Experimental Study of Spoofing Alerts in 4G LTE Networks

Gyuhong Lee

University of Colorado at Boulder, caixy6464@gmail.com

Follow this and additional works at: https://scholar.colorado.edu/csci_gradetds

Part of the Computer Sciences Commons

Recommended Citation
Lee, Gyuhong, "Experimental Study of Spoofing Alerts in 4G LTE Networks" (2019). Computer Science Graduate Theses & Dissertations. 204.
https://scholar.colorado.edu/csci_gradetds/204

This Thesis is brought to you for free and open access by Computer Science at CU Scholar. It has been accepted for inclusion in Computer Science Graduate Theses & Dissertations by an authorized administrator of CU Scholar. For more information, please contact cuscholaradmin@colorado.edu.
Experimental Study of Spoofing Alerts in 4G LTE Networks

by

Gyuhong Lee

B.S., Korea Military Academy, 2011

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Department of Computer Science

2019
This thesis entitled:
Experimental Study of Spoofing Alerts in 4G LTE Networks
written by Gyuhong Lee
has been approved for the Department of Computer Science

Prof. Sangtae Ha

Prof. Dirk Grunwald

Prof. Eric Wustrow

Date ________________

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.
Abstract

Modern mobile phones should receive and display alert messages, such as AMBER alerts, emergency alerts, and (unblockable) presidential alerts, under the mandate of the Warning, Alert, and Response Act of 2006.

In January 2018, an alert message that a ballistic missile was heading to Hawaii had been sent out in Hawaii state. Although it turned out to be caused by human error, it showed how this alert message could affect people. In addition, in October 2018, a test to broadcast unblockable presidential alert messages to all of the United States was conducted. These cases have raised concerns about what problems might arise when alert messages are misused or spoofed. In this thesis, we investigate the alert spoofing attacks in 4G LTE networks and provide 1) the detailed analysis of this attack, 2) design and implementation of such systems, 3) possible counter measures, and most importantly 4) the impact of such attacks in both indoor and outdoor environments.

Our attack can be carried out using a software defined radio, and our modifications to the open source NextEPC and srsLTE software libraries. We evaluate our attack in a controlled environment and estimate the impact of such an attack based on empirical measurements. We find that with only four malicious portable base stations of a single Watt of transmit power each, almost all of the 50,000-seat stadium can be attacked with 90% success rate. The true impact of such an attack would, of course, depend on the density of cell phones in the range: false alerts in crowded cities or stadiums could potentially result in cascades of panic.

Solving this problem will require large collaborative efforts among government, standardization organizations, network providers, and manufacturers. We also provide possible solutions to defend against such a spoofing attack.
Dedication

For my beautiful and glorious homeland, Republic of Korea, which I will forever devote my loyalty to.

For the Republic of Korea Ministry of National Defense and the Army Headquarter, supporting my study in the United States.

For my beloved parents, Yongmo Lee and Yunja Lee, and sister, Jieun Lee.
Acknowledgements

Parts of this thesis has been accepted to the following:


First of all, I would like to express my sincere appreciation and respect to Prof. Sangtae Ha, who has provided me with motivation, extensive knowledge and continuous support for related research. His guidance has always been helpful in the study and writing of this thesis. I could not imagine having a better mentor and teacher in my master’s course.

Besides my advisor, I extend my gratitude to the rest of my thesis committee: Profs. Dirk Grunwald and Eric Wustrow. Their comments and advice during the study have been a great help in improving the quality of this thesis.

Also, I express my special thanks to Dr. Jihoon Lee. I was able to learn so much from him on LTE systems. This research would not be possible without his help.

My sincere thanks also go to my lab mates, Dr. Jingsung Lee, Dr. Youngbin Im, Mr. Jaeyoung Oh, Mr. Insoo Lee, Ms. Hyoyoung Lim, Mr. Ibrahim Ayad, Ms. Parisa Rahimzadeh, Mr. Prasanth Prahladan, Mr. Siqi Chen, Mr. Zhang Liu, Mr. Sandesh Dhawaskar and Mr. Max Hollingsworth. Thanks to them, I was able to spend my tough master’s course pleasantly.
Contents

Chapter

1 Introduction 1

2 Background 7

3 Preliminary Investigation 10
   3.1 Building a CMAS-enabled LTE Testbed 10
   3.2 Identifying the Vulnerability 11
   3.3 Cases for CMAS Reception and Trust 14
   3.4 Impact of Signal Strength 17

4 Proof-of-concept Attacks 20

5 Evaluation 25
   5.1 Experimental Setting 25
   5.2 Success Rate 26
   5.3 Attack Coverage 27
   5.4 Practical Scenarios: Indoor and Outdoor 28

6 Potential Defenses 34
   6.1 Prevention: Digital Signature 35
   6.2 Mitigation 37
      6.2.1 Accepting only secure CMAS 37
Tables

Table

3.1 CBC configuration .............................................. 12
3.2 Cases for CMAS Reception and Trust .......................... 16
4.1 HW and SW systems used for the implementation ............ 20
4.2 Spoofing attack preparation .................................... 21
## Figures

Figure

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>A snapshot of false alert sent in Hawaii on Jan. 13, 2018</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>A snapshot of the first US nationwide test of a presidential alert performed on Oct. 3, 2018</td>
<td>2</td>
</tr>
<tr>
<td>1.3</td>
<td>Part of the captured SIB12 of the presidential alert on Oct. 3, 2018</td>
<td>5</td>
</tr>
<tr>
<td>1.4</td>
<td>Decoded message of the presidential alert on Oct. 3, 2018</td>
<td>6</td>
</tr>
<tr>
<td>2.1</td>
<td>LTE CMAS network architecture</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>CMAS call flow</td>
<td>9</td>
</tr>
<tr>
<td>2.3</td>
<td>Government alerts setting in mobile phones: (a) Android phones and (b) Apple's iPhones. Users have no way to disable or block Presidential alerts while they can manually disable AMBER and emergency alerts.</td>
<td>9</td>
</tr>
<tr>
<td>3.1</td>
<td>LTE testbed setup for a mobile phone to receive a CMAS notification</td>
<td>11</td>
</tr>
<tr>
<td>3.2</td>
<td>SIAP log messages of authentication failure: after five consecutive authentication failures, which takes about 40 seconds, the UE starts to search for a new cell in other frequency channels.</td>
<td>13</td>
</tr>
<tr>
<td>3.3</td>
<td>SIAP log message of a RLF: this UE was Active, but it becomes disconnected due to a sudden RLF</td>
<td>15</td>
</tr>
</tbody>
</table>
3.4 The Idle/Active life-cycle of a UE. The state of UE is moving counter clockwise.

CMAS spoofing is possible while the UE performs a cell search, prior to successful authentication with a trusted eNodeB.

4.1 Our Presidential Alert Spoofer scans a cell, gathers operator information, and sends a fake Presidential Alert to both idle and active UEs who are working on FDD/TDD frequencies in the US and Canada. It consists of one SDR device, one COTS LTE eNodeB, and 2 laptops.

4.2 S1AP log messages: our modification on NextEPC provides an interface to inject a user-defined message as a Presidential Alert.

4.3 Received multiple fake Presidential Alerts using Samsung Galaxy S8(left), and Apple iPhone X(right).

5.1 Testbed setup for evaluating the attack success rate. The transmission power levels of SecureNet eNodeB and Presidential Alert Spoofer can be controlled independently.

5.2 CDF as a function of $\alpha$. Because the cell reselection can be done when idle UEs wake up, the spoofing attack is much easier when a different PCI value is used.

5.3 CDF as a function of $\beta$. Using the same PCI value leads to more decoding errors observed at the receiver, which results in slightly more effective attacks.

5.4 Maximum attack distance between the UE and the malicious eNodeB according to RSRP value of the UE from the SecureNet eNodeB, when the transmit power of the malicious eNodeB is 0.1 Watt.

5.5 Indoor attack coverage measurements. (a) The outdoor RSRP distribution of AT&T is measured at Folsom Field, University of Colorado Boulder. (b) The attack coverage is simulated when $4 \times 1$-Watt malicious eNodeBs are located around the stadium: 90% of success rate can be achieved for idle UEs.
5.6 Outdoor attack coverage simulation. (a) The outdoor RSRP distribution of AT&T is measured at Folsom Field, University of Colorado Boulder. (b) The attack coverage is simulated when $4 \times 1$-Watt malicious eNodeBs are located around the stadium: 90% of success rate can be achieved for idle UEs.

6.1 Working procedure of digital signature implementation ........................................ 42
6.2 Network Architecture of eMBMS ........................................................................ 43
6.3 Security Architecture of eMBMS ......................................................................... 44
6.4 Implementation of eMBMS .................................................................................. 45
6.5 Reception Patterns of SIBs from Commercial eNodeBs ....................................... 46
6.6 Reception Patterns of SIBs from Malicious eNodeBs ........................................... 47
Chapter 1

Introduction

On Jan. 13, 2018, the Hawaiians received an emergency alert message like Figure 1.1. A correction message was not sent for about 38 minutes, and the Hawaiians had to go through chaos, scrambling to find shelters [66]. Although it turned out that this incident was caused by a message sent by human error, We thought this was a severe matter and came up with this concern: Can someone maliciously send alert messages to anyone? How can this attack be carried out? And how to prevent it?

Can we hack the servers that sent alert messages? This attack, however, is practically almost impossible. This is because servers installed in government agencies and others are well protected with sufficiently secured solutions. The core part of the cellular network is very difficult to attack, so what about attacking just the radio network part – between User Equipment (UE) and the cell tower? Of course, this attack is localized, but it would be very dangerous and enormous if it was
carried out in crowded places such as stadiums, theaters, shopping malls and subways.

In the US, the process of sending alert messages over the cellular network is officially called the Wireless Emergency Alerts (WEA) program, a government-mandated service in commercialized cellular networks. WEA was established by the Federal Communications Commission (FCC) in response to the Warning, Alert, and Response Act of 2006 to allow wireless cellular service providers to send geographically targeted emergency alerts to their subscribers. The Federal Emergency Management Agency (FEMA) is responsible for the implementation and administration of a major component of WEA called the Integrated Public Alert and Warnings System (IPAWS) \[62\]. IPAWS enables authorized public safety officials to send 90-character, geographically-targeted alerts to the public via commercial mobile service providers (CMSPs) \[60\].

This system can send three types of alerts: Presidential Alerts issued by the president to all of the United States; Imminent Threat Alerts involving serious threats to life and property, often related to severe weather; and AMBER Alerts regarding missing or abducted children. Considering the number of cell phone users and the nation-wide coverage of cellular networks, WEA over LTE was a natural step to enhance public safety immediately and effectively. In fact, recent rapidly moving fires have caused Emergency Services to consider using WEA instead of relying on opt-in alerting systems \[47\].

Figure 1.2: A snapshot of the first US nationwide test of a presidential alert performed on Oct. 3, 2018

The WEA sends alerts via the commercial mobile alert service (CMAS) standard \[1\]. These

\[1\] For clarity, we will use WEA to refer to the alert service and CMAS to refer to the underlying delivery technology.
alerts are delivered via the LTE downlink within broadcast messages, called System Information Block (SIB) messages. The SIB is broadcast by the cell to every UE that is tuned to the control channels of that cell. A UE obtains necessary access information, like the network identifier and access restrictions, from SIB messages and uses it for the cell selection procedure. Among the 26 different types of SIB messages, SIB12 contains the CMAS notification, which delivers the aforementioned alert messages to the UEs (greater detail in Chapter 2). On Oct. 3, 2018, the first national test of a mandatory Presidential alert was sent to all capable phones as shown in Figure 1.2. And the SIB12 of this message was successfully captured using the Mobileinsight [44] as shown in Figure 1.3. The SIB12 is not encrypted at all and the contents of message are encoded in GSM-7bit [2], but not encrypted as shown in Figure 1.4. This means that it’s very easy to create a fake SIB12, and anyone can’t determine the authenticity of the message.

The eNodeB broadcasts SIB messages to the UE, independently from the mutual authentication procedure that eventually occurs between them. Thus, all SIBs, including CMAS, are intrinsically vulnerable to spoofing from a malicious eNodeB. More importantly, even if the UE has completed its authentication and securely communicates with a trusted eNodeB, the UE is still exposed to the security threat caused by the broadcasts from other, possibly malicious, eNodeBs. This is due to the fact that the UE periodically gathers SIB information from neighboring eNodeBs for potential eNodeB (re)selection and handover.

The UE’s connection state can be classified into active and idle modes, depending on where the UE falls in the cell attachment process. Based on the UE’s state, we analyze its vulnerability to the spoofing attack. Next, we develop the attack model from our novel analysis and implement the CMAS spoofing attack system using commercial off-the-shelf (COTS) software defined radio (SDR) hardware and modified open source srsLTE [33] and NextEPC [48] software libraries. We then evaluate those attacks in a responsible and controlled manner: all tested phones are put into a radio-isolated shield box and the signal emitted by our malicious eNodeB is completely isolated to the outside. To the best of our knowledge, this is the first experimentally verified work that discloses the potential risk of CMAS spoofing. Note that LTE networks currently deployed in most countries
(e.g., countries in Europe, United States, and South Korea) have adopted public warning systems that follow the same architecture principles as CMAS [16], making them potential targets for the same attack.

We found via both experiment and simulation that a 90% success rate can be reached in 4,435\(m^2\) of a 16,859\(m^2\) building using a single malicious eNodeB of 0.1 Watt power, while in an outdoor stadium, 49,300 seats among the total 50,000 are hit with an attack, which itself has a 90% success rate using four malicious eNodeBs of 1 Watt power.

In summary, we make following major contributions:

- We identify security vulnerabilities of the WEA system and explain the detailed underlying mechanism stipulated by the LTE standard. We find that the CMAS spoofing attack is easy to perform but is challenging to defend in practice.

- We present our threat analysis on the CMAS spoofing attack, and implement an effective attack system using COTS SDR hardware and open source LTE software.

- We confirm that the CMAS spoofing attack can succeed in all 9 of the smartphones (from 5 manufacturers) that we tested.

- We evaluate our attack system using both SDR-based hardware prototype and measurement-based simulation. As one of the striking results, we demonstrate that four SDR-based malicious eNodeBs at 1-Watt of power can propagate their signal to 49,300 of the total 50,000-seat football stadium. Of the 49,300 seats affected, 90% will receive the CMAS message.

- We discuss possible solutions to prevent such a spoofing attack with thorough analysis and feasibility test, which can open the door toward collaborative efforts between cellular operators, government stakeholders, and phone manufacturers.
Figure 1.3: Part of the captured SIB12 of the presidential alert on Oct. 3, 2018
Figure 1.4: Decoded message of the presidential alert on Oct. 3, 2018
Here we review the background of the WEA service in the United States, whose underlying delivery architecture has been adopted in many other countries [16], and describe how these alert messages are delivered to a UE over the LTE network.

The 3GPP standardization body began a project in 2006 to define the requirements of CMAS in order to deliver WEA messages in the LTE network. The resulting technical specification, initially released in 2009, describes general criteria for the delivery of alerts, message formats, and functionality of CMAS-capable UEs [3].

Figure 2.1 illustrates the LTE CMAS network architecture. During an emergency, authorized public safety officials send alert messages to Federal Alert Gateways. The participating CMSPs
then broadcast the alert to the UEs, who will automatically receive the alert if they are located in or travel to the targeted geographic area. The cell broadcast center (CBC) is part of the core network and connected to the Mobility Management Entity (MME) which maintains the location information of the UEs attached to the network [7]. The eNodeB is the final step in communicating the alert to the UEs over the air.

Figure 2.2 shows a more detailed call flow of the CMAS procedure. An authorized official originates the alert with FEMA-approved alert origination software (step 1). The CMSP Gateway delivers emergency information to the CBC (steps 2 and 3), and the CBC performs geo-targeting which selects the eNodeBs where the alert is to be delivered (step 4) [6]. The CBC then identifies which MMEs need to be contacted and sends a Write-Replace Warning Request message containing the alert message and delivery attributes: Message Identifier, Serial Number, Tracking Area ID list, Warning Area, etc. (step 5) [3]. The MME sends a Write-Replace Warning Confirm message that indicates to the CBC that the MME has started to distribute the warning message to eNodeBs (step 6). If an eNodeB receives the Write-Replace Warning Request from its MME (step 7), it replies with a Write-Replace Warning Response message (step 8). A duplicated request can be detected by inspects the Serial Number at the eNodeB. If it is identified as a new alert, the eNodeB sends a paging signal with a CMAS indication to wake up all UEs in idle mode (step 9). The alert message is broadcast via a SIB12 message over the air (step 10) [13], and finally all UEs will receive the alert, irrespective of whether their connection state is either idle or active (step 11).

Among the three types of emergency alerts listed in Chapter 1, UEs may choose to turn off the notification of emergency alerts and AMBER alerts. However, the 3GPP mandated that the reception of Presidential Alerts is obligatory. Thus, cell phones have no option to disable Presidential Alert, as seen in Figure 2.3.

Because it cannot be disabled, this paper focuses on spoofing Presidential Alerts. Moreover, the attack can be performed without involving any of the IPAWS architecture or protocol described above. Instead, the attack begins with the injection of a fake CMAS message at the wireless stage from a rouge eNodeB (steps 9 and 10 in Figure 2.2).
Figure 2.2: CMAS call flow.

Figure 2.3: Government alerts setting in mobile phones: (a) Android phones and (b) Apple’s iPhones. Users have no way to disable or block Presidential alerts while they can manually disable AMBER and emergency alerts.
In this chapter we describe our end-to-end LTE CMAS testbed, as well as the current security issues discovered using our testbed. Next, we explain the threat methodology depending on the UE state and further derive the threat impact as a function of the signal power from a malicious eNodeB relative to the signal power from a trusted eNodeB.

### 3.1 Building a CMAS-enabled LTE Testbed

To analyze the specifics of CMAS alerts in detail, we built an LTE network in our lab and augmented it with CMAS capabilities.

The testbed, shown in Figure 3.1, consists of UEs, eNodeB, and MME. For the eNodeB, we use a COTS LTE small cell, Juni JL620 [37], which supports CMAS and 2x2 MIMO. We located the UEs inside an RF-isolated shield box, which prevents our experiments from unintentionally interfering with real devices. Our UEs communicate with the eNodeB via a pair of antennas also inside the box. For the LTE core including MME, we use the open source NextEPC [48] software.

In the testbed, the alert process starts with the CBC. The CBC is the go-between for the alert originator and the LTE core, or Evolved Packet Core (EPC). We implemented our own CBC with a REST (Representational State Transfer) API and changes to the EPC. Our implementation provides a number of configurable items summarized in Table 3.1.
3.2 Identifying the Vulnerability

An eNodeB broadcasts LTE system information through the Master Information Block (MIB) and SIB. Specifically, when a UE searches for an eNodeB, it searches for the eNodeB’s physical cell identifier (PCI) within a dedicated synchronization channel specified by the LTE standard [11]. After finding the PCI, the UE unscrambles the MIB which contains essential information such as the system bandwidth, system frame number (SFN), and antenna configuration in order to decode the SIB Type 1 message (SIB1). There are several SIB messages but only SIB1 has a fixed periodicity of 80 msec. Other SIB messages are dynamically scheduled by the eNodeB, and the scheduling information for other SIBs is encoded in the periodic SIB1. Each SIB message has a different role. For example, SIB2 has information about random access for initiating a data transfer and uplink power control.

3GPP specifies that the broadcast of CMAS messages is over the air through SIB12 [13]. However, unlike point-to-point messages in LTE, broadcasts of SIB messages are not protected by mutual cryptographic authentication or confidentiality, since the SIB contains essential information the UEs use to access the network before any session keys have been established. The contents of a
Table 3.1: CBC configuration

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageIdentifier</td>
<td>Type of CMAS alert [3]. ‘4370’ is the Presidential Alert, ‘4379’ is an AMBER alert.</td>
</tr>
<tr>
<td>SerialNumber</td>
<td>Identifier of a CMAS message to detect duplicates.</td>
</tr>
<tr>
<td>RepetitionPeriod</td>
<td>Defines the duration in seconds between broadcasts over the air.</td>
</tr>
<tr>
<td>NumberBroadcastRequests</td>
<td>Defines the number of times to broadcast over the air.</td>
</tr>
<tr>
<td>DataCodingScheme</td>
<td>Encoding scheme of the message content [2].</td>
</tr>
<tr>
<td>WarningMessageContents</td>
<td>Is the alert text to be shown to users.</td>
</tr>
</tbody>
</table>

CMAS message is a simple GSM 7-bit encoded text (the same format used in the traditional Short Message Service (SMS) [46]). Once a CMAS message has been received, there is no verification method for the message content. If an attacker can imitate eNodeB behavior closely enough to broadcast false CMAS notifications, then the UE will display them to the user.

A UE’s vulnerability to a fake CMAS alert depends on whether it’s in an active or idle state, illustrated in Figure 3.4. To affect the most UEs an attacker must consider different approaches for each state. Here we discuss idle UEs and active UEs separately:

**Idle mode UEs.** Reference Signal Received Power (RSRP) is the power of an eNodeB-specific reference signal recognized by the UE, which is typically used to make an eNodeB selection and handover decision. Normally whenever a UE in idle mode performs eNodeB selection (or reselection), it will associate with the eNodeB having the highest RSRP. Therefore, we can formulate the selection of a specific eNodeB (indexed by $k^*$) done by the idle UE (indexed by $i$) as follows:

$$k^* = \arg \max_{k \in C_i} \{\text{RSRP}_k\}, \quad (3.1)$$

where $C_i$ is the set of eNodeBs that are observed by the UE$_i$. If the RSRP of a malicious eNodeB is the strongest, the UE decodes the SIBs transmitted by the malicious eNodeB. The attacker does not need to have any user information (including security keys) of UE$_i$, which would be stored in the database of the network operator. Without having such user information, UE$_i$ will eventually reject the authentication process with the malicious eNodeB. However, UE$_i$ can receive a CMAS
message transmitted by the malicious eNodeB during this process, as shown in Case 3 of Table 3.2.

Since a UE tries its authentication to the network up to five times before it listens to other frequency channels, the malicious eNodeB can leverage this period to send a CMAS message. Figure 3.2 shows the S1AP message exchange between the MME and eNodeB when there is no user information in the Home Subscriber Server (HSS). After the MME replies with Attach Reject due to the failure in user authentication, the UE re-sends Attach Request four more times (totaling 5 requests), which takes 42.06 secs. This means a UE stays up to 42 seconds in the vulnerable "Searching" state shown in Figure 3.4. This allows an attacker plenty of time for the CMAS spoofing; assuming a 160–msec periodicity of SIB12 transmissions, 262 fake alerts can be received by a UE during this period.

<table>
<thead>
<tr>
<th>Time</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.575193274</td>
<td>S1AP/NAS-</td>
<td>106</td>
<td>SACK 1d-downlinkNASTransport, Attach reject (IMSI unknown in HSS)</td>
</tr>
<tr>
<td>1.500866657</td>
<td>S1AP</td>
<td>82</td>
<td>id-UEContextRelease, UEContextReleaseCommand [NAS-cause=normal-release]</td>
</tr>
<tr>
<td>1.561297499</td>
<td>S1AP</td>
<td>82</td>
<td>id-UEContextRelease, UEContextReleaseComplete</td>
</tr>
<tr>
<td>11.691385719</td>
<td>S1AP/NAS-</td>
<td>216</td>
<td>id-initialUEMessage, Attach request, PDN connectivity request</td>
</tr>
<tr>
<td>11.69189769</td>
<td>S1AP/NAS-</td>
<td>106</td>
<td>SACK 1d-downlinkNASTransport, Attach reject (IMSI unknown in HSS)</td>
</tr>
<tr>
<td>11.69966948</td>
<td>S1AP</td>
<td>82</td>
<td>id-UEContextRelease, UEContextReleaseCommand [NAS-cause=normal-release]</td>
</tr>
<tr>
<td>11.76119724</td>
<td>S1AP</td>
<td>82</td>
<td>id-UEContextRelease, UEContextReleaseComplete</td>
</tr>
<tr>
<td>21.011210376</td>
<td>S1AP/NAS-</td>
<td>216</td>
<td>id-initialUEMessage, Attach request, PDN connectivity request</td>
</tr>
<tr>
<td>21.01222666</td>
<td>S1AP/NAS-</td>
<td>106</td>
<td>SACK 1d-downlinkNASTransport, Attach reject (IMSI unknown in HSS)</td>
</tr>
<tr>
<td>21.019061859</td>
<td>S1AP</td>
<td>82</td>
<td>id-UEContextRelease, UEContextReleaseCommand [NAS-cause=normal-release]</td>
</tr>
<tr>
<td>21.02112266</td>
<td>S1AP</td>
<td>82</td>
<td>id-UEContextRelease, UEContextReleaseComplete</td>
</tr>
<tr>
<td>31.931212554</td>
<td>S1AP/NAS-</td>
<td>216</td>
<td>id-initialUEMessage, Attach request, PDN connectivity request</td>
</tr>
<tr>
<td>31.932140527</td>
<td>S1AP/NAS-</td>
<td>106</td>
<td>SACK 1d-downlinkNASTransport, Attach reject (IMSI unknown in HSS)</td>
</tr>
<tr>
<td>31.93978869</td>
<td>S1AP</td>
<td>82</td>
<td>id-UEContextRelease, UEContextReleaseCommand [NAS-cause=normal-release]</td>
</tr>
<tr>
<td>31.941033246</td>
<td>S1AP</td>
<td>82</td>
<td>id-UEContextRelease, UEContextReleaseComplete</td>
</tr>
<tr>
<td>42.081365586</td>
<td>S1AP/NAS-</td>
<td>216</td>
<td>id-initialUEMessage, Attach request, PDN connectivity request</td>
</tr>
<tr>
<td>42.0852149218</td>
<td>S1AP/NAS-</td>
<td>106</td>
<td>SACK 1d-downlinkNASTransport, Attach reject (IMSI unknown in HSS)</td>
</tr>
<tr>
<td>42.085966484</td>
<td>S1AP</td>
<td>82</td>
<td>id-UEContextRelease, UEContextReleaseCommand [NAS-cause=normal-release]</td>
</tr>
</tbody>
</table>

Figure 3.2: S1AP log messages of authentication failure: after five consecutive authentication failures, which takes about 40 seconds, the UE starts to search for a new cell in other frequency channels.

**Active mode UEs.** When a UE is in active mode, it securely communicates with one serving eNodeB. If it finds another eNodeB with a higher power level than the existing serving eNodeB, a handover procedure can be triggered: the UE measures the RSRP of the candidate eNodeB and sends the measurement report to the serving eNodeB. The serving eNodeB then makes a handover.

---

1. 1S1 Application Protocol (S1AP) is the 3GPP standard protocol between the MME and eNodeB [14].
2. 2HSS is the user subscription database located in the LTE core network.
decision based on the received measurement report. However, if the target eNodeB is not identified by the serving MME, the handover will eventually fail. Therefore, the handover procedure, even if caused by a malicious eNodeB, does not make a UE vulnerable to the CMAS spoofing attack.

As a consequence, the attacker first needs to disconnect the UE from its serving eNodeB. After the UE is released from the serving eNodeB, it will immediately try to attach to the strongest eNodeB, and thereafter, it can be attacked in the same way as idle mode UEs described in the section above.

When a communication error is detected on the established radio link between the UE and its serving eNodeB, it is referred to as a Radio Link Failure (RLF). The RLF can be detected by either the UE or eNodeB for various erroneous cases. A typical RLF is caused by reaching the maximum number of packet retransmissions in the Radio Link Control (RLC) layer of the LTE protocol stack. Jamming LTE signals can easily lead to an RLF in active UEs [45, 53].

Without any special jamming technique, however, a malicious eNodeB can jam the communication between a UE and its serving eNodeB simply by transmitting at a much higher power than the serving eNodeB. The malicious eNodeB overwhelms the serving eNodeB’s transmissions and causes an RLF. This transmission must be on the same frequency channel used by the serving eNodeB. Figure 3.3 shows a UE releasing resources allocated by its serving eNodeB with the cause of ‘failure-in-radio-interface-procedure’. This is a failure caused by an RLF. Once the RLF occurs, the radio connection of the UE releases. The UE attempts to attach to the higher-powered eNodeB and, thus, becomes vulnerable to spoofed CMAS messages just as an idle mode UE is.

3.3 Cases for CMAS Reception and Trust

With the LTE testbed, we performed a CMAS reception test. The results break down into three possible outcomes for CMAS alert reception, those being: the CMAS is not received, the CMAS is received and is known to be trustworthy, the CMAS is received and may be malicious. From those results we have identified three possible cases that determine whether the CMAS is received and is trustworthy. The cases and results are summarized in Table 3.2. Each case depends
Figure 3.3: S1AP log message of a RLF: this UE was Active, but it becomes disconnected due to a sudden RLF.

on where the UE currently is in the idle/active life-cycle. While testing each case, we continuously transmit the CMAS message once a second.

Simply put, if a UE is not listening to the eNodeB transmitting the CMAS message, the CMAS message will not be received by the UE. This is illustrated as the blue portion in Figure 3.4. It may seem obvious, but a necessary condition for the UE to receive a CMAS message is that it is tuned to the synchronization channels of the eNodeB that is transmitting the CMAS message. If the UE is listening to other frequency channels, or selects another cell which is not transmitting the CMAS message, then the message will not be delivered. We will not consider this scenario from now on.

Secure CMAS. In Case 1, the UE attaches to an eNodeB and is safely in the active state. To do this, the UE must be equipped with a valid Service Identity Module (SIM) card that is registered to that EPC. This case is the general scenario for phones receiving normal service from their provider. Because mutual authentication between the UE and the network has been successfully made, the UE can trust that the eNodeB is not malicious [10]. The CMAS reception is successful as we would expect, and we know that this CMAS message is trustworthy.

Unsecured CMAS. In Case 2, the UE has failed or is in the midst of failing to attach when the
Table 3.2: Cases for CMAS Reception and Trust

<table>
<thead>
<tr>
<th>Case</th>
<th>SIM Equipped</th>
<th>Authentication Success</th>
<th>CMAS Reception</th>
<th>Trustworthy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

eNodeB transmits the CMAS message. The CMAS message will still be received by the UE; this is the crux of the vulnerability. In order to demonstrate this, we deleted the SIM information from the EPC so that the user authentication would be unsuccessful. The UE is now in the unsecured range between the idle and active state due to the authentication failure. Even though the UE fails to reach the active state, we observe that the CMAS message is successfully received. This is because once the UE completes decoding SIB12 it delivers the contents to the application layer to be shown to the user. This is possible even after the authentication process has finally failed. Case 2 can lead the potential threat that *any malicious eNodeB can deliver fake CMAS messages while the UE is in between the eNodeB search and authentication procedures*. The red area in Figure 3.4 depicts this exploitable state of the UE.

Finally, in Case 3, the UE roams to an eNodeB which sends a CMAS message. To demonstrate this we removed the SIM card from the UE. No authentication is possible but the UE can make emergency calls such as 911. Even in this situation, we verified that the UE still receives the CMAS message which is potentially malicious.

As shown in Cases 2 and 3, CMAS spoofing can be done while the UE performs an eNodeB search, prior to successful authentication with a trusted eNodeB. These results are verified using 1 × JL620 COTS LTE small cell (no modification), 1 × open source NextEPC (modified with the CBC), and 9 different commercial LTE phones (Apple iPhone 8, X, and XS; Google Pixel 1; Huawei Nexus 6P; Motorola G5 Plus, and G6; Samsung Galaxy S7 Edge, and S8). Considering that the majority of UEs in cellular networks are in the idle state and UEs often transition

---

3In the 3GPP standard, there exists a corresponding state model on EPS connection management (ECM) consisting of ECM-CONNECTED and ECM-IDLE states. The idle mode is defined to conserve power due to the UE’s radio and network resources. Initially, a UE performs ‘network attach’ and then it can go to the idle mode by inactivity. For simplicity, we do not differentiate between ‘network attach’ and ‘wake-up from idle’, both of which have the same problem at this phase of the threat.
from the active to idle state due to an inactivity timer (around 10 seconds \([35]\)), almost all UEs are susceptible to this attack.

### 3.4 Impact of Signal Strength

Here, we provide an analysis to estimate the expected number of UEs who are attacked by the methods described in Chapter 3.2 as a function of the difference in the received power strength between the malicious eNodeB and the serving eNodeB originally chosen in Eq. (3.1).

Let \( r_\text{i} \) be the RSRP of UE \( i \) from its strongest normal eNodeB, while \( \rho_\text{i} \) be the RSRP of UE \( i \) from the malicious eNodeB. Let \( \delta_\text{i} \) be

\[
\delta_\text{i} = r_\text{i} - \rho_\text{i}.
\]

A UE that observes \( \delta_\text{i} \leq 0 \) has a higher or equal RSRP value from the malicious eNodeB than the serving eNodeB. Let \( Pr(\delta_\text{i} \leq x) \) be the probability that \( \delta_\text{i} \) is equal to or less than \( x \), and \( F(x) \) is drawn from the cumulative distribution function (CDF) of \( Pr(\delta_\text{i} \leq x) \). Further, let \( N \) be the total number of UEs which is given by

\[
N = N_{\text{idle}} + N_{\text{active}},
\]

where \( N_{\text{idle}} \) and \( N_{\text{active}} \) are the number of idle UEs and active UEs. Let \( N^f \) be the random variable representing the number of UEs which successfully receive a fake alert from the malicious eNodeB. Similarly, \( N^f_{\text{idle}} \) and \( N^f_{\text{active}} \) represent the number of UEs receiving the fake alert in idle mode and active mode. Then, the expected number of \( N^f \) can be expressed as

\[
E[N^f] = E[N^f_{\text{idle}}] + E[N^f_{\text{active}}] \quad (3.2)
\]

\[
= F(\alpha) \times N_{\text{idle}} + F(\beta) \times N_{\text{active}}, \quad (3.3)
\]

where \( \alpha \) is the RSRP difference required for an idle UE to select the malicious eNodeB, and \( \beta \) is the RSRP difference required to trigger the RLF for active UEs. From Eqs. (3.2) and (3.3), we can
derive the following:

\[
\alpha = F^{-1} \left( \frac{E[N^f_{\text{idle}}]}{N_{\text{idle}}} \right),
\]

(3.4)

\[
\beta = F^{-1} \left( \frac{E[N^f_{\text{active}}]}{N_{\text{active}}} \right).
\]

(3.5)

Equations (3.4) and (3.5) show that the ratio of the number of UEs receiving a fake alert to the total number of UEs can determine the threshold values of \( \alpha \) and \( \beta \).
Figure 3.4: The Idle/Active life-cycle of a UE. The state of UE is moving counter clockwise. CMAS spoofing is possible while the UE performs a cell search, prior to successful authentication with a trusted eNodeB.
Chapter 4

Proof-of-concept Attacks

In this chapter, we present the details of our Presidential Alert Spoofer system and describe how it works. Our system can be built with either an SDR device or a COTS eNodeB, and the list of hardware and software systems we used is summarized in Table 4.1.

**Attack preparation.** Our Presidential Alert Spoofer must first identify the existing eNodeBs in a given licensed frequency band. Each eNodeB can be uniquely identified at a given geographical position by the pair of ‘E-UTRA Absolute Radio Frequency Channel Number (EARFCN)’ and ‘Physical Cell ID (PCI)’. For each EARFCN, our Spoofer finds the eNodeB, and associated PCI, of which the RSRP is the strongest. Once the existing eNodeBs are listed, the Public Land Mobile Network (PLMN) information of each eNodeB is collected. Every LTE network has its own PLMN, which contains a three digit country code and two or three digits to identify the provider. The

<table>
<thead>
<tr>
<th>System</th>
<th>Hardware</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack Preparation</td>
<td>BladeRF 2.0† ($500)</td>
<td>OWL (modified) [23]</td>
</tr>
<tr>
<td></td>
<td>USRP B210‡ ($1,300)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Laptop (&lt; $1,000)</td>
<td></td>
</tr>
<tr>
<td>SDR-based Spoofer</td>
<td>BladeRF 2.0† ($500)</td>
<td>srsLTE (modified) [33]</td>
</tr>
<tr>
<td></td>
<td>USRP B210‡ ($1,300)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Laptop (&lt; $1,000)</td>
<td></td>
</tr>
<tr>
<td>COTS eNodeB-based Spoofer</td>
<td>JL620† (FDD)</td>
<td>NextEPC (modified) [48]</td>
</tr>
<tr>
<td></td>
<td>JLT621† (TDD)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Laptop (&lt; $1,000)</td>
<td></td>
</tr>
</tbody>
</table>

†, ‡, †, The system requires only one among these.
PLMN is periodically broadcast by the eNodeB in the SIB1 message making it possible to passively collect all of the observable PLMNs within receiving range. To launch an attack, our Presidential Alert Spoof er uses the same PLMN as an existing eNodeB such that the UEs will select our Spoof er during a eNodeB search. We use the OWL software [23] with an SDR device (USRP B210 [28] and BladeRF 2.0 [50]), to gather all the PLMNs. Table 4.2 is the attack preparation results measured in our lab across the top four US LTE operators. For each EARFCN, our Spoof er’s eNodeB tunes its radio to the same frequency with the same PLMN identifier, and starts to transmit a Presidential Alert continuously which contains a custom (attack) message.

<table>
<thead>
<tr>
<th>Operator</th>
<th>EARFCN</th>
<th>Duplex</th>
<th>PCI</th>
<th>PLMN</th>
<th>RSRP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT&amp;T</td>
<td>5110</td>
<td>FDD</td>
<td>415</td>
<td>310410</td>
<td>-100</td>
</tr>
<tr>
<td>Sprint</td>
<td>41374</td>
<td>TDD</td>
<td>265</td>
<td>310120</td>
<td>-102</td>
</tr>
<tr>
<td>T-Mobile</td>
<td>5035</td>
<td>FDD</td>
<td>312</td>
<td>310260</td>
<td>-120</td>
</tr>
<tr>
<td>Verizon</td>
<td>5230</td>
<td>FDD</td>
<td>229</td>
<td>311480</td>
<td>-105</td>
</tr>
</tbody>
</table>

**Attack execution with an SDR device.** We implemented the Spoof er using a USRP B210 and BladeRF to attack Frequency Division Duplex (FDD) systems. With an SDR, we can change the transmission frequency easily to target every cellular band. We added SIB12 support to the open source eNodeB software [33] to transmit CMAS messages. Since the attack can last about 42 seconds (described in Chapter 3.2) and we can send a CMAS message every 160 msec and a victim UE may receive up to 262 SIB12 transmissions.

**Attack execution with a COTS eNodeB.** We use a COTS eNodeB (Juni JLT-621 [37]) to target Time Division Duplex (TDD) systems. Our modification of NextEPC provides an interface to inject a user-defined Presidential Alert that broadcasts each second as shown in Figure 4.2. With this configuration, a victim UE may receive up to 42 transmissions of SIB12 from the COTS eNodeB. Any commercial LTE FDD/TDD eNodeB hardware can perform this attack, which may play a key role if an attacker wants to control multiple malicious eNodeBs in a coordinated manner.

**Attack verification.** In our lab environment, we verified that the fake Presidential Alert sent
by our SDR-based Spoofer was successfully shown in the FDD phones of AT&T, T-Mobile, and Verizon. With a TDD Sprint phone, we verified that our COTS eNodeB-based Spoofer also works successfully. Testing was performed on the nine different mobile phones listed in Chapter 3.3 (two of which are shown in Figure 4.3). The detailed conditions regarding the attack will be described in Chapter 5.

Affected devices. Through discussions with 3GPP of the SIB12 vulnerability described in Chapter 3.2, it became clear that the lack of authentication was a design choice by 3GPP, rather than an oversight. This design provides the best possible coverage for legitimate emergency alerts, but the trade-off leaves every phone vulnerable to spoofed alerts. As a consequence, all modem chipsets that fully comply with the 3GPP standards show the same behavior: the fake Presidential Alert is received without authentication. Once the LTE modem of the UE receives the fake alert, the operating system will display the alert to the user. Since our attack verification tests included many Android and iOS phones, we conclude that most (presumably all) LTE phones will be affected by the attack, regardless of the phone’s vendor or model.

---

1Note that emitting over-the-air signals on a licensed band spectrum is illegal. Our experiments are carried out with proper RF shielding.

2At the moment, an SDR-based LTE TDD system implementation is not available.

3According to a mobile market report, the worldwide market share of Android is 75.39% and that of iOS is 22.35% on March 2019.

4Since much of the LTE public warning system is inherited from 2G/3G, a similar attack is also possible in 2G/3G. Unfortunately, this is out-of-scope for this thesis.
CMAS-enabled MME

Figure 4.1: Our Presidential Alert Spoofer scans a cell, gathers operator information, and sends a fake Presidential Alert to both idle and active UEs who are working on FDD/TDD frequencies in the US and Canada. It consists of one SDR device, one COTS LTE eNodeB, and 2 laptops.

<table>
<thead>
<tr>
<th>Time</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>77.643929117</td>
<td>SACK id-downlinkNASTransport, Attach reject (IMSI unknown in HSS)</td>
</tr>
<tr>
<td>77.649933664</td>
<td>id-UEContextRelease, UEContextReleaseCommand [NAS-cause=normal-release]</td>
</tr>
<tr>
<td>77.651175039</td>
<td>id-UEContextRelease, UEContextReleaseComplete</td>
</tr>
<tr>
<td>77.813845964</td>
<td>SACK id-WriteReplaceWarning, WriteReplaceWarningRequest</td>
</tr>
<tr>
<td>77.820558354</td>
<td>id-WriteReplaceWarning, WriteReplaceWarningResponse</td>
</tr>
<tr>
<td>77.831089532</td>
<td>id-WriteReplaceWarning, WriteReplaceWarningRequest</td>
</tr>
<tr>
<td>78.840540843</td>
<td>id-WriteReplaceWarning, WriteReplaceWarningResponse</td>
</tr>
<tr>
<td>79.852827471</td>
<td>id-WriteReplaceWarning, WriteReplaceWarningRequest</td>
</tr>
<tr>
<td>79.860515254</td>
<td>id-WriteReplaceWarning, WriteReplaceWarningResponse</td>
</tr>
<tr>
<td>80.875585272</td>
<td>id-WriteReplaceWarning, WriteReplaceWarningRequest</td>
</tr>
<tr>
<td>80.880524681</td>
<td>id-WriteReplaceWarning, WriteReplaceWarningResponse</td>
</tr>
<tr>
<td>81.898361647</td>
<td>id-WriteReplaceWarning, WriteReplaceWarningRequest</td>
</tr>
<tr>
<td>81.900509912</td>
<td>id-WriteReplaceWarning, WriteReplaceWarningResponse</td>
</tr>
</tbody>
</table>

Figure 4.2: S1AP log messages: our modification on NextEPC provides an interface to inject a user-defined message as a Presidential Alert.
Figure 4.3: Received multiple fake Presidential Alerts using Samsung Galaxy S8(left), and Apple iPhone X(right).
Now, we evaluate the attack performance of our Presidential Alert Spoofer system. First, we show that the attack success rate depends on the spoofing signal strength. Second, we take propagation measurements of the AT&T LTE network and our SDR eNodeB in both indoor and outdoor environments. With these propagation measurements and the known success rate of the attack, we then evaluate the attack’s coverage.

![Diagram of testbed setup](image)

Figure 5.1: Testbed setup for evaluating the attack success rate. The transmission power levels of SecureNet eNodeB and Presidential Alert Spoofer can be controlled independently.

### 5.1 Experimental Setting

Figure 5.1 illustrates our experimental testbed setup, which consists of an EPC and eNodeB for a normal LTE system, a malicious eNodeB for spoofing, and cell phones for victim UEs. A signal attenuator receives the broadcast signals from two different sources and delivers the combined signal to a UE in a shielded box. We built an LTE test network, with an EPC and eNodeB, named SecureNet that assumes the role of the user’s original network. On the other hand, the malicious
eNodeB, part of the Presidential Alert Spoofer, is installed solely without any LTE core support. By using the signal attenuator, the signal power received at the UE can be precisely controlled for various practical scenarios. Within the experiments, all UE measurements were gathered with the Samsung Galaxy S8 and Motorola G6.

5.2 Success Rate

We evaluate the success rate of the Presidential Alert Spoofer as a function of $\alpha$ (or $\beta$), which is the RSRP difference between the SecureNet eNodeB and Presidential Alert Spoofer for an idle UE ($\beta$ for an active UE). We first attach the UE to SecureNet. For the idle UE case, we wait for the UE to enter the idle mode due to inactivity. The Spoofer broadcasts each new Presidential Alert message with a new Serial Number and different message content (described in Table 3.1). In doing so we can determine whether each Presidential Alert is successfully received\(^1\) and at what power configuration of $\alpha$ or $\beta$. We conducted 20 experimental trials for each value of $\alpha$ (or $\beta$) ranging from 0 dB to -25 dB.

The Spoofer may elect to use a different PCI than that of the serving eNodeB, appearing to be a new eNodeB. Or, the Spoofer may use the same PCI, appearing to be the existing eNodeB and interfering with the existing eNodeB’s PHY-layer control channel information\[^{68}\]. This decision has different impacts on the performance of the spoofing attack depending on the UE state (idle or active).

Figure 5.2 shows the empirical cumulative distribution function (CDF) of successful receptions of fake alerts as a function of $\alpha$ for idle UEs. When the Spoofer uses a different PCI and the received signal strength from the Spoofer is higher than that from SecureNet ($\alpha < 0$), the idle UE will consider the Spoofer as a new serving eNodeB by Eq. (3.1). Our experimental results verify this expectation; 50% of idle UEs can receive a fake message even at $\alpha = -1$, more than 90% of idle UEs can receive a fake message when $\alpha \leq -6$.

However, if the same PCI is used, the attack performance is significantly degraded. Because

\[^{1}\text{CMAS messages with repeated content may be ignored by the UE.}\]
the PCI is used to generate cell-specific reference signals [1], using the same PCI value will cause channel estimation errors at the UE due to collisions from the two transmitters. This, in turn leads to more decoding errors when receiving the SIBs. As a result, using the same PCI requires much higher attack power, as no UE is affected when $\alpha$ is greater than $-12$ dB. With $\alpha \leq -17$, 90% of idle UEs can still be attacked.

Figure 5.3 shows the CDF of successful fake message receptions as a function of $\beta$ (i.e., forcing disconnect) for active UEs. When the Spoofe uses a different PCI and the received signal strength from the Spoofe is higher than that from the SecureNet eNodeB, the active UE will start to consider the Spoofe as a target eNodeB for a handover, as described in Chapter 3.3. Because the Spoofe is not identified by SecureNet, a handover cannot be performed. Instead, we observed a RLF will occur when $\beta \leq -10$, which eventually leads to the reception of a fake alert. 90% of active UEs can receive a fake message when $\beta \leq -20$ assuming that a different PCI value is used for the Spoofe. Unlike the idle UE case, using the same PCI value results in higher decoding errors (and more RLFs) at a receiver. Thus, it shows better attack performance; 90% of receptions are successful with $\beta \leq -13$.

5.3 Attack Coverage

To determine the attack coverage, we performed measurements of RSRP over various distances between the UE and the eNodeB. We transmitted with a COTS eNodeB in the Educational Broadcast Service (EBS) band and at 0.1 Watt transmit power. The measurement was done with 70m of indoor space and 120m of outdoor space. From the measurements, we observed that the RSRP tends to decrease as the distance increases and the indoor RSRP is higher than that of outdoor. This is due to the fact that multiple signals from indoor reflections along with a line-of-sight signal can provide receiver diversity, thus exhibiting a smaller path-loss exponent than an outdoor environment [58, 32].

From Figures 5.2 and 5.3, we show the following results: If the malicious eNodeB’s PCI is

\[2\text{We have the spectrum license for the EBS band in our campus area.}\]
different from SecureNet and the idle UE's RSRP from the malicious eNodeB is 6dB higher than
SecureNet, the attack will be successful with 90% probability. When the RSRP difference is 1dB,
the attack to the idle UE is successful with 50% probability. On the other hand, in the case of
the same PCI configuration, the active UE will be attacked with 90% probability when the RSRP
difference is 13dB.

In order to figure out the maximum attack distance, polynomial regression was applied to
the measured values for both outdoor and indoor conditions. Figure 5.4 shows the relationship
between the RSRP from SecureNet and the attack distance by the malicious eNodeB in order to
achieve a 90% success rate for active UEs, a 90% success rate for idle UEs, and a 50% success
rate for idle UEs. For example, in the outdoor case shown in Figure 5.4(a), when the RSRP of an
active UE from SecureNet is $-100$ dBm, the attack can reach up to 23.4m centered at the malicious
eNodeB of 0.1 Watt power with 90% probability. For the idle UE, when the RSRP is $-100$ dBm,
it can be attacked up to 44.1m and 68.5m away from the malicious eNodeB of 0.1 Watt power
with 90% and 50% probability, respectively. Figure 5.4(b) shows the result in open space indoors,
where the maximum attack distance can be obtained similarly with the outdoor case. We observe
that the attack distance can be much greater than the outdoor case due to the smaller attenuation
characteristic of indoor buildings. For the active UE with $-100$ dBm RSRP, the attack radius is
55.2m with a 90% probability of success.

5.4 Practical Scenarios: Indoor and Outdoor

Since we do not use our spoofer outside of a shield box, we cannot directly measure its effect
on a large number of people. To evaluate the attack coverage according to its success rate, we use
actual RSRP measurements in the indoor and outdoor environments.

**Indoor Attack.** We placed our malicious eNodeB inside a campus building and measured the
RSRP of a dummy LTE signal (containing no CMAS message) in the EBS band with 0.1 Watt
transmit power. We also measured the RSRP of a nearby AT&T eNodeB, shown in Figure 5.5(a).
The RSRP does not attenuate consistently due to various obstacles, but generally the RSRP tends to
decrease as the distance from the AT&T eNodeB increases. We compared the two RSRPs throughout the building and indicated the attack coverage using measurements obtained from Chapter 5.2 depicted in Figure 5.5(b). As a result, in a building with a total area of about 16,859m$^2$, for idle UEs, the coverage for a 90% success rate was about 4,435m$^2$, whereas for active UEs, the coverage for a 90% success rate was about 2,955m$^2$.

**Outdoor Attack.** Without access to outdoor LTE equipment, we simulate the RSRPs of the spoofing eNodeB and the AT&T eNodeB with the NS-3 v3.29 network simulator [49]. For a scenario we assume a football game in which a large number of people are gathered in a restricted region. A group of attackers send fake alerts to the spectators inside the football stadium. We measured the RSRP of an actual AT&T eNodeB around the perimeter of our campus’ football stadium, shown in Figure 5.6. We used the simulator to estimate the RSRPs at the centers of each section in the stadium (Figure 5.6(a)). We simulated spoofers in four corners around the stadium, near but still outside of the ticketed area. Figure 5.6(b) shows which malicious eNodeB with a 1-Watt transmit power attacked each section. We observe that all sections, except one, are attacked by the malicious eNodeBs. This means that 49,300 among the total 50,000 seats will be hit with the attack, which itself has a 90% success rate, given that all UEs are in the idle state.
Figure 5.2: CDF as a function of $\alpha$. Because the cell reselection can be done when idle UEs wake up, the spoofing attack is much easier when a different PCI value is used.

Figure 5.3: CDF as a function of $\beta$. Using the same PCI value leads to more decoding errors observed at the receiver, which results in slightly more effective attacks.
Figure 5.4: Maximum attack distance between the UE and the malicious eNodeB according to RSRP value of the UE from the SecureNet eNodeB, when the transmit power of the malicious eNodeB is 0.1 Watt.
Figure 5.5: Indoor attack coverage measurements. (a) The outdoor RSRP distribution of AT&T is measured at Folsom Field, University of Colorado Boulder. (b) The attack coverage is simulated when 4 × 1-Watt malicious eNodeBs are located around the stadium: 90% of success rate can be achieved for idle UEs.
(a) Outdoor RSRP measurements of AT&T

(b) Attack coverage

Figure 5.6: Outdoor attack coverage simulation. (a) The outdoor RSRP distribution of AT&T is measured at Folsom Field, University of Colorado Boulder. (b) The attack coverage is simulated when $4 \times 1$-Watt malicious eNodeBs are located around the stadium: 90% of success rate can be achieved for idle UEs.
Defending against CMAS spoofing attacks requires careful consideration of several challenges. First, updates to the CMAS architecture could require expensive changes by cell phone manufacturers, operating system developers, government bodies, and cellular carriers. Coordinating such an effort would be difficult due to the fragmented nature of the network. Furthermore, updates must still support outdated devices, both on the user (UE) and infrastructure (eNodeB) side, as it could take years to replace or update old equipment already in use. In addition, any comprehensive defense must trade-off the protections provided with the availability of the system: if users cannot receive valid alerts due to complex protections, it may be more hazardous than if we continued to use the existing (but vulnerable) system.

With these challenges in mind, we discuss potential defenses in both prevention and mitigation. For prevention, we suggest adding digital signatures to alerts which is a secure and robust solution. For mitigation, we propose accepting only secure CMAS messages, and carrying messages over Evolved Multimedia Broadcast Multicast Service (eMBMS). We also discuss two possible mitigation solutions – the Received Signal Strength (RSS) based approach and eNodeB fingerprinting. We stress that neither of these defenses offer a magic solution, but instead hope they provide starting points for network operators and cell phone manufacturers to continue discussions.
6.1 Prevention: Digital Signature

We first consider digitally signing SIB12 messages to prevent spoofed messages, as discussed by 3GPP [1]. While conceptually simple, adding signatures is difficult for several reasons: First, operators and devices must agree on the key or keys that will be used to sign and validate messages. Second, devices must decide what to do if they receive a signature from an unknown key or an unsigned (e.g., legacy CMAS) message from the network. Finally, signatures must fit within the practical constraints of the network.

For key management, we leverage suggestions from 3GPP discussions [1], which suggest using the Non-Access Stratum (NAS) to send authenticated messages to the device. Because NAS provides message integrity between the eNodeB and UE (mediated by pre-shared keys in the UE SIM card), messages received in this way cannot be spoofed by a (physically) nearby adversary. However, sending alerts over this channel would limit their reception to devices that had established a NAS session. Instead, we recommend using this authenticated channel to send updates to the set of (public) keys that a device should trust. These keys could correspond to private ones held by the variety of local and national Cell Broadcast Entities (CBEs) that are authorized to issue such alerts, such as local law enforcement for AMBER alerts or the President for Presidential Alerts. If desired, these keys could be included in certificates that describe their scope, and be signed by central entities such as the carrier or traditional Certificate Authorities.

In the immediate term, networks or CBEs that do not support digital signatures will continue to send legacy ones, and it is also possible for devices to receive alerts from CBEs that they do not have the corresponding key for. If the device refuses to display such messages, it may leave the user in the dark about potentially legitimate alerts, while protecting them from false ones. If the alert is still displayed, then the addition of signatures provides no benefit. One solution is to allow legacy messages to be displayed until enough of the network has deployed signatures. Another approach could be to allow the user to see unverified alerts but with a warning indicating that the message could not be authenticated. We caution that such warnings must be designed carefully to
appropriately inform the user of the risks; previous research on the usability of HTTPS warning messages in browsers may be a useful starting point [30].

Finally, signatures must fit within the constraints of the network without adding significant overhead. CMAS alert messages are sent in an 82-byte field of the SIB12 message [3]. Even adding a short signature to this could limit the length of a useful message. Instead, we propose using additional pages in the existing SIB12 message to send a corresponding signature of a CMAS alert message, allowing for up to 82-bytes for the signature or any metadata (such as a time stamp or sequence number to prevent replay attacks). This is more than enough for a 64-byte ECDSA signature, though if smaller signatures are desired, BLS signatures could also be used to reduce the size in half [22]. As of May 2019, the FCC mandated that commercial mobile service operators must support alert messages that contain up to 360 characters of alphanumerical text on their LTE and future networks in the US [29]. As a result, adding a digital signature becomes applicable for the existing and future wireless emergency alert systems.

**Signature implementation.** To verify the feasibility of this scheme, we implement a simple digital signature for the Presidential Alert. We used the ed25519 digital signature [20] to sign a 4-byte time stamp along with the CMAS alert message (68 bytes overhead in total), and defined a new Message Identifier (see Table 3.1) which indicates that a digital signature is added. Once a signed message is received, the alert message can be displayed after verifying its signature. We implemented this by modifying the open source UE implementation running on a USRP B210 [33]. The resulting UE is not affected by the spoofing attack because it only accepts signed messages. Figure 6.1 shows the procedures taken for this implementation.

**Suggestion:** We recommend that the digital signature only be required for Presidential Alerts. Unlike AMBER and other alerts, the Presidential Alert has only one originator, making the key distribution much simpler. The private key need only be known by a single entity, rather than a large number of CBEs. In addition, Presidential Alerts are the only alerts that cannot be disabled by users, potentially making them a larger target for abuse.
6.2 Mitigation

The main reason why CMAS is currently sent through the SIB is due to the fact that the accessibility is important. This means that it is designed to deliver messages to as many people as possible, regardless of whether they are authenticated or subscribed to the different network. However, we can assume that the majority of UEs are under the coverage of their subscribed network and thus can receive CMAS messages. Even their UEs are in idle mode, using paging messages, it is possible to force them to be authenticated to their subscribed network. Therefore, we believe that providing security solutions at least to those who can authenticate to their subscribed network, can significantly reduce the effectiveness of this attack (Chapter 6.2.1, 6.2.2 and 6.2.3).

Another approach is to distinguish the difference between signals sent by the malicious eNodeB and signals sent by the legitimate eNodeB. If we can certainly fingerprint an attacker’s signal, it could be a very efficient defense, and here we suggest some ways to identify the characteristics of the attacker’s signal (Chapter 6.2.4).

6.2.1 Accepting only secure CMAS

Since the Presidential Alert Spoofe exploits the unauthenticated CMAS delivery in LTE, a UE may ignore all received SIB12 messages before it successfully authenticates the network. It could be implemented either 1) by the LTE modem firmware, or 2) by the operating system (e.g., Android and iOS) of the UE. As the LTE modem firmware maintains its own state including the UE authentication, it could decide whether each SIB12 reception is secure or not. Similarly, the OS could keep track of the authenticated state of the network, and it may choose not show an unsecured CMAS messages to the user. The major disadvantage of this approach is that the accessibility of trustworthy CMAS messages will be limited. Secure CMAS alert messages may not be received by:

- those who do not have a valid subscription from a home or visited network service provider,
- those who are temporarily in the unauthenticated state due to the UE attaching, idle exit, handovers, etc.
Nevertheless, the risk of the CMAS spoofing attacks could be significantly mitigated by this approach.

6.2.2 eMBMS based approach

eMBMS \[4,12\] is a point-to-multipoint interface that delivers single source content to multiple UEs located in a given geographical area in 4G/LTE networks. The purpose of eMBMS is to use network resources as efficiently as possible for multicast services such as real-time video streaming, file distribution, and emergency alerts to multiple UEs at the same time.

**eMBMS Overview.** To enable eMBMS in 4G/LTE networks, three new entities are required. The three entities and their roles are described below and the eMBMS network architecture is depicted in Figure 6.2.

- **Broadcast Multimedia Server Center (BM-SC)** is responsible for data delivery from the content provider to the MBMS Gateway, security of eMBMS user services (i.e., authentication, key generation and distribution, and data encryption), scheduling of eMBMS sessions, data synchronization using MBMS Synchronization (SYNC) protocol \[6\], and providing membership and proxy functions.

- **MBMS Gateway (MBMS GW)** is logically divided into two parts. One is for the user plane and the other is for the control plane. In the user plane, after receiving data from BM-SC, it distributes data to eNodeBs through IP multicast. In the control plane, it transmits and receives control signal such as the session start/stop signal to BM-SC and MME.

- **Multi-cell/multicast Coordination Entry (MCE)** is responsible for the admission control and conveying radio configuration data (e.g., radio resource allocation information, and scheduling information) for eMBMS to the eNodeB. The MCE can be located in each eNodeB, or can be located alone in the EPC which is connected to the eNodeB.

How to allocate the radio resource is a very important functionality of eMBMS. If it is impossible to allocate radio resources for eMBMS data flexibly, services demanding large bandwidth
such as video streaming cannot be transmitted in high quality. In the FDD mode, in one radio frame having 10 subframes, up to 6 subframes can be assigned for the eMBMS service. In the TDD mode, up to 5 subframes can be assigned. UEs have to know which subframes are assigned for the eMBMS service to distinguish eMBMS data from other data. For this, SIB13 is used. The SIB13 includes information about radio resource allocation, and so when the UE receives the SIB13, it can recognize which subframes are allocated for eMBMS.

**eMBMS Security.** Figure 6.3 shows the security architecture of eMBMS. It’s an end-to-end security approach – the BM-SC and the UE are responsible for the security of eMBMS. The BM-SC is responsible for authorizing users, generating and distributing secret keys to the UEs, and encrypting data. Also, the UE is responsible for registering to and de-registering from the eMBMS user service, requesting and receiving secret keys from the BM-SC, and decrypting data.

Specifically, the Bootstrapping Server Function (BSF), a part of Generic Bootstrapping Architecture (GBA) [8], and the UE have one pre-shared symmetric key, $K_s$. Using the bootstrapping procedure as defined by GBA, $K_{s\_int\_NAF}$ and $K_{s\_ext\_NAF}$ are derived from $K_s$ and they are stored in the UE and the BM-SC (Note: NAF stands for Network Application Function, int stands for internal, and ext stands for external). In the case of the UE, $K_{s\_int\_NAF}$ is stored in the Universal Integrated Circuit Card (UICC) and $K_{s\_ext\_NAF}$ is stored outside of the UICC.

Next, in the BM-SC and the UE, MBMS Request Key (MRK) is derived from $K_{s\_ext\_NAF}$, and MBMS User Key (MUK) is derived from $K_{s\_int\_NAF}$. Note that MRK and MUK are never transmitted over the air. The BM-SC and the UE authenticate each other using MRK in the HTTP Digest Authentication [54] procedure. Then, the BM-SC generates MBMS Service Keys (MSKs) for each service of eMBMS. The BM-SC transmits MSKs to the UE over Multimedia Internet KEYing (MIKEY) [18] messages protected by MUK. For each data traffic of eMBMS, the BM-SC generates MTKs, and distributes them to the UEs over MIKEY messages protected by MSK.

Consequently, the BM-SC encrypts data using MTKs and transmits encrypted data to the UEs. Then the UEs with MTKs can decrypt the encrypted data. Note that MRK and MUK are user specific, so each UE has different MRK and MUK. Therefore, the BM-SC must have MRKs
and MUKs for all UEs who register to the eMBMS user service. On the other hand, MSK and MTK are data specific, so all UEs who have registered to the eMBMS user service have the same MSKs and MTKs together.

**eMBMS Implementation.** In srsLTE release 18.12 (the latest release at the time of writing), only some user plane functionalities of MBMS GW are implemented. The BM-SC, MCE, and the control plane of MBMS GW are not implemented. It means that, using the current version of srsLTE, it cannot send control signals, such as the session start/stop signals, and it does not support any security functionality. Instead of implementing BM-SC, MCE and other functions of MBMS-GW, we just implemented the eMBMS data transmission and security related functions in the srsLTE.

Because of the absence of the control plane, we made the UE always listens to the eMBMS data transmissions from the MBMS GW without registering to the eMBMS user service, and the eNodeB always broadcasts the same SIB13 every 160ms. As a result, the UE could know which subframes will be assigned for the eMBMS data transmission. Because our objective is to check the feasibility of delivering alert message via eMBMS, we made the MBMS GW send a simple text and the UE receive it.

Our implementation is depicted in Figure 6.4. The UE successfully receives multicasted encrypted text from the MBMS GW and decrypts it. With this implementation, we verified that we can easily multicast encrypted alert messages via eMBMS to the UE. Even though this eMBMS approach cannot prevent SIB based attacks and needs modification of the network, we present a way to send highly secure messages to UEs.

### 6.2.3 Received Signal Strength (RSS) based approach

We can leverage the received signal strength (RSS) at the UE to determine if the eNodeB to which we are connected is a feasible distance away. Using a widely used path-loss model [32], we can estimate the distance to the eNodeB using the RSS value. Then compare this with the location provided by an Internet database [21] to determine whether the alert could have come from a trusted eNodeB or not.
We emphasize that client-only solutions are not as robust as a digital signature-based one, as a motivated attacker may still be able to spoof messages to some users. However, it has the advantage of not requiring any modifications on the network side, and can be implemented entirely through software updates on the UE, offering a potentially attractive short-term stopgap.

6.2.4 LTE eNodeB fingerprinting

Since every eNodeB broadcasts its network configuration through SIB messages, a UE may leverage such information to construct a fingerprint of the eNodeB during its normal operation. In particular, we observe that each eNodeB uses different SIB contents and patterns. The transmission patterns of SIB messages in terms of message types and periodicity can vary, and we observed that each operator exhibits its own transmission pattern. By monitoring the SIB transmission pattern of a certain eNodeB, we can link it to a specific operator that runs it. In addition, channel quality (e.g., RSRP) and cell load information can be measured by the UE and combined with location information to additionally classify the validity of the attached eNodeB. While an attacker may be able to mimic these signals, it may nonetheless be a significant engineering hurdle to perform a coordinated attack against a geographically diverse set of users.

Fingerprinting Implementation. By modification of open source software srsLTE, we could easily decode SIB1 and measured RSRP from eNodeBs. Since in SIB1, other SIBs’ scheduling information is included, we could record unique SIB transmission patterns across some US LTE carriers in Boulder, Colorado and our two types of spoofers (i.e. COTS type spoofer and SDR type spoofer). The results of our implementation are depicted in Figures 6.5 and 6.6.
Figure 6.1: Working procedure of digital signature implementation
Figure 6.2: Network Architecture of eMBMS
Figure 6.3: Security Architecture of eMBMS
Figure 6.4: Implementation of eMBMS
Figure 6.5: Reception Patterns of SIBs from Commercial eNodeBs
Figure 6.6: Reception Patterns of SIBs from Malicious eNodeBs
Chapter 7

Conclusion

In this thesis, we have identified security vulnerabilities of WEA over commercial LTE networks and found that a spoofing attack with fake alerts can be done very easily. Specifically, we presented our threat analysis on the spoofing attack, and implemented an effective attack system using COTS SDR hardware and open source LTE software. Our extensive experimentation confirmed that the CMAS spoofing attack can succeed in all tested smartphones in the top four cellular carriers in the US. Further, we have discussed several defenses, from which we believe that completely fixing this problem will require a large collaborative effort between carriers, government stakeholders, and cell phone manufacturers.
Bibliography


