Denial of Service Attack on LTE network

Chan Jun Yuan
National University of Singapore
School of Computing
+65 9792 6462
E0053015@u.nus.edu

Danny Ng Ming Xuan
National University of Singapore
School of Computing
+65 9232 8196
E0052972@u.nus.edu

Huang Chi Ying
National University of Singapore
School of Computing
+65 8179 3231
E0032341@u.nus.edu

Lai Qi Wei
National University of Singapore
School of Computing
+65 9785 1130
E0053019@u.nus.edu

Zachary Tang Tjun Chii
National University of Singapore
School of Computing
+65 9069 0249
E0031678@u.nus.edu

ABSTRACT
With prevalent adoption of 4G LTE services island-wide, security of cellular networks is of utmost importance. User Equipment such as mobile phones, would connect to cell towers based on the tower's signal strength. However, the LTE protocol is designed in such a way that the initial exchange of messages with the cell towers are unencrypted. As a result, the protocol is susceptible to attacks such as Denial of Service (DoS). In this project, we explore ways to perform attacks on UEs by simulating a bogus LTE network through the use of a Software Defined Radio. We perform a low-cost setup using LimeSDR Mini and srsLTE, an open source software suite, to simulate a malicious cell tower setup. We also discuss why these vulnerabilities exist and how to mitigate them.

Categories and Subject Descriptors
C.2.1 [**Network Architecture and Design**]: LTE architecture design; K.6.4 [**Security and Protection**]: Authentication, Denial of Service

General Terms
Experimentation, Security

Keywords
LTE 4G network, DoS attacks, Software Defined Radios

1. INTRODUCTION
4th Generation "Long Term Evolution" (4G LTE) is a type of 4G wireless broadband which succeeds GSM and CDMA in communication via mobile phones. 4G LTE was developed with the purpose of increasing the capacity and speed of wireless networks using new digital signal processing (DSP) techniques and modulations. In order to communicate using the 4G LTE protocol, mobile phones would first have to communicate with an evolved NodeB, which is a node that acts as an intermediary transceiver between the mobile phones and the core cellular networks.

In recent years, the usage of User Equipment (UE), such as mobile phones and tablets, has been steadily increasing. In fact, the number of mobile devices alone have been reported to greater than the entire human population back in 2014 [3], and the disparity will only grow larger. With the prevalence of adoption of 4G LTE services island-wide, security issues of cellular networks are not to be overlooked. UEs equipped with 4G LTE capabilities would automatically search for a suitable cell tower in an attempt to connect to the mobile network. UEs would choose cell towers with the best signal strength or the Reference Signal Received Power (RSRP), which is a power measurement level of signal [16].

To establish a connection with the network, UEs are required to perform the *Attach Procedure*. Only upon successful attachment would the UE be granted access to the LTE network services. However, the LTE protocol is designed in such a way that the initial exchange of messages with the cell towers are unencrypted. As a result, the protocol is susceptible to attacks such as Denial of Service (DoS) attacks.

2. LTE ARCHITECTURE
The LTE network architecture is designed using packet switched backbone, allowing for end to end IP connectivity [9]. To provide LTE services, telecommunication service providers in Singapore such as Singtel, M1 and Starhub operates on different LTE frequency bands ranging from Band 8 at 900MHz to Band 3 at 1800MHz to Band 7 at 2600MHz [7]. The LTE architecture is divided into two main sections: E-UTRAN (Evolved UMTS Terrestrial Radio Access Network) and the core network known as the Enhanced / Evolved Packet Core (EPC). This section examines these components in greater details. Figure 1 [12] below shows the LTE network Architecture.

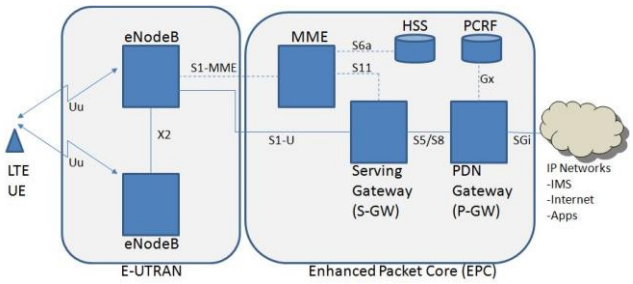


Figure 1. LTE network architecture

2.1 E-UTRAN

The E-UTRAN refers to the Radio Access Network (RAN) architecture which was primarily a part of the 3rd Generation Partnership Project (3GPP), as the 3GPP LTE physical layer specification. The E-UTRAN consists of mobile terminals known as User Equipment and the evolved NodeB (eNodeB), or commonly known as the LTE base stations.

2.1.1 User Equipment

User Equipment or UE refer to devices such as smartphones or tablets that are capable of utilizing the 4G LTE protocol for network communications. Therefore, UEs are equipped with Universal Subscriber Identity Module (USIM) card, which contains the International Mobile Subscriber Identity (IMSI) number that uniquely identifies a mobile subscriber, and the corresponding secret cryptographic key used to authenticate the UE upon connection with the core network. The key generated from the authentication process will then be used to encrypt the data communication between the UE and eNodeBs.

2.1.2 ENodeB

An eNodeB serves as the facilitator between the UE and the EPC, or the core network. The coverage area for an eNodeB is known as a cell. Each telecommunication service provider has divided regions into Tracking Areas (TA), which contain multiple cells. The eNodeBs would broadcast provider-specific Tracking Area Code (TAC), Mobile Country Code (MCC), Mobile Network Code (MNC) as well as the Cell identifier, allowing UEs to identify their serving network provider. UEs would then be able to inform the network about its mobility by performing the TrackingAreaUpdate (TAU) procedure in the TA [17].

ENodeBs communicate with UEs via the Radio Resource Control (RRC) protocol. Within a specific E-UTRAN, eNodeBs communicate with one another through the X2 interface, allowing them to perform load management as well as handover operations. The load management operation allows for the exchange of network traffic load information to manage traffic loads more efficiently among the eNodeBs. The handover operation allows eNodeBs to handover or transfer the UE to another eNodeB if the latter is deemed able to serve the UE better [11].

In addition, each eNodeB is connected to the Serving Gateway via the S1-U interface and the Mobility Management Entity in the EPC via the S1-MME interface. The two interfaces are collectively known as the S1 interface and lies between the eNodeB and the EPC.

2.2 Evolved Packet Core

The core network, known as EPC in the System Architecture Evolution (SAE), is responsible for the overall control of the UE and establishment of the bearers. The main logical nodes of EPC are:

- Packet Data Network Gateway (PDN GW)
- Serving Gateway (Serving GW)
- Mobility Management Entity (MME)
- Home Subscriber Server (HSS)

PDN GW - The PDN GW is responsible for the connection between EPC and external internet networks, acting as the point of entry and exit for packets. The PDN GW provides internet connectivity and allocates IP addresses for the UEs. The PDN GW also performs packet filtering, *lawful interception*[1] and packet screening for each UE.

MME - The Mobility Management Entity (MME) is the main component in the EPC that processes the signalling between the UE and the Core Network, using the Non-Access Stratum (NAS) protocols. The MME is responsible for paging initiation, authentication as well as allocating resources to the UE upon successful connection [11].

HSS - The HSS behaves like a database in the EPC, storing authentication information of mobile subscribers. It serves an important role during the initial authentication phase as security related information, such as the mapping between IMSI and the authentication key, which is required for the MME to determine if a UE is allowed to be connected to the network.

Serving GW - All IP packets from the UEs are transferred through the Serving gateway, which serves as the local mobility anchor for the data bearers when the UE moves between eNodeBs. It also retains the information about the bearers when the UE is in idle state and temporarily buffers downlink data while the MME initiates paging of UEs to re-establish the bearers.

2.3 LTE Attach Procedure

This section briefly describes how UEs are attached to the LTE network.

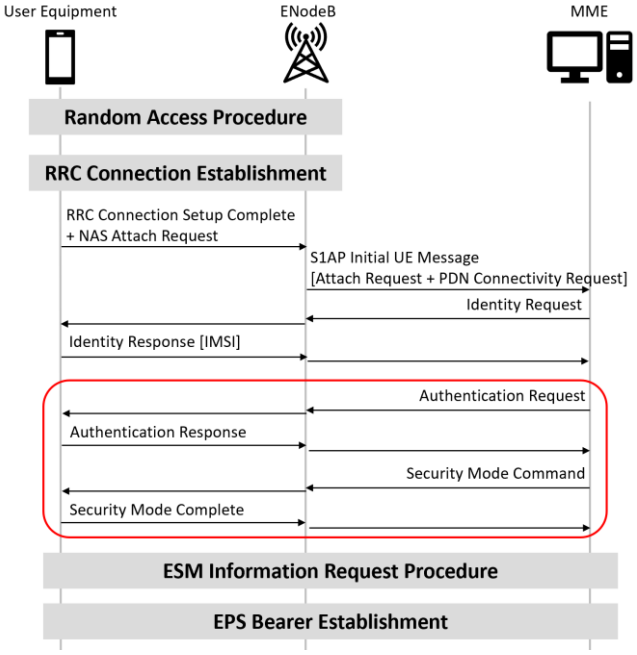


Figure 2. Attach Procedure

The UE is required to perform the LTE attach procedure before it is allowed to connect to the network. Figure 2 is a simplified figure to highlight the key aspects of this attach procedure. The following subsections details these procedures in the LTE protocol.

[1] Law enforcement agencies can wiretap subscribers.

2.3.1 Random Access Procedure

In this procedure, the UE decodes the synchronization signals sent by the eNodeB and synchronizes with it. Next, the UE will read the Master Information Block (MIB) broadcasted by the eNodeB to determine the downlink and uplink carrier information. Downlink is the frequency assigned to the UE and is used to transmit information from the eNodeB to the UE, while uplink is the frequency assigned to transmit information from UE to the eNodeB.

2.3.2 RRC Connection Establishment

Next, the UE would use the uplink frequency to initiate the RRC Connection request with the eNodeB. The *RRC Connection Establishment* allows the UE to inform the network of its purpose of the connection such as *Attach*, *TAU* or *Service Request*.

2.3.3 Attach Request

After the RRC connection is established successfully, to receive network connectivity from the network, the UE will send an *Attach Request* to the eNodeB, which will then forward it with PDN connectivity request to MME. The PDN connectivity request is used by the UE to request for the setup of default Evolved Packet System (EPS) bearer to the PDN. EPS bearer is a virtual connection between the UE and the PDN GW, which allows identification of data sent between these 2 endpoints.

In the case where the UE is connecting to the MME for the first time, it would attach its IMSI in the *Attach Request* message, bypassing the *Identity Request* procedure. Otherwise, it will attach the Globally Unique Temporary ID (GUTI) in the *Attach Request* message. The GUTI is a temporary identifier given to UE in the previous connection with the network.

2.3.4 Identity Request/Response

In this procedure, if the MME cannot find the GUTI sent by the UE, it will send an *Identity Request* message to the UE. The UE will then attach its IMSI in the *Identity Response* message.

2.3.5 Authentication and Setup Encryption

The procedures highlighted in red in Figure 2 shows the process of authentication and encryption. The MME challenges the UE by sending *Authentication Request* message to the UE which contains a random number (RAND) and authentication parameters (AUTN). The UE is required to compute RAND with the AUTN and the secret key stored in the USIM and attach it in *Authentication Response*. If the UE fails this authentication phase, the MME will reject UE's connection by sending an *Authentication Reject* message.

Once the authentication is completed successfully, the MME will initiate a NAS security procedure in which the encryption and integrity protection algorithm is attached to the message. From this point on, all messages will be encrypted based on the parameters provided in integrity protection algorithm.

2.3.6 ESM Request Procedure

The EPS Session Management (ESM) Request Procedure is used by the network to accquire protocol configurations like the Access Point Network (APN). The APN is used to set up a connection gateway between the external internet network and the mobile operator's network.

2.3.7 EPS Bearer Establishment

This procedure indicates that the *Attach Request* has been accepted by the network. An *Attach Accept* message, which consists of the IP address, is sent to the UE and data service is established. Next, the UE will reply with an *Attach Complete*, which signifies the conclusion of the LTE Attach Procedure. Once these procedures have been completed once, in subsequent requests, UE can perform data transfer over encrypted channels without the need for these security procedures again.

2.4 LTE Service/TAU Request Procedure

This section briefly describes how UEs performs *Service Request* and *TAU Request*.

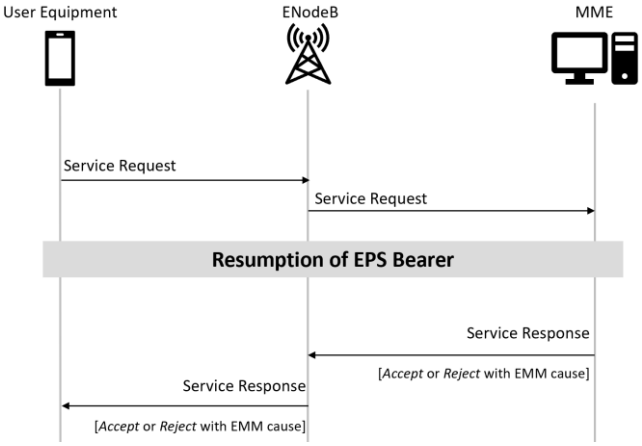


Figure 3. Service Request Procedure

2.4.1 Service Request/Response

UEs will go into RRC IDLE state to conserve power during inactivity. When it has pending data to be sent, it will send *Service Request* to the network. This message requests to resume previously established EPS bearers. The MME will then send the parameters to resume the existing bearers. The MME will reply with a *Service Response Accept* if it is successful and *Reject with EPS Mobility Management (EMM) cause* if it is not. The list of EMM reject cause can be found in the Appendix B.

2.4.2 TAU Request/Response

When a UE detects that it entered a new TA, it will send the *TAU request* to the network. The exchange of messages for *TAU request* and *response* is similar to the *Service Request* process. The establishment of EPS bearers is also similar to that of the *Service Request/Response* process.

2.5 Analysis of LTE Procedures

As discussed earlier, the *Authentication and Setup Encryption* process of the LTE *Attach Procedure* sets up strong encryption and mutual authentication. However, prior to that process, all messages are sent in unencrypted format. This opens up potential attack surfaces which includes passive and active attacks such as IMSI catcher where an attacker can utilize SDRs to eavesdrop packets such as *Attach Request* packets if they are on the same frequencies as the commercial mobile operators. Other forms of attacks include setting up a rogue eNodeB, to disrupt the services of the UE, and that can be achieved once the attacker obtains the configurations of the eNodeB. In order to facilitate the discovery of potential attack surfaces, we propose an experimental setup

(LTE workbench) to gain insights on the procedures of the LTE architecture.

3. EXPERIMENTAL SETUP

In this section, the experimental setup is discussed. To simulate a low-cost setup, open-sourced software and low-cost SDR are used. The experiment setup guide can be found in the Appendix C.

3.1 Software

This section will introduce the software used in the experimental setup.

3.1.1 srsLTE

srsLTE is an open source software suite that contains the required components such as srsUE, srsENB, srsEPC, to simulate a fully working LTE environment.

3.1.2 NextEPC

NextEPC provides the framework required to execute the EPC network. The rogue eNodeB setup in section 4 will be performed using NextEPC.

3.2 Hardware

This section will introduce the hardware used in the experimental setup.

3.2.1 SIM card Reader

The SIM card reader reads the parameters inside the chip in SIM card, and the parameters can be configured to access the simulated environment.

3.2.2 LimeSDR Mini

LimeSDR Mini is a full duplex software defined radio which is priced at around 150 USD and has the capability to receive and transmit frequency ranging from 10MHz to 3.5GHz. The LimeSDR Mini is used as the eNodeB in the environmental setup.

3.2.3 Mobile Phone

The mobile phones acts as the UE in this setup. Phones that were used for testing are *Samsung Galaxy S7, iPhoneX, LG G7 ThinQ.*

3.3 Findings

In this section, the findings of the experimental setup will be discussed.

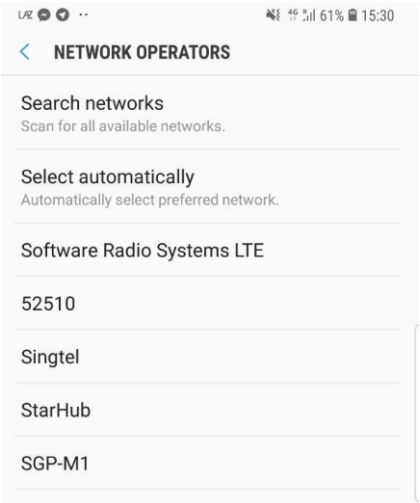


Figure 4. Screenshot LTE network discovery

Figure 4 shows the network name of LTE environment upon successful setup. In order to connect to the Software Radio Systems LTE network as shown above, a programmable USIM was configured to the configurations of our EPC. The configurations can be found in Appendix C. Once the USIM is configured successfully, the user will be able to access the internet and utilize any 4G LTE services. The analysis of the network traffic can also be performed on the EPC and eNodeB to visualise the flow of the aforementioned LTE procedures.

Referring to *LTE Attach Procedure* in Figure 2, after the UE receives the *identity request* message from the MME, it will reply with an *identity response* containing its IMSI. The figure below shows an example of an *identity response* packet.

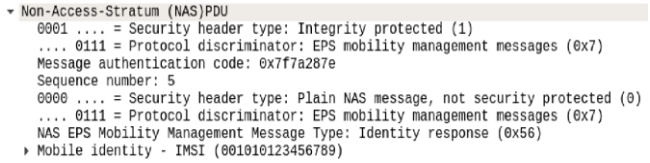


Figure 5. Identity Response Packet

From figure 5, it is clear that the *Identity Response* packet indeed returns the IMSI of the associated UE in plaintext. Therefore, a possible way to capture the IMSI can be achieved by triggering an *Identity Request* such that the UE will respond with its IMSI. The implication of transferring the IMSI in plaintext will be explored further in section 5.

4. ROGUE ENODEB SETUP

In this experiment, NextEPC was used to simulate the EPC instead of srsEPC due to the additional modules to handle *TAU requests*. Refer to appendix D for the rogue eNodeB setup.

In order to simulate a rogue eNodeB setup, LimeSDR Mini was configured to use the same uplink and downlink frequency as Singtel's eNodeB. This experiment was performed within a controlled environment, with the eNodeB in the faraday cage, to prevent it from interfering with the mobile operators' signals.

Mobile phones can access information regarding their mobile operators by going into the Service or Field Test Mode or by using a third-party application. For example, a Samsung phone allows users to access the Service Mode by dialling **#0011#* while an iPhone can access the Field Test Mode by dialling **3001#12345#**. Android phones can also download *LTEDiscovery* from the Google Play Store to access similar information. Figure 6 shows the screenshots for both the *Samsung* and *iPhone* Service Mode and Field Test Mode respectively, for different mobile operators.

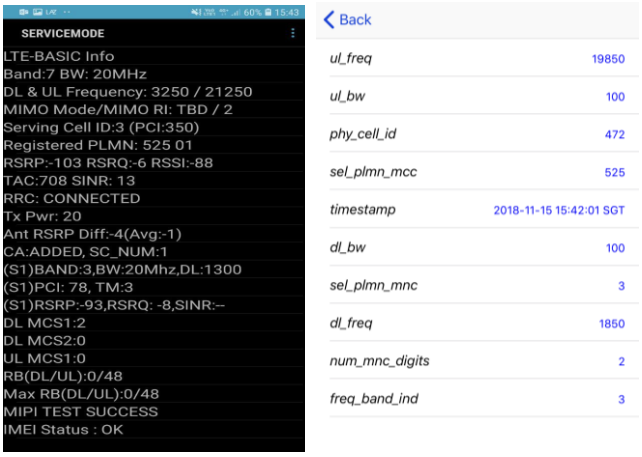


Figure 6. Service Modes of *Samsung* – Singtel (left) and *iPhone* – M1 (right)

From figure 6, information such as the frequency band, uplink and downlink frequency, TAC, Public Land Mobile Network (PLMN) which consists of the MCC and MNC can be found in the Service and Field Test Modes. These information are required to configure the eNodeB and EPC.

Once the eNodeB is configured according to the mobile operator's eNodeB configurations, the UE is then placed in the faraday cage. And since it is a shielded environment, the UE will only receive signals from the eNodeB and attempt to connect to it. As the UE is unable to determine the legitimacy of an eNodeB, it will send a *Service Request* message to our eNodeB, which will then forward it to the MME. However, this *Service Request* message will be rejected as it cannot resume the existing EPS bearers since none are established on our eNodeB. In that case, the MME will respond with a *Service Response Reject* message with EMM reject cause 9 which requires to UE to re-identify itself. Figure 7 below shows the *Service Response Reject* message.

```
Non-Access-Stratum (NAS)PDU
  0000 .... = Security header type: Plain NAS message, not security protected (0)
  .... 0111 = Protocol discriminator: EPS mobility management messages (0x7)
  NAS EPS Mobility Management Message Type: Service reject (0x4e)
▾ EMM cause
    Cause: UE identity cannot be derived by the network (9)
```

Figure 7. *Service Response Reject* Message

5. DOS VIA SERVICE REQUEST / TAU

In the previous section, it is apparent that the eNodeB can trigger responses from the UE using EMM causes.

```
Non-Access-Stratum (NAS)PDU
  0000 .... = Security header type: Plain NAS message, not security protected (0)
  .... 0111 = Protocol discriminator: EPS mobility management messages (0x7)
  NAS EPS Mobility Management Message Type: Service reject (0x4e)
▾ EMM cause
    Cause: EPS services and non-EPS services not allowed (8)
```

Figure 8. *Service Response Reject* with EMM Cause 8

A form of DoS attack that could be launched against UEs is through the *Service Request* procedure where attackers deny UE from accessing the LTE network completely [17]. According to the paper, this attack is achievable due to a vulnerability in the LTE's specifications, in which UEs would accept all "*Service Reject*" messages sent by the network without any authentication and integrity checks. Figure 8 shows EMM reject cause 8 which can be used to perform this attack.

Normally, a *"Service Response Accept/Reject"* message would be sent by the service provider's eNodeB to accept or disallow UEs from accessing services. However, due to the absence of checks as shown in the previous section, attackers can set up a rogue eNodeB that replies to the UEs with EMM reject cause 8, stating EPS and non-EPS services are not allowed [1], and UEs would accept and act according to the message. Under this reject cause, the UE would set its update status to "*EU3 ROAMING NOT ALLOWED*" and will treat the USIM as invalid. As a result, the UE will not look for and connect to LTE networks even if they are available, until it is restarted, or the SIM card is re-inserted. In addition, the UE will also go into the *"EMM-DEREGISTERED"* state, where the location of UE is not known to the MME and thus unreachable.

Above scenario can also be applied for the *TAU Request* message where a malicious eNodeB can reject the UE with EMM cause 8 and cause disruption to the UE's services. In this case, the TAC of the eNodeB has to be different from the mobile operator to trigger a *TAU request* from the UE as explained in *section 2.4.2*.

In *section 3.3*, the project explored how the *identity request* and *response* procedures trigger UE to reply its IMSI in plaintext. This creates opportunity for malicious attackers to perform either targeted attacks, preventing targeted UEs from using any forms of LTE services. Other forms of attack, after capturing the IMSI, includes location tracking with the use of IMSI, but will not be further explored due to the scope of this paper.

6. MITIGATIONS

In this section, suggestions on how 4G LTE network attacks can be mitigated will be presented, along with the justifications.

6.1 Authentication of Base Station

One suggestion is to make use of device-assisted network-based detection [4]. This approach is performed using the collective measurements provided by devices on the network. These devices are constantly collecting information such as the cell identifier and signal strength on the eNodeBs in the vicinity. These measurements will be uploaded to the network and data analysis will be performed to detect anomalies. For example, a rogue eNodeB would be transmitting much higher power signals to lure UEs to connect to them. If the configurations of the eNodeBs are similar, with the exception of signal strength, this might indicate that a rogue eNodeB is present. This approach will work as the network operator knows the network topology, hence they would be able to detect such anomalies. Once a rogue eNodeB is detected, the legitimate eNodeBs can broadcast information about it, preventing UEs from attempting to connect to it.

However, this form of implementation places the pressure on mobile operators and mobile phone developers to build a database of eNodeBs.

6.2 Protecting the IMSI

In the current 4G LTE implementation, if the attacker manages to trigger the UE to respond with its IMSI, the IMSI will be sent in plaintext, and this allows the attacker to perform different types of attacks. Hence, there is a need to protect the IMSI to protect users. One suggestion is to use probabilistic asymmetric encryption scheme [14], which relies on randomness. Using this scheme, encrypting the same IMSI multiple times generates different values. Each mobile operator will generate their asymmetric keys

and the public key will then be pre-provisioned in UEs along with the IMSI. Each time a UE needs to transmit its IMSI, it will generate a new pair of asymmetric keys. Using the mobile operator's public key and the newly generated private key of the UE, an encryption key is generated. This key is then used to encrypt the IMSI. As each encryption key is generated using a different keypair, encrypting the same IMSI will produce different values. This makes it hard for the attacker to identify the IMSI based on the value generated.

However, this would require mobile operators to change all existing USIM and can be difficult to implement. Therefore, this is a suggestion made for the 5G implementation specifications.

7. CONCLUSION

Despite the thoughts placed in 4G LTE protocol implementation specifications, there exists some flaws that can be exploited simply by setting up a rogue eNodeB, which is relatively easy to set up, and it is also reasonably priced. As seen in the experiments above, it is possible to capture the IMSI and deny all services of a targeted UE. Using a SDR with a stronger signal strength, a malicious attacker can potentially deny all services to all UEs in the same TA. However, due to the prevalence of adoption, it is difficult to implement mitigation factors in the current 4G LTE network. Therefore, it is important to design future network protocols with security in mind to reduce risks associated.

8. ACKNOWLEDGMENTS

Our team would like to express our deepest gratitude to Professor Anderson Hugh for his guidance and supervision throughout the entire project, without which our team would not be able to complete this project.

9. REFERENCES

[1] 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Non-Access-Stratum protocol for Evolved Packet System (EPS); Stage 3 (Release 10) https://www.arib.or.jp/english/html/overview/doc/STD-T63v10_10/5_Appendix/Rel10/24/24301-ab0.pdf

[2] An Analysis of DoS Attack Strategies Against the LTE RAN Retrieved from https://www.riverpublishers.com/journal/journal_articles/RP_Journal_2245-1439_323.pdf

[3] Boren, Z. D. (2014, October 07). There are officially more mobile devices than people in the world. Retrieved November 4, 2018, from https://www.independent.co.uk/life-style/gadgets-and-tech/news/there-are-officially-more-mobile-devices-than-people-in-the-world-9780518.html

[4] Detecting false base stations in mobile networks. Retrieved from https://www.ericsson.com/research-blog/detecting-false-base-stations-mobile-networks/

[5] Detection and Remediation of Attack by Fake Base Stations in LTE Networks. Retrieved from http://www.ijsce.org/wp-content/uploads/papers/v5i2/B2576055215.pdf

[6] EPC (Evolved Packet Core). Retrieved from http://www.3gpp.org/technologies/keywords-acronyms/100-the-evolved-packet-core

[7] E-UTRAN (Evolved Terrestrial Radio Access Network) (n.d.). Retrieved November 3, 2018, from http://ecee.colorado.edu/~ecen4242/LTE/e_utran.html

[8] Frequency Check https://www.frequencycheck.com/countries/singapore

[9] Henrydoss, J., & Boult, T. (2014). Critical security review and study of DDoS attacks on LTE mobile network. 2014 IEEE Asia Pacific Conference on Wireless and Mobile. doi:10.1109/apwimob.2014.6920286

[10] Location Disclosure in LTE Networks by using IMSI Catcher. Retrieved from https://brage.bibsys.no/xmlui/handle/11250/2462189

[11] LTE X2 Handover. Retrieved from http://www.3glteinfo.com/wp-content/uploads/2013/12/NMC.LTE-X2-Handover.v1.0.pdf

[12] Master LTE with the help of an LTE Network Diagram. Retrieved from https://www.rcrwireless.com/20140509/evolved-packet-core-epc/lte-network-diagram

[13] Netscout report on DoS trend 2018. Retrieved from https://www.netscout.com/threatreport

[14] Protecting 5G against IMSI catchers. Retrieved from https://www.ericsson.com/research-blog/protecting-5g-imsi-catchers/

[15] Protecting IMSI and User Privacy in 5G Networks. Retrieved from https://pdfs.semanticscholar.org/2161/d05e5858f3144948be8242d25a8711b696f9.pdf

[16] Reference Signal Received Power (RSRP). Retrieved from http://anisimoff.org/eng/rsrp_rsrq.html

[17] Shaik, A., Borgaonkar, R., Asokan, N., Niemi, V., & Seifert, J. (2017). Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems. doi:10.14722/ndss.20

[18] What is Paging in LTE? Retrieved from https://ltebasics.wordpress.com/2015/06/29/what-is-paging-in-lte/

10. APPENDIX A - GLOSSARY

3GPP – 3rd Generation Partnership Project
4G LTE – 4th Generation "Long Term Evolution"
APN – Access Point Network
AUTN – Authentication Parameter
CDMA – Code Division Multiple Access
DoS – Denial of Service
DSP – Digital Signal Processing
EMM – EPS Mobility Management
eNodeB – evolved NodeB
EPC – Enhanced/Evolved Packet Core
EPS – Evolved Packet System
ESM – EPS Session Management
E-UTRAN – Evolved UMTS Terrestrial Radio Access Network
GSM – Global System for Mobile Communication
GUTI – Globally Unique Temporary ID
HSS – Home Subscriber Server
IMSI – International Mobile Subscriber Identity
IRC – Internet Relay Chat
LTE – Long Term Evolution
MCC – Mobile Country Code
MIB – Master Information Block
MME – Mobility Management Entity
MNC – Mobile Network Code
PDN GW – Packet Data Network Gateway
PLMN – Public Land Mobile Network
RAN – Radio Access Network
RAND – Random Number
RRC – Radio Resource Control
RSRP – Reference Signal Received Power
SAE – System Architecture Evolution
Serving GW – Serving Gateway
TA – Tracking Area
TAC – Tracking Area Code
TAU - Tracking Area Update
UE – User Equipment
USIM – Universal Subscriber Identity Module

11. APPENDIX B – EMM REJECT CAUSE

Causes related to UE Identification

Cause #2 – IMSI unknown in HSS

Cause #3 – Illegal UE

Cause #6 – Illegal ME

Cause #9 – UE Identity cannot be derived by the network

Cause #10 – Implicitly Detached

Causes related to subscription options

Cause #5 – IMEI not accepted

Cause #7 – EPS services not allowed

Cause #8 – EPS services and non-EPS services not allowed

Cause #11 – PLMN not allowed

Cause #12 – Tracking area not allowed

Cause #13 – Roaming not allowed in this tracking area

Cause #14 – EPS services not allowed in this PLMN

Cause #15 – No suitable cells in tracking area

More causes can be found in Annex A and B from:
https://www.etsi.org/deliver/etsi_ts/124300_124399/124301/15.03.00_60/ts_124301v150300p.pdf

12. APPENDIX C – LTE EXPERIMENTAL SETUP (WORKBENCH)

The setup environment is as follows at the point of the experiment:

srsepc & srsenb – a laptop with USB 3.0 port
Operating System: Ubuntu 16.04.5 LTE
Software Defined Radio: Lime SDR Mini
Libraries: Soapy SDR v0.70, LimeSuite v18.10, srsLTE commit 1c015aab62a2958d5af5bed36cc859a4290d17e0

List of dependencies needed:

```
#Dependencies needed
sudo apt update
sudo apt install cmake g++ libpython-dev python-numpy swig git libsqlite3-dev libi2c-dev libusb-1.0-0-dev libwxgtk3.0-dev
freeglut3-dev libfftw3-dev libmbedtls-dev libboost-program-options-dev libboost-thread-dev libconfig++-dev libsctp-dev
```

Building SoapySDR from github source

```
#Building SoapySDR
cd ~
git clone https://github.com/pothosware/SoapySDR.git
cd SoapySDR
mkdir build; cd build;
cmake ../
make -j4
sudo make install
sudo ldconfig

#To verify that SoapySDR is installed properly run
SoapySDRUtil --info
```

Building LimeSDR from github source

```
#Building LimeSuite
cd ~
git clone https://github.com/myriadrf/LimeSuite.git
cd LimeSuite/build
cmake ../
make -j4
sudo make install
sudo ldconfig
cd ../udev-rules/
sudo bash install.sh

#To verify that you have installed LimeSuite properly run
LimeUtil --find

#To update firmware of LimeSDR mini
sudo LimeUtil --update
```

Building srsLTE from github source

```
git clone https://github.com/srsLTE/srsLTE.git
mkdir build; cd build
cmake ../
make -j4
sudo make install

#Config files are located at ~/.srs/ after running the following command
./srslte_install_configs.sh

#Enable IP masquerading run
cd ../srsepc/

#eg of INTERFACE_WITH_INTERNET_CONNECTION (eth0, wlp4s0); ifconfig to check
sudo bash srsepc_if_masq.sh INTERFACE_WITH_INTERNET_CONNECTION

#Verify it is forwarding correct
ping -I 172.16.0.1 8.8.8.8
```

Set CPU to performance level

```
sudo apt install cpufrequtils
sudo touch /etc/default/cpufrequtils
sudo sed -i "/GOVERNOR.*/d" /etc/default/cpufrequtils
test -s /etc/default/cpufrequtils && sudo sed -i '$a\GOVERNOR=\"performance\"' /etc/default/cpufrequtils || echo
"GOVERNOR=\"performance\"" | sudo tee /etc/default/cpufrequtils
sudo update-rc.d ondemand disable
sudo init 6
```

To simulate a network operator with cell tower, run in this order

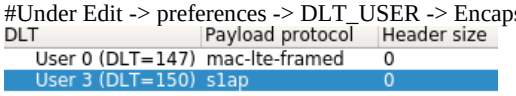
```
sudo srsepc
sudo srsenb

#Config files things to take note, config files are found in ~/.srs/, edit enb.conf and epc.conf
```

- Make sure tac, mcc, mnc is the same for both enb and epc
- mcc 001 and mnc 01 is used for testing network
- enable pcap files if you wish to see packet flow
- **ONLY IN** epc.conf, under [hss], change auth_algo to milenage

```
#To see packet flow, install wireshark and some configurations
sudo apt install wireshark

#Under Edit -> preferences -> DLT_USER -> Encapsulations Table, configure as follows
```

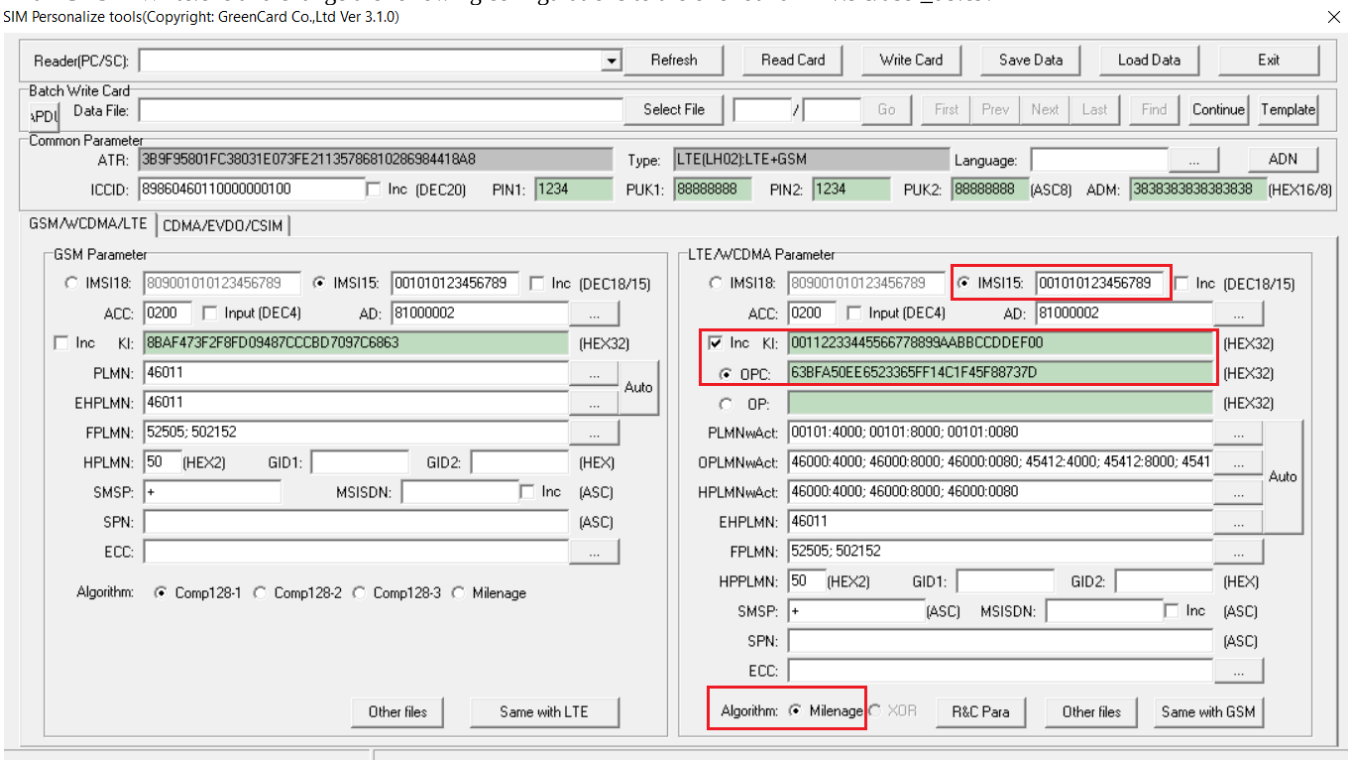


DLT	Payload protocol	Header size
User 0 (DLT=147)	mac-lte-framed	0
User 3 (DLT=150)	s1ap	0

Programmable sim card

#Download the programmable sim card program and run on a windows machine, the link can be found below
https://mega.nz/#!KG5VCSxI!W8-nxXkc5cqQ8Bk1GFvinyNltKJuMZ7kgTIk-xCKaBs

#Run GRSIMWrite.exe and change the following configurations to the one found in ~/.srs/user_db.csv



Mobile phone settings

#Make sure to enable data roaming on the mobile phone to be able to access internet, on Android Phones

Settings -> Connections -> Mobile Networks -> Data Roaming (enable)

#Add APN

Settings -> Connections -> Mobile Networks -> Access Point Names -> Add

APN = srsapn

#Save after this is done

APPENDIX D - MALICIOUS SETUP WITH NEXTEPC

Follow the guide on http://nextepc.org/guides/2-build/ to setup NextEPC.

In the setup, the NextEPC used is of commit c711e788d24e2e9bd5c9da0e4ca49bb8a570fa92.

Config file for NextEPC can be found in ~/nextepc/install/etc/nextepc/, edit the following fields according to the mobile operator's setup

Under (gummei, plmn_id), update mcc and mnc

Under (tai, plmn_id), update mcc and mnc

Under (tai), update tac

Edit the following file to send EMM cause 8 to simulate the DoS attack, ~/nextepc/src/mme/emm_sm.c

Case MME_EVT_EMM_MESSAGE and NAS_TRACKING_AREA_UPDATE_REQUEST

Change the EMM_CAUSE to EMM_CAUSE_EPS_SERVICES_AND_NON_EPS_SERVICES_NOT_ALLOWED. After changing you should see this.

```
if (!MME_UE_HAVE_IMSI(mme_ue))
{
    d_warn("[EMM] Service request : Unknown UE");
    rv = nas_send_service_reject(mme_ue,
        EMM_CAUSE_EPS_SERVICES_AND_NON_EPS_SERVICES_NOT_ALLOWED);
    d_assert(rv == CORE_OK,,
            "nas_send_service_reject() failed");
    FSM_TRAN(s, &emm_state_exception);
    return;
}

if (!SECURITY_CONTEXT_IS_VALID(mme_ue))
{
    d_warn("No Security Context : IMSI[%s]", mme_ue->imsi_bcd);
    rv = nas_send_service_reject(mme_ue,
        EMM_CAUSE_EPS_SERVICES_AND_NON_EPS_SERVICES_NOT_ALLOWED);
    d_assert(rv == CORE_OK,,
        "nas_send_service_reject() failed");
```

```
case NAS_TRACKING_AREA_UPDATE_REQUEST:
{
    d_trace(3, "[EMM] Tracking area update request\n");
    rv = emm_handle_tau_request(
            mme_ue, &message->emm.tracking_area_update_request);
    if (rv != CORE_OK)
    {
        d_error("emm_handle_tau_request() failed");
        FSM_TRAN(s, emm_state_exception);
        return;
    }

    if (!MME_UE_HAVE_IMSI(mme_ue))
    {
        d_warn("[EMM] TAU request : Unknown UE");
        rv = nas_send_tau_reject(mme_ue,
        EMM_CAUSE_EPS_SERVICES_AND_NON_EPS_SERVICES_NOT_ALLOWED);
        d_assert(rv == CORE_OK,,
                "nas_send_tau_reject() failed");
        FSM_TRAN(s, &emm_state_exception);
        return;
    }
```

#Recompile NextEPC after this is completed

Config file for srsenb can be found in ~/.srs/enb.conf, edit the following fields according to the mobile operator's setup

```
Under [enb], update tac, mcc and mnc
Under [rf], update dl_earfcn
```

Run the setup using the following commands

```
#In NextEPC directory, run
./nextepc-epcd

#In srsenb directory, run
sudo srsenb
```