We also perform more than 200 lab experiments to access the success probability of starting jamming within the specific time interval (i.e. 48ms). For this, we first use Qualcomm real time packet sniffing tool QXDM[19] to calibrate the time between performing inter-system switch and starting UL jamming. We modify the HACRF One source code to make jamming effective, and achieve UL and DL frequency jamming within 1ms after its initialization. We face two challenges in jamming specific LTE signaling message(s). From our experiments, we find that when we jam signals for more than 6 seconds the device internally triggers radio link failure, and if we continue jamming then the device switches to 3G network. To address this challenge, we systematically switch on and off jamming in an interval of 2.5s such that desired signaling messages remain blocked when they are re-transmitted on their time-out. The other challenge we face was regarding jamming UL signaling messages. We find that the device increases its UL transmit power (as high as 25dBm whereas our HACKRF One max UL transmit power is 15dBm) that renders UL jamming through low cost SDR device ineffective. To address this challenge, we perform DL jamming instead and block the TAU Accept message reaching towards the device. As the TAU procedure does not succeed after all, the network responds to retransmitted TAU request messages (as well as the spoofed message to be discussed in the next step below) even if it has received TAU request message earlier. Hence, we can successfully execute our attack step in practice.

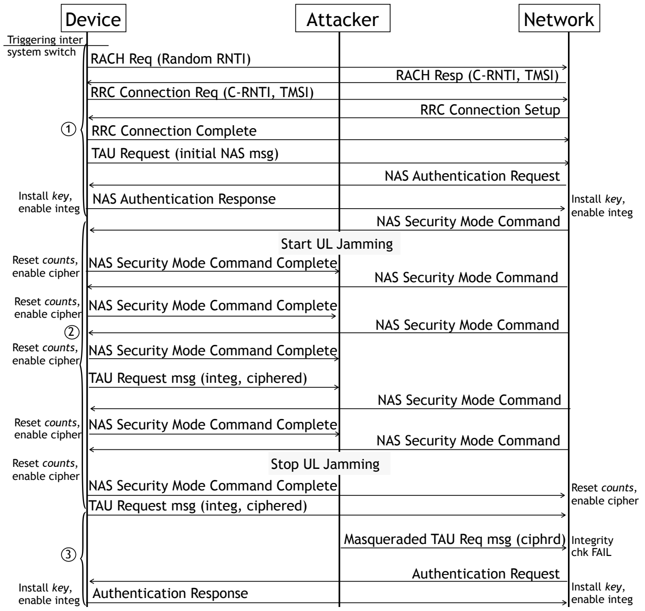

Device
Attacker
Network
Triggering inter system switch
RACH Req (Random RNTI)
RACH Resp (C-RNTI, TMSI)
RRC Connection Req (C-RNTI, TMSI)
RRC Connection Setup
①
RRC Connection Complete
TAU Request (initial NAS msg)
NAS Authentication Request
Install key, enable integ
NAS Authentication Response
Install key, enable integ
NAS Security Mode Command
Start UL Jamming
Reset counts, enable cipher
NAS Security Mode Command Complete
NAS Security Mode Command
Reset counts, enable cipher
NAS Security Mode Command Complete
NAS Security Mode Command
②
Reset counts, enable cipher
NAS Security Mode Command Complete
TAU Request msg (integ, ciphered)
NAS Security Mode Command
Reset counts, enable cipher
NAS Security Mode Command Complete
NAS Security Mode Command
Reset counts, enable cipher
Stop UL Jamming
NAS Security Mode Command Complete
Reset counts, enable cipher
TAU Request msg (integ, ciphered)
③
Masqueraded TAU Req msg (ciphrd)
Integrity chk FAIL
Authentication Request
Install key, enable integ
Authentication Response
Install key, enable integ

Fig. 4: Control plain attack main steps.

On practicality of jamming: We briefly discuss that our jamming works even if the attacker lacks LTE dedicated channel sniffing capability. We can always start jamming at desired signaling message with high probability. To evaluate this, we use three different methods, as discussed below:

Straw-man approach: The attacker makes a CSFB call towards victim, hangs-up the call as soon as the victim subscriber receives paging message, and starts jamming after waiting for 450ms (calculated according to Figure 5). We see that in this case the success probability of jamming is just 21%. This is mainly because the attacker hangs-up call while victim device was in the middle of call establishment procedure. This triggers location update procedure in 3G and the device does not release the connection towards LTE network.

Measured approach: To address the problem of straw-man approach, we let the control-plane call establishment procedure to be completed before hanging-up the call. The attacker lets the call establishment procedure to be completed before it hangs-up the call (just before call ringing). Hanging-up the call at this time triggers RRC connection release towards LTE network and the victim device immediately switches back to LTE network. The attacker starts the jamming after waiting for 450ms and gets the desired message blocked with the accuracy of 58%. The accuracy is halfed due to variable time of inter-system switch (i.e. how quickly LTE cell is selected).

Adaptive approach: Instead of calculating the jamming start time from call release event, we improve our results by sniffing the LTE broadcast RACH packet before making the jamming decision. Our results improve the jamming accuracy to 78% because in reality we cannot 100% predict when control-signaling message will arrive in future.

In summary, we show that the jamming at the desired occasion can be achieved with the accuracy of roughly 80% even if the attacker does not sniff LTE dedicated channel.

③ Spoofing location update messages through `keystream block` reuse Because the attacker has stopped jamming in step ②, the device initiated TAU request (on expiry of TAU timer T3430), we call it TAU_2, arrives at the network[6]. The attacker sniffs this TAU request message as well and retrieves the `keystream block` by xoring either

[5] Obviously, this message is also integrity protected, but we are interested to break the ciphering only to carry out our attack

[6] Careful reader should note that T3430 times out earlier than TAU Accept timer T3450 (default value 6s) at the network therefore network does not send TAU Accept message on receiving Security Mode Complete message

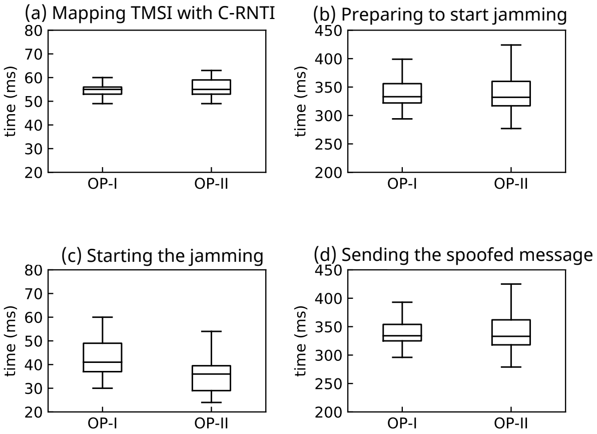


Fig. 5: Error margin (min, max, avg, and std) for different experiments. Time between (a) RACH Request to RRC Connection Request messages; (b) RRC Connection Request to Security Mode Command messages (c) Authentication Response to Security Mode Complete messages; (d) Security Mode Complete (5th try) to TAU Request (3rd try).

the contents of TAU_1 or TAU_2[7]. Recall that, he already gets hold of TAU request message (as initial NAS message) sent in plain text in step ①. Once he retrieves `keystream block` from the ciphered text, he encrypts his spoofed TAU request message that includes wrong device location identity by xoring the retrieved `keystream block`. He replaces his C-RNTI with victim's one[8] and immediately sends the spoofed message to the network. The network receives the spoofed TAU request message while it was waiting for TAU Complete message from device (as the network sends TAU Accept message after receiving TAU_2). According to LTE 3GPP standard, the network aborts previously received TAU message and processes the newly arrived message with different location identity (i.e. location information element). It has been stated in LTE NAS specification (3GPP TS 24.301[20]):

> "*If one or more of the information elements in the TRACKING AREA UPDATE REQUEST message differ from the ones received within the previous TRACKING AREA UPDATE REQUEST message, the previously initiated tracking area updating procedure shall be aborted if the TRACKING AREA UPDATE COMPLETE message has not been received.*"

The network decrypts the attacker originated TAU message and checks the integrity of the message. As the message contents were modified by the attacker, the TAU request fails the integrity check. The network finds that this is a special NAS message (4.4.4.3 Integrity checking of NAS signaling messages in the MME [20])) and it should be processed when the device fails the integrity check (*vulnerability 2* in Section V-A1). However, before accepting the message, the network successfully authenticates the victim device (by initiating the Authentication procedure), and sends TAU Accept message to the victim device. The victim devices replies with TAU Complete message to network that registers the spoofed device location identity in its database.

Experiment results: To make this step successful, the network must receive the attacker's spoofed message before the TAU Complete message arrives from the victim device. This is the time between receiving Security Mode Complete and TAU Complete messages (a device response to TAU Accept message of TAU_2). From Figure 5(d), we can see that the attacker has on average 370msec (15msec STD) to prepare and send its spoofed message to the network. For validating the impact of spoofed message, we modify the non-volatile memory of the LTE modem and used Qualcomm's service-programmer tool (QPST Service Programmer)[26], and AT-command tool (TeraTerm)[27] to send the spoofed message.

3) Attack damage: The consequence of our attacker is that the network updates the victim device location to erroneous tracking area. When the victim device enters in the idle state, it releases the RRC connection. The device relies upon the paging message from the network for the notification of its data packets during its idle state (e.g. if someone sends a text message, or voice call to victim). Because the attacker has registered the victim device on wrong location by hijacking TAU procedure, the victim device does not receive the paging message. Hence, the victim device remains unreachable for its incoming voice and data traffic.

Constraints: To realize the attack, the device must transition to idle state after performing the TAU procedure. The maximum time the victim device remains under attack is the time until it performs periodic TAU procedure (default value of 54 minutes). Note that, other LTE procedures such as Service Request procedure or VoLTE call establishment do not have any impact on our attack (i.e. they do not shorten the attack time).

Extending the attack period: The attacker can easily re-launch the attack to keep the victim device under attack even if the device updates its location through periodic TAU procedure, establishing a CSFB call, or even rebooting. After launching the attack for the first time, the attacker periodically pages the victim device by initiating a call towards him. If attacker's call generates the paging message, it means the victim subscriber has recovered from the attack. The attacker then re-launches the attack by following steps ① to ③, and keeps the victim subscriber under attack.

B. Designing LTE Service Outage Attack

We extend our location hijacking attack to bring more serious attack. In this variant of the attack, the attacker sends Detach Request message (with cause power off) instead of sending the spoofed TAU request message at step ② to the network. There are two scenarios that occur at the network. First, the network receives the device de-registration request in the middle of ongoing TAU procedure (i.e. the network is waiting for TAU Complete message from the device). Second, the detach request being sent by the attacker is bound to fail the integrity check at the network. The 3GPP standard explicitly discusses both these cases in LTE NAS standard[20]. The first case is defined as abnormal case for TAU procedure that requires the network to abort the TAU procedure and to process the Detach Request message from the device. It has been stated in [20]:

"*If the device receives a DETACH REQUEST message before the tracking area updating procedure has been completed,*

[7] Careful readers will argue that why the attacker needs to wait for second retransmitted TAU when he can create spoofed message at step ②. We do so to avoid the victim device transitioning back to *registered* state from *TAU init* state when the TAU timer T3430 expires while the spoofed TAU is being processed at the network. That can invalidate our attack in which the device initiated TAU rectifies the location identity

[8] C-RNTI spoofing is necessary so that LTE base station forwards the attacker's spoofed message towards victim's S1AP connection.

the tracking area updating procedure shall be aborted and the detach procedure shall be progressed."

While progressing the detach request message, the network finds the message has failed the integrity check. This is our second scenario and the 3GPP standard requires the Detach Request message (with cause power off) must be processed even of the message fails the integrity check (i.e. our *vulnerability 2* in Section V-A1). LTE NAS specification states [20]:

"*The procedure is completed when the network receives the DETACH REQUEST message. On reception of a DETACH REQUEST message indicating "switch off", the MME shall delete the current EPS security context.*"

We must point out that this special case only applies to Detach Request with reason power-off, otherwise, the network proceeds with the tracking area updating procedure first before progressing the detach procedure.

No LTE service When the network receives Detach Request message with cause power off, it re-authenticates the victim device first and then releases the device connection by deleting device sessions and freeing its IP address. The device (being unaware of its network registration has terminated) sends Service Request message (when it has some data to send or call to initiate). On receiving the Service Request message from the victim device, the core network rejects the request with error cause #43 (Invalid EPS bearer identity). On receiving the Service Reject message with error cause #43, the device enters into *deregistered* state, according to 3GPP NAS specification[20] that states:

"*The UE shall abort any ongoing ESM procedure related to the received EPS bearer identity, stop any related timer, and deactivate the corresponding EPS bearer context locally (without peer to peer signalling between the UE and the MME).*"

Now the victim needs to manually register the device with network (by rebooting device or by turning on/off the device airplane mode), otherwise LTE service remains unavailable.

VI. Attacking LTE Data Plane

Overview We demonstrate key reinstallation attacks on VoLTE and Cellular IoT (CIoT) in 5G that use LTE data-plane to deliver their voice and IoT data packets, respectively. The root cause of these attacks is the vulnerability found at LTE Packet Data Convergence Protocol (PDCP layer), responsible of encrypting data packets. The PDCP layer resets the `count` values without updating the `key` when it establishes a new data association (i.e. PDCP entity). We discover that whenever the LTE subscriber dials/receives a VoLTE call indication, it creates a VoLTE PDCP entity. The PDCP entity is destroyed when the VoLTE call is released. In this way, the attacker can easily establish a PDCP entity at victim device by simply calling victim. In attacking VoLTE, he uses chosen-voice packets to decrypt complete victim's private conversation (both UL/DL speech packets). The attacker first records his conversation (both encrypted and non-encrypted packets) with victim, and xors it with victim's encrypted private conversation (e.g. victim conversation with his friend) that he records later-on. This deciphers encrypted victim private voice packets. Because LTE data packets are not integrity protected, the attacker can even modify and can inject his own packets by using the retrieved `keystream block`. We extend our attack to 3GPP standardized IoT solution. We attack CIoT "PS data off" feature that enables LTE IoT applications to establish their network association as required. LTE based IoT compliant devices send activate/deactivate "PS data off" request and get their PDCP entity established/destroyed. The attacker uses the PDCP vulnerability of resetting `counts` and can decrypt/inject/modify IoT data packets.

Roadmap We first discuss key reinstallation attack on VoLTE. We provide the feasibly analysis from LTE standard followed by detail attack procedures in LTE operational network. Later, we discuss attack damage and the limitations. Lastly, we extend this attack and design an attack on CIoT.

A. VoLTE Data Packets Decoding Attack

1) Feasibly analysis from LTE standard: Following, we discuss PDCP vulnerability that we exploit in attacking LTE data-plane through VoLTE.

PDCP entity establishment resets `counts` without updating the `key` LTE PDCP layer manages a number of PDCP entities that carry user plane data belonging to different radio bearers (i.e. radio connection type). Through PDCP entity, the device and network can discern different types of traffic, and can meet quality of service requirements. When LTE registers with the core network, it creates a default PDCP entity bound to default LTE bearer. In case, the device initiates VoLTE call connection, it establishes a new PDCP entity for VoLTE call through Activate Dedicated Bearer procedure. The device uses VoLTE PDCP entity to send and receive VoLTE data packets, and destroys it when the call is terminated. LTE data packets are encrypted at PDCP layer that maintains `count` value for every PDCP entity in both UL and DL direction. The ciphering algorithm, as shown in 2(right), takes `count` value as an input in calculating `keystream block`. We find that every time a PDCP entity is established, the `counts` are reset to zero (refer to Section 7, 3GPP PDCP specification [28]). Resetting `count` values on establishment of PDCP entity is vulnerable that gives birth to key reinstallation attack. The attacker can dial periodic VoLTE calls towards the victim, and gets `count` values reset at victim device without triggering an update to data encryption `key`.
Vulnerability 3: PDCP establishment procedure resets `counts` without modifying the `key`.

2) Detailed attack procedure: We first discuss our system settings followed by step by step attack procedure.

Attack settings: In evaluating our attack, we consider victim device sends and receives its voice call using VoLTE (not CSFB). The attacker who knows the victim's phone number gets the victim TMSI by initiating paging towards victim device (procedure discussed in pre-condition of section V-A2). As the purpose of our attack is to discuss key reinstallation attacks at LTE data plane, we consider that the network operator has not enabled data encryption at IMS telephony application (i.e. end-to-end VoLTE encryption is disabled). From our study, we find that in reality there are a number of network operators that indeed has not enabled end-to-end VoLTE encryption. These include operators from South Korea[29] and Germany[30] to name a few. Note that

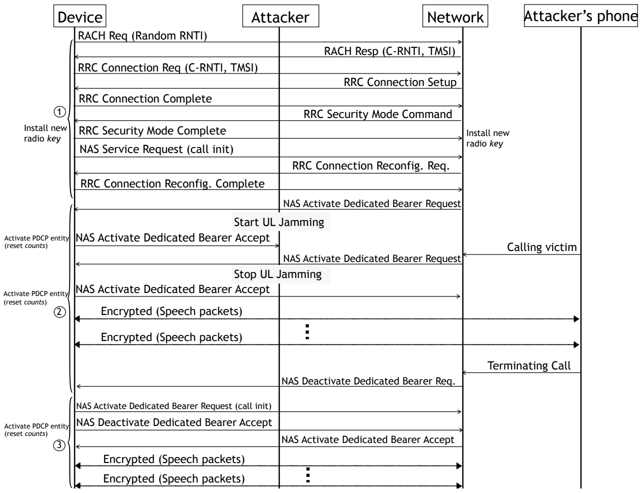

Device
Attacker
Network
Attacker's phone
RACH Req (Random RNTI)
RACH Resp (C-RNTI, TMSI)
RRC Connection Req (C-RNTI, TMSI)
RRC Connection Setup
RRC Connection Complete
RRC Security Mode Command
①
Install new radio *key*
RRC Security Mode Complete
Install new radio *key*
NAS Service Request (call init)
RRC Connection Reconfig. Req.
RRC Connection Reconfig. Complete
NAS Activate Dedicated Bearer Request
Start UL Jamming
Activate PDCP entity (reset *counts*)
NAS Activate Dedicated Bearer Accept
Calling victim
NAS Activate Dedicated Bearer Request
Stop UL Jamming
Activate PDCP entity (reset *counts*)
NAS Activate Dedicated Bearer Accept
②
Encrypted (Speech packets)
Encrypted (Speech packets)
Terminating Call
NAS Deactivate Dedicated Bearer Req.
NAS Activate Dedicated Bearer Request (call init)
Activate PDCP entity (reset *counts*)
NAS Deactivate Dedicated Bearer Accept
NAS Activate Dedicated Bearer Accept
③
Encrypted (Speech packets)
Encrypted (Speech packets)

Fig. 6: VoLTE call attack main steps.

for testing purpose, we can turn off VoLTE encryption by modifying configuration file at the device side[31].

Our attack procedure consists of 3 main steps, as shown in Figure 6.

① **Preparing to block victim originated VoLTE call** In this step the attacker aims to block victim originated VoLTE call so that he could establish his call with victim to get his chosen-voice packets encrypted. When the victim originates VoLTE call during device idle state, the victim device performs RACH procedure and receives the C-RNTI. The attacker maps the victim C-RNTI with its corresponding TMSI so that he can uniquely identify victim's signaling messages over the air, as discussed in Section V-A2 step ①. After RACH procedure, the device establishes RRC connection and enters into connected state. It then performs RRC Security Mode Command procedure and activates radio security. The RRC Security Mode procedure installs new radio security `key` at device with which data plane packets are encrypted. Readers should not confuse RRC Security Mode procedure with NAS Security Mode procedure as we have discussed during control-plane attack (Section V)). Because the victim has triggered VoLTE call in idle state, the device sends the NAS Service Request message with indication of VoLTE call. The network reconfigures RRC connection by updating the VoLTE packets quality of service class and VoLTE call data throughput values. This is the time (from RACH to RRC Connection Reconfiguration procedures) that the attacker can take in preparing to block victim originated VoLTE call.
Experiment results: From our experiments, as shown in Figure 7a, we find that the attacker has an average 130ms to prepare to block the victim subscriber VoLTE call progress.

② **Getting chosen voice packets encrypted with radio `key` after blocking victim originated VoLTE call** Once the network reconfigures RRC connection, it sends Activate Dedicated Bearer Request to victim device. The device activates PDCP entity for VoLTE call and sends Dedicated Bearer Complete message to the network. The attacker who is sniffing the radio channel on victim's C-RNTI blocks the VoLTE call setup by starting UL jamming so that the Dedicated Bearer Complete message does not reach the network. Network makes several tries (not sown in Figure 6) to activate dedicated bearer and aborts VoLTE call setup after successive

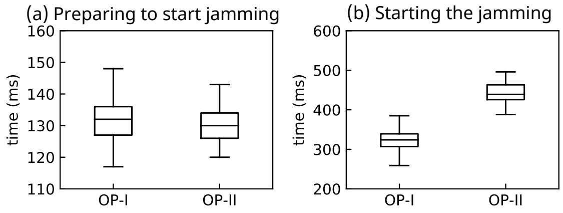


Fig. 7: Error margin (min, max, avg, and std) for different steps. (a) Preparing to start jamming for the device that has made a VoLTE call after initiating the Service Request (SR). That is the time between Service Request and RRC Connection Reconfiguration Request messages; (b) starting the UL jamming to block VoLTE call, i.e. the time between RRC Connection Reconfiguration Request to Activate Dedicated Bearer Request messages.

call setup failures[9]. Meanwhile, the attacker calls the victim device. When the attacker sniffs Activate Dedicated Bearer Request message with indication of mobile terminating call, he stops UL jamming and lets victim device to complete VoLTE call setup procedure for the attacker initiated voice call. The call setup procedures activates new PDCP entity for the VoLTE call and resets the `count` values. Once the call is completed, the attacker speaks with victim[10] for some time and records both encrypted (exchanged between victim subscriber and the network), and unencrypted voice packets (received by attacker's telephony application). Once enough packets have been captured, attacker hangs-up the VoLTE call. The call termination action triggers Deactivate Dedicated Bearer procedure. As a result, the LTE modem at victim device disables VoLTE PDCP entity.
Experiment results: First, we find out the error margin the attacker has to start UL jamming. Figure 7b shows that on average the attacker can take 320ms to start UL jamming after identifying the victim. That is, the time between RRC Reconfiguration and Activate Dedicated Bearer procedures. Through our adaptive jamming approach (as discussed in Section V-A2, step ②), the attacker can jam with around 80% success probability.

③ **Decrypting victim's private conversation** Recall that, the victim's original call was failed due to jamming and he has received the call from the attacker instead, therefore, it is highly probable that he will now re-initiate his failed call. However, in reality the victim may not initiate his failed call immediately and may delay his call for sometime. In this case, the network releases the RRC Connection on expiry of inactivity timer (default value of 10 seconds). This is undesirable situation for attacker because on releasing the RRC connection, the victim device enters into idle state and its RRC security association with the network is destroyed. The device later re-initiates the security association (hence gets a new `key`) with the network when it transitions from idle to connect state. Therefore, the attacker must not let the victim subscriber device enter into idle state and should keep it in connected state until the the victim sends/receives his private VoLTE call. To achieve this, the attacker periodically initiates the call towards the victim device but hangs-up the call before the attacker phone rings (i.e. call alerting message at attacker side). The periodicity of the call depends upon the inactivity

[9] Even if the victim device switches to CSFB to initiates its phone call, the call will not succeed due to UL jamming and is failed eventually.

[10] The victim believes he was called in error, i.e. wrong number call

Noise	Speech samples		
	best	average	worst
0%	100%	100%	100%
20%	90%	85%	80%
40%	80%	70%	60%
60%	70%	55%	40%

TABLE IV: Accuracy of speech packets decryption with presence of noise

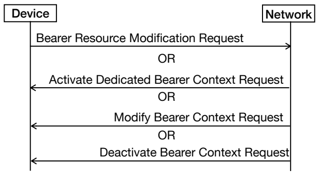

Device
Network
Bearer Resource Modification Request
OR
Activate Dedicated Bearer Context Request
OR
Modify Bearer Context Request
OR
Deactivate Bearer Context Request

Fig. 8: Activation/deactivation of data connection as required

timer[11] and should be less than the timer value. When the victim initiates the VoLTE call after a while, its device setups Dedicated Bearer for VoLTE call and establishes the PDCP entity by reseting `count` values. The attacker who is sniffing the radio channel captures the encrypted voice packets (at both UL and DL directions) exchanged between victim and his friend (i.e. callee). The attacker has now two sets of voice packets encrypted with same `keystream block` as well as non-encrypted voice data packets as he received when speaking with the victim at step ②. The attacker now successfully decrypts the victim's private voice data packets by xoring all recorded packets.

Experiment results: To evaluate the accuracy of decrypting voice packets, we consider 3GPP encryption algorithm, SNOW3G, and method of encryption/decryption. The implementation of SNOW3G is available as opensource, CryptoMobile[32]. We collected a number of voice samples regarded these between attacker and victim, and between victim and his friend. Considering all samples have *same* length, we validate the accuracy of our decryption when different level of noise is added. Table IV shows our results. We conclude that for high accuracy of VoLTE packets decryption, the position of voice and noise packets plays an important role. Even when the overall noise is recorded 60%, we can still decrypt 70% of the voice packets if the noise is spread among two voice samples (chosen-voice sample and its encrypted version) such that encrypted and non-encrypted noise packets align with each other; hence canceling each other out, and maximizes the decryption efficiency.

3) Attack damage: The attack damage in this attack is now obvious where the victim can decrypt the VoLTE call (both incoming and outgoing voice legs) made between victim and his friend. This is a serious privacy breach to LTE subscribers who use VoLTE for enhanced voice quality. Because LTE data packets are not integrity protected, the attacker can even inject his own packets, modify the call conversation, and even re-direct the call to a different destination.

Constraints: We note down two constraints we face. First, the device must originate and receives its calls through VoLTE, and should not trigger CSFB call after step ②. This is because CSFB call releases the LTE RRC connection and switches the device to 3G for the voice call. Even if the device comes back to LTE later but its RRC `key` gets updated. From our experiments we find out that when the network enables VoLTE control plane connection during device registration, all incoming and outgoing calls are made through VoLTE as long as the device is under acceptable radio condition (i.e. -115dBm or better). This LTE device feature addresses our concern. Second, from our experiments with OPI and OPII, we discover that every time a new PDCP entity is established (through Activate Dedicated Bearer procedure), the network assigns a different data radio bearer identity (drb-identity, which is an input parameter to ciphering algorithm, refer to Figure 2(b)) without changing the `key`. The drb-identity wraps around to starting value after 24 tries. In order to launch key reinstallation attack, we are required that both victim's private conversation and attacker's chosen voice packets are encrypted using same drb-identity. To achieve this, we launch 23 silent calls with the victim (i.e. hang-up the call before the victim device rings) before recording victim's private conversation.

B. Designing Cellular IoT Attacks in 5G

Background: 5G–especially for cellular IoT– is designed for low-power, wide-area networks (LP-WAN). To meet the design goals, two optimizations for CIoT are defined by 3GPP standards – the control-plane CIoT optimization and the data-plane CIoT optimization. The control-plane CIoT uses LTE signaling radio bearer to send and receive data packets; whereas the data-plane CIoT establishes CIoT data connection with LTE core network for sending and receiving its data packets. For data-plane CIoT, 3GPP has introduced the "PS data off" feature (since 3GPP release 14 in 2017) to enable IoT device service (in UL) and LTE network (in DL) to exchange PS data packets only when it is required. The device can also register some other IoT services and makes them "PS data off" exempt, meaning these services can always transmit and receive IP data packets. Take an example of audio video surveillance IoT device (e.g. all in one Canary surveillance IoT device) that provides 24/7 video surveillance but only records audio or sends alerts as some event occurs (such as road accident or house burglary[33]). In this example, PS data off feature is enabled for audio recording and its transmission, while video surveillance is exempted from this feature.

Key reinstallation attack on CIoT: We consider audio video surveillance CIoT application to design our attack. When audio video IoT device registers with LTE network, it registers video service to be PS data off exempt and audio service to be non-exempt. After registration, the network installs radio `key` to encrypt both audio and video data packets and reset `counts` for audio and video PDCP entities. The device starts sending its video packets to the network by encrypting them using `key` and video PDCP entity `count` value. When an event occurs for which the device is required to provide audio stream to the network, the IoT device sends Bearer Resource Modification Request with reason "deactivation request of PS data off" to the network, as shown in Figure 8. The network sends Activate Dedicated Bearer Context Request message to the device. The device establishes audio PDCP entity by reseting `count` values and responds with Activate Dedicated Bearer Context Response message. Thereafter, it uses existing `key` (as installed during device RRC connection) and its `count` to encrypt audio data packets to be transmitted to the network. In case the event reporting has concluded, the device sends Bearer Resource Modification Request by including the cause as "activation request of PS data off" to the network and gets its audio data bearer deactivated. The attacker who is sniffing LTE radio channel records all encrypted packets first and later decrypts them once his chosen-audio packets are encrypted on re-establishment of audio PDCP entity (as the event occurs). For this he needs to know the triggering condition of an event to re-establish the PDCP entity. The

[11] The attacker knows the network configured timer value as the attacker controlled phone is associated with the same operator as of victim device.

event triggering conditions can be found in IoT device manual (e.g. for Canary IoT device event is an anomalous activity, such as someone entering the house at mid-night, which is configurable according to application scenario).

Useful applications: In hindsight, one may question about the usefulness of this attack because the attacker can listen the audio anyway (considering a conversation scenario when the attacker is closer to victim). Nevertheless, there exists numerous applications in which the attacker and victim are not closed enough and the attacker is interested to spy by decrypting audio conversation. Take an example of wireless spy microphones which were installed at quite places of Canadian airports[34] to record the audio conversation between travelers. An other example would be event based automatic audio recording between traveler and homeland security official at the airport. Although, today audio and video systems are installed and managed separately, in future, IoT devices are expected to provide both video and audio surveillance at a single device[35]. By showing the feasibility of key reinstallation attacks over CIoT, we call for attention in developing future 5G security procedures.

VII. Proposed Remedies and Discussion

LTE base station, and core-network entities are 3GPP compliant and any vendor specific implementation, conflicting with the LTE standard, may fail inter-operability between devices and the network functions. Therefore, our proposed remedies should be 3GPP standard compliant. Further, we seek to mitigate discussed vulnerabilities without introducing any other security loopholes.
To achieve these goals, our solution is based on following three security design principles.

1) Make key installation and count reset procedures atomic (**P1**); i.e. either both succeed or none
2) Message failing the integrity is always re-executed (**P2**)
3) Derive fresh key on establishing new PDCP entity (**P3**)

A. Remedies

Bounding `key` installation and `count` reset procedures: One of the root causes of control-plane attacks is the disjoint execution of key installation and count reset procedures. To address this, we bound LTE NAS Authentication (that installs the `key`), and NAS Security Mode Command (that resets `count`) procedures. That is, we perform LTE Authentication procedure whenever Security Mode Command procedure fails (making security procedures atomic). In LTE Authentication procedure, the network sends Authentication Request message by starting timer T3460 (default value of 6s). The timer is stopped when the network receives Authentication Response message from the device. In our solution, we stop T3460 when the network receives Security Mode Complete message from the device instead of stopping at Authentication Response. It means, the Authentication procedure fails if the device Security Mode Command procedure fails; hence bounding these two procedures. Our approach addresses *vulnerability 1* based on principle *P1.*

Enforcing integrity protection for all signaling messages once security has been established: The other root cause of control-plane attacks is that certain messages (i.e. TAU and Detach Request) are partially accepted even if their integrity check fails. Although, the network authenticate the device afterwards but does not validate whether the received signaling message was indeed originated by the authenticated device or not. We mitigate this vulnerability by enabling the device to not accept any signaling messages failing the integrity check if the security association has already been established. Instead the network *rejects* the message and re-authenticates the device. We should point out that present 3GPP specifications generate integrity failure message response for selected signaling procedures. To provide the integrity check failure feedback for all types of signaling messages, we propose that the network should reject the signaling message with error cause # 9 (UE identity cannot be derived by the network). On receiving this error message, the device re-registers with the network after executing both authentication and security mode procedures. Our standard compliant solution may arguable delay LTE service for a couple of seconds, but it enforces LTE security at all times; hence mitigating *vulnerability 2* based on principle *P2.*

Installing fresh `key` on establishment of new PDCP entity: The main reason of data-plane attacks is that the `count` values are reset on establishment of PDCP entity. Although LTE network operators update the drb-identity value every time a new PDCP entity is installed, their solution does not mitigate the vulnerability because the drb-identity value wraps around without renewing the `key`. In our solution, we run RRC Security Mode procedure to update the radio `key` whenever the device establishes a new PDCP entity. This addresses *vulnerability 3* based on principle *P3.*

B. Security Analysis through Prototyping

We provide the security analysis of our proposal by developing a proof of concept prototype without creating interaction between victim and attacker. We use AT commands to take certain actions emulating the network enforcing above principles to mitigate vulnerabilities 1 to 3. Although there exists hundreds of AT commands, only few have privilege to execute over commercial handsets. We create our prototype by using those AT commands which our program can execute over commercial phones (such as Google Pixel or Samsung Glaxy devices). We design two sets of experiments.
In the first experiment, we check whether the subscriber device is under jamming attack or not. If the signals are jammed to launch key reinstallation attacks by resetting `counts`, we re-activate LTE bearers that re-establish the security by renewing `key`. When the device makes a voice call through CSFB, our program checks for LTE registration by running "*at+creg?*" and "*at+cgdcont?*" commands. "*at+creg?*" tells whether the device is PLMN registered and if true then whether it is registered with LTE network or not. The "*at+cgdcont?*" outputs the IP and APN name that explains with which cellular radio access technology the device has camped-on. For example, fast.tmobile.com tells the device is registered with LTE APN over T-mobile carrier network. Thereafter, our program sends "*at+cgdata="PPP", 1*" command to establish the data connection with the network. If the data request is not entertained, the device AT command returns error. It means the device data connection request has failed due to jamming. On receiving the error message, our program waits for 2 seconds before running "*at+cgdata*" command again. If the error persists then our approach is to renew the `key` by

re-activating LTE service. We run "*at+cops=2*" immediately followed by "*at+cops=0,1*" to force the device to reselect LTE network and perform re-authentication procedure.

In our second set of experiment, our goal is to renew the `key` whenever the PDCP entity is established. We do so by first checking whether a new PDCP entity has been activated or not, and if the outcome is true then we re-activate LTE service and manually add LTE bearer information (that was deleted due to LTE service re-activation). This is done by first executing "*at+cgdcont?*" that outputs list of currently defined PDP contexts. As soon as the new bearer is added (i.e. PDCP entity is established), our program re-activates LTE service (through at+cops commands) and then adds the bearers by running "*at+cgact=1,"IPv6","apn name","0.0.0.0",0,0*". This command gets Packet Data Protocol (PDP) context activated for the bearer which was deleted earlier in re-activating LTE security. In short, our approach re-activates the security as needed and does not introduce any other security vulnerability.

C. Discussion

We now briefly discuss that by using our solution design principles, 3GPP can permanently fix LTE key reinstallation vulnerabilities stemming from their standards. To make security procedure (i.e. Authentication and Security Mode Command procedures) atomic, we propose that the network should send the counter values to device within integrity protected SMC Request message, instead of letting the device to provide initial `count` values to the network. The network always monotonically increases the `counts` for all follow-up SMC request messages that it sends to device. It ensures that `counts` always increase for every retransmitted message, hence addresses key reinstallation *vulnerability 1*. To address *vulnerability 2*, we propose to introduce a new error cause representing the message failure due to integrity check. When the network finds the message integrity check has failed, it rejects the message by sending a response to device that includes message drop cause (i.e. integrity check failure). The device should re-send the control-plane message to the network by integrity protecting the message with unused `count` value. To thwart LTE data-plane key reinstallation attacks, our proposal is to run RRC Security Mode Command procedure to renew data radio `key` before establishing the PDCP entity. It means that for every time PDCP `counts` are reset, the new `key` is used to encrypt the data packets (addressing *vulnerability 3*). In short, by introducing minor changes in LTE NAS and RRC standards, LTE key reinstallation attacks can permanently be addressed.

VIII. Related Work

Key reinstallation attacks: Closest to our work is key reinstallation attack in WiFi[3], [1], [36]. Mathy and et al. [3] has recently shown a variant of key reinstallation attack in WiFi. Their work exposes design and implementation issues in WiFi security protocols that reinstall an already-in-use key. [1] discusses passive and active attacks due to keystream reuses in WEP. [36] shows key recovery attack on WEP. In contrast, our work although in the similar direction is different than all above works. We show key reinstallation attacks in LTE, even though LTE never reuses the same key (all keys are chained in forward direction), employs separate keys and counters for encryption and integrity protection.

Other works related to key reinstallation attacks are count reset due to power failure[4]; use of static counter due to implementation bugs[12]; faulty state machine transitions leading to count resets[37], [38]; count resets through routing protocols[11]; and side channel attacks on CBC mode with a block cipher[39]. Contrary to these works our work study LTE design flaws that resets counter values welcoming key-reinstallation attacks. Our attacks are neither implementation bugs nor brought due to careless design choices. We show that seemingly working security protocols have security loopholes when certain signaling messages are re-transmitted.

LTE security: A number of other works discuss LTE security issues. [7], [40], [13] conduct LTE protocol vulnerability analysis and show real impacts on LTE subscribers. LTEInspector[7] tool cannot detect key reinstallation vulnerabilities mainly because re-transmission of lost message is a valid behavior of every protocol and it is hard to automate impact of these lost messages over LTE security. LTEInspector[7] findings on detach request with IMSI are different than what we disclosed in this paper. [40] discusses LTE inter-protocol vulnerabilities in which the adversary can spoof LTE messages. Their attacks were not practical because the network rejects the attacker originated message (as C-RNTI and TMSI does not match). [13] shows the man in the middle attacks to exploit LTE layer two vulnerabilities. Attacks discussed in [13] are mostly passive in nature that map device activities and perform website fingerprinting. In contrast, in this paper, we present novel attacks in LTE and make them practical through key stream reuse.

Other works such as [41] conducts experimental validation to prove that LTE temporary identity can disclose subscriber location. [42] discusses privacy attacks in which signalling information is leveraged to infer user privacy information. [43] shows that current cellular infrastructures exhibit security loopholes (off-path TCP hijacking) due to their NAT/firewall settings. [44], [45] study insecurity in mobile data charging. [31], [46] discuss how a subscriber can inject control-plane traffic into user-plane and can get free data service. Different to all above works, we do not discuss security vulnerabilities due to misconfigurations or operator side bugs. Rather, we present first work that discusses security issues in LTE due to key reinstallation vulnerability.

IX. Conclusion

This is the first work to best of our knowledge that shows key reinstallations attacks in LTE. The re-transmission of certain signaling messages resets the counter values multiple times that lead to reuse of key stream block for ciphering of plain text messages. In consequence, the attacker can launch attacks at both LTE control and data planes. In control-plane, the attacker can hijack LTE location update procedure, and can de-register the victim device from the network. Through data-plane attacks, he can decrypt voice over LTE calls, and Cellular IoT IP data packets.

Update: We seek to permanently fix these vulnerabilities for not being carried forward to 5G standard documents. Currently, we are teaming-up with a device vendor and planning to submit *LTE change request document* to 3GPP standard body. We believe our recommendations will be included in the future LTE release version.

References

[1] Borisov, Nikita and Goldberg, Ian and Wagner, David. Intercepting mobile communications: the insecurity of 802.11. In *ACM Mobicom*, 2001.

[2] Karlof, Chris and Sastry, Naveen and Wagner, David. TinySec: a link layer security architecture for wireless sensor networks. In *ACM SenSys*, 2004.

[3] Vanhoef, Mathy and Piessens, Frank. Key reinstallation attacks: Forcing nonce reuse in WPA2. In *ACM CCS*, 2017.

[4] Zenner, Erik. Nonce generators and the nonce reset problem. In *International Conference on Information Security*, pages 411–426. Springer, 2009.

[5] Shannon, CE. (1948), "A Mathematical Theory of Communication", Bell System Technical Journal, vol. 27, pp. 379-423 & 623-656, July & October. 1948.

[6] Kahn, David. The codebreakers. *New York, NY: Scribner*, 1996.

[7] Hussain, Syed Rafiul and et al. LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE. In *NDSS*, 2018.

[8] 3GPP. TS33.401: 3GPP SAE; Security architecture, Sep. 2016.

[9] Vernam, Gilbert Sandford. Secret signaling system (U, 1919). *U.S. Patent*, 131071.

[10] Mason, Joshua and et al. A natural language approach to automated cryptanalysis of two-time pads. In *ACM CCS*, 2006.

[11] p. y. o. Aumasson, Jean-Philippe et al., booktitle=International Conference on Information and Communications Security. A note on a privacy-preserving distance-bounding protocol.

[12] Böck, Hanno and et al. Nonce-Disrespecting Adversaries: Practical Forgery Attacks on GCM in TLS. *IACR Cryptology ePrint Archive*, 2016:475, 2016.

[13] Rupprecht, David and et al. Breaking LTE on Layer Two. In *IEEE S&P*, 2018.

[14] IntelliJudge: WaveJudge LTE packets sniffer. http://www.sanjole.com/our-products/intellijudge-lte/.

[15] ThinkRF: Real-Time Spectrum Analyzer. https://www.thinkrf.com/real-time-spectrum-analyzers/.

[16] Lichtman, Marc and et al. LTE/LTE-a jamming, spoofing, and sniffing: threat assessment and mitigation. *IEEE Communications Magazine*, 54(4):54–61, 2016.

[17] Naseef, M. Vulnerabilities of LTE and LTE-Advanced Communication White Paper. 2014.

[18] Lichtman, Marc and et al. Detection and mitigation of uplink control channel jamming in LTE. In *IEEE Milcom*, 2014.

[19] QXDM– LTE packets capturing tool. https://www.qualcomm.com/media/documents/files/qxdm-professional-qualcomm-extensible-diagnostic-monitor.pdf.

[20] 3GPP. TS24.301: Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3, Jun. 2016.

[21] Tu, Guan-Hua and et al. How voice calls affect data in operational LTE networks. In *ACM CCS*, 2013.

[22] Tu, Guan-Hua and et al. Detecting problematic control-plane protocol interactions in mobile networks. *IEEE/ACM Transactions on Networking*, 24(2):1209–1222, 2016.

[23] MM, Galib Asadullah. *Robust wireless communications under co-channel interference and jamming*. PhD thesis, Georgia Institute of Technology, PhD thesis, 2008.

[24] Yan, Qiben and et al. MIMO-based jamming resilient communication in wireless networks. In *IEEE Infocom* , 2014.

[25] Zeng, Huacheng and et al. Enabling jamming-resistant communications in wireless MIMO networks. In *IEEE CNS*, 2017.

[26] QPST— Qualcomm service programmer tool. https://github.com/botletics/SIM7000-LTE-Shield/tree/master/SIM7000%20Documentation/Firmware%20Updater%20Tool/QPST%20Tool.

[27] Tera-Term-A Terminal Emulator. http://ttssh2.sourceforge.jp/index.html.en.

[28] 3GPP. TS36.323: EUTRA Packet Data Convergence Protocol (PDCP) specification, Sep. 2016.

[29] Kim Dongkwan, Dissecting VoLTE: Exploiting free data channels and security problems in Master Thesis KAIST, 2016.

[30] How secure are your VoLTE and VoWiFi calls? in Technical Report from ERNW GmbH, 2107.

[31] Kim, Hongil and et al. Breaking and fixing volte: Exploiting hidden data channels and mis-implementations. In *ACM CCS*, 2015.

[32] Cryptography for mobile network – C implementation and Python bindings.

[33] Foggia, Pasquale and et al. Audio surveillance of roads: A system for detecting anomalous sounds. *IEEE transactions on intelligent transportation systems*, 17(1):279–288, 2016.

[34] Border agency to eavesdrop on travellers' conversations.

[35] The Internet Of Things, IT and audio-visual integration.

[36] Stubblefield, Adam and et al. A key recovery attack on the 802.11 b wired equivalent privacy protocol (WEP). *ACM transactions on information and system security (TISSEC)*, 7(2):319–332, 2004.

[37] Beurdouche, Benjamin and et al. A messy state of the union: Taming the composite state machines of TLS. In *IEEE Security and Privacy, year=2015*.

[38] De Ruiter, Joeri and Poll, Erik. Protocol State Fuzzing of TLS Implementations. In *USENIX Security Symposium*, 2015.

[39] Vaudenay, Serge. Security Flaws Induced by CBC Padding—Applications to SSL, IPSEC, WTLS... In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 534–545. Springer, 2002.

[40] Raza, Muhammad Taqi and et al. Exposing LTE Security Weaknesses at Protocol Inter-layer, and Inter-radio Interactions. In *SecureComm*, 2017.

[41] Hong, Byeongdo and et al. GUTI Reallocation Demystified: Cellular Location Tracking with Changing Temporary Identifier. 2018.

[42] Shaik, Altaf and et al. Practical attacks against privacy and availability in 4G/LTE mobile communication systems. 2015.

[43] Qian, Zhiyun and Mao, Z Morley. Off-path TCP sequence number inference attack-how firewall middleboxes reduce security. In *IEEE S&P*.

[44] Peng, Chunyi and et al. Mobile data charging: new attacks and countermeasures. In *ACM CCS*, 2012.

[45] Peng, Chunyi and et al. Real threats to your data bills: Security loopholes and defenses in mobile data charging. In *ACM CCS*, 2014.

[46] Li, Chi-Yu and et al. Insecurity of voice solution volte in LTE mobile networks. In *ACM CCS*, 2015.