The Vulnerability and Enhancement of AKA Protocol for Mobile Authentication in LTE/5G Networks

Teng Fei, *Student Member, IEEE*, Wenye Wang, *Fellow, IEEE*

***Abstract*—The Long-Term Evolution (LTE)/5G network connects much of the world's population to provide subscriber's voice calls and mobile data delivery, with security provided by the Authentication and Key Agreement defined by 3GPP, which makes the LTE/5G network more secure than all its predecessors. Primarily due to the access limitations of LTE systems, the vulnerabilities of AKA protocol and potential attacks have not received much investigations, which is essential to LTE users with tremendous amount of cellular services. In this study, we focus on two questions: i) what are the vulnerabilities that can be exploited to carry out attacks in practice? and ii) how to design an enhanced AKA protocol against such attacks? We examine the detailed procedures of Evolved Packet Core EPS-AKA protocol by 3GPP, and have identified three types of attacks with respect to *catching*, *location tracking*, and *jamming*. We have designed and implemented attacks with commercial equipment to evaluate their threats in practice. In addition, we propose an enhanced AKA protocol that essentially relies on the asymmetric encryption rather than symmetric in the AKA protocol, and additional digital signatures to countermeasure these attacks and to mitigate the newly found vulnerabilities through formal verification.**

***Index Terms*—Long-Term Evolution (LTE), Authentication Protocol, Analysis and Verification.**

I. Introduction

The Long-Term Evolution (LTE) is the latest widely deployed communication technology that is equipped with high data rate, low delay, and sophisticated security mechanisms. It has created a lot of enthusiasm in both industry and academia. All those novel features and huge amount of subscribers enable new applications such as short message service (SMS), video streaming and authentication, e.g., google verification. By the year of 2023, the total number of subscribers for LTE networks will grow from 5.2 billion in 2019 to 5.7 billion in 2023, among which 46 percent will be LTE subscribers [1]. According to Citrix's latest mobile analytic report [2], 52 percent of all the data usage comes from watching videos, mainly going to YouTube/Google Video, Netflix, etc.

Because of the explosive usage of smart phones for a variety of applications, and more importantly, for text message authentications, LTE subscribers expect security guarantees against malicious attackers. Mutually authenticating subscribers and their carriers for establishing secure channels is one of the most crucial approach to protect subsequent communications. For network generations 3G and 4G, this is achieved by using multiple Authentication and Key Agreement (AKA) protocol in which cryptographic keys will be generated during authentication process and will be used to ensure integrity and encryption. Such strong security protocols make the LTE networks the most secure communication technology compared to its predecessors, such as Global System for Mobile Communications (GSM) in which all devices in the communication environment are assumed to be trustful. In GSM networks, the International mobile subscriber identity (IMSI), a permanent 15-digit number uniquely assigned by an operator [3], is transmitted in plaintext and can be sniffed over the air for leaking identification of subscribers. Also, a rogue base station [4] that costs no more than $200 can be easily setup and trap surrounding subscribers, since the subscribers in the GSM network do not authenticate a network operator.

To tackle the aforementioned vulnerabilities identified in the GSM network, the 3GPP designed an enhanced protocol, namely UMTS-AKA, for UMTS (Universal Mobile Telecommunications Service) network. The UMTS-AKA transmits the TMSI (Temporary Mobile Subscriber Identity) through the air interface instead of the IMSI compared with the GSM network. Since the TMSI changes much more frequently than the IMSI, it becomes harder for an eavesdropper to track a specific subscriber. Also, it is more robust against rogue base stations, since the user equipment (UE) is designed to verify the authentication challenge received from network operators [5]. These improvements protect the UMTS network from efficient attackers in the GSM network, which makes the UMTS-AKA a more secure authentication protocol than the AKA protocol in GSM networks.

However, the TMSI used in UMTS networks still leaves subscribers traceable since the 3GPP does not specify when and how to update temporary identifiers. Therefore, attackers can collect TMSIs that could be persistent for hours or days [6] and reuse the same TMSI, which could lead to the leakage of subscriber's location. In [7], it is found that an attacker can replay messages between the subscribers and the network, which led to severe consequences such as wrong billing and service downgrade. To address these issues, this end, the 3GPP proposes a new authentication protocol to strengthen the authentication process for LTE networks, called EPS (Evolved Packet Core) AKA, in which the GUTI (Globally Unique Temporary Identity) [8] is created to replace TMSI to solve the traceable problem. The major difference between the GUTI and the TMSI is that the GUTI changes in seconds and minutes instead of hours and days. Moreover, the EPS-AKA deploys a serving network ID (SNid) for a home network to protect

This work was supported in part by the NSF under Grant CNS-1824518.

subscribers from the replay and redirection attacks. These enhancements make the LTE authentication the most secure one compared with previous authentication protocols.

With more and more emerging applications and new attacks, the LTE authentication is found to be insufficient, such as "IMSI Paging Attack" [9], in which attackers exploited the paging requests used by operators to inform incoming calls or data services to subscribers. The same procedure is also implemented with IMSI or TMSI, which can lead to the leakage of subscriber's privacy since these identities are transmitted in the plaintext. Furthermore, Denial-of-Service attacks become possible in the LTE network [10], in together of other attacks, showing that the LTE authentication is not sufficient and vulnerable under both passive and active attackers.

The LTE authentication, however, continues to be an open issue in that adversaries can keep exploring vulnerabilities of LTE networks that have not been verified and take the advantage of increasing capacity and processing power to carry out attacks in practical systems. Therefore, our work aim to investigate the fundamental vulnerabilities in the AKA (and EPS-AKA) protocol by examining 3GPP technical specifications closely, and derive proof-of-concept attacks accordingly, and propose the solution with formal verification. In other words, we focus on two questions: i) what are the vulnerabilities that can be exploited to carry out attacks in practice? and ii) how to design an enhanced AKA protocol again such attacks? Our main contributions can be summarized as follows:

- We identify vulnerabilities with respect to serving network identity, UE identity, authentication tokens, and location leakage in the LTE network, assuming that an adversary with minimum privilege in the sense that an individual is able to perform attacks without collaborating with others or have insiders in operators.
- We perform three types of proof-of-concept attacks with respect to catching, location tracking, and jamming, and implement them in an LTE testbed with commercial devices, including oneplus5 smartphones and Amarisoft OTS 100 as an LTE base station, by adding system information block (SIB5) messages wrt latest specifications.
- We propose a new AKA protocol that essentially relies on the asymmetric encryption rather than symmetric in the AKA protocol, and additional digital signatures to countermeasure these attacks in practice. In addition, We also formally verified that the enhanced AKA protocol is able to address the identified vulnerabilities with the verification tool Proverif.

This paper is organized as follows. In Section II, we introduce the LTE architecture and basics of LTE authentication protocol. In Section III, we describe the adversary model, the newly found vulnerabilities, and three types of attacks that can be performed with commercial devices based on the vulnerabilities. In Section IV, we introduce the proposed AKA protocol with enhanced features, along with formal verification to ensure that our approach meet the security goals in expectation. In Section V, we describe related works and limitations of our solution. Finally, in Section VI, we conclude the paper.

II. Preliminaries

We explain in this section how authentication and key agreement protocols are achieved in the LTE network, following as closely as possible the specification 3GPP TS 33.401 [11].

A. LTE Architecture

We consider a simplified version of the LTE architecture, as shown in Fig. 1, involving components required to set up connections between subscribers and networks.

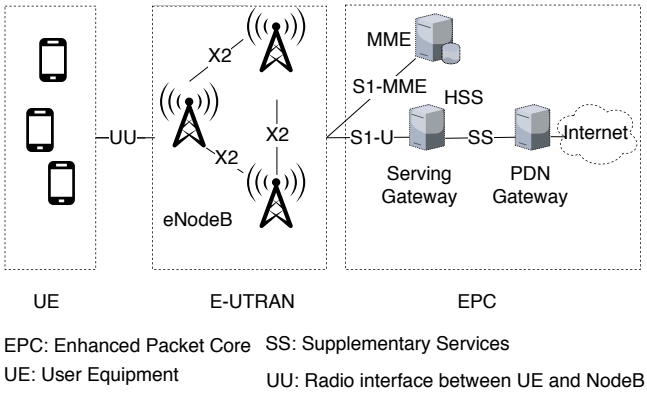


Fig. 1: LTE architecture.

The UE (User Equipment) refers to mobile devices that are equipped with a SIM card, which stores the IMSI, preshared key, and sequence number in it [12]. The E-UTRAN is a geographical area that is consisted of several hexagonal cells, which has a eNodeB (the base station) that can provide LTE services to UEs located in the cell. The EPC (Evolved Packet Core), which consists of HSS (Home Subscriber Server) and MME (Mobility Management Entity), is in charge of providing UEs with network services. The Mobility Management Entity (MME) is responsible for authentication of the mobile device [13] and the HSS stores subscriber's identities along with the cryptographic keys.

B. AKA Protocol

The EPS-AKA protocols, which is used exchangeablly with the AKA protocol thereafter without loosing generality, are specified by the 3GPP to enable secure channels establishment and mutual authenticate between the LTE networks and subscribers. We now describe these protocols in an informal way for readers to easily understand the LTE authentication process, which starts with the initial attach of UE.

As shown in Fig. 2. when a UE initiates the attach procedure, it first sends out the attach request message that contains its identity, which enables the eNodeB to request authentication material from the HSS. Then the HSS will begin to generate the quintet authentication vector [RAND,xRES,CK,IK,AUTN] based on the received identity. As a start, the HSS will begin to generate a random number

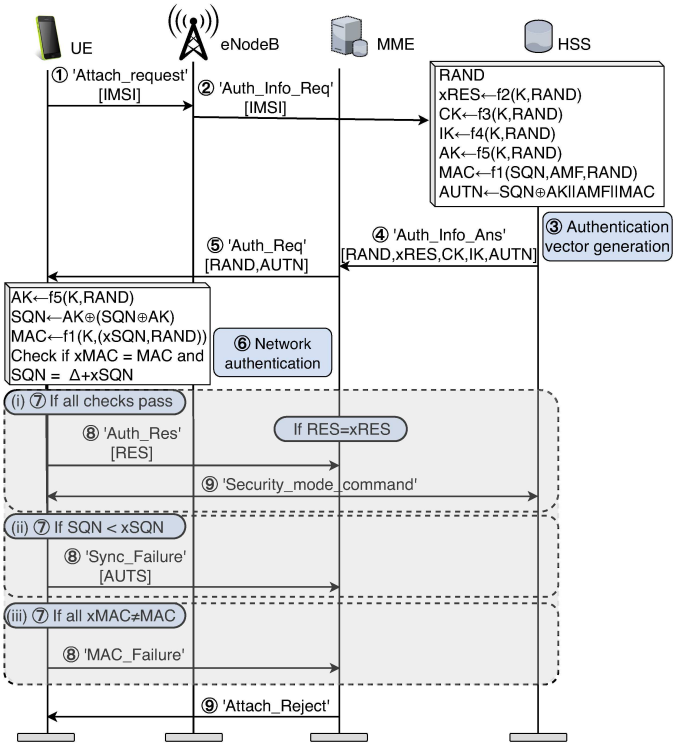


Fig. 2: The EPS-AKA procedure in the LTE network.

called RAND in order to provide randomness for later authentication materials. The xRES is the expected authentication response that should be compared with the response from the UE in order to authenticate the subscribers. The CK and IK are the ciphering key and the integrity key respectively that are used to provide secrecy and integrity. The MAC is calculated by the f1 function by using the concatenation of the SQN, AMF, RAND. The SQN is the sequence number that is incremented by one after each authentication process in order to provide freshness to prevent replay attacks. The AMF is the Access and Mobility Management Function that is configured in operator's database in the Authentication center and USIM. The MAC and the AK along with AMF, SQN will be used as the input to generate the AUTN (Authentication Token). After receiving the authentication vectors sent from HSS, MME will only send AUTN and RAND to the eNodeB in an authentication request message.

Upon receiving the request, the subscriber checks its authenticity and freshness by extracting the xSQN and MAC from AUTN and checks whether the MAC is the correct. If the extracted MAC is not the same as the one stored in the UE, it will reply "MAC_Failure". It also checks if the authentication request is fresh, otherwise the subscriber replies "Sync_Fail" as well as another re-synchronization token called AUTS.

There could be potential vulnerabilities to disclose the information without encryption in the above procedures. Next, we will explore the potential vulnerabilities and attacks in detail and show the messages that can leak subscriber's privacy.

III. Vulnerabilities and Practical Attacks in the LTE Authentication

One of our main objectives of this study is to find out the vulnerabilities that can be exploited to carry out attacks in commercial LTE networks. In this section, we first describe the adversary model, then explore the newly found vulnerabilities. Finally, we elaborate the three types of attacks that can be performed with commercial devices based on the vulnerabilities.

A. Adversary Model

We model the LTE network with the following components including UE, eNodeB, MME and HSS. Among these components, the HSS is absolutely secure and can not be impersonated by an adversary, and there is no way for an adversary to retrieve the messages in the HSS. On the contrary, the remaining components, UE and eNodeB, can not make such promises, which indicates that either UE, the eNodeB or the MME might be a potential adversary. The channels between the UE, eNodeB and the MME is not secure and can be sniffed by the eavesdroppers. Through the eavesdropping, an adversary will be capable of stealing messages between the channels in order implement attacks in the LTE network.

Regarding the aforementioned adversary, we assume it with minimum privilege by which means that an individual is able to perform attacks by using off-the-shelf equipment without collaborating with others or have insiders in operators. The adversary, which can be either a malicious subscriber or a rogue base station, is able to eavesdrop on the downlink broadcast messages transmitted from the legitimate LTE cell to the victim UE. Rather than being a pure eavesdropper, the adversary can also be an active attacker. As an active attacker, it is able to inject, modify and transmit the messages between UE and the network. This is achieved by establishing a rogue base station with the help of the open sourced srsLTE library as the software and USRPX310 as the hardware. To generate attacks to an LTE eNodeB, we will use USRPX310 [14] connected to a laptop through an Ethernet cable to play as the attacker. We will monitor the attacker's signalings by a virtual terminal connected to a PC, run the srsUE application to simulate a LTE subscriber; and we implement the srsENB and srsEPC application together to simulate a rogue LTE cell.

B. Vulnerabilities in the LTE Authentication

The LTE network is designed based on the UMTS network, which aims to provide more functionality and better performance, and also has backwards compatibility at the same time. 3GPP has been making continuous efforts to enhance the AKA protocol that is used by UE and the network to authenticate each other. However, we find that there are still several vulnerabilities that will result in severe attacks, such as identity response message, authentication request message and authentication response message.

Serving network identity disclosure. The SN (serving network) identity is broadcasted in the air within plaintext MIB and SIB messages in order to be identified by the UE, which needs SN's identity to decide whether it should attach to the

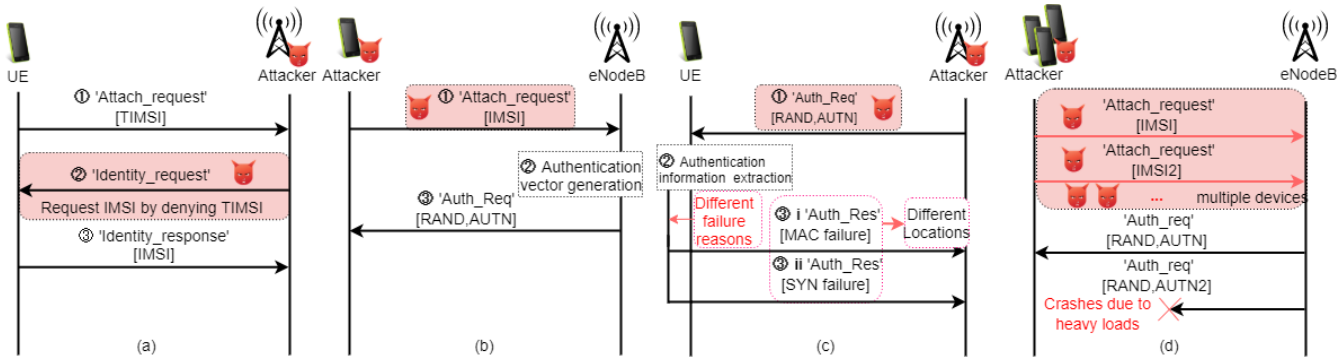


Fig. 3: LTE vulnerabilities and attacks.

network (random access, RRC connection and etc.). These identities can be exploited by adversaries to implement fake base stations, which require a real cell identity that are sniffed through the spectrum from a legitimate eNodeB.

UE identity disclosure. In the LTE network, the 3GPP designed the Globally Unique Temporary Identity (GUTI) to prevent identity disclosure. However, the IMSI is still transmitted in plaintext during initial attach [12]. The GUTI are also traceable, which will further leads to leakage of subscriber's location. In [15], the GUTI has also been allocated all around the world and the results show that the GUTI does not change or fit for some regular patterns.

Authentication token leakage. The key point of AKA protocol in the LTE network is primarily based on the challenge-response mechanism. The challenge is an authentication request message that consists of a random number and an authentication token called AUTN. Since the AUTN is transmitted in the plaintext, it can be sniffed by the passive attacker.

Location leakage. The different responses for authentication failure can lead to subscriber's location leakage. Attackers can repeatedly send the same AUTN to the subscriber, if the user is in the cell of the rogue base station, the authentication failure message will be 'SYN failure', otherwise, the messages will be 'MAC failure'. Based on different authentication failure responses, we can know whether the victim is in the cell or not.

Duty imbalance. For control and batteries considerations, most of the computation during the authentication is completed on the network side. However, this authentication based on the sole control of network side also gives chances for the attacker to perform a special kind of jamming attack. By forcing many mobile devices attach to the network at the same time, the network will suffer from overwhelming traffic and finally impacts the overall performance.

C. The Attacks In Practice

We perform three types of proof-of-concept attacks with respect to catching, location tracking, and jamming, and implement them in an LTE testbed with commercial devices, including oneplus5 smartphones and Amarisoft OTS 100 as an LTE base station, by adding system information block (SIB5)

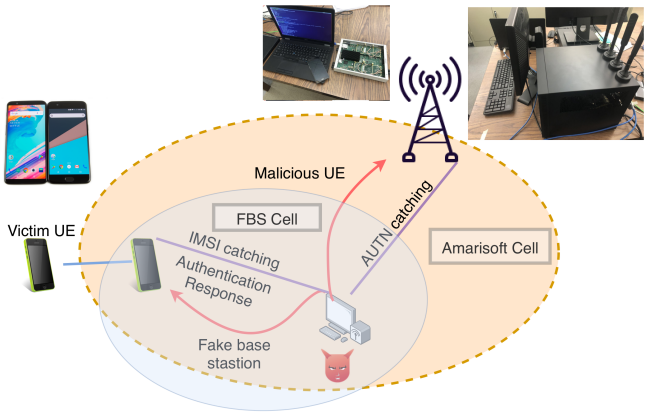


Fig. 4: The experimental set up: The orange cell are operated by the Amarisoft cell and the blue cell are operated by our rogue base station.

messages w.r.t latest specifications. We also show a simplified version of the experimental set up in Fig. 4.

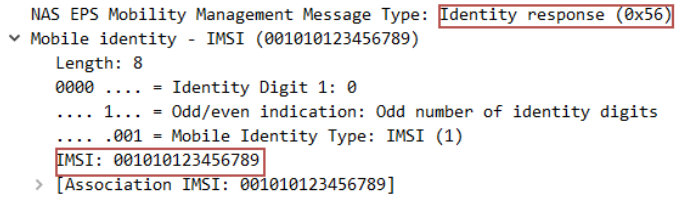
1) Catching Attacks: IMSI and AUTN: The first type of attacks is *catching attacks* which are mainly caused by the disclosure of identity response messages and the authentication request messages sent from the network side. The former is referred to as IMSI catching attack, and the latter is referred to as AUTN caching attack.

IMSI Caching Attack. (*UE identity disclosure*) To implement our attacks, we utilize a mechanism in the LTE network called cell selection absolute priority [16]. The absolute priority mainly refers to high priority frequencies that are mainly transmitted in the SIB5 (System Information Block) messages according to [16]. This approach has been mentioned in [17], however, we use only one SIB5 message instead of four SIB messages. In this way, if we operate the rogue eNodeB to broadcast at a higher priority frequency than the legitimate eNodeB, we can manage to force the UE to attach to our eNodeB.

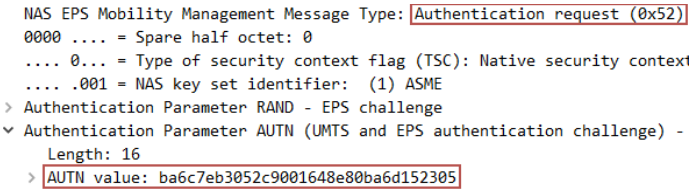
To build our eavesdropper, we modified srsUE application to sniff the SIB messages. As long as we find a cell and retrieve the broadcasting messages, we shut down the srsUE and turn on the srsLTE ENB and EPC. Once the UE tries to build a

connection to our base station, it will send the attach request message. Since our base station does not have the temporary identifier to respond, it will send an identity request message and the UE will reply with the identity response message with its IMSI. The attack is shown in Fig. 3(a) and the Identity Response message is shown in Fig. 5(a).

AUTN Catching Attack. (*Authentication token leakage*) After getting the IMSI of the UE, we turn the srsLTE to a malicious UE with this IMSI. Since this IMSI is also stored in the HSS of the commercial network, the EPC will accept the attach request and calculate the authentication material for the UE with our IMSI. Once received the material, the eNodeB will send our UE with authentication request message that contains the RAND and AUTN. The attack is shown in Fig. 3(b) and the capture AUTN is shown in Fig. 5(b).



(a) Successful IMSI catching attack. The rogue base station received the identity response message that contains subscriber's IMSI sent from the victim.

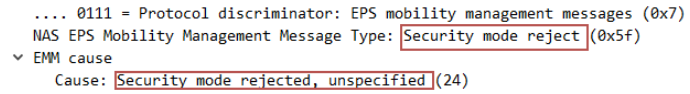


(b) Successful AUTN catching attack. The attacker receives the authentication request message using caught IMSI. The message contains the AUTN and the AWF (9001) that belongs to a legitimate subscriber.

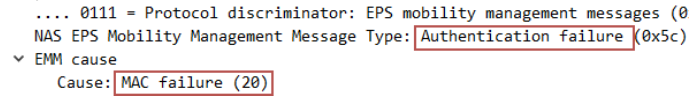
Fig. 5: The attacker successfully catches subscriber's IMSI and AUTN (Authentication token).

2) Location-Tracking Attack (Location leakage): The attack exploit the different authentication response messages to infer the location of the victim UE as shown in Fig. 3(c). It is performed under the assumption that the subscriber's IMSI and its corresponding AUTN are acknowledged by the adversary, which are received through our IMSI and AUTN catching attacks.

When the target UE is inside the cell of our rogue base station, since the MAC of the AUTN is generated by the target's preshared key, the MAC verification will be successful. However, since the SQN in the UE is changed, the target notices that the sequence number extracted from the AUTN is smaller so the target UE will know that the authentication request this time is out-of-sync. So after receiving the *security mode command* message from the adversary, the UE will send back the *security mode reject* message. According to 3GPP specification [18] section 5.4.3.5, the reject is caused actively by the UE, so we can assume that this failure is caused by the



(a) The UE is inside the cell. The authentication failure message indicates that the security mode command is not accepted by the UE, which means that the authentication fails on the UE side because of desynchronization.



(b) The victim is outside of the cell. The authentication failure message indicates that the MAC received by the UE is not the same as calculated by the UE, which means that the authentication fails on the UE side because of failure on MAC verification.

Fig. 6: The differences between authentication failure responses indicate whether the victim is inside a cell or not.

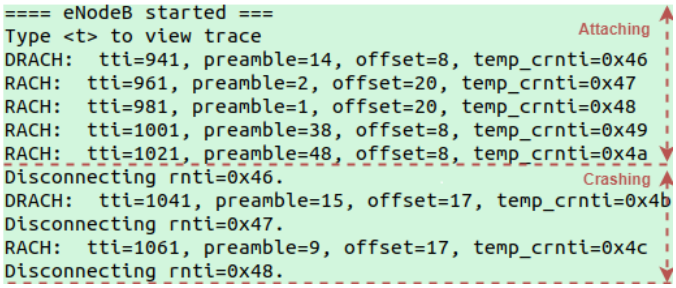


Fig. 7: The eNodeB crashes and disconnects UEs.

desynchronization and thus the victim is inside the cell. The captured message is shown in Fig. 6(a).

When the UE is outside the cell of the rogue base station, it can not receive the message from base station anymore. So what we are trying to do is to see if we can receive the synchronization failure messages again. We use another UE and put it inside the cell. We assume the adversary does not acknowledge any credentials of the UE. We operate our rogue base station to attract the UE and use the same AUTN, RAND to authenticate the UE. Since two UEs do not share the same credentials, this time the UE will reject the authentication with MAC failure message and we know that the victim is not in the cell. The captured message is shown in Fig. 6(b).

3) Jamming Attack (Duty imbalance): We use the srsLTE as the LTE cell, and we make it to transmit on the frequency of the highest priority compared with a nearby commercial cell, which will make the surrounding UEs try to attach to our rogue base station as shown in Fig. 3(d). Since our base station has limited resources compared with a commercial one, it crashes and keeps disconnecting the UEs as shown in Fig. 7. From the figure, we can observe that multiple devices are trying to have random access to the base station, however, the base station is crashed due to the heavy traffic load.

This attack is pretty practical since the LTE requires more base stations as the cell range becomes smaller. As a result, the femtocells become more popular and have various forms for the LTE network. They can be a special designed chip card

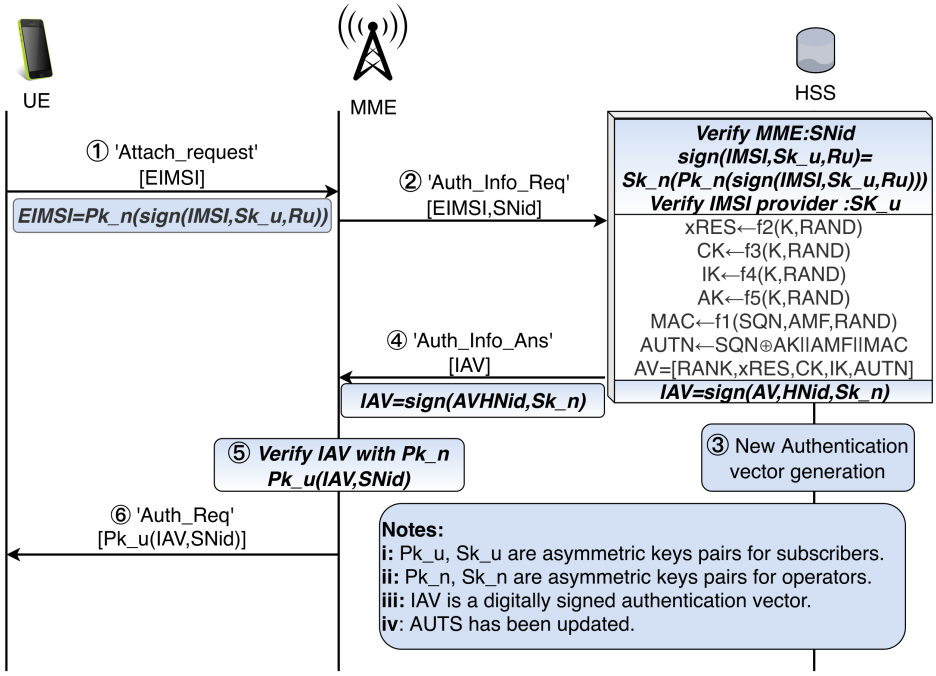


Fig. 8: An enhanced AKA protocol based on the original AKA protocol (Part I).

or your own smart mobile devices. These devices have lower computation resources than a real eNodeB and thus can be easily crashed by lots of traffic.

IV. An Enhanced AKA Protocol and Verification Analysis

In this section we propose a new AKA protocol to address the potential attacks explained in Section II.C. We add asymmetric encryption and digital signatures to provide UE with privacy and integrity. We believe that our solution can be easily adopted to real AKA implementations by any LTE network operators. Recall that subscribers store their public-private pairs in the USIM and operators store its key pairs in the Auc (Authentication Center). The private keys are only known by subscribers and operators, compared by public keys known by anyone. These key pairs will be used in the proposed solution to enable asymmetric encryption and digital signatures. The details of the proposed approach is shown in Fig. 8 and Fig. 9. The new procedures compared with original ones are in the bold font with a blue background.

The UE and network have a preshared key in the original AKA protocol. In order to provide better security and privacy, we design our own AKA protocol that will need public-private key pairs in the UE and HSS. The UE has a key pair that Pk_u serves as the public key and Sk_u serves as the private key. Any entities that have enough capabilities can encrypt a message as long as it possess or acknowledge the public key. However, only the entities that have the private key are able to decrypt those encrypted messages. The HN also stores such public-private keys pairs. We assume that the UE and HSS know each other's key pairs.

Those key pairs are distributed separately as public and private keys. Speaking of the distribution of private keys, since these keys are unique for different UEs, they will be assigned to each UE in the SIM card while the K and IMSI are stored in the SIM card. Regarding the public keys, those keys can be distributed using public certification, the authority (in our case, the network operator) provides a certificate (which binds identity to the public key) to allow key exchange without real-time access to the public authority each time. We can initial put a public key into a SIM card before it is assigned to a UE, and for each time the network operator desires the refreshment of the public keys, it can exploits the public certification mechanism to refresh the public keys after a security mode is set up with the UE.

The first vulnerability we want to tackle is the IMSI catching problem during the initial attach. In the previous AKA protocol, the IMSI is exposed since it will be transmitted in the spectrum in identity response messages. In our solution, We first sign the IMSI with UE's private key to provide authenticity for subscribers by enabling HSS to ensure that the received IMSI is sent from an legitimate subscriber. Since it is not secure to expose the IMSI and private keys in the open spectrum, we encrypt identity response messages using network's public key and called the encrypted token EIMSI. As a result, only the HSS is able to decrypt the attach request message. We also put a random number generated by the UE in the EIMSI in order to defeat the active attacker, who want to expose the subscriber's location by triggering attach requests several times in order to retrieve two same EIMSIs. The new identity response message will be transmitted to the MME.

Once MME receives the attach request message, it originally

send it directly to HSS. However, a rogue MME might be able to perform man in the middle attack between the UE and HSS according to our vulnerabilities. Consequently, we send the EIMSI along with the serving network ID (SNid) to keep the authenticity of the MME. Even though the SNid is forged by an adversary, it can not forge a correct EIMSI thus can not be accepted by the HSS. The authentication information request messages will be forwarded by the MME to the HSS.

The HSS will receive the authentication information request message then it will verify the SNid to determine whether the message is from a legitimate MME or not. Then the HSS will decrypt the first layer of received EIMSI. This procedure has already authenticate the HSS in a superficial manner since it will not be able to decrypt the message if it does not have the required private keys. After decrypting the first layer, the HSS will verify the digital signature inside the EIMSI with subscriber's public key and it can confirm that the message is from the legitimate UE since the provided private keys are also lying in the database of the HSS. In this way, we manage to make the HSS acknowledge the IMSI in the identity response message without exposure and also make sure that it comes from the legitimate UE. After that, our protocol enters the original AKA protocols, which generate the session keys and authentication tokens AUTN and the random number. However, we do not send the authentication token and random number directly to the MME, since there is a vulnerability that the channels between HN and SN might be compromised. As a result, we append the AV (authentication vector) with destination's SNid and then sign them with network's private key. This decision has two benefits. First, the SN id can help the MME to verify the message's authenticity by knowing the HNid and the HNid can be used later for the UE to make sure that the IAV is sent from a legitimate HN. Notice that we will not need to authenticate MME since we already confirm that in the first step. The HSS will forward the authentication information answer message to the MME.

The MME will verify the authenticity of the authentication information answer message with the network public key. If the MME determines that the message is sent from a legitimate HN (home network), then it will encrypt the authentication challenge with subscriber's public key and then sent the challenge the UE. The received message will be decrypted by the private key of the UE, which will verify the source of the messages by the SNid (Serving network id). And the HSS (Home Subscriber Server) will also be verified by the HNid contained in the IAV. After fully authenticate the source of the message, the UE will start the SQN and MAC verification just like in the EPS-AKA.

If all the SQN and MAC verification are successful, we do not need to do anything with current LTE authentication procedure. However, if the authentication fails, we need to consider that the differences between the authentication failure response messages will reveal the location of a specific subscriber. So the authentication failure message should not be recognized and this message should be authenticated by HSS since there might be adversaries keeping sending the same message that was sniffed in the spectrum. In order to address above concerns, we forward the authentication response message with the SNid to make sure that it is the failure message sent from the current session. We also generate the random number from UE to make failure message random, and encrypt it with the operator's public key such that no third party can decrypt the message and force a new one. The random number will also be appended with the message for HSS to do the XOR operation to reveal the EMM cause.

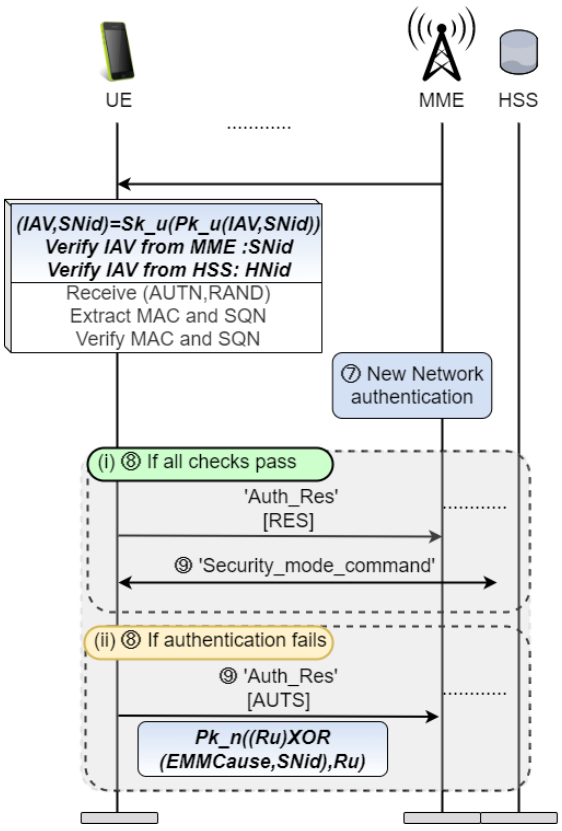


Fig. 9: An enhanced AKA protocol based on the original AKA protocol (Part II).

A. Security Analysis

Our solution successfully addresses the vulnerabilities and attacks proposed in previous sections. First, the attacker is not able to receive the IMSI from attach request messages anymore from the first attack, since we digitally sign the IMSI and also encrypt the message with network's public key even during the initial attach.

Second, the attacker is not able to receive the unencrypted AUTN or use it to perform the second AUTN catching attack. We encrypt the authentication vector with subscriber's public key, so the attacker can not get the AV. And the attacker is not able to replay the message since we append the serving network ID in it. If the UE detects repeated same authentication challenge in a short time with same SNid, it will drop the session since that authentication challenge might come from an adversary.

```
-- Query not attacker(imsi1[])
Completing...
Starting query not attacker(imsi1[])
goal reachable: attacker(imsi1[])

The attacker has the message ~M_983 = imsi1.
A trace has been found.
RESULT not attacker(imsi1[]) is false.
```

(a) The IMSI leakage.

```
-- Query not attacker(Auth_REQ)
Completing...
Starting query not attacker(Auth_REQ)
goal reachable: attacker(Auth_REQ)

The attacker has the message ~M_4093 = Auth_REQ.
A trace has been found.
RESULT not attacker(Auth_REQ) is false.
```

(b) The AUTN leakage.

```
-- Query not attacker(imsi[])
Completing...
Starting query not attacker(imsi[])
RESULT not attacker(imsi[]) is true.
```

(c) IMSI leakage gets fixed.

```
-- Query not attacker(Auth_REQ)
Completing...
Starting query not attacker(Auth_REQ)
RESULT not attacker(Auth_REQ) is true.
```

(d) AUTN leakage gets fixed.

Fig. 10: Anonymity problems found in the LTE network (Fig (a), (b)). In these cases, the attacker is able to retrieve the IMSI and AUTN which should only be known by the subscriber and network. Anonymity problems fixed by our solution (Fig (c), (d)). The attacker is not be able to retrieve the IMSI and AUTN anymore.

We also fix the third attack location leakage caused by the different authentication failure response. The EMM cause is now random and encrypted to make sure that the attacker is not able to identify the detailed EMM cause. Meanwhile, the SNid is also appended in the message for two purposes: 1) The network will be able to make sure that the response comes from a specified eNodeB, which helps to prevent receiving a fake response. 2) The network will be able raise a warning if too many authentication failures come from the same eNodeB, since there might be a rogue base station involved.

B. Formal Verification

In this study, we formally verify our solution using ProVerif [19], which is an automatic cryptographic protocol verifier with a preset attacker model (so called Dolev-Yao model). We will first introduce the basic structure of the Proverif.

1) Modeling: The Proverif provides an environment that a Delov-Yao style [20] adversary is considered to be part of the model. This adversary has the ability to receive, send and modify messages transmitted over the open channel. And it can not violate any cryptography functions, which means that it can not crack an encrypted message without a proper key. There are three major entities in the LTE network in terms of communication purposes, the UE, the SN (Serving network) and HN (Home network). Unlike the modeling in [9] and [21], we do not combine the serving network and home network as one entity since the data sent from the HN is not exactly the same as the data sent from the SN. The channels in the LTE network are simplified to two channels between UE ,SN and HN respectively, called US and HS channel.

2) Typed Pi Calculus: We use typed Pi calculus to precisely demonstrate the solution that we used to solve the vulnerabilities. The Table II shows the syntax that we used in our implementation. For a more detailed introduction, readers can turn to [19] for a comprehensive version. The main process, denoted by P and Q, is the entities that can send and receive messages, which are UE, SN and HN in our implementation. !P means that the process P can repeat itself , which enables a UE to send messages to the network multiple times. Besides main process, the syntax also contains the bitstring as predefined type, which is exactly the same type of the messages transmitted in the LTE network. So we define all the IMSI and authentication materials as the bitstring type. All the defined variable can be set to private in order to prevent attacker from acknowledging it directly. After defining basic entities and messages, we need to transmit those materials between different entities. For example, the message out(M,N):P implies that the message N is sent from P through channel M.

```
The attacker has the message Auth_RES.
A trace has been found.
RESULT not attacker(Auth_RES) is false.
```

(a) The authentication response leakage.

```
-- Query not attacker(sres[])
Completing...
Starting query not attacker(sres[])
RESULT not attacker(sres[]) is true.
```

(b) The authentication response leakage gets fixed.

Fig. 11: The location tracking problem get fixed by our solution. The attacker is not be able to retrieve authentication response message.

3) Results and Observations: The result of ProVerif with general explanation is summarized in Table I. First of all, we can see that the attacker can catch the IMSI and AUTN in Fig. 10(a) and 10(b), which proves that there is a IMSI and AUTN leakage problem in current LTE networks. After implementing our solutions, we run the proVerif tool again. This time the attacker fails to acknowledge those materials according to the verification results in Fig. 10(c) and 10(d).

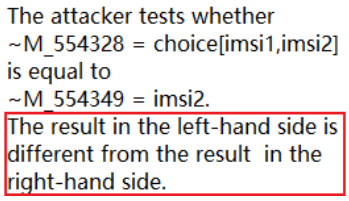
The authentication failure messages (which we indicate it using sres) is being encrypted and not being revealed anymore as shown in Fig. 11.

Proverif output	Output interpretation	The specific meaning in our experiment	Notable Implication If Any
RESULT not attacker is True.	The attacker has not been able to obtain the variable in the query.	The attacker can not reveal the subscriber's IMSI.	None
RESULT not attacker is False.	The attacker has been able to obtain the variable in the query.	The attacker can reveal the subscriber's IMSI.	IMSI Catching Location Tracking
RESULT observational equivalence is True.	The attacker can not distinguish the two variables in different sessions.	The attacker can not distinguish two subscribers.	None
RESULT on two sides are different.	The attacker can distinguish the two variables in different sessions.	The attacker can distinguish two subscribers.	AUTN Catching Location Tracking

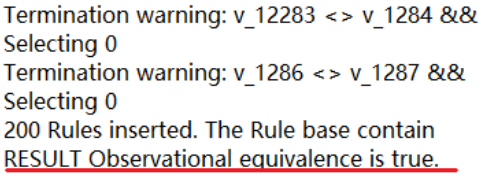
TABLE I: Proverif output interpretation.

P , Q ::=	Main process.
P | Q	Parallel composition
!P	replication
let R=P	Sub process.
free A: K	Declare a new variable
If M= N then P else Q	Conditional
in(M, x : t); P	message input
out(M, N) ; P	message output

TABLE II: The basic ProVerif calculus syntax.



(a) The IMSI is traceable since the attacker is able to distinguish cases that there are two different subscribers in the system.



(b) IMSI traceable fixed: The attacker is not able to tell the difference between two subscribers because of random identifiers.

Fig. 12: The attacker is not able to track the subscriber anymore after implementing our solution.

We also prove that the IMSI traceability problem is solved by our solution. From Fig. 12(a), we can see that the attacker is able to distinguish two subscribers from multiple communication sessions. After adapting our solutions, we can see that the attacker can not differentiate the subscribers anymore, as shown in Fig. 12(b).

C. Performance evaluation

We have proved that our solution reaches the security goals that are set up for the vulnerabilities that we found. However, there are still concerns regarding whether the asymmetric encryption will spend too much computation resources of the system and slow down the running time. In our solution, we try to make our protocol as efficient as possible by striking exactly the vulnerabilities in the LTE network. In order to prove that our solution is efficient even after applying public key algorithm, we simulate 5 AKA protocols and compare their running time.

Unlike the verification part, we do not treat SN and HN as separate parts since the channels between the SN and the HN are usually wired connections, which cost much less time during the authentication procedure compared with that in the air interfaces between the SN and UE. As a result, it does little difference whether we simulate the running time in that part whether or not.

To start simulating the authentication in the LTE network, it is critical to consider how to model some basic elements, such as, the KDF function. The AKA protocol depends on the KDF function to generate keys and tokens during the process, but the original algorithms for commercial networks are quite complicated [5] to follow. Fortunately, the 3GPP provides a test KDF algorithm with similar cost of resources [22] as well as easier implementation. Our KDF is implemented specifically according to this algorithm.

We also need to consider how to model the symmetric and asymmetric encryption algorithm in the other four AKA protocols since there are no such algorithms involved in the original AKA protocols. In order to address this problem in our implementation, we choose the Advanced Encryption Standard (AES) as symmetric encryption and RSA (Rivest–Shamir–Adleman) as asymmetric encryption. Our metric for evaluating the performance is time so we record the program running time on the UE side.

	Time Cost on UE side (Nano Seconds)	Increase of time cost compared with Original AKA protocol
Original AKA	48,659,834	-
EIMSI-AKA	275,378,223	4.6592
PMSI-AKA [21]	48,723,232	0.0013
HEPS-AKA [23]	351,840,504	6.2306
PEPS-AKA [24]	137,356,487	1.8227

TABLE III: The above table is the time consumption of five AKA protocols. The EIMSI-AKA is our solution proposed in this paper.

We conclude our result in table III. The first column shows the five AKA protocols that we have implemented, and the second column records the running time (in nano seconds) on the UE side for a whole authentication procedure, which starts from the attach request sent from UE to the

authentication success appeared on the network side. In the third column, we show the increase running time comparing to the original AKA protocol. From the table we can see that the time consumption increase order is AKA<PMSI-AKA<PEPS-AKA<EIMSI-AKA<HEPS-AKA. Our solution uses public key algorithm and digital signatures. As a result, the running time for our solution is quite high. This is quite reasonable since asymmetric encryption always costs much more system resources. However, we cost much less time than the HEPS-AKA since we implement these algorithms accurately at the found vulnerabilities instead of covering the entire LTE network. Our solution does not do the best in terms of time among the protocols in the table, however, though the PMSI-AKA is quite efficient according to the simulation results, it does not reach the security goal that we set up. First, although the PMSI is used to tackle the IMSI catching issue, it does not prevent this attack in initial attach that transmits the IMSI in the plaintext message. And it does not have any protection for the AUTN, which makes the solution fail to satisfy our security goal.

V. Related work

Regarding IMSI catching issues, Broek et al. [21] proposed a light-weighted enhanced LTE AKA protocol to protect subscriber's IMSI by using an updated VIMSI that is changed every time when subscriber is authenticated by the network operator. However, the IMSI catching issue is only addressed when UE has been authenticated by the network before, which the IMSI still be vulnerable when the UE initially connects to the network for the first time. This issue will be a potential vulnerability in terms of subscriber's location and privacy.

Regarding linkability issues, Borgaonka et al. [25] found the vulnerability that the SQN might be revealed if there is a replay attack generated by a rogue base station. They proposed a solution that uses a symmetric encrypted SQN. However, even with the secured SQN, the different authentication failure response messages can still expose subscriber's location.

Compared with previous works, our solution successfully addresses the aforementioned problems. We use the asymmetric public key to encrypt the IMSI, even during initial attach, which makes it impossible for an attacker to get the IMSI. The IMSI traceablity problem is also fixed by appending random numbers generated by the UE to the identity. And we also address the location tracking by making the authentication failure response message random and encrypted. As a result, the adversary who learns the victim's location by telling the difference by those messages can not do that anymore.

VI. Discussions

In this section, we discuss the limitations, which are the vulnerabilities that we do not address in this work and propose possible mitigation. We also have a short discuss on practical issues when considering implementing our enhancement to the current LTE network.

A. Limitations

First, the jamming attack can happen if there is a task overflow on the network side. In order to protect subscribers from this jamming attack, we propose two mitigation. First, a self-organizing network (SON) should be a brilliant solutions to address this vulnerability. This kind of network is able to detect a traffic overflow on a single base station and assign it to surrounding base stations. We can also consider the mobile data offloading, which the base station should decide to assign some tasks to the UE. If the UE does not have enough computation resources and energy, it can offload the tasks to its connected devices, such as smart watches or a remote server.

Second, rogue base stations exploit the serving network identity disclosure to attract the legitimate UEs. We can sign the broadcasting messages to protect its integrity, so that the UE provided with a certificate can tell whether the messages are coming from a legitimate base station or not.

Finally, our solution verification relies on the result of the formal verification tool ProVerif. Though the tool has been used by several applications, the result heavily depends on the modeling of the LTE AKA protocol, which limits our work to a simulation level. For future work, a real and practical implementation with the commercial devices should be considered, which will make the solution more convincing.

B. Practical Issues

Compared with the original AKA protocol, the base station now needs extra computation resources for asymmetric and digital signature algorithms. According to [23], the time consumption for asymmetric decryption, which is 0.646 ms/bytes, is an extra consumption in our new protocol since there is no decryption algorithm in the original one. Since most decryption happens at the network side, the base station might need to put more computation resources on those algorithms. In terms of power consumption, the asymmetric cryptography has a power consumption of 11.6uWs/byte. So the base station should also prepare for a larger power consumption compared with previous conditions.

C. Implementation in 5G Networks.

The 5G standard proposes a new 5G-AKA protocol with an asymmetric randomized encryption. The SUPI is used instead of GUTI to assure the secrecy of subscribers' identities, which is a very similar approach compared with this paper. And the rest of 5G AKA protocols exploit the same structure as the AKA protocols in LTE networks. This same structure has pros and cons, which is on one hand, the vulnerabilities in LTE networks remain in 5G networks. For example, there is still no randomness in authentication failure messages and this will induce location tracking attacks. However, on the other hand, our solution in LTE networks can benefit from the similar structure, which make it very easily to be adopted into current 5G network implementation.

VII. Conclusion

We identify four vulnerabilities in LTE networks and perform three types of proof-of-concept attacks with commercial devices. In order to address the vulnerabilities, we propose a new AKA protocol that essentially relies on the asymmetric encryption and digital signatures. We also formally verified that the enhanced AKA protocol is able to address the identified vulnerabilities with the verification tool Proverif while the system performance remains on an acceptable level.

References

[1] Cisco. Cisco annual internet report (2018–2023) white paper. Online, 2020. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[2] Citrix. Mobile analytics report 2015. Online, 2020. https://www.citrix.com/content/dam/citrix/en$_us/documents/products-solutions/citrix-mobile-analytics-report-february-2015.pdf$.

[3] 3GPP. Numbering, addressing and identification. Technical Specification (TS) 23.003, 3rd Generation Partnership Project (3GPP), 2015. Version 12.5.0.

[4] Simone Margaritelli. How to build your own rogue gsm bts for fun and profit. https://www.evilsocket.net/2016/03/31/how-to-build-your-own-rogue-gsm-bts-for-fun-and-profit/. Accessed: 2016-03-13.

[5] 3GPP. Security architecture. Technical Specification (TS) 33.102, 3rd Generation Partnership Project (3GPP), 2013. Version11.5.1.

[6] Myrto Arapinis, Loretta Ilaria Mancini, Eike Ritter, and Mark Ryan. Privacy through pseudonymity in mobile telephony systems. In *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014*, 2014.

[7] Y. Huang, C. Y. Shen, S. Shieh, H. Wang, and C. Lin. Provable secure aka scheme with reliable key delegation in umts. In *2009 Third IEEE International Conference on Secure Software Integration and Reliability Improvement*, pages 243–252, July 2009.

[8] 3GPP. Digital cellular telecommunications system. Technical Specification (TS) 33.220, 3rd Generation Partnership Project (3GPP), 2012. Version 11.4.0.

[9] Myrto Arapinis, Loretta Mancini, Eike Ritter, Mark Ryan, Nico Golde, Kevin Redon, and Ravishankar Borgaonkar. New privacy issues in mobile telephony: Fix and verification. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 205–216, New York, NY, USA, 2012. ACM.

[10] J. Henrydoss and T. Boult. Critical security review and study of ddos attacks on lte mobile network. In *2014 IEEE Asia Pacific Conference on Wireless and Mobile*, pages 194–200, Aug 2014.

[11] 3GPP. Security architecture. Technical Specification (TS) 33.401, 3rd Generation Partnership Project (3GPP), 2018. Version 15.5.0.

[12] 3GPP. Characteristics of the Universal Subscriber Identity Module (USIM application). Technical Specification (TS) 31.102, 3rd Generation Partnership Project (3GPP), 2017. Version 12.5.0.

[13] 3GPP. Evolved Packet System (EPS); Mobility Management Entity (MME) and Serving GPRS Support Node (SGSN) related interfaces based on Diameter protocol). Technical Specification (TS) 29.272, 3rd Generation Partnership Project (3GPP), 2018. Version 15.5.0.

[14] Wikipedia contributors. Universal software radio peripheral — Wikipedia, the free encyclopedia, 2018. [Online; accessed 17-December-2018].

[15] Byeongdo Hong, Sangwook Bae, and Yongdae Kim. Guti reallocation demystified: Cellular location tracking with changing temporary identifier. In *Symposium on Network and Distributed System Security (NDSS). ISOC*, 2018.

[16] 3GPP. User Equipment (UE) procedures in idle mode. Technical Specification (TS) 36.304, 3rd Generation Partnership Project (3GPP), 2012. Version 9.9.0.

[17] Altaf Shaik, Jean-Pierre Seifert, Ravishankar Borgaonkar, N. Asokan, and Valtteri Niemi. Practical attacks against privacy and availability in 4g/lte mobile communication systems. In *23rd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, California, USA, February 21-24, 2016*, 2016.

[18] 3GPP. Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS). Technical Specification (TS) 24.301, 3rd Generation Partnership Project (3GPP), 2018. Version 15.4.0.

[19] Bruno Blanchet and Vincent Cheval. Proverif: Cryptographic protocol verifier in the formal model. https://www.ettus.com/all-products/x310-kit/ . Accessed: 2019-05-24.

[20] D. Dolev and A. C. Yao. On the security of public key protocols. In *Proceedings of the 22Nd Annual Symposium on Foundations of Computer Science*, SFCS '81, pages 350–357, Washington, DC, USA, 1981. IEEE Computer Society.

[21] Fabian van den Broek, Roel Verdult, and Joeri de Ruiter. Defeating imsi catchers. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 340–351, New York, NY, USA, 2015. ACM.

[22] 3GPP. Common test environments for user equipment. Technical Specification (TS) 33.401, 3rd Generation Partnership Project (3GPP), 2018. Version 15.5.0.

[23] J. Zhou, M. Ma, and S. Sun. A hybrid authentication protocol for lte/lte-a network. *IEEE Access*, 7:28319–28333, 2019.

[24] S. B. M. Baskaran, G. Raja, A. K. Bashir, and M. Murata. Qos-aware frequency-based 4g+relative authentication model for next generation lte and its dependent public safety networks. *IEEE Access*, 5:21977–21991, 2017.

[25] Ravishankar Borgaonkar, Lucca Hirschi, Shinjo Park, and Altaf Shaik. New privacy threat on 3g, 4g, and upcoming 5g aka protocols. Cryptology ePrint Archive, Report 2018/1175, 2018. https://eprint.iacr.org/2018/1175.