Breaking and Fixing VoLTE

Conference Paper · October 2015

DOI: 10.1145/2810103.2813718

Breaking and Fixing VoLTE: Exploiting Hidden Data Channels and Mis-implementations

*Hongil Kim† *Dongkwan Kim† Minhee Kwon† Hyungseok Han†
Yeongjin Jang§ Dongsu Han† Taesoo Kim§ Yongdae Kim†

† KAIST, Republic of Korea
§ Georgia Institute of Technology, USA

{hongilk, dkay, minhee.k, han456, dongsu_han, yongdaek}@kaist.ac.kr,
{yeongjin.jang, taesoo}@gatech.edu

ABSTRACT

Long Term Evolution (LTE) is becoming the dominant cellular networking technology, shifting the cellular network away from its circuit-switched legacy towards a packet-switched network that resembles the Internet. To support voice calls over the LTE network, operators have introduced Voice-over-LTE (VoLTE), which dramatically changes how voice calls are handled, both from user equipment and infrastructure perspectives. We find that this dramatic shift opens up a number of new attack surfaces that have not been previously explored. To call attention to this matter, this paper presents a systematic security analysis.

Unlike the traditional call setup, the VoLTE call setup is controlled and performed at the Application Processor (AP), using the SIP over IP. A legitimate user who has control over the AP can potentially control and exploit the call setup process to establish a VoLTE channel. This combined with the legacy accounting policy (e.g., unlimited voice and the separation of data and voice) leads to a number of free data channels. In the process of unveiling the free data channels, we identify a number of additional vulnerabilities of early VoLTE implementations, which lead to serious exploits, such as caller spoofing, over-billing, and denial-of-service attacks. We identify the nature of these vulnerabilities and concrete exploits that directly result from the adoption of VoLTE. We also propose immediate countermeasures that can be employed to alleviate the problems. However, we believe that the nature of the problem calls for a more comprehensive solution that eliminates the root causes at mobile devices, mobile platforms, and the core network.

Categories and Subject Descriptors

C.2.0 [**Computer-Communications Networks**]: General—*Security and protection*; C.2.1 [**Computer-Communications Networks**]: Network Architecture and Design—*Wireless communication*

Keywords

VoLTE; Accounting; Security; Cellular Networks

1 Introduction

Due to the increasing demand for data-centric services, mobile network operators are quickly moving towards high-speed networks. With higher bandwidth and lower latency, Long Term Evolution (LTE) has become the dominant cellular network technology in recent years. One distinctive feature of LTE is the way it delivers data such as voice and SMS; LTE operates through packet-based switching, whereas traditional cellular networks (e.g., 2G or 3G) rely on circuit-based switching for their voice service. To reliably serve voice calls on a packet-switching only network, mobile network operators have adopted, deployed and recently started a service for end-users, called Voice-over-LTE (VoLTE) [17], which is similar to a Voice-over-IP (VoIP) service in spirit.

Today, mobile network operators are aggressively deploying VoLTE services: by April 2015, 16 operators in 7 countries had commercially launched VoLTE services, and 90 operators in 47 countries are investing to deploy VoLTE services in the near future [13]. Despite this fast-moving trend, little research has been conducted to systematically examine security issues in the upcoming VoLTE services, not only in terms of their end-facing interfaces but also their cellular infrastructure.

The use of packet-switching in VoLTE opens a large attack surface that has not been seriously considered thus far. In circuit-switching mobile networks, the signal processing is conducted by a communication processor (CP) in a mobile phone, whose detailed implementation is proprietary to a few chip manufacturers. On the contrary, VoLTE-compatible devices perform signal processing purely in their application processor (AP): e.g., initiating Session Initiation Protocol (SIP) [29]. As with general-purpose computers, VoLTE relies completely on Internet Protocol (IP) for packet delivery. Consequently, well-known offensive techniques targeting the IP are also applicable for abusing or attacking VoLTE-based devices.

Several studies exploring potential attacks and countermeasures in SIP and VoIP services have been conducted, including, for example, works on breaking authentication [8], bypassing accounting [36], mounting man-in-the-middle attacks [35], introducing various attacks by a hacker [24], and even standardizing basic security issues such as credentiality, integrity and authenticity, by communities [4, 5, 7, 34].

Since VoLTE operates on the cellular network, it is not only exposed to VoIP-related issues, but also inherits security issues from the cellular network, such as adversaries being able to interpose

*These two authors contributed equally

CCS'15, October 12–16, 2015, Denver, CO, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3832-5/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2810103.2813718.

the signal processing itself. For example, unlike VoIP services, the LTE network provides a communication channel (which is called a *bearer*) with guaranteed bandwidth, once a VoLTE call is established. As it becomes easy to interpose call signaling with VoLTE functionality, an adversary can create and utilize private communication channel for peer-to-peer data exchange, which is not a supported feature in cellular networks. Furthermore, since most operators do not charge the use of a dedicated channel for VoLTE service, an adversary can utilize it without being charged.

More seriously, its implementation caveats in authentication and session management for VoLTE make its infrastructure vulnerable, and, therefore, an adversary can easily mount various attacks that bypass the security policy of VoLTE.

Regarding problems with user equipment (UE), VoLTE opens a security loophole whereby an adversary can make a call without suitable permission for voice, since the current permission model for Android devices is only suitable for legacy circuit-switching calls.

In this paper, we first present problems of commercially deployed VoLTE services in 5 operators in the United States and Korea [1]. These problems are mainly caused by legacy policies and the immature software infrastructure of VoLTE. To show the motivation for addressing these problems, we demonstrate various attacks that (1) piggyback a hidden, free data channel (e.g., free extra bandwidth to the adversary), (2) bypass VoLTE's accounting system (e.g., direct calling bypassing the charging server), and (3) abuse the VoLTE service (e.g., caller spoofing, overbilling attack)

In addition, we propose immediate and potential countermeasures for these problems. For immediate solutions, 1) operators may deploy DPI (Deep Packet Inspection) for detecting a hidden data channel, 2) strict session management on both the phone and operator side is required, and 3) cellular gateways have to be fixed to prevent hidden channels. As a longer term solution, we suggest changing the accounting policy of operators, and tighter security implementation at the mobile devices. To completely eliminate problems, however, cellular operators, device manufacturers, and mobile platform providers must draw up a comprehensive solution.

To summarize, we make the three following contributions:

- To the best of our knowledge, this is the first attempt to analyze the security loopholes of commercially deployed VoLTE services. We found three previously unknown security issues on VoLTE: 1) hidden data channels in VoLTE services, 2) misimplementation of the cellular operators, and 3) fundamental problems in the mobile devices.

- To show the motivation for addressing these problems, we successfully demonstrate an attack that enables a free data channel, and various abuse attacks including call spoofing and denial-of-service. At the time of the submission, all bugs and exploits are reported to operators.

- We proposed effective, immediate countermeasures, and further devise a long-term and comprehensive solution that can eliminate current security issues in the VoLTE service.

The rest of the paper is organized as follows. §2 presents an overview of the VoLTE system with the network architecture and call setup procedure including the accounting policy of real operators. In §3, we bring up current problems and threats of deployed VoLTE services through an analysis of VoLTE call flow. We explain the details of hidden data channel attacks and implementation details along with the measurement results in §4. §5 introduces possible attacks caused by several implementation flaws in the VoLTE service. In §6, we provide countermeasures of our attacks and discuss immediate and fundamental solutions. §7 includes related work, and we conclude our study and present directions for future work in §8.

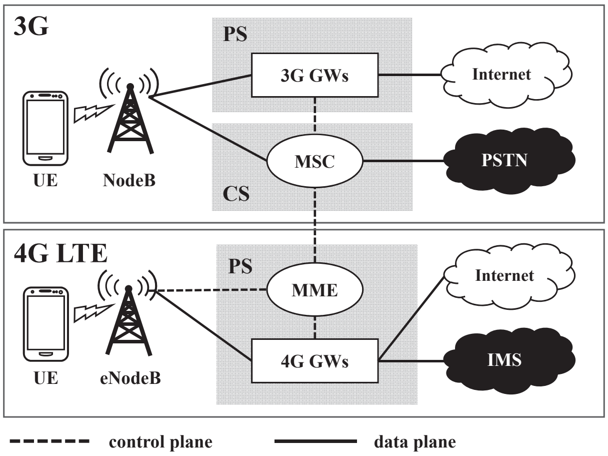

Figure 1: Two-folded architecture of 3G and LTE networks. Mobility Management Entity (MME) in LTE stands for user mobility.

2 Background

2.1 LTE Network Infrastructure

A cellular network comprises two architectural components: an access network that the UE connects to, and a core network that supports its cellular infrastructure. Figure 1 illustrates the two-folded architecture of 3G and LTE networks. The access network (left side) is a radio connection where the UE accesses a base station (e.g., NodeB in 3G and evolved NodeB, shortly eNodeB in LTE). On the other hand, the core network (gray region) handles service-level connections such as voice calls and the Internet (e.g., PSTN in 3G and IMS [3] in LTE). IMS stands for IP Multimedia Subsystem, and it offers IP-based voice calls and multimedia services by using the SIP (bottom), and PSTN stands for Public Switched Telephone Network, a typical public telephone network (top). 4G GWs in the core network consist of Serving Gateway (S-GW) and P-GW. S-GW is a mobility anchor for inter eNodeB handover and relays the traffic between 2G/3G systems and P-GW. Meanwhile, the P-GW manages PDN connection between a UE and a service. It is also responsible for packet filtering and charging, which are crucial functions for preventing accounting bypass and service abuse attacks.

The major difference between 3G (top) and LTE (bottom) networks is in the way they deliver data in the core network. The 3G network separates network domains into packet-switching for the Internet connection and circuit-switching for phone calls. The mobile switching center (MSC) in the circuit-switching domain transfers voice calls, and the 3G gateways enable data communication in the packet-switching domain. In contrast, the LTE network only operates through the packet-switching domain; as it does not have a circuit-switching domain, its voice calls either fall back into the 3G network (also known as Circuit Switched Fallback, or CSFB in short) or the LTE provides a VoLTE solution to transfer both voice calls and data to the packet-switching domain, which does not require any fallback to the 3G circuit-switching network.

[1] We performed experiments on 6 operators, but we failed to connect to one operator's VoLTE service, unlike its advertised service coverage

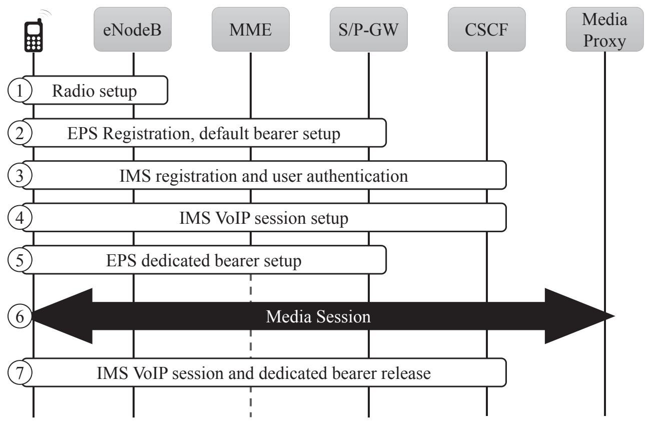
eNodeB
MME
S/P-GW
CSCF
Media Proxy
1 Radio setup
2 EPS Registration, default bearer setup
3 IMS registration and user authentication
4 IMS VoIP session setup
5 EPS dedicated bearer setup
6 Media Session
7 IMS VoIP session and dedicated bearer release

Figure 2: Overview of Epacket-switching and IMS protocols in VoLTE; registration and call setup between a UE and a LTE network.

2.2 VoLTE Service

VoLTE service is introduced to deliver voice calls over the packet-switching based LTE network. The service utilizes an IMS network based on SIP, similar to VoIP service over the LTE network. To establish a voice call, a UE follows standard procedures as depicted in Figure 2.

To connect to the LTE network, ① a UE first contacts the eNodeB, and then ② the UE registers itself to the Evolved Packet System (EPS), establishing an Internet Protocol (IP) connection, which is identified by a *default bearer*. Note that this IP address is different from the one used for data connection. In other words, every phone supporting VoLTE is assigned two IP addresses: one for voice and the other for data [2]. Once the UE has an IP connection to VoLTE, ③ the UE connects to the IMS network and the IMS server authenticates whether the device is allowed for the VoLTE service.

If authenticated, ④ the UE can make a voice call through the SIP signaling service, provided by Call Session Control Function (CSCF) servers. When a call session is established, ⑤ a *dedicated bearer* is created to identify voice-related traffics and ⑥ all voice packets are transferred through this dedicated bearer. ⑦ Upon call termination, the bearer used for the voice session is released.

Note that two bearers are used to enable a connection in the VoLTE service. The default bearer established during the EPS registration is for call signaling. Once the default bearer has been established, all the incoming and outgoing SIP packets are bound to this bearer. According to the 3GPP specification [1], this default bearer has the highest priority among all possible bearers for voice or data services. The main reason for this different prioritization is to support the QoS of a phone call, similar to circuit-switching based routing. When a phone call is made, the IMS Packet Data Network (PDN) temporarily creates a dedicated bearer that has a lower priority than the default one. This dedicated bearer, however, has a higher priority than the bearers for data services. Although the dedicated bearer has the same IMS PDN address as the default bearer, it operates by different rules that make voice packets with the negotiated media port go through the dedicated bearer.

2.3 VoLTE Signaling Protocol

Call Signaling. Figure 3 illustrates the call setup procedure between two UEs, UE-A and UE-B. This can be considered as a more detailed version of Figure 2 from ④ to ⑦, corresponding to the VoIP protocol. To initiate a VoLTE call to UE-B, UE-A first generates an `INVITE` message and sends it to a SIP server [2]. The `INVITE` message contains a description of the caller's phone number, IP address, and media characteristics: a port number, encoding scheme, and QoS parameters for media communication. Upon receiving the `INVITE` message, the SIP server responds to UE-A with a `TRYING (100)` message, and then it relays `INVITE` to UE-B after checking if the message is valid. Upon receiving the `INVITE` message, UE-B responds with `RINGING (180)` and `SESSION PROGRESS (183)` to indicate its call session is being processed. In response to the `SESSION PROGRESS` message, UE-A sends a progress ack (`PRACK`) message.

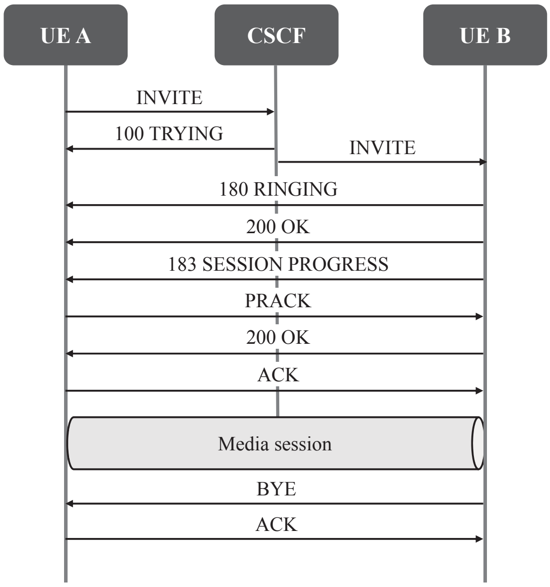

Figure 3: VoLTE signaling (SIP) flow of Call setup and tear down

If the user accepts the call at UE-B, it sends an `OK (200)` message containing information similar to the `INVITE` message. When the SIP server receives `OK (200)`, it routes this message to UE-A and starts charging the calling session. As soon as UE-A receives `OK (200)`, an end-to-end media session for voice data is established between UE-A and UE-B. This media session contains a dedicated bearer for both UE-A and UE-B, and a media proxy in the IMS network. Typically, this media session is implemented using RTP (Real Time Protocol) on top of UDP. When either of the UEs wants to terminate the call, it sends a `BYE` message to the SIP server. Upon receiving such a message, the SIP server stops charging the call session and routes `BYE` to the other UE, and terminates the media proxy.

Call Management in the UE. After a call session is established, UEs transfer voice data to each other through the established media channel. The smartphones we used for our experiment have two processors: an application processor (AP) for running the smartphone operating system (e.g. Android) and user applications; and a communication processor (CP) for handling radio access and radio-related signaling.

Call signaling is handled by a SIP client running in the AP. This SIP client binds its socket on a specific port (default: 5060) to communicate with SIP servers. Meanwhile, the CP has a digital signal processing (DSP) module that handles radio communication with

[2] In this paper, for simplicity, we use the SIP server to represent all Call Session Control Function (CSCF) servers in the IMS network.

Table 1: Characteristics of VoLTE services on tested carriers

	US-1	**US-2**	**KR-1**	**KR-2**	**KR-3**
Network protocol	IPv6	IPv6 + IPSec	IPv4	IPv4	IPv6
Transport protocol for SIP	TCP & UDP	TCP & UDP	UDP	UDP	UDP
Encryption algorithm for IPSec	-	AES	-	-	-
Capability of changing SIP source port	✓	✘	✓	✓	✓
Existence of a media proxy	✘	✓	✘	✓	✓
Capability of sending random data through media session	✓	✓	✓	✓	✓
Capability of changing QoS parameter specified in `INVITE`	✘	✘	✘	✘	✘
Free use of audio data	✓	✓	✓	✓	✓

base stations as well as audio data from a speaker and a microphone. Upon receiving voice traffic from a radio channel, the CP processes the voice packets and forwards only audio data to the AP, which lessens the computational burden of the AP.

3 Empirical Analysis of VoLTE Services

In this section, we describe our empirical security analysis on the current implementation of VoLTE services by commercial, deployed mobile cellular network operators. We examined five major carriers [3] in the United States and Korea, the two countries with the highest LTE penetration ratio for VoLTE service [20].

For our security analysis, we do not assume a privileged adversary that has physical access to a core network. Instead, we consider an adversary who is legitimately subscribed to an LTE network. Therefore, the adversary considers the actual implementation of cellular networks as a blackbox. This implies that the adversary can only access the public information and infer implementation by analyzing the behaviors of the VoLTE service. Also, the adversary is allowed to have full permission of her mobile devices to have access to raw sockets or the device interface in the Android kernel. In short, the adversary may have root access to the phone.

In the rest of this section, we first describe an interesting accounting policy that we discovered from the VoLTE carriers: discrepancy of the accounting policy and its cost model. After understanding the accounting policy, we explain a hidden, free data channel that we found in the current VoLTE protocol, and how to exploit the discrepancy. We then describe several security problems that we found in the commodity VoLTE services.

3.1 Accounting of VoLTE Calls

While VoLTE is implemented on the packet-switching network (i.e. it runs on IP), it applies a legacy time-based charging policy as in 3G networks. Traditionally, a voice call delivered through the circuit-switching network is charged according to the duration of time it occupies the channel (e.g. $15 for 225 minutes). In contrast, data connection through a packet-switching network is charged based on byte-usage (e.g. $15 for 1GB). As VoLTE utilizes only a packet-switching network, it consumes the same byte-usage as for the data connection. Despite this, it is quite odd that the majority of carriers still charge the voice service by time duration. This *discrepancy* could complicate accounting procedures, since different IP addresses are used for both data and voice, and their accounting units are different. Furthermore, many operators recently provide unlimited calls among VoLTE users as a default pricing plan, which could be used as a free data channel, if exploited.

3.2 Analyzing Hidden Data Channel

With the problems outlined above as motivation, we conducted a thorough empirical analysis of current VoLTE services. To determine how accounting on VoLTE could be bypassed, we first analyzed the 3GPP specifications for the VoLTE protocol. We then examined the implementation of target carriers (five carriers) by examining actual traffic between a UE and the cellular core in order to check if there exists any exploitable vulnerability.

3.2.1 Potential Free Data Channels

As mentioned above, the accounting for VoLTE call starts when a SIP server receives `OK (200)` from UE-B, the receiver. Thus, if UE-B does not send `OK (200)` to the SIP server, then the call may not be charged.

An adversary may also consider either sending data by encapsulating it into SIP messages (e.g. messages before `OK (200)` such as `INVITE`) or directly sending `OK (200)` to UE-A, the sender, bypassing the SIP server. Since each UE already has an IP address for its default VoLTE bearer, UE-A can send SIP messages directly to the other UE using this address if it's not blocked by the carrier's implementation.

Furthermore, according to the 3GPP specification [3], the QoS parameter for voice traffic is specified in the `INVITE` message. Thus, if an adversary could manipulate this parameter, she would be able to increase the bandwidth for sending a large amount of data.

Note that the above potentially free data channels could be easily blocked or detected by SIP network flow analysis. However, what if an adversary embeds the data in the media session? Detecting this requires significant implementation effort, as the carrier needs to check if the data in the media session are voice or not.

In summary, an adversary may try to 1) squeeze data into SIP packets, 2) send data directly to the receiver, or 3) send data over the media session.

3.2.2 Empirical Analysis

With the knowledge of the signaling protocol of VoLTE, we analyzed the actual call flow in the top five carriers. The characteristics of the service of each carrier are summarized in Table 1. Note that each carrier supports different smartphones for VoLTE and still only a limited number of models support VoLTE. For the analysis, we used the following four models: Samsung Galaxy S5, S4, and LG G3. In each experiment, we used at least two different models among them.

Transport protocol. To send/receive SIP messages, Korean operators use UDP only, whereas U.S. operators use both UDP and TCP.

[3] We intentionally anonymized the names of carriers to protect them until all operators fix the problems we identified.

We discovered that the U.S. operators send response messages such as ACK or PRACK using UDP, while TCP is used for all other SIP messages. One of the U.S. operators protects SIP messages using IPsec with AES encryption. Fortunately, we were able to change the encryption algorithm of IPsec from AES to Null by modifying configuration file for SIP in the UE side. It should also be noted that we were able to analyze call flows in plaintext. Furthermore, we identified that an IPsec daemon running on the phone automatically wraps packets with a specific SIP port into the IPsec tunnel. By utilizing this daemon, we were able to send SIP messages to a SIP server.

Changing a SIP source port. Since the native SIP client in UE-A is already listening on a pre-defined port, other custom applications cannot bind to this port. Therefore, we checked if SIP servers accept other source ports, and found that all operators except one U.S. operator allow that. This could be problematic if a malicious app initiates VoLTE sessions. (See §5.1 for more details.)

Media proxy. We also checked whether a proxy for relaying media data exists. The results of our analysis indicated that two of the operators do not utilize a media proxy, resulting in UEs being able to directly transfer media data. In this case, when a UE sends an INVITE message to a SIP server, the UE receives a response message with the other UE's IP address. Therefore, a UE can collect another UE's IP address by randomly sending INVITE messages to the SIP server. Furthermore, if a media proxy exists, it could be used to detect malicious behaviors over the media session by inspecting packets.

Sending data through a media session. When a call is established, a dedicated bearer is also created for the media session. Voice packets are then sent directly through this bearer from CP in UE-A. However, we were able to send packets through this bearer in AP with the IP address and port number specified in SIP messages for all operators. In other words, the Android device does not have proper access control for using the dedicated bearer for the media session. Furthermore, transmitted data over the media session were not charged since operators provide unlimited calls for VoLTE users.

Manipulating QoS negotiation. We investigated whether we could manipulate QoS parameters specified in INVITE to acquire higher bandwidth. However, we discovered that even if we changed the QoS parameters in INVITE, the actual QoS level of the dedicated bearer remained unchanged. In other words, for all the operators, SIP servers do not consider the QoS parameters in INVITE (See §4.2.1 for details.)

Summary. As we discussed, there may exist multiple hidden data channels in VoLTE. In §4, we show how we exploit these channels and which operators are open to these channels.

3.3 Implementation Problems of VoLTE

Throughout the analysis of VoLTE call flow, we discovered several security problems associated with both the UEs and the operators.

Permission model mismatch. This is an interesting issue we discovered. In general, mobile devices have separate permissions to regulate the behavior of an application for security reasons. For example, in an Android device, an application should have the permission, *android.permission.CALL_PHONE*, to call other people. However, we discovered that this permission could be violated due to the implementation of the VoLTE interface. To verify this, we developed an Android application that has only one permission, *android.permission.INTERNET*, which enables the application to send data to the Internet. With this permission, the application can send SIP messages to the SIP server to call other people. This shows that the current Android permission model cannot distinguish SIP messages from data communication. We also found that when the call is initiated by the application, *the calling state is not displayed on the phone*. Therefore, a user would not be able to recognize that her phone is now calling.

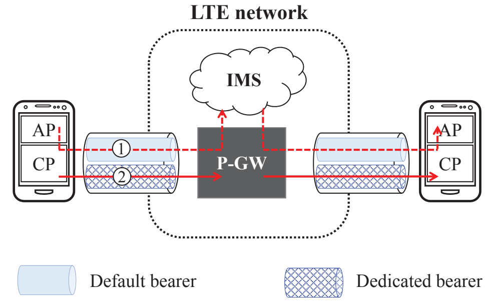

Figure 4: Flow of (1) SIP tunneling and (2) RTP tunneling

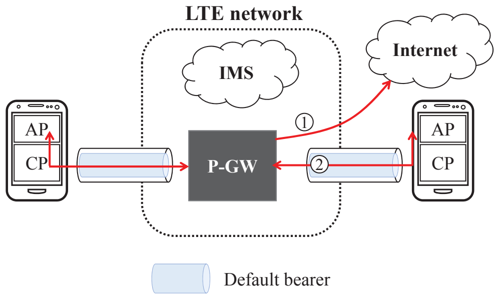

Figure 5: Flow of direct communication channel

Direct communication. When UE-A makes a call, it sends an INVITE to a SIP server. The SIP server then manages the call session between the UEs. In cellular networks, direct phone-to-phone data communication is blocked using NAT (Network Address Translation), even if two mobile devices know each other's IP address. We discovered that with two of the operators, sending SIP messages directly from UE-A to UE-B was possible; furthermore, the call session was successfully established. As a result, the communication could not be accounted as discussed in §2.3. This problem comes from the inappropriate access control of the default bearer for SIP signaling in P-GWs of the LTE network.

No authentication. A SIP server should authenticate every SIP message to determine whether it is from a valid user. However, we discovered that two of the Korean operators do not perform proper user authentication. As a result, we could make a call with a fake phone number by sending a manipulated INVITE.

No session management. In addition, we determined that, apart from one operator in Korea, none of the operators manages call sessions correctly. As a result of incorrect session management in SIP servers, an adversary can create multiple SIP messages to make a phone call to many people simultaneously. It should be disallowed, as more than one call sessions should not be established. Furthermore, this could be used to launch control plane DoS attacks. Core networks are able to manage a limited number of bearers, and if compromised phones start sending multiple INVITE messages which generate a number of bearers, it will deplete the resources in LTE networks. When UE-A sends an INVITE, the dedicated bearers are established both for UE-A and UE-B, even if UE-B does not respond to the call.

Summary. VoLTE potentially has several security problems as we discussed in this section. In §5, we discuss how these vulnerabilities can be exploited among different operators.

4 Exploiting Hidden Data Channels

From the analysis of the accounting policy and VoLTE call flow in §3.2, we showed the possibility of hidden channels that an adversary can utilize to bypass accounting. These channels are classified into channels that reside in the VoLTE call flow (i.e. SIP and RTP tunneling) and a direct communication channel utilizing a VoLTE default bearer.

SIP/RTP tunneling are potentially free channels in VoLTE call service, as in Figure 4. ① In SIP tunneling, the payload is embedded in SIP messages, and these messages are sent through the default bearer. Meanwhile, ② RTP tunneling carries the payload through the established dedicated bearer, as explained in §2.2. Strictly speaking, any protocol can be used for data delivery through the media session. However, we utilize RTP tunneling as all the operators we tested encapsulated voice data with RTP. Note that the voice data transmission using this channel is handled in the CP, and most of the details on its implementation remain proprietary.

Direct communication is another channel that directly sends one UE's data to another UE. Figure 5 illustrates the flow of direct communication: ① phone-to-Internet and ② phone-to-phone. Since the default bearer for VoLTE signaling messages is always established as long as the device is turned on, a UE can easily send data through this bearer to the Internet or another UE unless P-GW blocks it.

4.1 Exploitation

We implemented our own sending and receiving modules to verify the hidden channels in operational networks. Figure 6 illustrates the sending module (left) and receiving module (right), which are connected through the IMS network.

As 3GPP specifications give some freedom to operators, and it is not clear if all operators follow 3GPP specifications [3, 18] completely, our modules take this implementation-specific deviance into account. For example, for the operator using IPsec, we utilized the established IPsec tunnel to send our data instead of sending SIP messages directly. In the case of direct communication, we do not require any additional implementation since we can open a socket at each side and transfer packets directly, if possible.

4.1.1 Sending Module

First, our sending module should have more functionality than the native VoLTE calling application in a mobile phone; it should be able to vary its parameters such as the sender's phone number, IP address, and port number to create arbitrary media sessions. The sending module consists of SIP Parser, SIP Handler, SIP Tunneling Sender and RTP Tunneling Sender.

① **SIP Parser** extracts common headers and carrier-specific headers in the packets obtained from native VoLTE apps. For example, an `INVITE` message contains the caller's phone number, IP address, and routing information such as IP addresses of SIP servers. SIP Parser automatically parses this information and stores it in its database separated by each operator for later recreation of the SIP message in our sending module.

② **SIP Handler** manages the exploitation. When we initiate our attack, it takes configuration values: operator's name, phone number, and phones' IP address and port number for tunneling. By simply modifying these configuration values, the SIP Handler can generate SIP messages for each operator. It also randomly generates parameters (e.g. branch, tag, and calling ID) that distinguish each call session to guarantee freshness. SIP Handler triggers either SIP Tunneling Sender or RTP Tunneling Sender for each test.

③ **SIP Tunneling Sender** establishes a SIP tunnel when it receives a signal from SIP Handler. It first fragments a file to be transferred into several blocks. Then it embeds the fragmented blocks inside the SIP messages. Since there exists a maximum number that we can fragment (otherwise blocked by SIP servers in IMS) and the size of each block cannot exceed the MTU, the SIP Handler may need to fragment the file into multiple blocks. For convenience, we place the data block at the end of the body in `INVITE`.

④ **RTP Tunneling Sender** is more complicated than SIP tunneling. In the case of RTP tunneling, fragmentation of a given file is the same as in SIP tunneling. However, we first must establish a media session to transfer data. Therefore, the RTP Tunneling Sender generates an `INVITE` and follows the native call flow until it gets an `OK (200)` message from the callee. It then extracts the IP address and the port number of the callee from the established media channel. It transfers the data blocks wrapped as an RTP packet to the extracted IP address and the port number. In native calling apps, voice is wrapped and sent from the CP. However, as described in §3.2.2, we discovered that audio packets from the AP to the receiver are routed correctly as well. To distinguish our packets from others, we add an identifier at the beginning of the payload. We also add the sequence number and timestamp after the identifier for our performance evaluation.

4.1.2 Receiving Module

The receiving module receives the data blocks in RTP packets sent from the sending module through the IMS network. Our receiving module consists of the SIP Tunneling Receiver, RTP Tunneling Receiver, Measuring Engine, and Data Storage.

❶ **SIP Tunneling Receiver** parses SIP messages and extracts our data from received RTP packets. Since SIP messages are processed in the AP, as described in §2.2, the SIP Tunneling Receiver can capture incoming packets and parse them in real time. As we placed the data blocks at the end of the body in the `INVITE`, the receiver can easily extract the fragmented blocks and reassemble them. The receiving module opens a raw socket to capture packets because a SIP daemon is already running on the device. For the operator using IPsec, we could easily extract SIP messages out of ESP packets since we changed the encryption algorithm to Null.

❷ **DIAG** is QualComm's proprietary diagnostic protocol. It has a command that can be used to mirror every received packet to the RTP Tunneling Receiver via DIAG interface in the Android kernel as introduced by Delugre [10]. To initiate mirroring from CP to the DIAG interface, the mobile device has to be connected to a laptop once. After this, the laptop can be disconnected.

This step is necessary since the data we sent through the media channel are only processed in CP, but not forwarded to AP. Therefore, in order to receive and process packets directly, we need to utilize the DIAG command.

In addition to the DIAG command, one may consider Android radio interface layer (RIL) to receive audio data. The problem with the RIL interface is that some mobile devices do not export incoming voice to the AP. Instead, it transfers incoming voice directly to a speaker. Because of this limitation, we chose to use the DIAG command.

❸ **RTP Tunneling Receiver** utilizes the DIAG interface. After the mobile device receives the DIAG command, RTP Tunneling Receiver starts receiving all network packets through the DIAG interface in the Android kernel. If the received packet is not corrupted and contains the identifier we set in the sending module, it accepts the packet. Finally, it extracts and sends data blocks to Data

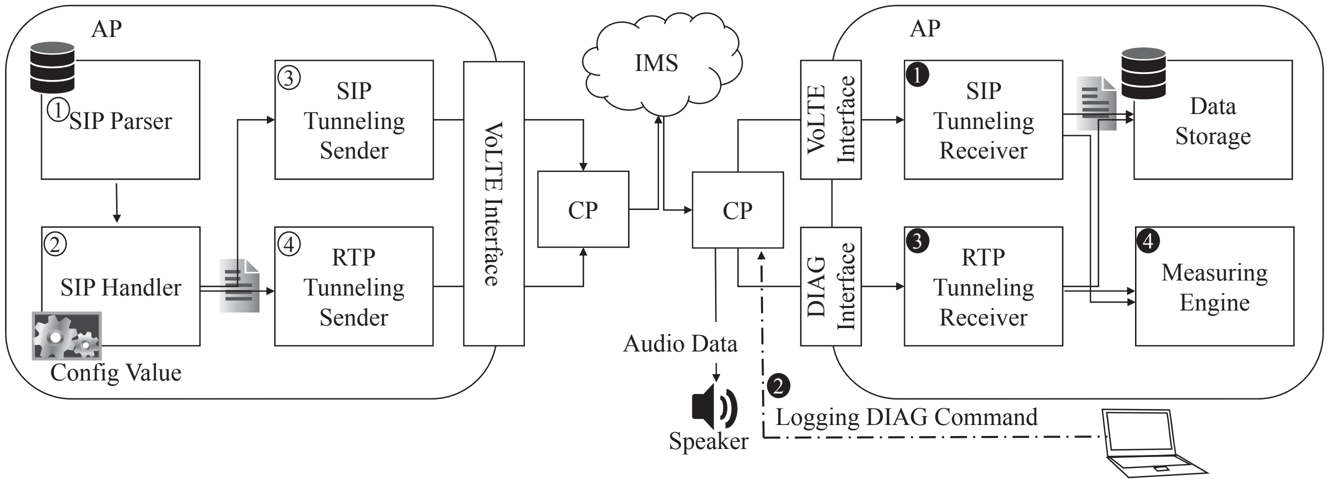

Figure 6: Diagram of Sending and Receiving Module

Storage while a sequence number and timestamp are passed to the Measuring Engine.

❹ **Measuring Engine** receives a sequence number, timestamp, and data size from the receivers to evaluate the network performance of our tunneling. Note that we did not measure the performance for the SIP tunneling since it might cause denial of service to the SIP servers in the IMS network (See §5 for more details.)

4.1.3 Challenges and Limitations

During our implementation, we encountered some challenges. First, many operators do not follow the specifications [3, 18] for either the flow or the structure of SIP messages. For example, some operators do not transfer `RINGING` or `SESSION PROGRESS` during call setup. Some operators even simply modify or remove header fields for their own purposes. Consequently, much work had to be done to ensure our sending module adjusts operators' individual VoLTE features and obtain results thereupon.

The second challenge is that at the receiver's side, the device automatically closes the calling session by sending a `BYE` message when it does not receive RTP packets for a certain period (typically 10 seconds). Therefore, we had to wrap the data blocks into RTP packets to keep the session alive.

Finally, the receiving module requires a mobile device to be connected with a laptop once to send the DIAG command. However, once logging setup is complete, the device does not need to be connected with the laptop until it is powered off. We discovered that the DIAG interface in the Android kernel does not accept DIAG commands. We tried to send the DIAG command from the kernel to eliminate the one time connection to a laptop, but it was not successful. In contrast, from our laptop, we could send DIAG commands through the USB connection. There might be a protecting mechanisms in the CP that blocks DIAG commands from the mobile device since they are usually sent from control software in a laptop.

4.2 Evaluation

4.2.1 Media Channel Properties

We first measured the characteristics of the media channel during the call as in Table 2. The experiment is conducted with OPTis-S[4] software from Innowireless. These media channel characteristics represent bearer information set by operators as well as the QoS parameter for bandwidth designated in the body of an `INVITE` message.

Table 2: Media channel characteristics

	US-1	US-2	KR-1	KR-2	KR-3
Qos Param. (Kbps)	38	49	41	41	49
Bandwidth (Kbps)	38/49	49	65	65	65
Latency (sec)	0.1	0.1	0.1	0.1	0.1
Loss rate (%)	1	1	1	1	1

Table 3: Feasibility of our accounting bypass attacks in each operator

	Hidden Channel	US-1	US-2	KR-1	KR-2	KR-3
VoLTE Call Service	SIP Tunneling	✓	✓	✓	✓	✓
	RTP Tunneling	✓	✓	✓	✓	✓
Direct Communication	Phone to Phone	✓	✘	✓	✘	✘
	Phone to Internet	✘	✓	✓	✘	△

When a mobile device establishes a media channel, the network sends a request for bearer creation with QoS information. We analyzed this request and extracted bandwidth, latency, and loss rate for the media channel. However, some operators do not specify this information in the message. In this case, we use the QoS class identifier (QCI) value in the message to identify the channel characteristics, as described in [1]. In most operators, as shown in Table 2, the bandwidth specified in the bearer request is different from the `INVITE` message.

4.2.2 Hidden Data Channel Measurements

We measured the network performance with the sending and receiving modules for the hidden data channel. The experiment was conducted on the same five operators as in §3. The feasibility of accounting bypass in each hidden data channel is shown in Table 3. The table indicates that if we can send data through a certain channel, it is not charged. In the case of SIP tunneling and RTP tunneling, all operators are open to free data transfer. However, the

[4] OPTis-S is a software that enables mobile device manufacturers to analyze control-plane messages.

Table 4: Measurement results of RTP tunneling in each target operator.

	US-1	US-2	KR-1	KR-2	KR-3
Throughput (Kbps)	37.90	36.93	45.76	39	50.48
Latency (sec)	0.52	0.02	0.10	0.32	0.30
Loss rate (%)	1.44	1.74	0.77	0.65	0.73

result of direct communication is different among operators. In case of US-1, for example, phone-to-phone communication is available while phone-to-internet access is prohibited. The triangle mark in KR-3 means that a free data channel is available for IPv4, but not for IPv6. Since direct communication originates from implementation flaws or P-GW blocking policy, it can vary among operators. Through a feasibility analysis, we found that phone-to-phone direct communication is available for operators that do not have a media proxy.

We also measured the actual network performance for each operators, which includes throughput, latency, and loss rate, as shown in Table 4. While the information of network performance is included in the bearer creation request, we conducted this experiment to derive the actual performance. Since transferred data are wrapped with a RTP header upon UDP, we added an additional header containing an identifier, sequence number, and timestamp. To measure throughput, we calculated received packet bytes per unit time. For latency, the receiver computes the time difference and delay with the first two packets to sync its time with the sender.

Note that in our hidden data channel, we can send data as fast as possible. However, since the bandwidth is limited, more packets will be dropped when we increase the throughput. Therefore, by fitting the loss rate to 1% (i.e. using the same loss rate as in Table 2) by delaying or varying the payload size, we can obtain the actual throughput.

For the experiment, we sent 200,000 packets and averaged the results. Since multiple SIP messages can damage SIP servers in the IMS network, we did not measure the performance for SIP tunneling. As can be seen in Table 4, the results are different from the media channel characteristics in §4.2.1. The discrepancy may originate from several factors: number of users, cellular network status, or signal strength to the cell tower.

We also measured the performance of direct communication: phone-to-phone and phone-to-internet. The best result of phone-to-phone communication was 16.84 Mbps in one of the Korean operators. In the case of phone-to-internet, the best result was 21.55 Mbps for the same operator. This high throughput comes from the bandwidth of the default bearer for VoLTE signaling being configured in the same manner as the default bearer for the data service. However, as described in §2.2, data transmission through the default bearer for VoLTE signaling has the highest priority. Therefore, if a malicious user utilizes the VoLTE default bearer for data transmission, she will be guaranteed higher performance than normal users.

5 Exploiting VoLTE Mis-implementation

In §3.3, we described several security problems that an adversary can exploit to carry out malicious behaviors/activities. In this section, we discuss possible attacks that could be exploited using the discovered vulnerabilities. Table 5 describes the vulnerabilities with possible attacks and discusses if they are exploitable in each operator.

5.1 Permission Model Mismatch

In our experiment, an application with only Internet access permission can make a call. In addition, this calling activity is not displayed on the screen; thus, a user may not know that her device was making a call. Consequently, if a malicious application is installed on a victim's device, an adversary can conduct a couple of attacks by exploiting this vulnerability.

Denial of Service on Call is an easy way to block a victim's phone. With a malicious application installed on the victim's phone, an attacker can cause the phone to make calls to designated number(s) repeatedly. This activity would result in the victim not being able to receive any incoming calls. Furthermore, because the call is not displayed on the screen, the victim would not be cognizant that her phone is in a calling state. Therefore, this can cause denial of service on calls to the victim.

Overbilling is another powerful attack. If a malicious application installed on a victim's phone can send an `INVITE` message that initiates an expensive video call, severe overbilling of the victim can occur.

The permission model mismatch problem shows that the current permission model used by mobile phones cannot handle the All-IP environment properly. In the case of 3G networks using a circuit-switching network for voice calls, call permission and data permission are completely separated.

5.2 Direct Communication

Direct communication is an implementation problem originating from the policy configured in P-GW. Even though direct phone-to-phone communication should be blocked because it can allow attacks such as overbilling attacks, some operators do not prevent this access in VoLTE.

Free Video Call is a useful application of direct communication. Because direct phone-to-phone communication bypasses the IMS network, and only goes through P-GW, one can directly send an `INVITE` message for a video call to another party. On the callee's side, the phone only replies to the source IP address of the received `INVITE`, and no SIP server is involved in the procedure. Further, because all accounting related to VoLTE calls is handled in the IMS network, by using this method one can talk to others without being charged. One requirement is that the sender has to open a microphone, but this does not pose a difficulty because the user can simply root the phone. In the case of Korea, video calls from operators cost 1.66 times the price of a voice call. In the case of the U.S., the operators charge for both data and voice.

Caller Spoofing is a severe issue related to direct communication. Because packets are only routed through P-GW, there is no authentication between the caller and the callee. Accordingly, one can send a manipulated `INVITE` to spoof the victim. If the adversary modifies the phone number in the `INVITE`, the modified number will be on the screen of the victim's phone. Therefore, the victim would believe that the call is from that number. As a result, an adversary can exploit direct phone-to-phone communication for voice phishing by simply changing a few bytes in the `INVITE`.

5.3 Lack of Authentication

Absence of authentication is one of the three threats originating from mis-implementation of the IMS network.

Caller Spoofing is also feasible even when SIP messages go through SIP servers. When an adversary generates a modified `INVITE` message and sends it to the SIP server, if the SIP server only checks if the phone number is valid, the server is vulnerable to caller spoofing. When this is successful, it can also cause a calling fee to be imposed on the person who owns the modified phone number.

Table 5: Threats and possible attacks in each operator

Weak Point	Threats	US-1	US-2	KR-1	KR-2	KR-3	Possible Attack
UE	Permission Mismatch	Vulnerable for all Android devices					Denial of Service on Call, Overbilling
P-GW	Direct Communication	✓	✘	✓	✘	✘	Free Video Call, Caller spoofing
IMS	No Authentication	✘	✘	✓	✓	✘	Caller spoofing
	No Session Management	✓	✓	✓	✘	✓	Denial of Service on Core network, Cellular P2P

Therefore, this attack can be considered a simple yet powerful attack. In fact, not only `INVITE`, but also a `BYE` message can be used for caller spoofing. If operators do not properly authenticate users, `BYE` messages can be transmitted to terminate the victim's on-going call.

We found that two operators in Korea are vulnerable to caller spoofing. Other operators prevent spoofing using either of the following two methods: verifying the caller's phone number with the IP address or with International Mobile Station Equipment Identity (IMEI), a unique identifier for mobile devices.

5.4 Lack of Session Management

Absence of session management is another issue of the IMS network. Since some operators do not manage call sessions, one can send multiple `INVITE` messages to the SIP server.

Denial of Service on Core network is a possible attack in this case. When the SIP server receives an `INVITE` message, it should open a session for each message and manage each session independently. If the number of `INVITE`s are too large, this can damage the SIP server and paralyze the IMS network for VoLTE service.

In general, a user can only call once at a time with the native calling app in the mobile phone. However, with our sending module, we can transmit virtually an unlimited number of `INVITE` messages. In our analysis, if UE-A sends `INVITE`, the dedicated bearers among UE-A, the P-GW, and UE-B are all established, even when UE-B does not answer the call. Since the cost of the bearer activation and release procedure is expensive among control-plane procedures, multiple `INVITE`s can overload the P-GW, causing a denial of service on the core network.

We conducted our experiment only sending 2-4 `INVITE` messages, and checked whether a call session for each message is created. We found that except for one operator in Korea, all the other operators in the experiment are vulnerable. The most important aspect of our attack is that we can commit a denial of service attack with only one mobile device whereas such an attack usually requires a huge number of bots. Of course, we were not able to verify the cost of each bearer in the operator, or whether this attack actually shuts down the SIP server.

Cellular P2P is a more complicated application but is still feasible. Since there is no session management, users can send multiple INVITE messages to create several call sessions. When multiple call sessions are established among users, people can share files as torrents through RTP tunneling. Even though the 1% loss rate is high, we can utilize the reliable UDP protocol, as in [9]. The throughput is adequate and there is still enough speed to transfer files as a torrent because there are many peers in the cellular network, and these peers do not usually shutdown their phones. Therefore, people can share movies or other content when they are sleeping. If implementation is seriously concerned, Cellular Tor can also be available to evade censorship on the cellular networks.

6 Countermeasures

In this section, we discuss solutions for both free hidden channels and VoLTE service abuse attacks. Since these solutions are quite intuitive, they can be easily applied to the commercial cellular networks. Some attack vectors originate from mis-implementation of operators, while others are derived from fundamental problems of the VoLTE system. These fundamental problems are more difficult to obviate than the other attacks. To prevent these attacks, we suggest more difficult yet comprehensive solutions for securing the overall VoLTE service.

6.1 Immediate Solution

Filtering. The main cause of direct communication is inappropriate access control of user-initiated requests at P-GWs in cellular networks. The purpose of the VoLTE default bearer is call signaling; therefore, the P-GW should filter out all other packets except SIP messages. In our analysis, however, some operators do not follow the standard for call related services. For example, one operator in Korea provides conference calls using their proprietary protocols on top of HTTP although an IETF standard [21] provides a conference call solution using the SIP protocol. This inconsistency of service implementation can create difficulties in correctly managing the access control in the P-GW.

In addition, operators should block all packets travelling directly from a UE to another UE, and only allow packets from UEs to the SIP server/media proxies, and vice versa. As a result, free data channels as well as free video calls and caller spoofing could be blocked.

Strict Session Management. This is another requirement for securing VoLTE. SIP tunneling, denial of service, and cellular p2p are all possible attacks resulting from the absence of session management. If a SIP server carefully inspects SIP messages originating from UEs, it can block SIP tunneling. For example, it can check whether an invalid field or content exist in the header and the payload. If the result of the check is unsatisfactory, it should reject the requests and respond to UEs with an error message.

Further, to protect against SIP tunneling and denial of service attacks, operators should limit the number of SIP messages from a UE within a certain period. If an adversary exceeds a certain threshold, operators should block the adversary and inspect her activities. Furthermore, the SIP server should check if a UE is already calling another UE. If so, it should block other call originating messages (i.e. `INVITE` messages). This policy can detect and prevent other attacks such as cellular p2p definitely, and intuitively because a prerequisite of these attacks is that an adversary should be able to send multiple `INVITE` messages.

UE Verification. To prevent call spoofing, SIP servers should verify the source of SIP messages. One possible way to facilitate this is to add a UE's unique identity such as IMEI to SIP messages. SIP servers would then be able to cross-check the unique identity with the phone number. Unless a unique identity is stored securely in the mobile device, an adversary can easily intercept this information by remotely installing a malicious application.

Another solution is to bind the phone number with the IP address of a UE. The operators can validate the user-initiated SIP messages by checking UE's IP header with the parameters (such as IP address, phone number, and device unique identity) in SIP messages because this information is already stored in the operators' server. IP spoofing is also possible, but we found that all the operators we experimented with have an IP spoofing prevention mechanism for the data interface. Even though we have not tested IP spoofing for the VoLTE interface, the operators might have already installed a similar mechanism in VoLTE. Therefore, cross-checking is a strong yet easy to implement solution.

However, it cannot prevent caller spoofing in cases where an adversary spoofs the victim's IMS registration procedure, such that the SIP server stores spoofed parameters. A UE registers itself to a SIP server by exchanging REGISTER messages, and the SIP server stores several parameters that can identify the UE. Thus, if an adversary can spoof the registration, she can obtain the full permission of the victim's phone address.

Deep Packet Inspection (DPI). Since the RTP tunneling exploits the media dedicated bearer during a VoLTE call, operators can recognize if a user is utilizing the media channel through traffic monitoring. This monitoring job can be done by applying a deep packet inspection (DPI) solution in P-GWs or media proxies. However, if an adversary disguises the data into voice-like traffic, then the DPI solution will not be sufficient to prevent it.

Accounting Policy. All the free channels we exploited in this paper resulted from time-based accounting. This problem can be resolved by changing the accounting policy of VoLTE service to a byte usage-based scheme. Of course, even in this case, an adversary can still exploit a voice channel. However, she cannot bypass the accounting. This seems quite natural, but it would be difficult for the operators to change the current time-based policy since it is deeply related to the revenue. 70% of total revenue in the operators is still from voice and SMS [12].

6.2 A Long-term and Comprehensive Solution

Permission model mismatch is a severe problem that is prevalent in current VoLTE-compatible mobile devices. Unlike the previous call mechanism, as VoLTE is IP-based, the current permission model of mobile devices cannot handle it. Strictly binding sockets from any application to the data interface would be one way to prevent it. In this way, SIP packets from an application cannot reach the SIP server. The operators, in the same manner, should block packets from the data interface. One limitation of this solution is that deploying the solution would not be easy since all the firmware of mobile devices should be updated. Furthermore, data encryption (e.g. IPsec or TLS for signaling, and sRTP for media data) should be deployed as specified in the 3GPP specifications [4, 5].

However, even with strict binding and encryption, an adversary can still utilize tunneling since she has all permission for her phone. Another way to resolve the problem is to process both call signaling and voice data transmission in the CP as a traditional circuit-switching call does. The CP only allows legitimate call related requests and blocks all other packets utilizing the VoLTE interface from the AP. Thus, an adversary cannot send manipulated packets through the VoLTE interface. Furthermore, protecting the CP with hardware security modules such as TrustZone or secure storage [6] may also be required to prevent the adversary from intercepting SIP messages in the CP.

7 Related Work

Accounting Issues in Cellular Network. Several research groups have studied cellular accounting issues [14, 15, 25–27, 32]. There are two main attacks related to accounting: accounting bypass and overbilling.

Peng et al. demonstrated DNS port abuse for free-data [25]. In [26], the authors uncovered an accounting bypass by source IP address spoofing because a mobile data charging system was only based on the packet header. Go et al. wrapped their payload in a TCP retransmission packet and utilized some ISPs that did not charge a fee to ensure fairness [14, 15].

In addition to accounting bypass, overbilling attacks, also occur. Go et al. pointed out that TCP retransmission could also be used to impose a data fee on a victim. In [25], a large quantity of spam data was sent to a victim using VoIP and a malicious phishing link. The unfair billing practices of some operators were illuminated in [27, 32].

While much research has been conducted on accounting bypass and overbilling, our work is fundamentally different in terms of the interface we used. Previous works only covered accounting issues related to the data interface, whereas our work is focused on the VoLTE interface. Furthermore, previous research studies assumed that the overbilling attacks were committed by first installing a malicious application on the victim's phone. In contrast, in our call spoofing attack, the adversary only needs her own device.

VoIP Tunneling for Censorship Resistance. Much research has been conducted on VoIP tunneling, with a specific focus on avoidance of censorship. In general, VoIP services such as Skype are widely used as tunneling protocols. Skypemorph sends Tor traffic through UDP ports via the Skype video call channel [22]. In [33], the authors utilized RTP downstream to avoid censorship. In the case of Freewave, data are converted to acoustic signal data and loaded into normal VoIP packets to hide the data [19].

Our work also utilizes a tunneling protocol and data concealment. However, our focus is on accounting bypass as well as discovery of mis-implementation problems. In other words, the main goal of our work for tunneling is different from that of previous works. In fact, in contrast to previous works, which applied tunneling on the Internet, our work is the first to apply tunneling in cellular networks.

DoS Attack on Cellular Network. Various DoS attacks in cellular networks have also been investigated. In general, most research on DoS attacks has been related to the GSM Network [11, 16, 23, 28, 30]. Enck et al. suggested that sending SMS to certain phone numbers compiled by a hit-list would massively affect the cellular core network [11]. Traynor et al. [30] demonstrated that degrading the cellular network service was possible with a cellular botnet. Mulliner et al. [23] described how malformed SMS messages could force mobile phones to be rebooted, and would finally overload the network. Golde et al. [16] introduced a DoS attack because of mis-authentication in the GSM network. Traynor et al. [31] presented DoS attacks exploiting the setup and teardown process of the radio interface in the GPRS/EDGE network. The UMTS network is still vulnerable to a targeted DoS. While Enck et al. utilized a phone number on the GSM network, Qian et al. made a hit-list by fingerprinting IP addresses to carry out a targeted DoS on the core network of UMTS [28].

In contrast to these attacks on GSM networks, the DoS attack proposed in this paper is against the VoLTE network, and can be initiated by a significantly small number of mobile devices.

8 Future Work and Conclusion

We conducted an analysis of the VoLTE network of five operators in the United States and South Korea. To the best of our knowledge, we are the first to analyze security issues on the VoLTE network. The key contribution of our study is that VoLTE transition is not simple, but involves the EPC core (3GPP standards), OS support at UE, hardware interface redesign, and cellular accounting policies. The main lesson is that it negates the common belief that VoLTE is a simple transition because VoIP is well understood on the Internet. In contrast, it shows that architectural aspects of cellular networks make the problem much more complex. Although we discovered a few implementation bugs that are easy to fix, the core problem is complicated processes, involving accounting, access control, session management, and EPC-UE interaction. This is evidenced by the response from ISPs, Android, and US/KR CERTs to our responsive disclosure. It requires greater attention because a systematic security analysis of new architecture is always necessary to make the architecture robust.

In this paper, we considered security issues and possible attacks related to VoLTE call service after legitimate IMS registration. However, an attacker can also utilize a SIP `REGISTER` message to perform other attacks. If there are vulnerabilities in the registration phase, an attacker can control all access to a victim's VoLTE service. For example, she can carry out an imposter attack or even wiretapping. We plan to investigate scenarios such as this in future work. In this work, we concentrated on the problems and vulnerabilities discovered in five operators; however, more problems and vulnerabilities may be present in these and other operators. As more and more operators provide VoLTE services, it is essential that more security analyses be conducted on VoLTE networks.

9 Acknowledgment

We thank the anonymous reviewers for their helpful feedback. This research was supported by (1) the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2015-R0992-15-1006) supervised by the IITP (Institute for Information & communications Technology Promotion), and (2) Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the MSIP (No. NRF-2013R1A1A1076024). Yeongjin Jang and Taesoo Kim were partly supported by the NSF DGE-1500084, ONR N000141512162, and DARPA-15-15-TC-FP-006.

10 References

[1] 3GPP. ETSI TS 23.203. Policy and charging control architecture, 2012.

[2] 3GPP. ETSI TS 23.221. Architectural requirements, 2011.

[3] 3GPP. ETSI TS 23.228. IP Multimedia Subsystem (IMS) Stage 2, 2011.

[4] 3GPP. ETSI TS 33.203. Access security for IP-based services, 2011.

[5] 3GPP. ETSI TS 33.210. Network Domain Security (NDS); IP network layer security, 2011.

[6] T. Alves and D. Felton. Trustzone: Integrated hardware and software security. *ARM white paper*, 3(4):18–24, 2004.

[7] J. Arkko, G. Camarillo, A. Niemi, T. Haukka, and V. Torvinen. Security mechanism agreement for the session initiation protocol (SIP), 2003.

[8] J. Beekman and C. Thompson. Breaking Cell Phone Authentication: Vulnerabilities in AKA, IMS, and Android. In *WOOT*, 2013.

[9] T. Bova and T. Krivoruchka. Reliable UDP protocol. *draft-ietf-sigtran-reliable-udp-00.txt*, 1999.

[10] G. Delugre. Reverse engineering a Qualcomm baseband. *CCC*, 2011.

[11] W. Enck, P. Traynor, P. McDaniel, and T. La Porta. Exploiting Open Functionality in SMS-Capable Cellular Networks. In *Proceedings of the 12th ACM conference on Computer and communications security*, pages 393–404. ACM, 2005.

[12] Ericsson. What is voice over LTE?, January 2013.

[13] Global mobile Suppliers Association and others. Evolution to LTE report, 2015. [Online; accessed 11-May-2015].

[14] Y. Go, E. Jeong, J. Won, Y. Kim, D. F. Kune, and K. Park. Gaining Control of Cellular Traffic Accounting by Spurious TCP Retransmission. In *Proceeding of the Network and Distributed System Security Symposium (NDSS)*, 2014.

[15] Y. Go, D. F. Kune, S. Woo, K. Park, and Y. Kim. Towards Accurate Accounting of Cellular Data for TCP Retransmission. In *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*, page 2. ACM, 2013.

[16] N. Golde, K. Redon, and J.-P. Seifert. Let Me Answer That for You: Exploiting Broadcast Information in Cellular Networks. In *Proceedings of the 22nd USENIX conference on Security*, pages 33–48. USENIX Association, 2013.

[17] GSM Association. Voice and Video calls over LTE. [Online; accessed 14-May-2015].

[18] GSM Association. VoLTE Service Description and Implementation Guidelines, Version 1.1, 2014.

[19] A. Houmansadr, T. J. Riedl, N. Borisov, and A. C. Singer. I want my voice to be heard: IP over Voice-over-IP for Unobservable Censorship Circumvention. In *NDSS*, 2013.

[20] IDATE. in World LTE market, 2014. [Online; accessed 11-May-2015].

[21] A. Johnston and O. Levin. Session Initiation Protocol (SIP) Call Control-Conferencing for User Agents, 2006.

[22] H. Mohajeri Moghaddam, B. Li, M. Derakhshani, and I. Goldberg. Skypemorph: Protocol Obfuscation for Tor Bridges. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 97–108. ACM, 2012.

[23] C. Mulliner, N. Golde, and J.-P. Seifert. SMS of Death: From Analyzing to Attacking Mobile Phones on a Large Scale. In *USENIX Security Symposium*, 2011.

[24] F. Özavci. VOIP Wars: Return of the SIP, 2013.

[25] C. Peng, C.-y. Li, G.-H. Tu, S. Lu, and L. Zhang. Mobile Data Charging: New Attacks and Countermeasures. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 195–204. ACM, 2012.

[26] C. Peng, C.-Y. Li, H. Wang, G.-H. Tu, and S. Lu. Real Threats to Your Data Bills: Security Loopholes and Defenses in Mobile Data Charging. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 727–738. ACM, 2014.

[27] C. Peng, G.-h. Tu, C.-y. Li, and S. Lu. Can We Pay for What We Get in 3G Data Access? In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 113–124. ACM, 2012.

[28] Z. Qian, Z. Wang, Q. Xu, Z. M. Mao, M. Zhang, and Y.-M. Wang. You Can Run, but You Can't Hide: Exposing Network Location for Targeted DoS Attacks in Cellular Networks. In *NDSS*, 2012.

[29] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, et al. SIP: session initiation protocol, 2002.

[30] P. Traynor, M. Lin, M. Ongtang, V. Rao, T. Jaeger, P. McDaniel, and T. La Porta. On Cellular Botnets: Measuring the Impact of Malicious Devices on a Cellular Network Core.

In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 223–234. ACM, 2009.

[31] P. Traynor, P. McDaniel, T. La Porta, et al. On Attack Causality in Internet-Connected Cellular Networks. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, pages 1–16. USENIX Association, 2007.

[32] G.-H. Tu, C. Peng, C.-Y. Li, X. Ma, H. Wang, T. Wang, and S. Lu. Accounting for Roaming Users on Mobile Data Access: Issues and Root Causes. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 305–318. ACM, 2013.

[33] Q. Wang, X. Gong, G. T. Nguyen, A. Houmansadr, and N. Borisov. Censorspoofer: Asymmetric Communication using Ip Spoofing for Censorship-Resistant Web Browsing. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 121–132. ACM, 2012.

[34] Z. Wang. IMS Security Framework. *3GPP2 S. S0086-B, Version*, 2, 2008.

[35] R. Zhang, X. Wang, R. Farley, X. Yang, and X. Jiang. On the feasibility of launching the man-in-the-middle attacks on VoIP from remote attackers. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, pages 61–69. ACM, 2009.

[36] R. Zhang, X. Wang, X. Yang, and X. Jiang. Billing Attacks on SIP-Based VoIP Systems. *WOOT*, 7:1–8, 2007.