Call Me Maybe: Eavesdropping Encrypted LTE Calls With ReVoLTE

David Rupprecht, Katharina Kohls, and Thorsten Holz, *Ruhr University Bochum;*
Christina Pöpper, *NYU Abu Dhabi*

https://www.usenix.org/conference/usenixsecurity20/presentation/rupprecht

This paper is included in the Proceedings of the
29th USENIX Security Symposium.

August 12–14, 2020

978-1-939133-17-5

Call Me Maybe:
Eavesdropping Encrypted LTE Calls With ReVoLTE

David Rupprecht
Ruhr University Bochum
david.rupprecht@rub.de

Katharina Kohls
Ruhr University Bochum
katharina.kohls@rub.de

Thorsten Holz
Ruhr University Bochum
thorsten.holz@rub.de

Christina Pöpper
NYU Abu Dhabi
christina.poepper@nyu.edu

Abstract

Voice over LTE (VoLTE) is a packet-based telephony service seamlessly integrated into the Long Term Evolution (LTE) standard and deployed by most telecommunication providers in practice. Due to this widespread use, successful attacks against VoLTE can affect a large number of users worldwide. In this work, we introduce ReVoLTE, an attack that exploits an LTE implementation flaw to recover the contents of an encrypted VoLTE call, hence enabling an adversary to eavesdrop on phone calls. ReVoLTE makes use of a predictable keystream reuse on the radio layer that allows an adversary to decrypt a recorded call with minimal resources. Through a series of preliminary as well as real-world experiments, we successfully demonstrate the feasibility of ReVoLTE and analyze various factors that critically influence our attack in commercial networks. For mitigating the ReVoLTE attack, we propose and discuss short- and long-term countermeasures deployable by providers and equipment vendors.

1 Introduction

Millions of people worldwide use the latest widely deployed mobile communication standard LTE daily. Besides high-speed Internet access, LTE also provides the packet-based telephony service VoLTE. VoLTE promises low call-setup times and high-definition voice quality while being seamlessly integrated into the standard call procedure. With more than 120 providers worldwide and over 1200 different device types supporting VoLTE [23], it is an essential part of our communication infrastructure. At the same time, the use of VoLTE is fully transparent to the user and improves the call quality without requiring any further interaction. Consequently, any practical vulnerability in the VoLTE standard has far-reaching consequences for users all over the world, without them even realizing that they may be affected.

LTE not only improves the performance of prior mobile network generations, but it also defines a series of fundamental security aims to protect further the sensitive information of phone calls, web browsing, etc. One crucial aspect of these security aims is providing data confidentiality [8] for all voice calls, which protects LTE communication from eavesdropping. This is achieved by implementing publicly reviewed encryption algorithms like AES that protect the radio-layer transmission. In addition, VoLTE can establish an *additional* layer of security that further protects all signaling messages (IPsec tunnel) and voice data (SRTP). We will later see how these additional security features must be considered in the design of our attack. Breaking these protection mechanisms and thus the data confidentiality of LTE, allows us to recover the information of an arbitrary phone call. In a setting where the underlying mobile network generation promises strong security aims, this might reveal highly sensitive information that was assumed to be protected.

While prior work demonstrates that the aims of location and identity privacy [13,43] and an attacker can break the integrity of user data [38], a technical report by Raza and Lu [36] recently indicated that the data confidentiality of LTE might contain a fundamental flaw. By jamming particular messages and reinstalling a key, the authors introduce a concept that theoretically allows eavesdropping on a VoLTE connection. Although their work presents the foundation for breaking the essential security aim—data confidentiality—of the LTE communication standard, their work only covers a theoretical evaluation of the attack vector. It lacks any evidence that the concept is actually feasible in a real-world setup and at a sufficiently large scale.

In this work, we build upon the concept of key reinstallation and break the data confidentiality aim of LTE in a commercial network setup. This attack vector is the starting point for ReVoLTE: An attack concept that uses a *passive* downlink sniffer instead of active jamming, and provides insights on numerous adjustments to the technical requirements and challenges of a real-world implementation of the attack. ReVoLTE is a *layer-two* attack that allows us to Reuse Encrypted VoLTE traffic to eavesdrop on an encrypted voice call. Keystream reuse can occur when two calls are made within one radio connection.

Consequently, an attacker can decrypt the first call when she instantly calls the victim after the first call ended. Even though the specification states that the network is in charge of preventing such key reuse, we find multiple networks reusing the same keystream for subsequent calls. In addition to proving the general feasibility in commercial networks, we further provide an extensive experimental evaluation of all technical and operational requirements that allows us to understand the attack vector better.

With millions of users potentially being at risk, we argue that it is crucial to analyze LTE key reuse attacks beyond their theoretical concept. By developing a better understanding of the open attack vectors in our current mobile network generations, we can avoid the same issues in the specification and implementation of upcoming standards. With that said, *we can find the same attack vector in the upcoming 5G networks.* Therefore, we additionally take a defensive perspective to analyze and discuss short- and long-term countermeasure concepts that protect from or circumvent the threat of REVOLTE. In summary, our contributions are as follows:

- **Attack with Real-World Impact.** We analyze keystream reuse under real-world considerations and present a practical attack called REVOLTE. REVOLTE completely breaks the confidentiality aim of LTE and allows an attacker to eavesdrop phone calls.
- **Preliminary and Real-World Experiments.** We conduct several preliminary experiments to evaluate the various conditions that influence REVOLTE. In particular, we conduct real-world experiments with three operators on keystream reuse and find two of them vulnerable. Further, we assess the use of so-called comfort noise, transcoding, and robust header compression.
- **Discussion of Countermeasures.** Our experimental evaluation of REVOLTE provides clear evidence that the confidentiality aim of LTE is at risk. We thoroughly discuss potential mitigations that can be deployed by the providers and elaborate on how users can protect themselves.

Disclosure Process. The keystream reuse vulnerability exploited by REVOLTE is an implementation flaw and affects a large number of deployments. Following the guidelines of responsible disclosure, we have reported the vulnerability via the GSMA CVD program (CVD-2019-0030) and actively work together to fix the problem.

2 Preliminaries

In this section, we introduce the basics of LTE networks with a focus on security establishment and encryption features. Furthermore, we take a closer look at the technical background of the VoLTE standard.

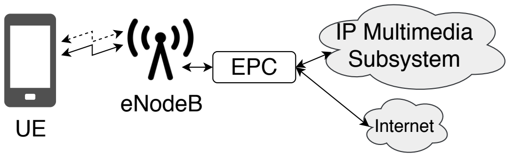

Figure 1: LTE network with IMS

2.1 LTE and IMS Network

When establishing a VoLTE connection with an LTE network, a series of different components assures the communication between a user's device and the core network components (cf. Figure 1). In the following, we introduce all entities that are relevant in the context of the proposed attack.

User Equipment (UE). The UE is the user's endpoint of the connection (e.g., a smartphone) and provides the technical functionality for accessing the LTE network. It implements the VoLTE stack that allows to access VoLTE services at the IP Multimedia Subsystem (IMS). On the second layer of the network stack, the *radio layer*, the UE connects to one of the base stations in the current radio cell. On the third layer, the UE further executes the authentication and key agreement procedure with the Evolved Packet Core (EPC) and IMS. In our attack, we eavesdrop the VoLTE call for the victim's UE.

Evolved NodeB (eNodeB). eNodeBs are the base stations in an LTE network and are responsible for controlled resource allocation for all UEs in their cell. Furthermore, an eNodeB applies encryption to user and control plane data and can use additional compression for user plane packets. In this work, we locate a sniffer in the range of the eNodeB and thus can receive all frames.

EPC. The EPC is the LTE core network and responsible for the authentication and key agreement, and mobility management. The EPC also forwards user plane traffic to the correct packet data network, e. g., the Internet in case of web browsing. In the case of a VoLTE call, the packet data network is the IP Multimedia Subsystem (IMS).

IMS. The IP Multimedia Subsystem (IMS) is the IP-based telephone service for LTE and consists of different subcomponents. One of the critical functions is the Proxy Call Session Control Function (P-CSCF) that manages the incoming and outgoing VoLTE calls.

2.2 VoLTE

The VoLTE specification allows using the packet-based LTE network and IP protocols to establish voice and media calls. To this end, VoLTE uses modified Internet domain protocols: the Session Initiation Protocol (SIP) to signal the call flow, the Real-Time Transport Protocol (RTP) to transport the actual voice data, and the RTP Control Protocol (RTCP) to control the RTP connection. ReVoLTE enables an attacker to decrypt the encrypted payload of the RTP packets. In a VoLTE setting, these protocol messages are treated as user data with special transmission requirements. Two important characteristics, the *multimedia codecs* and *robust header compression*, influence the way data is transmitted in a VoLTE call. Furthermore, the concept of data bearers allows matching the specific transmission requirements of VoLTE calls.

2.2.1 Codecs and Comfort Noise

Multimedia codecs help to transform signals between different representations and are a core component for mobile communication. The technical characteristics of a codec depend on its main goal and can either optimize the data consumption or the perceived call quality (maximizing both would be optimal but unrealistic). Once translated into the target representation, VoLTE uses RTP to transmit data in packets. There are three possible codec options for a VoLTE call: Enhanced Voice Services (EVS), Adaptive Multi-Rate (AMR), and Adaptive Multi-Rate Wideband (AMR-WB).

All three codecs are optimized to save bitrate in periods where one calling partner is silent. In these periods, *comfort noise* is generated based on a transmitted seed sent by the silent calling partner. Comfort noise saves bitrate as the seed is smaller and transmitted on a lower frequency. For example, the AMR-WB codec encodes the seed of the comfort noise with 40 bit every 160 ms. Actual voice is encoded with 477 bit every 20 ms in the high-quality mode (23.85 kbit/s) [5].

Transcoding converts the voice data with a particular codec sent by one calling partner into another codec that is sent to the other calling partner. Although this results in the same audio content (i. e., what the calling partner *hears*), it destroys the bit pattern of the encoded voice data. Transcoding can happen when the call is routed via an IP exchange (IPX) or when radio-layer problems enforce a downsampling.

2.2.2 Robust Header Compression

Robust Header Compression (ROHC) is a technique to save transmission bits in the headers of IP, TCP, UDP, and RTP packets, and is primarily used in the context of wireless transmissions with high bit-error rates. The compression saves bandwidth by removing redundancies from similarities in packet headers of the same connection endpoints. Furthermore, compression becomes possible through the possibility of predicting parts of the information across protocols.

The eNodeB can activate ROHC for radio transmissions with different profiles that define the compressed data of the IP packet. In the context of VoLTE, two profiles are commonly used: Profile 1 compresses RTP, UDP, and IP headers and only transmits the payload of the RTP data with a ROHC small header. Profile 2 compresses UDP and IP headers and only carries the UDP payload again with a small ROHC header.

Table 1: Exemplary assignment of radio data bearers to their purpose, and radio bearer IDs.

Bearer	Purpose	Bearer ID
DRB1	Internet	1
DRB2	SIP (IMS)	2
DRB3...32	RTP (temporary)	3..32

The ReVoLTE attack extracts a keystream from the sniffed radio packet and sent plaintext. The ROHC influences the transmitted radio packets and is thus vital to consider a possible compression for the keystream computation.

2.2.3 Radio Connection and Radio Data Bearers

An active radio connection transports data over the air between the UE and the eNodeB. After reaching the threshold of an inactivity timer, the eNodeB switches an active connection into the idle mode to save resources. When reactivating the radio connection, both parties derive a new key which is used for encrypting the data. For the ReVoLTE attack, the two subsequent calls must take place within one radio connection, as only then the same encryption key is reused.

Part of the active radio connection are multiple radio bearers, which represent a logical link between the UE and the eNodeB and match certain transmission requirements. In case of a VoLTE-capable UE, three radio data bearers are required to provide Internet access and additional functionality for VoLTE voice calls. Table 1 provides an exemplary overview of the bearers used for a radio connection. The default bearer (`DRB1`) transmits the Internet data. A second data bearer (`DRB2`) is used for the SIP signaling traffic sent to the IMS. In case of a phone call, a third (dedicated) data bearer transports the voice traffic. This bearer is only established for the phone call and is immediately removed after the call. The eNodeB selects the used bearer ID and, thus, depends on the implementation. ReVoLTE targets the dedicated voice bearer and exploits the fact that the same bearer ID (DRB3) is reused for a second call within the same radio connection.

2.3 LTE Security

The LTE security aims include mutual authentication and data confidentially. A provably secure Authentication and Key Agreement (AKA) achieves the first aim on layer three (Non-Access Stratum (NAS)) between the EPC and UE. For this work, we focus on the radio-layer encryption, as it is crucial to understand the attack vector of ReVoLTE.

2.3.1 Radio Layer Encryption

Radio-layer encryption protects all user and control plane data transmitted on the connection between the UE and the

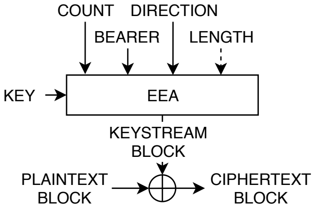

Figure 2: Encryption in LTE

eNodeB (cf. Figure 2). To this end, the Packet Data Convergence Protocol (PDCP) applies the encryption algorithm (EEA) that the Radio Resource Control (RRC) security mode command selects and activates. Besides Advanced Encryption Standard (AES) in counter mode (EEA2), Snow3G (EEA1) and ZUC (EEA3) are alternative ciphers. To encrypt a packet, its plaintext gets XOR-ed with a keystream block that the encryption algorithm generates *for each* packet individually, which results in the ciphertext representation. The following, input parameters document the standard setup for encryption algorithms on the radio layer:

- Key (128-bit): LTE introduces a key hierarchy and uses separate keys for different domains. The root key (k_{asme}) for all keys is the key derived by the AKA. As VoLTE data is user data, the key is the user plane key (k_{up}), which is established for each new radio connection.
- Count (32-bit): For user data, the count consists of the PDCP sequence number + PDCP hyperframe number[1]. The length of PDCP sequence number is individually configured for a bearer during the setup. The following PDCP sequence number length are possible: 5, 7, 12, 15, and 18 bit.
- Bearer (5-bit): The bearer identity depends on the used bearer. Table 1 gives an overview of the possible input parameters.
- Direction (1-bit): The direction bit defines if the data is either sent uplink or downlink.
- Length: The length defines the length of the keystream block. However, this input parameter does not influence the keystream generation itself.

Count, *bearer*, and *direction* represent the initialization vector of the underlying encryption algorithm and lead to a deterministic keystream, i. e., reusing the same information results in the same keystream. According to the *specification*, the eNodeB should avoid the keystream reuse [10][5.3.1.2]. However, the REVOLTE attack exploits an incorrect *implementation*, in which affected eNodeBs reset the count and reuse the bearer identity for a second call, which eventually leads to reusing the same keystream.

2.4 VoLTE Security

Besides the LTE security measures, VoLTE itself implements further security measures on layers three and four. While the encryption of user plane data is optional but recommended on layer two, the additional VoLTE security measures on higher layers of the protocol stack are *optional* and depend on the network configuration of a specific country. In particular, we discuss an additional AKA with the IMS, the IPsec protection of SIP messages, and the protection of RTP traffic.

2.4.1 Additional AKA

When the UE connects to the IMS via the SIP register procedure, both parties perform an additional AKA. Again, this AKA establishes mutual authentication and a key based on the shared key on the SIM card. The established key can protect SIP messages with an IPsec tunnel that can be operated in two modes: Authentication Header (AH) ensures the authentication and integrity of the IP payload. The Encapsulating Security Payload (ESP) additionally encrypts the IP payload.

2.4.2 Secure Real-Time Transport Protocol (SRTP)

While the security measures of higher layers of the protocol stack only secure the signaling messages of SIP, the RTP data (media plane) can be secured via the encryption and integrity protection of the SRTP protocol. According to the media plane protection specification [7], two scenarios are possible: 1) either the data is protected between the UE and the IMS, which is called end-to-access edge protection, or 2) both UEs protect their data with an end-to-end solution. Enabling media protection is optional and must be supported by the IMS and UE.

As REVOLTE focuses on decrypting the media plane, additional encryption beyond the second layer can hinder the success of the attack. Therefore, we analyze the occurrence of additional AKA and the use of SRTP in a series of preliminary experiments (see Section 4). In our experiments, we can verify that—despite the availability of an additional layer of security—the tested networks do not enable media plane protection.

3 ReVoLTE Attack

The goal of the REVOLTE attack is to recover the encrypted contents of a recorded VoLTE call to eavesdrop the conversation eventually. To this end, we decrypt the voice packets of an over-the-air transmission to recover the original plaintext of the voice stream. REVOLTE exploits a keystream reuse [36] that appears when two subsequent calls take place during one active radio connection. In those cases, the packets of the first call are encrypted with the *same* keystream as the packets of the second call. REVOLTE makes use of this reuse, i. e., the attack recovers the initial keystream by conducting a

[1] We note that the hyperframe number of the PDCP as specified in [6] is not the same hyper *system* frame number as specified in [9].

second call within a short time window after the initial (target) call. Once the keystream is recovered, the attack allows us to decrypt and access the contents of the recorded target call. In the following, we first introduce the general attack concept and its core components. Furthermore, we provide details on the technical and operational aspects of REVOLTE, and discuss the many practical challenges introduced by the representation of VoLTE data.

3.1 Attack Concept Overview

The attack concept of REVOLTE consists of three core components. (i) The *technical* aspects of the attack summarize the attack vector and the required steps to exploit the keystream reuse. (ii) The *operational* component summarizes all points of the attack that relate to conducting the attack, i. e., the required capabilities (attacker model), the procedure of steps (attack procedure), and the monitoring of VoLTE calls (data recording). (iii) Assuming a successful attack operation, the adversary receives *data* that needs to be processed in the subsequent steps. As introduced in Section 2.2, VoLTE traffic contains specific transmission characteristics, e. g., the use of comfort noise, or multimedia codecs, which add additional challenges for the processing of data that we need to consider. In the following, we first explain the underlying attack vector in more detail and introduce the steps required to derive the VoLTE plaintext in cases of keystream use. Using this as the technical foundation of the attack, we then describe the operational aspects of the attack and discuss the various challenges introduced by the specific elements of VoLTE voice streams.

3.2 Technical: Attack Vector

Whenever a UE connects to a base station, a *new* user plane key gets negotiated for the radio connection. While the general concept requires new keys for new connections, a keystream reuse can occur when two subsequent VoLTE calls take place within one radio connection. In this case, the eNodeB signals that the same input parameters, i. e., the direction, bearer id, and the count, shall be used with the freshly installed key for both calls and thus *the keystream is reused*. As a consequence, the same keystream encrypts a packet of the first call (*target call*) *and* a packet of the second call (*keystream call*), both with the same corresponding count.

The attacker exploits the keystream reuse by XOR-ing the recorded ciphertexts of the target call with the keystream derived from the second keystream call, as summarized in Figure 3. The keystream call allows the attacker to extract the keystream by XOR-ing the sniffed traffic with the keystream call plaintext. The keystream block is then used to decrypt the corresponding captured target ciphertext. The attacker thus computes the target call plaintext.

Exploiting the keystream reuse is the central attack vector of REVOLTE. The required steps are comparably simple and only have a minor influence on the real-world feasibility of the attack. Much more challenging aspects of its feasibility are the *operational* steps for recording traffic in the required way, and countering the challenges of the VoLTE-specific *data* representation.

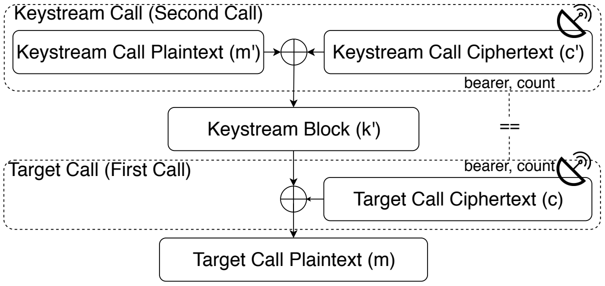
Keystream Call (Second Call)
Keystream Call Plaintext (m')
Keystream Call Ciphertext (c')
bearer, count
Keystream Block (k')
==
Target Call (First Call)
bearer, count
Target Call Ciphertext (c)
Target Call Plaintext (m)

Figure 3: REVOLTE Attack vector overview: the attacker can decrypt the packets of the recorded target call since it uses the same keystream as the second adversarial keystream call.

3.3 Operational: Attack Procedure

The operational aspects of the attack determine the steps required for successful decryption of the target call in a real-world setting. More precisely, these aspects define the attacker model and the required steps of the attack procedure that include everything beginning with the ability to record a VoLTE call right up to the decryption step.

3.3.1 Attacker Model

The attack consists of two main phases: the *recording phase* in which the adversary records the target call of the victim, and the *call phase* with a subsequent call with the victim. For the first phase, the adversary must be capable of sniffing radio-layer transmissions in downlink direction, which is possible with affordable hardware for less than $1,400 [1]. Furthermore, the adversary can decode recorded traffic up to the encryption data (PDCP) when she has learned the radio configuration of the targeted eNodeB. However, our attacker model does not require the possession of any valid key material of the victim. The second phase requires a Commercial Off-The-Shelf (COTS) phone and knowledge of the victim's phone number along with his/her current position (i.e., radio cell).

3.3.2 Attack Procedure

As REVOLTE aims to recover the encrypted contents of a voice call, its two attack phases first cover the recording of this *target call*, before the subsequent *keystream call* allows to exploit the keystream reuse and to gather all information required to decrypt the target call. Figure 4 depicts the specific procedures of both attack phases, which we describe in the

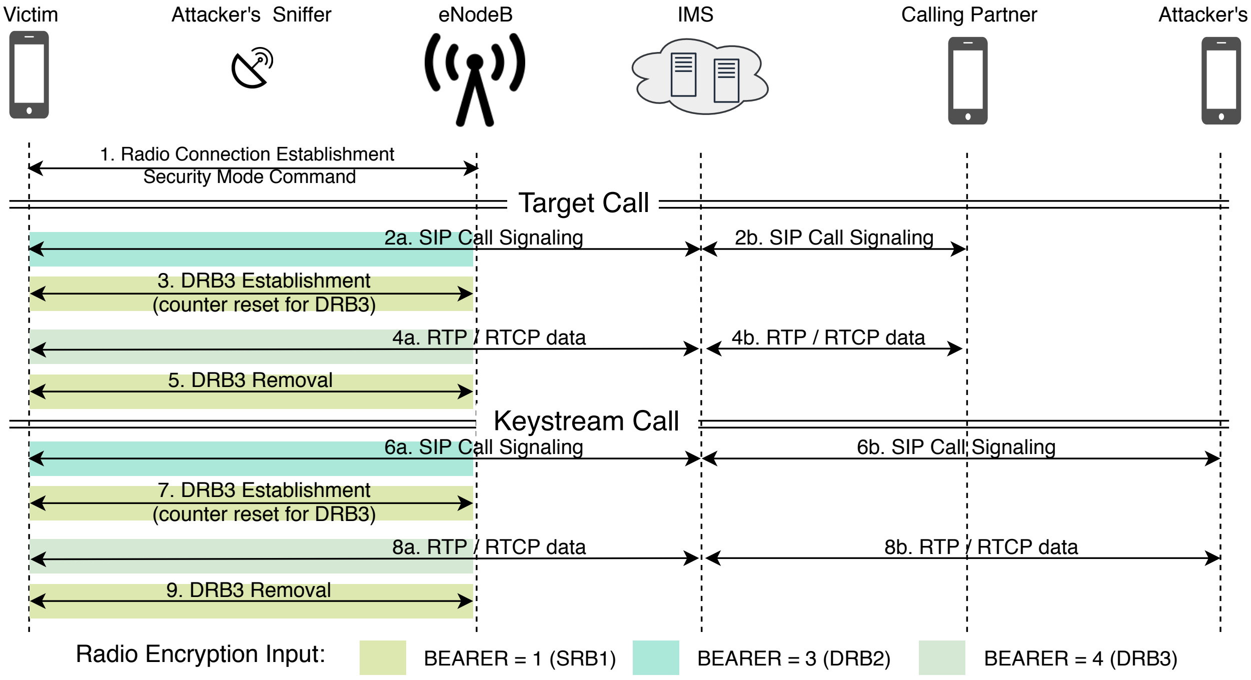

Figure 4: REVOLTE sequence diagram: The target call is encrypted with the same keystream as in the keystream call.

following. Please note that we highlight the input parameters of different bearers with distinct colors.

Target Call. Before the actual call takes place, the victim's UE establishes a connection with the eNodeB in its radio cell, which uses the two default bearers DRB1 and DRB2 for an Internet and an IMS connection. The security mode command generates a new user-plane key and activates the encryption for all data bearers; the user-plane key remains valid for the entire radio connection.

After this preliminary procedure, a standard VoLTE call establishment works as follows. SIP messages establish the call between the victim and the IMS (2a.), and the IMS forwards the call to the calling partner (2b.). Note that for REVOLTE it does not make a difference whether it is an incoming or outgoing call, as the call establishment procedure is the same in both cases. Besides the two standard bearers of the radio connection establishment (1.), the VoLTE connection requires a third *dedicated bearer* that transports the voice data between the eNodeB and the UE (3.). This dedicated bearer DRB3 transports the RTP data (4.), i. e., it provides the data relevant for the REVOLTE attack. When the phone call ends, the dedicated bearer DRB3 is removed again (5.).

The adversary monitors the target call by placing a downlink sniffer in the same radio cell that the victim's UE connects to. We explain later how an attacker can decode the sniffed data up to the encrypted PDCP layer (Section 3.4.1).

Keystream Call. The adversary uses the downlink sniffer to detect the end of the target call, i. e., when no more data occurs on DRB3. In response, she initiates the keystream call, where the attacker's UE dials the victim's UE (6.). Again, we see the same call setup procedure as for the target call (2. and 3.). At this point, one crucial thing happens: The second VoLTE call requires another dedicated bearer DBR3 to transport the voice data (7.). Since the subsequent keystream call occurs directly after the initial target call and uses the same radio connection, the count for the dedicated bearer resets, and all input parameters are the same as in the target call. As this results in the same keystream, all RTP data (8.) is encrypted in the same way as the voice data of the target call. As soon as a sufficient amount of keystream data was generated, the adversary cancels the call (9.).

Benefits. At this point, we emphasize two fundamental differences to the keystream reuse introduced previously in the technical report by Raza and Lu [36] that help to create a more realistic attack setup and procedure. First, we do not depend on *jamming*, i. e., we do not actively interfere with the transmission spectrum of the providers, but only use a passive downlink sniffer that does not change the transmissions of the radio cell. Second, the downlink sniffer allows recognizing the beginning and end of the target call, which allows initiating the keystream call immediately afterward.

3.4 Data

While the technical and operational capabilities of the adversary define the exact process to exploit the attack vector, particular additional challenges specific to VoLTE transmissions influence the process of eventually decrypting the recorded target call. In the following, we discuss the influencing factors for an *exact keystream computation* and, in the following step, for a *complete decryption*.

3.4.1 Radio Layer Sniffing and Decoding

An LTE sniffer samples the physical frequencies of a transmission and decodes radio-layer channels up to the Medium Access Control (MAC) layer. For the attack, we require to access decrypted information of PDCP. However, the configuration for decoding the MAC frames to PDCP frames is configured by the encrypted RRC layer. That means that the attacker cannot decode the data up to the PDCP layer correctly, even if the information is unencrypted as the configuration is missing. In particular, the RRC reconfiguration message when adding the dedicated voice bearer is responsible for this configuration. Part of this configuration is mapping between the Logical Channel ID (LCID) and bearer identity, the Radio Link Control (RLC) mode, PDCP sequence number length, and the used ROHC profile.

Both academic work and commercial products demonstrate the feasibility of sniffing and decoding LTE signals up to the MAC layer. Bui et al. [15] describe how to build a downlink analyzer based on srsLTE [21]. Commercial sniffers also implement the uplink sniffing functionalities [2]. For our experiments, we utilize the downlink sniffer *Airscope* by Software Radio Systems [3]. In preliminary experiments, we show that the configuration remains stable for an eNodeB. An attacker can hence learn the configuration before the attack and decode MAC frames up to the PDCP frames correctly (see Section 4.1.1).

3.4.2 User-Plane Key Reuse

TThe keystream reuse occurs when the target and keystream call use the *same* user-plane encryption key. As this key is updated for every new radio connection, the attacker must ensure that the first packet of the *keystream call* arrives within the active phase after the *target call*. Consequently, the keystream call must begin to ring before the inactivity timer at the victim's UE initiates a switch into the idle mode. However, the victim can wait as long as she/he wants to pick up the call, as the SIP messages being exchanged during ringing keep the radio connection open. Our experiments on the RRC inactivity timer show that all providers use 10 sec as a threshold.

3.4.3 Exact Keystream Computation

A successful attack depends on the extraction of the exact radio-layer keystream between the victim's UE and the eNodeB. Although the adversary knows the packet contents *sent* during the keystream call (Step 8. in Figure 4), these packets pass many different entities on their transmission path until they are encrypted with the keystream. Consequently, one central requirement for the attack is a plaintext that remains predictable during the entire transmission process until reaching the radio layer. Influencing factors with the ability to change the plaintext are transcoding, media encryption, ROHC, and plaintext-ciphertext mapping (cf. Figure 5).

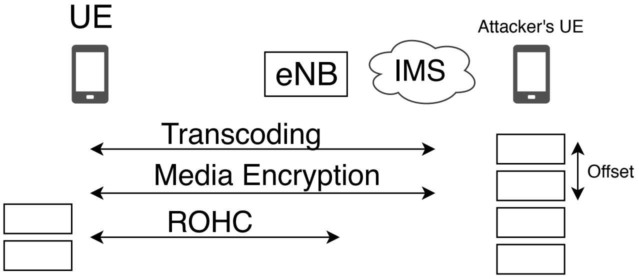

Figure 5: REVOLTE challenges for computing the exact keystream.

Transcoding. Transcoding destroys bit patterns within the packets sent by the attack. For extracting the exact keystream, REVOLTE depends on a predictable plaintext and, therefore, the attacker data must be the same as the data transmitted over the radio layer during the keystream call (between 8b and 8a in Figure 4). We analyze the influence of transcoding between shared and different providers in Section 4.1.2.

Media Encryption. Additional media plane encryption is a feature of the SRTP protocol and must be supported by the IMS and the UE, which makes it optional to use. When the network uses end-to-access edge encryption for the media plane, the sent data receives an additional layer of encryption between the UE and the IMS. This additional encryption destroys the bit pattern, which prevents the adversary from extracting the exact keystream. Our experiments demonstrate that no additional media encryption is enabled and used in all tested networks. Thus, we do not expect this to affect the attack's success.

Robust Header Compression. During the keystream call, the attacker can access the complete IP packet, including the IP, UDP, and RTP headers along with the encoded voice signal. ROHC can compress these headers before transmitting the encrypted packet between the UE and the eNodeB; the network policy defines which headers are affected by this compression. With an active ROHC, the adversary cannot use the entire packet (IP, UDP, and RTP) to calculate the keystream. Depending on the ROHC profile, the attacker can only use the RTP payload or the UDP payload for the keystream calculation. All tested providers use ROHC during VoLTE calls, which needs to be considered to extract the keystream.

Plaintext-Ciphertext Mapping. For computing the keystream, the packet containing the plaintext must be XOR-ed with the corresponding radio-layer ciphertext. Therefore, the sent and received packets at the UE must be

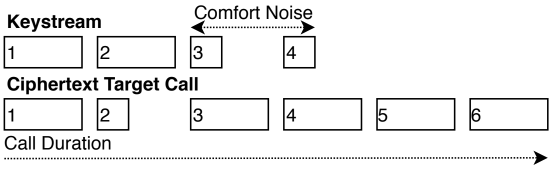

Figure 6: REVOLTE complete decryption.

mapped to the packets recorded on the radio layer, i. e., the packets of the dedicated voice bearer. Although the dedicated bearer for all voice data helps to distinguish the different packet streams, packets can still get lost or reordered on the path between the attacker's UE and the radio layer. For example, the first RTP packet sent by the adversary towards the victim is not necessarily also the first packet received at the radio layer. We analyze the mapping between plaintexts and ciphertexts in Section 4.

3.4.4 Complete Decryption

Each frame is associated with a count and encrypted with an individual keystream that we extract during the keystream computation. As the same count generates the same keystream, the count synchronizes the keystreams with encrypted frames of the target call. XOR-ing the keystreams with the corresponding encrypted frame decrypts the target call. Figure 6 depicts the synchronization between those two vectors and possible challenges. In particular, the call duration and comfort noise challenge a *complete* decryption.

Call Duration. All three VoLTE codecs use a fixed sampling rate for sending packets to the calling partner; this applies to the keystream and target call in uplink and downlink direction. That said, there are no options to fit more keystream data into the adversary's subsequent call, as both calls produce the same packet rates. As we aim to decrypt the complete call, the keystream call must be as long as the target call to provide a sufficient number of packets (Figure 6), as otherwise we can only decrypt a part of the conversation.

Comfort Noise. Comfort noise is a mechanism of the VoLTE codec that reduces the bit rate of the transmission. Whenever voice inactivity is detected, the codec generates noise following a specific seed that receives a periodical update. In contrast to standard voice packets, comfort noise encoding uses a fraction of bits and saves bandwidth in comparison to "real silence". For example, the AMR-WB codec encodes comfort noise packets with 40 bit to 477 bit.

When the attacker sends or receives comfort noise, these packets limit the amount of information that can be put into the packet. This can be a problem if the corresponding target packet is *not* a comfort-noise packet. One workaround is to create a keystream call with similar voice activity, resembling the standard and comfort noise pattern of the target call.

4 Experiments

As outlined in Section 3, a series of different network and protocol characteristics influences how packets are transmitted and, eventually, the way of decrypting the recorded target call. Despite the general concept for exploiting the attack vectors, a better understanding of these influencing factors is crucial to provide an attack concept that works on paper *and* under real-world conditions. Therefore, we conduct several preliminary experiments that provide insights into all relevant influencing factors in a commercial network. Based on the results of these preliminary experiments, we then conduct REVOLTE in a real-world setup and demonstrate its feasibility.

4.1 Preliminary Experiments

Within our preliminary experiments, we first analyze if and how eNodeBs implement the key bearer identity assignment. We then analyze the radio-layer configuration, including the use of robust header compression. In a third step, we take a closer look at further influencing factors that affect the representation of information in packets, including the codecs of VoLTE, mapping mechanisms, and media encryption.

In our preliminary experiments, we do *not* focus on the VoLTE implementation details of the different phones (i.e., basebands), as they are not critical for the success of the attack. According to the specification, the eNodeB is responsible for selecting input parameters that are used for the encryption, e. g., bearer identity, or sending the phone to idle mode. The phone must follow this setup, as otherwise the inter-operability is not given and a phone call cannot be established. Consequently, we first focus on network and eNodeB configurations.

4.1.1 Radio Layer Configuration

Among other parameters, the selected bearer identity and the radio-layer configuration influence the data, which we need to know to decode the transmitted information successfully. Furthermore, it defines the use of Robust Header Compression (ROHC). To test this, we analyzed the radio-layer configurations of three providers in Europe using commercial Android phones with VoLTE support. In our experiments, we conduct multiple phone calls, debug the connection with SCAT [24, 45], and manually inspect the recorded traces.

Bearer ID Reuse. One central requirement for the REVOLTE attack is the reuse of the same bearer identity within one radio connection. We test eNodeBs on the key reuse and find two providers vulnerable (cf. Table 2), i. e., the eNodeBs of providers P01 and P03 reuse the same bearer identity for two subsequent calls, which makes them vulnerable to the REVOLTE attack. However, the eNodeB of provider P02 increments the bearer identity and renews the key when it comes

Table 2: Radio Layer Configuration of dedicated VoLTE bearer (DRB3 for P01 and P03).

Provider	P01	P02	P03
Bearer ID Reuse	yes	no	yes
RLC Mode	UM	UM	UM
RLC Seq Len (ul/dl)	5 bits	10 bits	10 bits
PDCP Seq Len	7 bits	12 bits	12 bits
ROHC Profile	1 & 2	1 & 2	1 & 2
RRC Idle time (sec)	10	10	10

close to a bearer identity wrap around, which implements the correct behavior.

Configuration. The information we are looking for is part of the RRC reconfiguration message, which is sent for the establishment of the dedicated voice bearer. Our results show that *all* tested providers use the unacknowledged RLC[2] (cf. Table 2). The RLC influences parameters of the keystream generation, e. g., provider P01 uses smaller sequence numbers than providers P02 and P03, which affects the count calculation of the encryption algorithm. Furthermore, all three providers use an RRC inactivity timer of 10 sec, which means that the keystream call must arrive within 10 sec after the target call.

ROHC. Besides the RLC, we find that all providers deploy ROHC in profiles 1 (RTP/UDP/IP) and 2 (UDP/IP). This is a setup in which only the payload of RTP and RTCP packets is transmitted with a smaller ROHC header. Consequently, we need to take this header compression into account when computing the keystream. Due to the compression, the plaintext differs from the original plaintext sent by the attacker. However, we can utilize the RTP payload (profile 1) or the RTCP packet (profile 2) to reconstruct the keystream and not use the entire plaintext, namely the IP/UDP/RTP(RTCP) packet.

4.1.2 Transmission Characteristics

One critical aspect of REVOLTE is the process of deriving the correct keystream from the second call (i.e., the keystream call performed by the adversary). The VoLTE codecs, the offset between sent and received data, additional media encryption, and the data send in DRB3 are factors that can prevent an adversary from computing the correct keystream. In our real-world experiments, we use phones equipped with SIM cards from different providers and let them call each other for 8 times. Within these different combinations of providers, we automatically answer the incoming calls with delays in a range of 1 s to 8 s to find out possible offsets between the packets sent by the attacker and packets received by the victim. For all calls, we take a look at the codecs and possible transcoding, and check the ordering of sent and received RTP/RTCP data. In particular, we have manually inspected the traces recorded with SCAT, which contain the SIP and RTP/RTCP streams.

Table 3: Offset (in packets) between sent and received data for 8 subsequent calls and data in the dedicated bearer (DRB3 for P01 and P02)

From/To	P01	P02	P03	DRB3 Data
P01	0	0	0	RTP, RTCP
P02	0	0	0	RTP
P03	16-23	0	0	RTP

Offset and Dedicated Bearer Data. Table 3 shows the offset between the sent and received data for different provider configurations. The only combination of providers that requires further coordination by the adversary is for calls between providers 3 and 1, where initial RTP packets are lost during the transmission. For our increasing answering delay, we measure offsets ranging from 16 to 23 packets without any correlation to the increasing answering time. While an attacker can statistically evaluate the packet offset, she can use one of the other providers with a fixed offset of 0 packets. Furthermore, we find that only the first provider includes RTCP data in the dedicated VoLTE bearer DRB3. All other providers send the RTCP packets within DRB2. If the attacker computes a keystream for provider 1, she needs to consider RTP *and* RTCP packets. In the case of provider 2 and 3, she only needs to consider RTP packets.

Codecs and Media Encryption. The characteristics of the transmission codec influence the representation of information in packets, and using different codecs also leads to varying transmission characteristics that the adversary needs to take into account. We find only one single codec (AMR-WB) in our measurements where transcoding is not enabled. Furthermore, an enabled media encryption adds a layer of security that can destroy all information required for REVOLTE, which means the attack would not be feasible anymore. However, we found that none of the tested providers enables media encryption in practice.

4.2 Real-World REVOLTE

Based on the insights of our preliminary experiments, we verify the feasibility of REVOLTE in two real-world commercial networks (P01 and P03). In the following, we document the experimental setup and the steps taken to conduct the end-to-end attack.

4.2.1 Experimental Setup

Our experimental setup consists of three UEs, a laptop running Xubuntu 18.04 controlling the downlink sniffer and the attack orchestration, and an Ettus USRP B210 (cf. Table 4).

[2] RLC is a layer-two protocol above the MAC and below the PDCP layer; it defines the transmission mode for upper-layer protocol data units (PDU) (acknowledged (AM), unacknowledged (UM), transparent (TM)).

Table 4: Overview of phone configuration

Role	Phone	OS v.	Provider
Calling Partner	OnePlus 6T	9.0	P01
Attacker	Xiaomi Pocophone F1	9.0	P03
Victim	Sony Xperia X	8.0	P03

In a real-world scenario, the attacker controls only one UE and the downlink sniffer. The victim possesses one UE, and the calling partner controls the other UE. The adversary wants to eavesdrop on the call between the victim and the calling partner.

UEs. We use three Android phones with a rooted OS for automation and analysis purposes, but without effect on the attack. All phones are VoLTE capable with a Qualcomm baseband, which allows us to use SCAT [35, 45] for reading information from the diagnostics interface. In particular, SCAT enables us to capture the plaintext packets of the keystream call. The UEs of the adversary and the victim equip SIM cards of the same provider to prevent any RTP/RTCP offsets. To emulate the audio activity of a phone call, we play voice samples of the LibreSpeech Corpus [34] through the speakers of the laptop, which are nearby the phones' microphones.

Downlink Sniffer. We are mainly interested in the RTP/RTCP plaintexts of the adversary's keystream call, which allows us to reconstruct the keystream used in the target call; for debugging and evaluation purposes, we also record the traces of the calling partner and victim. To this end, we use a downlink sniffer that records the transmissions of the target and the keystream call. Besides the USRP as the hardware component, we use the commercial Airscope software [3] that uses the software stack of srsLTE [21] and performs real-time radio decoding for LTE downlink traffic. Airscope provides us with decoded MAC frames, and we use the radio-layer configuration of the preliminary experiments for correct decoding up to the PDCP layer. For a timely execution of the keystream call, we implement a live call and hang-up detection that uses the radio-layer identity Radio Network Temporary Identifier (RNTI) to distinguish phone calls in the monitored radio cell.

4.2.2 Experimental Procedure & Results

The procedure to conduct the ReVoLTE attack is as follows:

1. Downlink Sniffing. We start Airscope to analyze the cell of provider 3 and capture the downlink traffic.

2. Conducting the Target Call. The orchestration script initiates a phone call towards the victim's UE, and the laptop begins playing the audio sample as soon as the call is answered. This triggers the call detection mechanism, which results in recording the downlink traffic using Airscope. All frames of this recording are saved for the later decryption and ignored for now. After 10 s, the call ends.

3. Conducting the Keystream Call. The termination of the target call again triggers the call detection script, which instructs the adversary's UE to begin the keystream call. Again, the victim's UE answers the call and holds it for 10 s, and we monitor the downlink traffic. Furthermore, the adversary saves the RTP/RTCP packets received in the UE.

4. Decrypting the Target Call. In the final step of the attack, we decrypt the target call following the approach depicted in Figure 3. In the first step, we compute the keystream blocks for each packet of the keystream call. Therefore, we XOR the payload of the RTP packets with frames recorded during the keystream call. In the second step, we attempt to decrypt the keystream call by XOR-ing the computed keystream blocks with the recorded frames of the target call.

Result. We can decrypt 89 % of the binary representation of the target call successfully. This includes the voice data sent in the downlink direction, which directly resembles the spoken words of the conversation. The main reason for information loss in the decryption is the fact that we do not capture all radio ciphertext packets with the downlink sniffer. In particular, we lose 3 % in the target call, and 8 % in the keystream call. However, there is no information loss due to a false mapping between the plaintext and ciphertext. Therefore, we do not expect that the order of packets changes for different RLC modes.

Along with the results of our preliminary experiments, the successful real-world attack of ReVoLTE in a commercial network demonstrates the feasibility of the attack and emphasizes that given configurations do *not* prevent from the attack. Consequently, we can fully break the confidentiality aim of LTE with ReVoLTE.

5 LTE and 5G Defenses

To get a better understanding of the underlying problem and the exploited flaw of our attack, we first discuss whether it is a specification or implementation flaw. In particular, we point out that even though the security parts clearly state to avoid keystream reuse, the actual protocol specification does not prevent it. Based on these insights, we then discuss different types of countermeasures and evaluate them regarding their deployment requirements. We focus on fast deployment and sustainable mitigation options, as they help all stakeholders to prevent the substantial privacy issues of ReVoLTE efficiently.

5.1 Root Cause Analysis

The specification forbids the reuse of the keystream but does not specify an implementation, respectively. In particular, the security paragraph of the RRC specification states the following: “The eNB is responsible for avoiding reuse of the COUNT with the same RB identity and with the same KeNB” [10][5.3.1.2]. Despite being documented in the security paragraph, the rest of the protocol specification does not document measures for avoiding keystream reuse.

In particular, when going through the procedure of releasing and adding a new bearer, neither the RRC nor the PDCP specification indicates how to avoid possible keystream reuse. The RRC specification [10, 11] is responsible for the management of data bearers, i. e., the RRC reconfiguration messages sent in downlink direction can add, release, or modify bearers. When the UE receives such a reconfiguration message for adding a data bearer, it adds a new PDCP entity and configures it with the current security configuration [10][5.3.10.3]. A new PDCP entity causes a *reset of the count variable*. More precisely, the hyper frame number and the sequence numbers are set to 0 [6][7.1]. While the count starts over again, the security configuration including the k_{up} *remains the same*. This results in the keystream reuse.

Root Cause. Adding a PDCP entity for the VoLTE data bearer in the same radio connection resets packet counts for a second time, which introduces the keystream reuse for a subsequent call along with reusing the same bearer identity. We argue that the specification must clarify the problems of keystream reuse, in particular in the procedure parts of the specification. This is also part of the current deployment of 5G networks, which resembles the LTE specification.

5.2 Suggested Countermeasures

The security parts of the specification make not only the eNodeB responsible for avoiding keystream reuse, but they also suggest how to *avoid* the keystream reuse. In particular, the paragraph states: “In order to avoid such reuse, the eNB may e. g. use different radio bearer identities for successive radio bearer establishments, trigger an intra-cell handover or by triggering a transition from RRC_CONNECTED to RRC_IDLE or RRC_INACTIVE and then back to RRC_CONNECTED.” Those three mechanisms have different consequences and may be suitable for different use cases, which we assess in the following.

Radio Bearer Identities. Using different radio bearer identities mitigates the threat of keystream reuse, as a separate input parameter changes the output keystream for the subsequent call. Further, it is low-cost mitigation, as no additional messages are exchanged and no key derivation function is triggered. However, the radio bearer identity is only defined as a 5-bit field, which means that incrementing it only works for 32 new bearers. A simple bearer identity wrap-around is not allowed, as it results again in keystream reuse. In this case, the underlying key material must be changed.

Intra-Cell Handover. An inter-cell handover allows transferring a phone from one cell to another while the phone stays connected. With an intra-cell handover, the target and the origin cell are the same. Using an intra-cell handover as mitigation works, as the handover procedure has a built-in key reuse avoidance. Based on the next hop chaining counter (NNC), which is sent in an RRC Reconfiguration message, the old key (k_{enb}) and a new key ($k_{enb'}$) are derived. As the input material differs from the one used before, the keystream reuse is mitigated. However, using an intra-cell handover comes with the cost of an additional run of the key derivation function.

Switching between RRC Idle/Connected. Another possibility suggested by the specification is to switch back and forth between the RRC connected and RRC idle states, which can be achieved by the RRC connection release and the RRC connection establishment. The eNodeB sends the phone into RRC idle mode with RRC connection release. The phone then triggers an RRC connection establishment, as it needs to send data to the network. A new key for the radio connection is established when the RRC establishment carries an uplink NAS message, which increases the uplink NAS count. Again, this derives a new key ($k_{enb'}$), which is then used for the connection. In general, most RRC connection establishment procedures carry a NAS uplink message. Thus this procedure helps to mitigate the threat. However, sending the phone to idle mode does increase the latency, which should be avoided for the VoLTE calls.

5.3 Encryption of RTP Traffic

A successful REVOLTE attack requires that no additional media encryption is active [7]. Even though the adversary can attack and decrypt the radio layer encryption, such additional encryption via SRTP prevents access to any voice data. Media encryption conforms with the specification, but support by the IMS and UE is optional. However, our preliminary experiments in Section 4 demonstrate that none of the tested providers makes use of this additional layer of protection.

Using media encryption as a countermeasure to REVOLTE does not depend on any additional specification process, nevertheless, the baseband of the UE must implement it. When implemented, the encryption itself only poses a minor overhead, as we can assume that the respective algorithm, e. g., AES, is implemented in hardware. However, the key exchange is performed via the SDP protocol as part of the SIP protocol and thus brings some additional overhead.

As a long-term solution, vendors and providers both must make better use of the media encryption specification. This includes signaling the encryption support through the baseband, as well as providing all required features in the IMS.

5.4 Conclusion: Suggested Defenses

The REVOLTE attack is a threat to the confidentiality of phone calls and introduces severe privacy issues, which emphasizes the need for a practical countermeasure. As a concrete suggestion for a realistic countermeasure setup, we provide a conclusion of the above options.

As a short-term defense, we recommend increasing bearer identities; when reaching a wrap-around, we suggest deriving a new key with an intra-cell handover. However, switching from RRC connected to idle and back again introduces latency and an obsolete overhead for VoLTE calls.

A long-term solution, we recommend specifying mandatory media encryption and integrity protection for VoLTE. This provides long-term mitigation for known issues, e. g., key reuse, and missing integrity protection on the radio layer, and introduces an additional layer of security.

6 Discussion

As LTE is a fundamental part of our communication infrastructure, open attack vectors in its implementation affect millions of users worldwide. Therefore, discussing the real-world feasibility, possible attack scenarios and potential mitigation helps to get a better understanding of the impact of REVOLTE.

6.1 Real-World Application

Our experiments demonstrate the practical feasibility of REVOLTE in a real-world environment. Our realistic setup includes COTS phones that connect to standard commercial networks, and we record traffic using the downlink sniffer *Airscope* [3]. An adversary needs to invest less than 7000 $ to create a setup with the same functionality and, eventually, the ability to decrypt downlink traffic.

While our downlink REVOLTE is already feasible, a more sophisticated adversary can improve the attack's efficiency by extending the setup with an uplink sniffer, e. g., the WaveJudge5000 by SanJole [2] where we can exploit the same attack vector, and access both directions simultaneously.

6.2 Is the Victim on a Call?

For a targeted attack, the adversary needs to know if the victim is currently on a call; only if this is the case, she/he can start the keystream call right after the target call ends. Technically, this can be achieved by matching the phone number to the internal network identifiers, such as the radio layer identity (RNTI), i. e., if a victim's RNTI has an active voice bearer, the attacker knows that a call is ongoing. Prior work demonstrates that matching a public identifier with an internal network identity is feasible in mobile networks, e. g., Shaik et al. [43] demonstrate that is is possible to map the phone number to the TMSI. Further, Jover [27] and Kohls et al. [30, 38] show how an uplink or downlink sniffer can match the TMSI to the RNTI. Such stepping stone attacks allow an adversary to assess if the victim is currently on a call.

6.3 Attack Severity

The severity of the attack depends on the number of vendors using an incorrect implementation that enables to exploit the keystream reuse, as well as on the distribution of vulnerable eNodeBs. To estimate this, we sample 15 different eNodeBs with a wide geographical distribution, which is important as providers tend to deploy the same vendor within one region. Our results show that 12 of the sampled eNodeB are prone to REVOLTE. Because only a small number of vendors exists, which provide large deployments, we estimate that a high number of users are potentially affected.

6.4 User Interaction

We can exploit the keystream reuse of VoLTE when we manage to place the adversarial keystream call right after the initial target call took place. While we can demonstrate the *technical* feasibility of REVOLTE in different real-world setups and discuss their challenges, user interaction remains one mandatory factor of the *operational* aspects of the attack. In general, we can structure this user interaction in three steps:

1) Recognize Incoming Call. This step is rather simple, but still decides whether the attack can be successful. For answering the keystream call, the victim must recognize the call. We can assume that the victim is in the proximity of the phone and thus recognizes the incoming call as he just hang up the previous call.

2) Answer Call. The likelihood to answer the incoming call depends on human factors. For example, answering the phone depends on the caller identity [16]. If the caller identity is known or fits a particular pattern, e. g., area code, we can assume that it is likely that the call gets answered. The adversary can influence this by identity spoofing, which is a common attack in telephony networks [18]. Identity spoofing can exploit a variety of different attack vectors based on SS7 [42] or SIP spoofing [28, 46]. We argue that an attacker who is capable of performing such an attack can increase the likelihood that the victim answers the incoming call. Note that SS7 identity spoofing requires additional capabilities for an attacker, i. e., SS7 network access. In contrast, SIP spoofing does not require additional capabilities as only the attacker's phone must be manipulated.

3) Hold Call. To generate sufficient keystream material for the final decryption, the keystream call must be as long as the initial target call. Therefore, the adversary must keep up the conversation with the victim for a certain amount of time, depending on the recorded target call. In the context of telephony fraud, different techniques on the basis of social engineering exist, e. g., scam and robocalls are a well-known problem [18, 32, 40]. Besides these rather simple approaches, more advanced techniques use artificial intelligence to impersonate the known voice of a specific person [44]. Obviously, there is a wide range of different options to keep up the malicious keystream for the required amount of time.

Conclusion: User Interaction. Even though REVOLTE depends on user interaction—a factor we cannot influence despite an elaborate and successful technical attack concept—a large body of prior work indicates that we can assume a sufficient rate of "collaboration". To further increase the chances of a successful attack, the adversary can influence individual factors that motivate users more to answer and hold an incoming call. Overall, we conclude that user interaction is a critical but manageable aspect of REVOLTE.

6.5 Ethics

At all times, we ensure the privacy of users and ensure that we only process data related to our experiments. To ensure the privacy of uninvolved users for recorded traces, we a) never sniffed broadcast channels (e. g., the paging channel), and b) only analyze data related to our own radio identifier. We learn this by using the Qualcomm debug (SCAT) interface.

6.6 Disclosure Process

To mitigate the threat of eavesdropping, we have informed providers about the attack vector through the GSMA CVD process [4]. The GSMA requested all equipment vendors to reveal implementation details on keystream reuse mitigation and to provide patches for affected base stations. By the date of publication, the affected vendors should have provided patches, and providers are requested to install and configure them securely. However, we need to consider the large number of providers worldwide and their large deployments. It is thus crucial to raise awareness in the whole telecommunication industry to support long-term mitigation.

7 Related Work

REVOLTE extends the idea of key reinstallation attacks by an elaborate concept that covers all technical challenges of conducting the attack in real-world scenarios. In the following, we discuss the core differences between our keystream reuse and prior attack concepts, and summarize existing specification and implementation flaws in the context of LTE. Furthermore, as one core component of the attack, we outline existing options to record the traffic of an LTE radio cell.

Key Reinstallation Attacks. In 2018, Raza and Lu [36] introduced the theoretical foundation for our work. In their technical report, the authors examine key reinstallation attacks on the LTE control and user plane with an *active* radio attacker. Such key reinstallation attacks enable an adversary to *deny the service* for a user by either hijacking the location update or the deregister procedure. As part of their work, they discovered that keys are reused for user plane traffic in case of two subsequent VoLTE calls of one radio connection.

In contrast to their work, we make use of the keystream reuse to fully *decrypt the call* of a victim that we previously recorded. On the one hand, this requires a much more elaborate attack concept that is capable of countering all technical challenges implied by the protocol and transmission characteristics of VoLTE. By taking this into account, we manage to successfully conduct the attack in different commercial networks and with realistic voice signals in the calls. Our attack is feasible with a *passive* radio sniffer and a normal phone. On the other hand, our practical evaluation of different networks and attack scenarios allows us to provide an in-depth discussion of the attack vector. Furthermore, we discuss possible short- and long-term defenses against such a critical security and privacy threat.

Overall, we emphasize the importance of a *practical* perspective in this context, as otherwise neither the impact of the attack for our communication infrastructures nor the consequences for future mobile generations become accessible for future research.

Specification Flaws. In the context of radio layer vulnerabilities, Rupprecht et al. demonstrated that missing integrity protection of user plane data allows an active attacker to redirect a victim to a malicious website or even to impersonate a user towards the network and vice versa [38, 39]. The presented ALTER attack breaks the mutual authentication aim and, eventually, also affects the confidentially aim, as all subsequent DNS and TCP traffic can be intercepted. While ALTER and REVOLTE both highlight flaws on the layer two of the protocol stack, ALTER uses a more restrictive attacker model that depends on an *active* Man-in-the-Middle (MitM) adversary. In contrast, REVOLTE invades the privacy of VoLTE calls solely depending on *passive* downlink sniffing.

Further exploits of specification flaws focus on location and identity privacy and manage to localize a victim either using an active or passive attacker model [26, 27, 37, 43]. In the context of REVOLTE, we can use such attacks for verifying if a victim is in the proximity of the attacker, which provides certainty about the success chances of a targeted attack. Another direction of research is the formal verification of the LTE specification. Hussain et al. [25] introduce a symbolic model that is capable of checking critical LTE procedures; by applying their tool, they have identified different flaws that allows for denial of service or relay attacks. Basin et al. [13] and Cremers et al. [19] use a Tamarin prover to analyze the 5G AKA, which is comparable to the LTE AKA. While such work demonstrates the general security of the AKA, REVOLTE exploits the keystream reuse *after* the initial AKA.

Implementation and Configuration Flaws. While specification flaws introduce security issues in the foundations of LTE, implementation flaws are examples of an insecure realization of the specification. Kim et al. [29] propose the tool LTEFuzz, which allows to find vulnerabilities in different

LTE implementations. Furthermore, configuration flaws introduce vulnerabilities in cases where providers setup network parameters in an insecure way. Chlosta et al. [17] analyze multiple network configurations and find configuration flaws that enable an adversary to impersonate a victim to the network.

LTE Cryptography. LTE (gen. four) encrypts radio transmissions with secure encryption algorithms, e. g., AES. In contrast, the second generation (GSM) specifies three encryption algorithms, two of which have been broken in the meantime, which allows a passive adversary to eavesdrop phone calls. The A5/2 algorithm is purposely weak and already prohibited to use [12, 22, 33]; the stronger but still insufficient algorithm A5/1 can be broken by consumer hardware and rainbow tables [14, 20, 41]. In scenarios where the phone or the provider does not support VoLTE, GSM calls are still used and in case of A5/1 encryption, the call can be eavesdropped.

VoLTE Security. The security of VoLTE implementation was analyzed by Kim et al. and Li et al. [28, 31]. They found attacks that allow caller identity spoofing and billing bypass. In contrast to our work, the authors analyzed an active client attacker exploiting vulnerabilities in the core network/IMS configuration and found identity spoofing or billing bypasses.

8 Conclusion

Data confidentiality is one of the central LTE security aims and a fundamental requirement for trust in our communication infrastructure. We introduced the REVOLTE attack, which enables an adversary to eavesdrop and recover encrypted VoLTE calls based on an implementation flaw of the LTE protocol. Our attack builds upon a previously introduced keystream reuse and extends it with an elaborate attack concept that enables eavesdropping attacks in real-world commercial networks. In a series of preliminary experiments, we analyze the different protocol and transmission characteristics of VoLTE and provide an in-depth evaluation of network configurations. Based on these insights, we conduct the REVOLTE attack in a commercial network with a setup that costs less than 7000 $. Our results emphasize the need for short-term solutions that avoid the exploitation in current mobile generations and long-term solutions that help to provide data confidentiality for upcoming generations that currently indicate the same vulnerability.

Acknowledgments

This work was supported by the German Federal Ministry of Education and Research (BMBF Grant 16KIS0664 SysKit) and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2092 CASA – 390781972. Most of all, we give thanks to Muhammad Taqi Raza and Songwu Lu for their foundational work on the topic of keystream reuse. Further, we thank our shepherd Yongdae Kim for the valuable comments and guidance towards the camera-ready version, and Nils Fürste, Bedran Karakoc, and Seokbin Yun for performing several tests. We thank Domonkos Tomcsányi for the helpful discussion on testing large deployments. Software Radio System's Airscope is a central component for this research project, and we would like to thank the SRS team for providing us insights into their software tools.

References

[1] Ettus Research USRP B210. `https://www.ettus.com/product/details/UB210-KIT`. [Online; accessed 02-Mar-2020].

[2] Sanjole - WaveJudge4900A. `http://www.sanjole.com/brochures-2/WaveJudge4900A-LTEHandout-Feb11-2012.pdf`, 2018. [Online; accessed 02-Mar-2020].

[3] Software Radio Systems - Airscope. `http://www.softwareradiosystems.com/products/`, 2018. [Online; accessed 02-Mar-2020].

[4] 3GPP. GSMA Coordinated Vulnerability Disclosure Programme). `https://www.gsma.com/security/gsma-coordinated-vulnerability-disclosure-programme/`. [Online; accessed 02-Mar-2020].

[5] 3GPP. Speech codec speech processing functions; Adaptive Multi-Rate - Wideband (AMR-WB) speech codec; Frame structure. TS 26.201, 3rd Generation Partnership Project (3GPP), 12 2009.

[6] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); Packet Data Convergence Protocol (PDCP) specification. TS 36.323, 3rd Generation Partnership Project (3GPP), 01 2010.

[7] 3GPP. IP Multimedia Subsystem (IMS) media plane security. TS 33.328, 3rd Generation Partnership Project (3GPP), 12 2010.

[8] 3GPP. Service requirements for the Evolved Packet System (EPS). TS 22.278, 3rd Generation Partnership Project (3GPP), 10 2010.

[9] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2. TS 36.300, 3rd Generation Partnership Project (3GPP), 03 2011.

[10] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol

specification. TS 36.331, 3rd Generation Partnership Project (3GPP), 06 2011.

[11] 3GPP. 5G; NR; Radio Resource Control (RRC);. TS TS38.331, 3rd Generation Partnership Project (3GPP), 2018.

[12] Elad Barkan, Eli Biham, and Nathan Keller. Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication. In *Annual International Cryptology Conference*, pages 600–616. Springer, 2003.

[13] David Basin, Jannik Dreier, Lucca Hirschi, Saša Radomirovic, Ralf Sasse, and Vincent Stettler. A Formal Analysis of 5G Authentication. In *Conference on Computer and Communications Security (CCS)*, pages 1383–1396. ACM, 2018.

[14] Alex Biryukov, Adi Shamir, and David Wagner. Real Time Cryptanalysis of A5/1 on a PC. In *Workshop on Fast Software Encryption (FSE)*. Springer, 2000.

[15] Nicola Bui and Joerg Widmer. OWL: A Reliable Online Watcher for LTE Control Channel Measurements. In *Workshop on All Things Cellular: Operations, Applications and Challenges (ATC)*. ACM, 2016.

[16] Mario Callegaro, Allan L McCutcheon, and Jack Ludwig. Who's calling? The Impact of Caller ID on Telephone Survey Response. *Field Methods*, 22(2):175–191, 2010.

[17] Merlin Chlosta, David Rupprecht, Thorsten Holz, and Christina Pöpper. Lte security disabled — misconfiguration in commercial networks. In *Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*. ACM, 2019.

[18] Federal Communications Commission. Caller id spoofing. `https://www.fcc.gov/consumers/guides/spoofing-and-caller-id`. [Online; accessed 02-Mar-2020].

[19] Cas Cremers and Martin Dehnel-Wild. Component-Based Formal Analysis of 5G-AKA: Channel Assumptions and Session Confusion. In *Symposium on Network and Distributed System Security (NDSS)*. ISOC, 2019.

[20] Jovan Dj. Golić. Cryptanalysis of Alleged A5 Stream Cipher. In *Theory and Application of Cryptographic Techniques (EUROCRYPT)*. Springer, 1997.

[21] Ismael Gomez-Miguelez, Andres Garcia-Saavedra, Paul D. Sutton, Pablo Serrano, Cristina Cano, and Doug J. Leith. srsLTE: An Open-source Platform for LTE Evolution and Experimentation. In *Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization (WiNTECH)*. ACM, 2016.

[22] GSM Association Security Group. Industry Initiative to Withdraw A5/2 Briefing Paper. `http://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_44_Tallinn/Docs/S3-060541.zip`. [Online; accessed 02-Mar-2020].

[23] GSMA. VoLTE (Voice over LTE)). `https://www.gsma.com/futurenetworks/technology/volte/`. [Online; accessed 02-Mar-2020].

[24] B. Hong, S. Park, H. Kim, D. Kim, H. Hong, H. Choi, J. P. Seifert, S. J. Lee, and Y. Kim. Peeking over the Cellular Walled Gardens - A Method for Closed Network Diagnosis. *IEEE Transactions on Mobile Computing*, 2018.

[25] Syed Rafiul Hussain, Omar Chowdhury, Shagufta Mehnaz, and Elisa Bertino. LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE. In *Symposium on Network and Distributed System Security (NDSS)*. ISOC, 2018.

[26] Syed Rafiul Hussain, Mitziu Echeverria, Omar Chowdhury, Ninghui Li, and Elisa Bertino. Privacy Attacks to the 4G and 5G Cellular Paging Protocols Using Side Channel Information. In *Symposium on Network and Distributed System Security (NDSS)*. ISOC, 2019.

[27] Roger Piqueras Jover. LTE Security, Protocol Exploits and Location Tracking Experimentation with Low-Cost Software Radio. *CoRR*, abs/1607.05171, 2016.

[28] Hongil Kim, Dongkwan Kim, Minhee Kwon, Hyungseok Han, Yeongjin Jang, Dongsu Han, Taesoo Kim, and Yongdae Kim. Breaking and Fixing VoLTE : Exploiting Hidden Data Channels and Misimplementations. In *Conference on Computer and Communications Security (CCS)*. ACM, 2015.

[29] Hongil Kim, Jiho Lee, Eunkyu Lee, and Yongdae Kim. Touching the untouchables: Dynamic security analysis of the lte control plane. In *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019.

[30] Katharina Kohls, David Rupprecht, Thorsten Holz, and Christina Pöpper. Lost Traffic Encryption : Fingerprinting LTE/4G Traffic on Layer Two. In *Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*. ACM, 2019.

[31] Chi-Yu Li, Guan-Hua Tu, Songwu Lu, Xinbing Wang, Chunyi Peng, Zengwen Yuan, Yuanjie Li, Songwu Lu, and Xinbing Wang. Insecurity of Voice Solution VoLTE in LTE Mobile Networks. In *Conference on Computer and Communications Security (CCS)*. ACM, 2015.

[32] Najmeh Miramirkhani, Oleksii Starov, and Nick Nikiforakis. Dial one for Scam: A large-scale Analysis of Technical support Scams. In *Symposium on Network and Distributed System Security (NDSS)*. ISOC, 2016.

[33] osmocom Security. Withdrawal of a5/2 algorithim support. http://security.osmocom.org/trac/wiki/A52_Withdrawal. [Online; accessed 02-Mar-2020].

[34] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: An ASR corpus based on Public Domain Audio Books. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, April 2015.

[35] Shinjo Park, Altaf Shaik, Ravishankar Borgaonkar, Andrew Martin, and Jean-Pierre Seifert. White-Stingray: Evaluating IMSI Catchers Detection Applications. In *Workshop on Offensive Technologies (WOOT)*. USENIX Association, 2017.

[36] Muhammad Taqi Raza and Songwu Lu. On Key Reinstallation Attacks over 4G/5G LTE Networks: Feasibility and Negative Impact. Technical report, University of California, Los Angeles, 11 2018. https://www.researchgate.net/publication/328927054_On_Key_Reinstallation_Attacks_over_4G5G_LTE_Networks_Feasibility_and_Negative_Impact [Online; accessed 02-Mar-2020].

[37] David Rupprecht, Adrian Dabrowski, Thorsten Holz, Edgar Weippl, and Christina Pöpper. On Security Research towards Future Mobile Network Generations. *IEEE Communications Surveys & Tutorials*, 2018.

[38] David Rupprecht, Katharina Kohls, Thorsten Holz, and Christina Pöpper. Breaking LTE on Layer Two. In *IEEE Symposium on Security & Privacy (SP)*. IEEE, 2019.

[39] David Rupprecht, Katharina Kohls, Thorsten Holz, and Christina Pöpper. IMP4GT: IMPersonation Attacks in 4G NeTworks. In *Symposium on Network and Distributed System Security (NDSS)*. ISOC, February 2020.

[40] Merve Sahin, Aurélien Francillon, Payas Gupta, and Mustaque Ahamad. SoK: Fraud in Telephony Networks. In *IEEE European Symposium on Security and Privacy (EuroSP)*. IEEE, 2017.

[41] Security Research Labs. Kraken: A5/1 Decryption Rainbow Tables. https://opensource.srlabs.de/projects/a51-decrypt, 2010. [Online; accessed 02-Mar-2020].

[42] Hemant Sengar, Ram Dantu, Duminda Wijesekera, and Sushil Jajodia. SS7 over IP: signaling interworking vulnerabilities. *IEEE Network*, 20(6):32–41, 2006.

[43] Altaf Shaik, Ravishankar Borgaonkar, N. Asokan, Valtteri Niemi, and Jean-Pierre Seifert. Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems. In *Symposium on Network and Distributed System Security (NDSS)*. ISOC, 2016.

[44] Catherine Stupp. Fraudsters Used AI to Mimic CEO's Voice in Unusual Cybercrime Case. https://www.wsj.com/articles/fraudsters-use-ai-to-mimic-ceos-voice-in-unusual-cybercrime-case-11567157402. [Online; accessed 02-Mar-2020].

[45] The Computer Security Group at Berlin University of Technology. SCAT: Signaling Collection and Analysis Tool. https://github.com/fgsect/scat. [Online; accessed 02-Mar-2020].

[46] Patrick Ventuzelo, OL Moal, and Thomas Coudray. Subscribers Remote Geolocation and Tracking using 4G VoLTE Enabled Android Phone. In *Symp. on Information and Communications Security (SSTIC)*, 2017.

Acronyms

3GPP	3rd Generation Partnership Project
AES	Advanced Encryption Standard
AKA	Authentication and Key Agreement
AMR	Adaptive Multi-Rate
AMR-WB	Adaptive Multi-Rate Wideband
COTS	Commercial Off-The-Shelf
eNodeB	Evolved NodeB
EPC	Evolved Packet Core
EVS	Enhanced Voice Services
IMS	IP Multimedia Subsystem
LCID	Logical Channel ID
LTE	Long Term Evolution
MAC	Medium Access Control
MitM	Man-in-the-Middle
MME	Mobile Management Entity
NAS	Non-Access Stratum
P-CSCF	Proxy Call Session Control Function
PDCP	Packet Data Convergence Protocol
RLC	Radio Link Control
ROHC	Robust Header Compression
RRC	Radio Resource Control
RTCP	RTP Control Protocol
RTP	Real-Time Transport Protocol
RNTI	Radio Network Temporary Identifier
SDR	Software Defined Radio
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
SRTP	Secure Real-Time Transport Protocol
UE	User Equipment
VoLTE	Voice over LTE