Putting LTE Security Functions to the Test: A Framework to Evaluate Implementation Correctness

David Rupprecht
Horst Görtz Institute for IT-Security
Ruhr-University Bochum, Germany
david.rupprecht@rub.de

Kai Jansen
Horst Görtz Institute for IT-Security
Ruhr-University Bochum, Germany
kai.jansen-u16@rub.de

Christina Pöpper
New York University Abu Dhabi
christina.poepper@nyu.edu

Abstract

Long Term Evolution (LTE) is the most recent generation of mobile communications promising increased transfer rates and enhanced security features. It is todays communication technology for mobile Internet as well as considered for the use in critical infrastructure, making it an attractive target to a wide range of attacks. We evaluate the implementation correctness of LTE security functions that should protect personal data from compromise.

In this paper, we focus on two security aspects: user data encryption and network authentication. We develop a framework to analyze various LTE devices with respect to the implementations of their security-related functions. Using our framework, we identify several security flaws partially violating the LTE specification. In particular, we show that *i*) an LTE network can enforce to use no encryption and *ii*) none of the tested devices informs the user when user data is sent unencrypted. Furthermore, we present *iii*) a Man-in-the-Middle (MitM) attack against an LTE device that does not fulfill the network authentication requirements. The discovered security flaws undermine the data protection objective of LTE and represent a threat to the users of mobile communication. We outline several countermeasures to cope with these vulnerabilities and make proposals for a long-term solution.

1 Introduction

Mobile communication has become an integral part of our daily life. Most applications running on today's smartphones or tablets are based on the ability to wirelessly communicate with the surrounding network infrastructure. Long Term Evolution (LTE) is the latest mobile communication standard and predicted to comprise about 2.3 billion subscribers by 2019 [29].

The security of today's mobile communication is essential for the protection of personal data to prevent compromise of users as well as service providers. Security-related changes need to be carefully evaluated in order to maintain strong protection, especially since LTE is considered as the communication technology for, e. g., critical infrastructures, emergency services, and law enforcement [12]. The increasing dependency on LTE makes it an attractive subject to a wide range of attacks and any vulnerability could have serious impact on the underlying services.

By addressing several security issues of the previous mobile communication specifications, in particular Global System for Mobile Communications (GSM) and Universal Mobile Telecommunications System (UMTS), LTE raises the bar in terms of security and applies, e. g., mutual authentication, longer encryption keys, extended key hierarchies, and a public review of the used encryption algorithms. However, individual deployments across different manufactures can lead to gaps between specification and implementation. These implementation flaws can be exploited by an adversary circumventing security features defined in the specification. All these attack vectors need to be examined carefully to evaluate the actual security of LTE devices.

Prior work has addressed several aspects in this landscape. For instance, in 2016 Shaik *et al.* [28] presented attacks against the location privacy of mobile subscribers and the availability of the network based on vulnerabilities in the LTE specification. Further security flaws manipulating the firmware of the baseband chip were identified by Komaromy and Golde [9, 31]. Similar work has been done by Broek *et al.* [30] introducing a testing framework solely focusing on implementations of the SMS layer of GSM. The latter attacks directly focus on the implementation rather than the specification—an approach that we leverage on in this work by developing a framework that analyzes implementation correctness. To date, there is no testing framework openly available for investigating the security of LTE devices.

We develop such a testing framework and focus on a

testing approach with two main security aspects: (i) user data encryption and (ii) network authentication. We use the gained insights to develop a test framework for LTE devices that enables us to semi-automatically test LTE devices on implementation flaws in their security functions. As we will show, *all* tested devices violate the specification with respect to security-relevant ciphering indication. Furthermore, implementations of common LTE devices differ to the extent that network authentication can be broken by a new MitM attack.

In summary, our work provides the following contributions:

- We identify which parts of the LTE specification are susceptible to implementation flaws. We narrow down our investigations to security-critical functions, in particular user data encryption and network authentication.
- We develop a lost-cost framework for testing modern off-the-shelf LTE devices to identify implementation flaws for about 1200 €. The framework is based on commercially available Software Defined Radios (SDRs) [10] and the OpenAirInterface project [11]. To the best of our knowledge, this is the first testing environment for the LTE protocol stack available for academic purposes.
- We tested a total of 10 devices from six different manufacturers. Our test results reveal an exploitable attack vector. As a proof of concept, we launch a MitM attack against one particular LTE device. With this attack it is possible to intercept the data traffic and violate the user privacy.
- We outline countermeasures to reinforce the security and the protection of user privacy. Moreover, we suggest to make changes in the selection of security algorithms for upcoming generations of mobile communication.

We plan to make our framework publicly available once the review process is finished to enable comprehensive tests on the security of available LTE equipment. We have also informed the manufacturer of the vulnerable LTE device who created CVE-2016-3676 [18] and fixed the issue.

2 LTE Background

In this section, we introduce the LTE system architecture which is often technically referred to as Evolved Packet System (EPS). In the remainder of this paper, we mainly use the more common term LTE. Contrary to GSM and UMTS, LTE is based on packet switching rather than circuit switching.

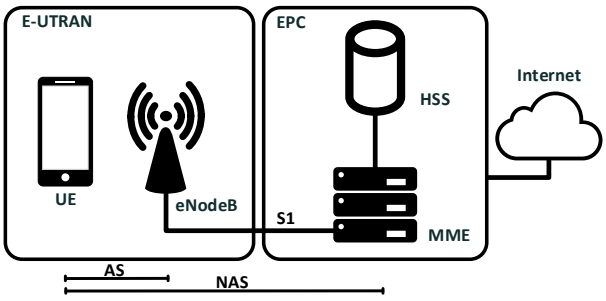

Figure 1: The LTE architecture at least comprises an eNodeB, an MME, and an HSS. An LTE-capable UE can communicate with the network.

2.1 LTE Architecture

The LTE infrastructure is a combination of several components and their interconnections. We briefly introduce the functionality of the network entities and highlight security properties. Notably, we confine this list to components that are of interest in our security analysis.

User Equipment (UE). The UE provides applications and services to the user and is responsible for the data transmission to and from the network. The UE holds a Universal Subscriber Identity Module (USIM), which stores the International Mobile Subscriber Identity (IMSI) uniquely identifying each user. Moreover, a pre-shared key K is stored on the USIM to derive further keys used during the authentication procedure.

Evolved NodeB (eNodeB). The deployed base stations that realize radio-level access points to LTE networks are called eNodeB. Furthermore, each eNodeB is involved in the process of radio management data encryption and integrity protection as well as user data encryption.

Mobility Management Entity (MME). The MME handles the establishment of new connections and the authentication process. It is assigned to manage mobility data of connected UEs, where it applies encryption and integrity protection.

Home Subscriber Server (HSS). The HSS stores the authentication information of the mobile subscribers. Thus, it plays a central role during the initial authentication procedure of an unconnected UE providing the MME with security-related user information.

An overview including the interconnections of the individual components is depicted in Figure 1. The LTE architecture can be divided into two logical domains. The Evolved UMTS Terrestrial Radio Access Network (E-UTRAN) encompasses connected UEs and the eNodeB base stations, whereas the Evolved Packet Core (EPC) contains, e. g., the MME and the HSS, among other components. The two domains are connected via the S1 interface. Furthermore, there are two

Table 1: Security algorithm options, according to [3, 9.9.3.23 NAS security algorithms].

Encoding	Integrity	Ciphering	Algorithm
X000X000	EIA0	EEA0	NULL
X001X001	128-EIA1	128-EEA1	SNOW 3G
X010X010	128-EIA2	128-EEA2	AES
X011X011	128-EIA3	128-EEA3	ZUC

important logical connections between the network components, which are specified as follows.

Access Stratum (AS). The AS represents the direct connection between UEs and eNodeBs and comprises all messages, i. e., *user data* that are exchanged on the radio layer to physically access an LTE network.

Non-Access Stratum (NAS). The NAS is the logical connection between the UE and the MME. The communication between these two parties takes place via NAS *management data*, e. g., to initiate communication sessions including mobility and identity management.

2.2 LTE Security

On both AS and NAS layer, a variety of security procedures are implemented to provide protection against malicious attacks. For instance, the AS layer is responsible for the security of user data and radio management data, whereas the NAS layer handles the security of mobility management messages.

2.2.1 Security Algorithms

Protecting messages on the AS and NAS layers can be achieved with a choice of predefined security algorithms. Two important selections from an algorithm set have to be made to determine the encryption and the integrity protection of exchanged messages. By now, four different security algorithms are defined in the specifications, two of which are based on well-established ciphers. Each algorithm is encoded in a byte vector as shown in Table 1. The encoding is:

$$\text{X}[\#\text{EIA}]\text{X}[\#\text{EEA}],$$

where X is a formatting wildcard.

Notably, not all possible encodings are assigned to actual algorithms but are reserved for future use, i. e., EIA4-7 and EEA4-7. Moreover, the EPS Integrity Algorithms (EIAs) and the EPS Encryption Algorithms (EEAs) can be arbitrarily combined, for instance using 128-EIA1 with 128-EEA2.

A special entity of both algorithm sets is EIA0 and EEA0. These algorithms are void operations leaving the data unencrypted and/or provide no integrity protection. According to the LTE specification, EIA0 is only allowed when emergency bearers are established. The choice of an actual instantiation for the encryption and the integrity protection algorithm is up to the network provider and can be based on territorial affiliation. EIA3/EEA3 (ZUC) has been designed in China and is the only allowed algorithm for LTE in China [25].

2.2.2 Security Procedures

In order to fulfill important security objectives including confidentiality and integrity protection, security procedures are implemented. Several entities of the LTE architecture are involved in the process, e. g., UE, eNodeB, MME, and HSS. In the following, we describe the transition from a detached UE to the state of successful connection establishment. While we highlight individual steps, we again confine ourselves to only present security-relevant parts. An overview of the whole process is given in Figure 2.

When a UE initiates a new connection to an LTE network, a *physical* connection to the eNodeB is first established on the radio layer. For simplicity, the process is omitted here. On the other hand, the *logical* connection to a network is started by a UE sending an Attach Request message to the MME including a permanent identity. In particular, when the Attach Request only contains a temporary identity, an unprotected Identity Request message [3, 4.4.4.2] can force the UE to reveal its IMSI. This is not included in the depicted protocol flow.

In addition, the Attach Request contains the Security Capabilities that define the supported security algorithms of the UE. For the sake of simplicity, we omit details about the LTE key hierarchy and refer to the standard [2, 6.2]. All keys that are involved in the attach procedure are derived from the same permanent long-term key K. After receiving the Attach Request, three phases are executed to establish a secure connection.

1. Authentication and Key Agreement (AKA). The goal of AKA is to establish mutual authentication between the UE and the LTE network as well as to derive a common session key. The process involves the MME, which requests information stored at the HSS. Note that both HSS and the UE share the same long-term key K. The sequence of exchanged messages is as follows.

(1) The MME sends an Authentication Information Request to the HSS, which holds the pre-shared key K of the requesting user identified by the included IMSI.

(2) The HSS starts a challenge-response run between the network and the UE by computing an Authentication Vector (AV) and sending it back to the MME via an Au-

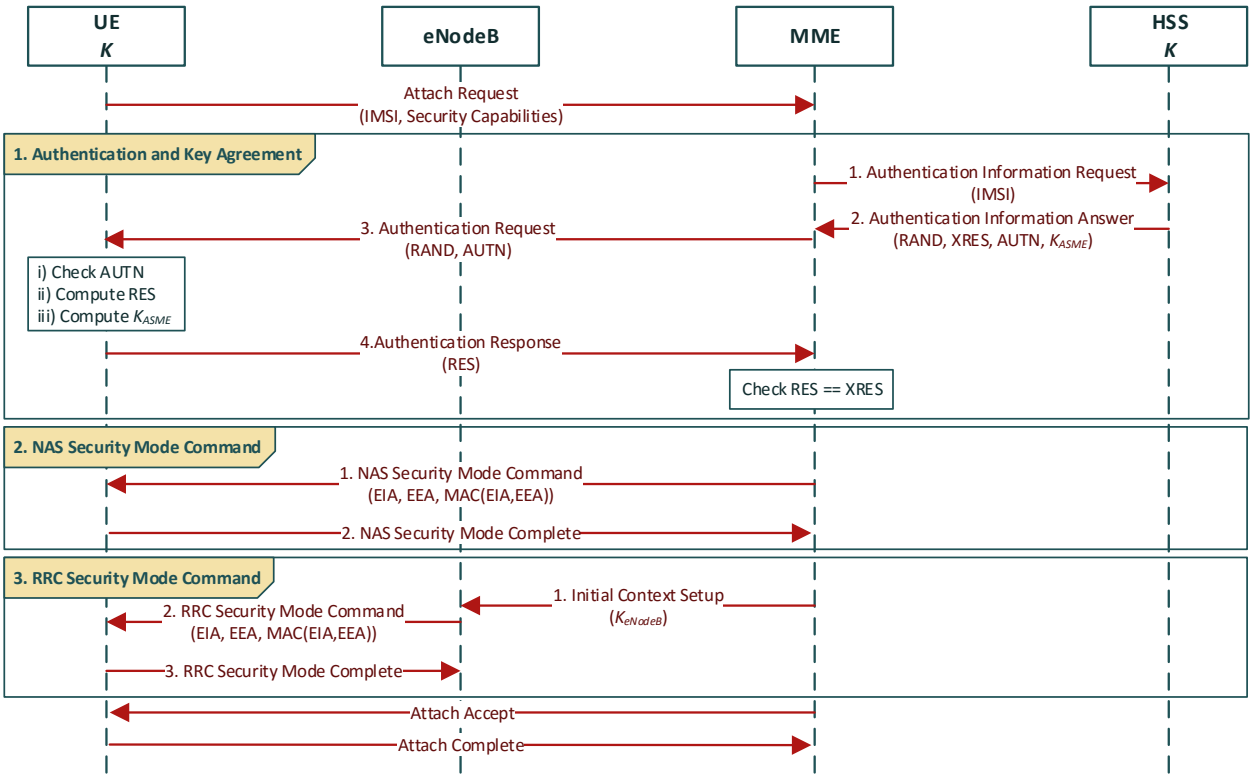

Figure 2: Security procedures in the LTE Attach Procedure.

thentication Information Answer. The AV contains the following parameters:

- Random Number (RAND)
- Expected Response (XRES)
- Authentication Token (AUTN)
- Intermediate Key (K_{ASME}),

with K_{ASME}—Access Security Management Entity being derived from the long-term key K and AUTN containing, e. g., a Sequence Number (SQN), which is synchronized to the current state of the UE.

(3) The MME stores XRES and K_{ASME} and forwards an Authentication Request to the UE with the challenge RAND as well as AUTN.

(4) The UE (i) verifies AUTN by checking the range of the SQN, (ii) computes Response (RES), (iii) and its own intermediate key K_{ASME}. The computed RES is sent back to the MME.

When XRES and RES are equal, the UE has intuitively proved that it possesses the same long-term key K and thus has authenticated itself against the network. Otherwise, the AKA procedure is aborted.

2. NAS Security Mode Command. The second phase performs the negotiation of the security algorithms, which are later used for *management* data encryption and integrity protection on the NAS layer. The communication involves the MME and the UE and is as follows.

(1) The MME sends a NAS Security Mode Command, which contains the selected EIA and EEA. The message itself is integrity protected with a Message Authentication Code (MAC) based on the selected EIA and a key derived from K_{ASME}.

(2) Upon reception, the UE checks the MAC and responds with a NAS Security Mode Complete signaling that the selected algorithms are accepted.

After successful negotiation, the selected EIA and EEA are applied to establish encryption of management data. Thereby, the NAS Security Mode Complete message and all subsequently exchanged messages on the NAS layer should be protected by the selected integrity and encryption algorithms.

3. Radio Resource Control (RRC) Security Mode Command. The third phase negotiates the security algorithms on the AS layer. It enables the encryption of *user* data and protects all messages exchanged on the AS layer. The involved parties are the MME, the eNodeB, and the UE. The flow of messages is as follows.

(1) The activation of AS security is initiated by the MME using key material derived from K_{ASME}. The derived key K_{eNodeB} is sent to the eNodeB via an Initial Context Setup message.

(2) The eNodeB selects algorithms for EIA and EEA specifically for the AS layer protection. The RRC Secu-

rity Mode Command message protects the selection with a MAC using a key derived from K_{eNodeB}.

(3) When the UE successfully verifies the MAC, the UE sends an RRC Security Mode Complete message back to the eNodeB confirming the selection.

All subsequent AS layer messages are encrypted and integrity protected. Finally, the MME sends an Attach Accept to the UE, which is answered by an Attach Complete concluding the LTE attach procedure.

3 Security Analysis

In this section, we perform a security analysis of the LTE specification tailored towards the aspects of ciphering indicators and network authentication. In particular, we analyze how the network authenticates itself against a UE. First of all, we define the attacker model that we consider.

3.1 Attacker Model

Our attacker model follows the model defined by Shaik *et al.* [28] and comprises a passive and an active attacker that differ in their capabilities and restrictions.

The passive attacker can observe arbitrary communication between the UE and the LTE network over the radio layer. Furthermore, this type of attacker can decode messages but cannot retrieve secret keys or break applied security mechanisms. Consequently, the passive attacker remains silent with respect to the communication flow.

The active attacker can additionally intercept, relay, modify, drop, or delay messages, without knowing the key material of devices not owned by the attacker. Moreover, our attacker can deploy a fake LTE base station impersonating a real LTE network with the objective of breaking the privacy of user data.

We make use of SDRs [10] and OpenAirInterface [11] to set up a rouge eNodeB realizing such an attacker. This allows us to implement major parts of the involved network entities in software and to be able to interact with a commercial network on all layers.

3.2 User Data Encryption

In general, LTE traffic can be distinguished into two types: user data and management data. We put our focus on the user data. *User data* encryption is activated by the RRC Security Mode Command which determines the encryption algorithm for messages on the AS layer.

Since a service provider, respectively the configuration on the eNodeB, can choose which EEA is used on the AS layer, EEA0 could indeed be selected. When the UE accepts this choice and replies with an RRC Security Mode Complete, all subsequent communication is unencrypted.

For this scenario, the LTE specification mandates that the user should be informed when the confidentiality of user data is not provided [2, 5.2]. Furthermore, the document [4, 14.0] states that the ciphering indicator should be configurable by devices with a user interface and ignore the settings of the network operator, which are set in the SIM card.

Test and Evaluation Framework. Even though this problem is known for feature phones [6, 15, 26], we are investigating *whether* EEA0 or undefined choices are accepted by modern LTE smart phones and *how* different vendors implement the ciphering indicator.

3.3 Network Authentication

To establish a secure communication link to an LTE network, mutual authentication between the UE and the network is a mandatory requirement. Without authentication, either part of the link cannot fully trust the other party. The *user authentication* towards the network is implicitly given since XRES must equal RES at the end of the AKA. This is the case, when both parties share the same pre-shared key K. We therefore focus on the *network authentication* highlighting important aspects of the procedure and outlining a MitM attack, where a UE accepts an attacker-controlled rogue LTE network as a serving network.

While mechanisms for user authentication can be recognized easily, the situation for network authentication is less clear. Intuitively, the authentication of the network towards the UE should be provided by the AUTN in the Authentication Request sent by the network during the AKA procedure depicted in Figure 2. When the UE receives the Authentication Request, it tries to verify the freshness of the AUTN by checking if the included SQN is within a certain range and thus can be accepted. This could be a small time or value frame and should prevent replay attacks. Moreover, the actual implementation of the SQN and the acceptance ranges are within the responsibility of the network provider.

However, we can show that the AUTN and the included SQN are not sufficient to provide network authentication alone. A simple SQN keeps its freshness as long as both parties are not too far out of sync, which can be exploited by an attacker-controlled fake base station. When a UE tries to attach to this network, the attacker can separately start a new attach procedure with an authentic commercial network using it as an oracle to obtain an AUTN for the victim UE. In a next phase, the attacker forces insecure parameters rendering the communication exposed to eavesdropping. The performed steps for this attack are illustrated in Figure 3 and are as follows.

(1) The UE tries to establish a new connection by

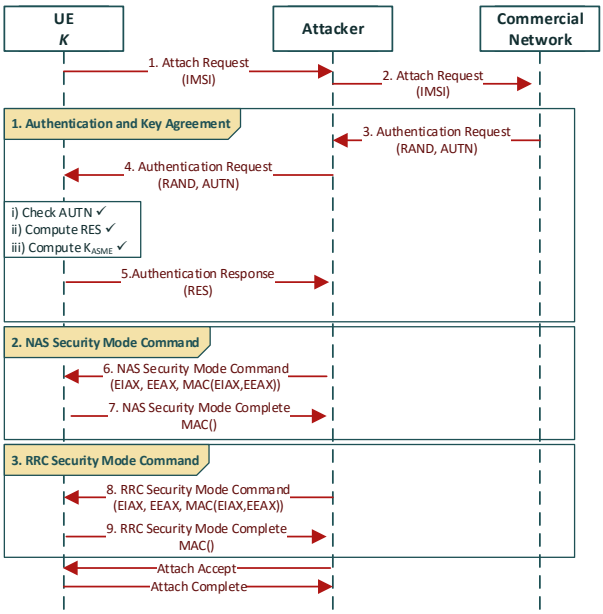

Figure 3: Active attack against the LTE Attach Procedure using a commercial network as an oracle.

sending an Attach Request including its identity. This message is received by our rogue base station.

(2) The attacker starts a new attach run by generating an Attach Request with the identity of the victim and sends it to a real commercial network that is serving the respective UE.

(3) An Authentication Request with a freshly generated RAND and a valid AUTN is returned to the requester. Notably, the attacker does not need to complete the whole attach process with the authentic network.

(4) The attacker receives the Authentication Request including a fresh RAND and AUTN and forwards it unaltered to the victim UE.

(5) The UE checks the validity of the AUTN by, e. g., checking the freshness of the SQN. The RES is computed and the Authentication Response is sent back to the fake base station. Notably, this message is not forwarded and thus never received by the authentic network resulting in a connection clearing.

At this point, the UE cannot distinguish between the rogue network that answered with a valid RAND and AUTN in the Authentication Request and an authentic commercial network. As a result, the Authentication Request on its own *does not* provide network authentication. The UE proceeds with the attach procedure as if it is connected with a commercial network.

However, the authentication of the network is given by the following NAS Security Mode Command message that is sent by the network. It includes a verifiable MAC, which is computed with the derived key K_{ASME}. Thus, the usage of the MAC provides key confirmation and the UE is assured that the network possesses the same long-term key K and is thereby authentic.

An attacker can now try to prevent this key confirmation step by choosing an integrity protection algorithm that allows to send a valid MAC without knowledge about the long-term key K.

(6) The attacker chooses an EIA and an EEA, tries to predict the MAC, and sends the NAS Security Mode Command message to the victim UE (we will explore the conditions for success of this step later). For instance, an attacker could select EIA0 and EEA0 while setting the MAC to null or select undefined algorithms, such as EIA7, with an arbitrary MAC.

(7) If the UE accepts the chosen EIA and EEA, it answers with a NAS Security Mode Complete message and a MAC that is computed with the selected EIA.

In particular, this answer is a security-critical part of the protocol and thus the main focus of our later applied testing. A NAS Security Mode Reject aborts the attach procedure, whereas a NAS Security Mode Complete indicates the success of the attack and the attach process is continued.

(8) The attacker selects security algorithms (EIA, EEA) for the AS layer via an RRC Security Mode Command. The MAC is chosen to, e. g., 0 or any acceptable value for which the attacker does not need any key material.

(9) Again, the UE signals the acknowledgement of the selected RRC algorithms with an RRC Security Mode Complete message.

As a result, the key confirmation of the NAS/RRC Security Mode Command is completely circumvented. The network authentication is not concluded leaving the UE connected to a fake LTE network.

Test and Evaluation Framework. In this work, we test *how* different implementations react to chosen values for EIA and EEA including malformed Security Mode Command messages. Additionally, we test the behavior when facing an altered sequence of messages during the attach process.

4 Test and Evaluation Framework

We designed and developed a test framework to evaluate the implementation compliance of the security procedures on different LTE devices. At first, we introduce the developed framework. Second, we discuss the parameters we consider for our testing approach. Third, we provide details on the tested devices with important characteristics.

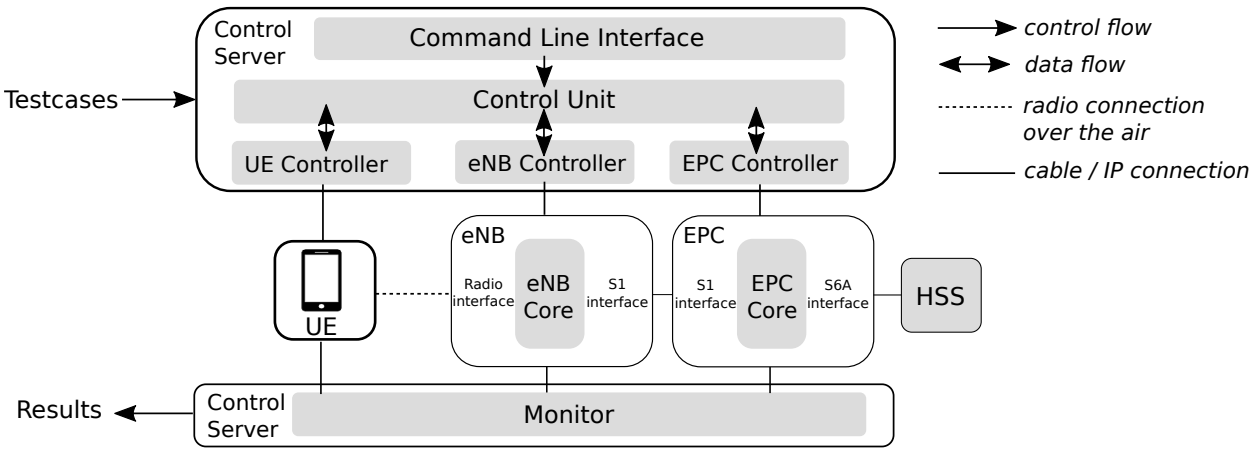

Figure 4: Conceptional design of our LTE test and evaluation framework.

4.1 Framework Design and Development

An integral part of our work is the developed testing framework. Using our framework we can semi-automatically test any LTE-capable device against security vulnerabilities with respect to network authentication and the ciphering indicator feature. The structure of our test and evaluation framework is depicted in Figure 4.

A testcase file allows us to define how messages are manipulated during our testing. The control unit decodes the next action and configures the involved controller elements, i. e., UE controller, eNodeB controller, and EPC controller. The core components eNodeB core and EPC core can manipulate the communication flow, e. g., the Security Mode Command message. In order to supervise the interfaces of each component, a monitoring system is implemented. The UE tries to connect to the eNodeB over the radio interface while the behavior of the chipset is analyzed. Based on logs and interface traces, results are generated to identify possible vulnerabilities of the tested device.

All components are realized based on the software project OpenAirInterface [11]. The UE is an actual LTE-capable device, while the eNodeB is built up on a USRP B210 from Ettus Research [10] that cost about 1200 €. The USRP B210 is chosen due to the variable clock that can be tweaked to the required LTE sampling frequency. The different network components are running on standard off-the-shelf notebooks with Linux-based operating systems.

A distributed and modular design was chosen to ease the deployment of individual LTE network entities on different host systems. Our framework uses the radio layer as input vector which makes the testing process fully transparent for the LTE device. In particular, we use a testing approach tailored towards security-relevant functions defined by the specification. This means that we specifically focus on potential vulnerabilities that we identified in Section 3. We can intervene in any communication step of the LTE protocol stack varying the content as well as the order of messages. The testing is performed *semi*-automatic due to user-enforced reconnects required by some of the UEs. For part of the tested devices, we were able to achieve full automated testing.

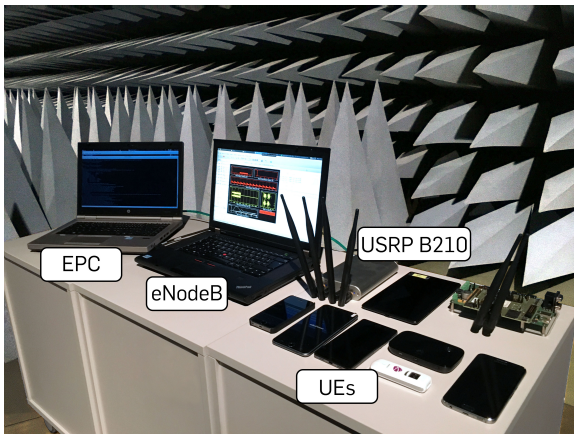

Figure 5: Our experimental testing setup within the anechoic chamber.

Our framework follows a modular approach and can be easily extended to test further parameters not considered so far to broaden our coverage of the LTE protocol stack. Our setup is depicted in Figure 5. We conducted our experiments in an anechoic chamber that prevents leakage of signals to the outside world. This allows to test authentic setups that are otherwise prohibited by law.

	Information Element	Type/Reference
NAS Header	Protocol discriminator	Protocol discriminator 9.2
	Security header type	Security header type 9.3.1
	Message authentication code	Message authentication code 9.5
	Sequence number	Sequence number 9.6
NAS Message	Protocol discriminator	Protocol discriminator 9.2
	Security header type	Security header type 9.3.1
	Security mode command message identity	Message type 9.8
	Selected NAS security algorithms	NAS security algorithms 9.9.3.23
	NAS key set identifier	NAS key set identifier 9.9.3.21
	Spare half octet	Spare half octet 9.9.2.9
	Replayed UE security capabilities	UE security capability 9.9.3.36

Figure 6: Structure of a NAS Security Mode Command message [3, 8.2.23].

4.2 Test Cases

We initially focused on the NAS Security Mode Command, which is the determining step for the data encryption as well as the network authentication. Figure 6 shows the structure of a NAS Security Mode Command highlighting the message parts that are tested.

The security header type of the NAS header indicates how the message is secured. All possible values for this field can be found in [3, 9.3.1]. The MAC field contains the calculated MAC, when the header type mandates that the NAS message is integrity protected. The NAS security algorithms field indicates the selected algorithms, i. e., EIA for integrity protection and EEA for ciphering/encryption as shown in Table 1.

Within our testing framework we can actively intercept the communication at any point of the LTE attach procedure. To reduce the complexity and to remain within reasonable time frames, we avoid performing dumb testing with a complete coverage. We narrowed down our test cases to analyze potentially security-critical scenarios.

We recall that an attacker has no access to the cryptographic keys and hence has no means to compute valid MACs. We therefore fixed the MAC to 0 since this does not require any key material. Furthermore, we mainly targeted the test cases to use EEA0 for ciphering. From an attacker's perspective, this represents the desired scenario in which no encryption is applied—as allowed by the specification.

After narrowing down the tested cases, we test the security header type and EIA, which have a length of 4 bits, respectively 3 bits. All possible values are tested totaling in $2^3 \cdot 2^4 = 128$ test cases. We have chosen to also test the security header type because this field indicates different security contexts.

4.3 Tested Devices

For the tests we selected a wide range of devices, which are listed in Table 2. The set includes standard smartphones available on the market as well as development and test devices. For each device, we gathered information about the chipset type, the baseband version, and the operating system. The baseband version is the firmware version of the baseband chipset.

The process of attaching to a new network including the authentication procedures is handled by the baseband firmware running on the baseband chip. For this reason, we tried to select diverse baseband chip manufactures. All tested devices are shown in Table 2.

5 Experimental Results

In order to process our findings, we analyzed logfiles and interface traces from the monitoring component within the control server of our framework. In the following, we highlight interesting findings regarding the two security aspects and describe the implementation of a MitM attack against one of the tested devices.

5.1 User Data Encryption

We tested all devices on whether a NAS/RRC Security Mode Command message with EEA0-EEA7 is accepted or refused. We found out that all devices accepted EEA0-EEA2 and refused other choices for the Security Mode Commands.

When a UE accepts EEA0, the messages on the respective layer are sent unencrypted and thus can be eavesdropped by a passive attacker. The specification mandates the support of EEA0 [2, 5.1.3.2], however as best practice service providers are encouraged to use one of EEA1-EEA3.

Furthermore, we examined the behavior of the UEs when the communication is unencrypted. Surprisingly, none of the tested devices informs the user about the lack of encryption. According to the specification, the choice of EEA0 for *mobility management* and *radio management* data is specification conform and its indication is optional. However, the user *needs* to be informed when EEA0 is chosen for *user* data. Failing to comply leads to a violation of the LTE specification.

We expected to see some kind of notification or alert on the display, when the UE accepts to communicate unencrypted. Even though the acceptance of EEA0 is specification conform, the lack of indication is not. Without such indications, the user is completely unaware that the connection is not confidential. Any attacker could easily eavesdrop the communication, thus violating the privacy of user data.

Table 2: Tested LTE devices, their build-in chipsets including the version of the baseband and the operating system. The last column indicates the results of our network authentication test.

Device	Chipset	Baseband Version	Operating System	Result
Huawei E3276 USB Dongle	HiSilicon Hi9620	22.250.04.00.00	Web0S 12.005.01.00	✗
Huawei E5776 Wi-Fi AP	HiSilicon Hi9620	22.265.11.00.00	Web0S 15.100.09.00	✓
Huawei P8 Lite	HiSilicon SoC Kirin 935	22.126.12.00.00	Android 5.0.1	✓
Telit EVK2 LE910	Qualcomm MDM9215	17.00.523	—	✓
Google Nexus 5	Qualcomm Snapdragon 800	M8974A-2.0.50.1.16	Android 4.4.4	✓
Google Nexus 7	Qualcomm Snapdragon S4 Pro	G002.43.0_1210	Android 6.0.1	✓
Apple iPhone 5s	Qualcomm MDM9615M	6.01.00	iOS 9.2.1	✓
Apple iPhone 6s	Qualcomm MDM9625M	1.14.00	iOS 9.1	✓
Samsung Galaxy S4	Qualcomm MDM9615M	I9505XXUHOJ2	Android 5.0.2	✓
Ulefone Be Touch 2	Mediatek MT6752	MD.LWTG.MP.V20.P2	Android 5.1	✓

5.2 Network Authentication

The network authentication depends on a combination of messages. The AUTN contains authentication information from the network, while the MAC of the Security Mode Command provides key confirmation concluding the network authentication. Consider an attacker tries to compute a valid MAC without knowledge of the used key. This can be achieved by forcing to use no integrity protection by selecting EIA0 or EIA4-EIA7. However, unlike for the ciphering algorithm, the specification prohibits the use of a different integrity protection algorithm other than EIA1-EIA3 [2, 5.1.4.2].

Surprisingly, the HiSilicon Hi9620 chipset accepts a Security Mode Command with the choice of EIA0. As a result, the MAC can be set to 0 and is therefore valid for EIA0. This leads to the fact that the network has not authenticated itself to the UE. Notably, selecting EIA0 is acceptable only for emergency bearers and should be refused with a Security Mode Reject in all other cases.

Due to the identified lack of network authentication, we are able to deploy a MitM attack against the LTE baseband implementation of the Huawei E3276 USB Dongle. A detailed description of the MitM attack is shown in the Appendix A.

6 Discussion and Countermeasures

On the basis of our findings, we discuss the (in)security of LTE implementations and outline countermeasures against the discovered vulnerabilities of the network authentication. First, we detail limitations we faced during our analysis.

6.1 Limitations

The software project OpenAirInterface that we utilized to build our testing framework is still under development and pervasive changes are made on a frequent basis. In its current state, only a simple LTE network can be deployed without the functionality for Voice-over-LTE calls or text messages. These features are still in a planning phase. Consequently, our work focuses on management messages of the LTE network.

For fully automatic testing, tools for the automatic reconnection are still missing. Currently, we need to manually reconnect to the LTE network in order to start new test cases. An exception to this is the Huawei E3276 USB Dongle and the Telit EVK2 LE910, which can be tested fully automated due to the availability of a serial connection. For testing other UEs, such as smartphones, reconnections must be triggerable by the control server.

6.2 User Data Encryption

The specification only mandates to inform the user about the lack of encryption, i. e., EEA0 is applied to *user* data, whereas the same choice for *management* data demands no notification. However, we argue that management data can have the same privacy-critical relevance and are worth to protect as much as the user data.

Furthermore, we suggest to make the applied encryption type more visible to the user. This can be achieved by triggering a notification when a connection without encryption algorithm is established. Notably, a warning pop-up had already been implemented for unencrypted calls in iOS 5 as reported by 9to5Mac [5]. We presume that several users were annoyed by such warnings and consequently this feature was dropped. However, we argue that an opt-in feature to display the currently used encryption type would not harm the usability [6].

As a long-term solution, we support the drop of EEA0 for LTE connections. We acknowledge that this might be a problem for countries prohibiting encryption for their citizens by law. However, user data could still be intercepted via a lawful interception interface specified for

LTE networks [1]. As a result, for future mobile communication generations encryption should become the standard.

6.3 Network Authentication

The specification clearly states that EIA0 should not be accepted by a UE and thus needs to be rejected for normal connections. For emergency bearers, the choice of EIA0 is still specification conform. Nevertheless, we suggest that a UE could offer the option to display the currently applied integrity protection.

We identified that the Huawei E3276 USB Dongle accepts EIA0 during the attach procedure, which is clearly an implementation flaw. A simple code review of the baseband chipset firmware should overcome this vulnerability and protect against our MitM attack. However, the dongle was given, e. g., to customers of one of the largest network operators in Germany and is therefore widespread [22].

We admit that the identified network authentication problem can potentially not be overcome by protocol changes since an attacker can always act as a simple relay of important values. Therefore, we again stress that implementation correctness is essential to establish network authentication and ultimately the security goals of LTE. The developed framework helps to evaluate and verify the implementation correctness.

7 Related Work

We summarize related work and draw connections to our work. A detailed overview of LTE security is given by Forsberg *et al.* [13]. Security research for mobile communications can be divided into different domains. One branch is targeted at the analysis of specification-based vulnerabilities. The authors of [16] found a security-critical specification flaw in the LTE handover key management. Further attacks against the privacy of radio transmissions are presented in [14]. Shaik *et al.* [28] highlighted privacy and availability problems in the protocol specifications of LTE.

A different research line analyzes the implementation correctness to demonstrate vulnerabilities in commercial products. Our work can be categorized in this domain. Weinmann [31] outlined problems in the GSM protocol stack and tested them with different devices. Another vulnerability was found in the baseband chipset firmware of the Samsung Galaxy S6 discovered by Golde *et al.* [9]. Similar to these, our flaw shows that incorrect implementation can lead to serious problems, undermining the user's security.

Early work on fuzzing mobile phones was done by Mulliner and Miller in [24, 23] with the focus on exchanged text messages. To open a broader analysis surface, fuzzing tools were developed for implementations of the GSM protocol stack by Broek *et al.* [30] and Welte [17]. These approaches focus on GSM implementations and can not be used for the LTE stack. Using our framework we can analyze the LTE stack of various implementations.

A variety of jamming and DoS attacks that can force UEs to perform a reconnection to, e. g., fake base stations are presented by Lichtman *et al.* [21] and Xiao *et al.* [32]. The authors of [7] and [8] evaluate the effects of DoS attacks by overloading eNodeBs with Attach Requests. A framework for large scale security analysis against LTE DoS attacks was developed by Jermyn *et al.* [19]. The authors of [20] give an overview of current research directions against LTE DoS and jamming attacks. In our case jamming can be used by an attacker to enforce the connection to the fake base station.

8 Conclusion

The most recent mobile communication standard LTE inherits security procedures that should protect the privacy of users and their data. By performing a comprehensive security analysis of the LTE specification, we identified two aspects which are potentially susceptible to implementations flaws. In particular, we focused on user data encryption and network authentication. In order to examine the implementation correctness, we designed the first low-cost testing framework to test devices on their LTE specification compliance. We tested several modern smartphones and other LTE-capable devices with our own LTE network based on commercial SDRs and the OpenAirInterface software project.

Our approach revealed that *all* tested devices allowed unencrypted connections without issuing any warning to the user. This is a violation of the specification and represents a threat since network providers are free in their choice of encryption. Furthermore, we identified a more severe problem with the network authentication, omitting the necessary steps in which the network authenticates itself against a UE. Consequently, we were able to launch a MitM attack against the Huawei E3276 USB Dongle allowing the attacker to actively intercept the communication (current firmware versions are not susceptible to this attack [18]).

To counter mis-implementations, we suggest the displaying of notifications for unprotected data transmissions and ultimately the drop of EEA0 and EIA0 as choices for security algorithms. Since LTE is more and more integrated in critical infrastructure, we encourage manufactures to fix the identified security issues and to thoroughly verify the implementation correctness of upcoming mobile communication generations. Our devel-

oped testing framework can serve as a foundation for this and can be used for further security testing.

9 Acknowledgments

This work has been supported by the Franco-German BERCOM Project (FKZ: 13N13741) co-funded by the German Federal Ministry of Education and Research (BMBF) and by the DFG Research Training Group GRK 1817/1.

References

[1] 3GPP. 3G security; Lawful interception requirements. TS 33.106, 3rd Generation Partnership Project (3GPP), 12 2009.

[2] 3GPP. 3GPP System Architecture Evolution (SAE); Security architecture. TS 33.401, 3rd Generation Partnership Project (3GPP), 06 2011.

[3] 3GPP. Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3. TS 24.301, 3rd Generation Partnership Project (3GPP), 06 2011.

[4] 3GPP. Service aspects; Service principles. TS 22.101, 3rd Generation Partnership Project (3GPP), 06 2011.

[5] 9TO5MAC. iOS 5 Unsecured Calls warning lets users know if they are talking on unencrypted networks. http://www.9to5mac.com/2011/06/07/ios-5-unsecured-calls-warning-lets-users-know-if-they-are-talking-on-unencrypted-networks/. [Online; accessed 12-May-2016].

[6] ANDROULIDAKIS, I., PYLARINOS, D., AND KANDUS, G. Ciphering Indicator approaches and user awareness. *Maejo International Journal of Science and Technology 6*, 3 (2012), 514–527.

[7] BASSIL, R., CHEHAB, A., ELHAJJ, I., AND KAYSSI, A. Signaling Oriented Denial of Service on LTE Networks. In *Mobility Management and Wireless Access* (2012), MobiWac '12, ACM Press, pp. 153–158.

[8] BASSIL, R., ELHAJJ, I. H., CHEHAB, A., AND KAYSSI, A. Effects of Signaling Attacks on LTE Networks. In *Advanced Information Networking and Applications Workshops* (March 2013), WAINA '13, IEEE Computer Society, pp. 499–504.

[9] DARREN PAULI. Samsung S6 calls open to man-in-the-middle base station snooping. http://www.theregister.co.uk/2015/11/12/mobile_pwn2own1/, 2015. [Online; accessed 12-May-2016].

[10] ETTUS RESEARCH. USRP Bus Series - USRP B210. http://www.ettus.com/product/details/UB210-KIT. [Online; accessed 12-May-2016].

[11] EURECOM. OpenAirInterface. http://openairinterface.eurecom.fr/, 2015. [Online; accessed 12-May-2016].

[12] FIRSTNET. FirstNet: First Responder Network Authority. http://www.firstnet.gov/network. [Online, accessed 12-May-2016].

[13] FORSBERG, D., HORN, G., MOELLER, W., AND NIEMI, V. *LTE Security*. NSN/Nokia Series. Wiley, 2012.

[14] FORSBERG, D., LEPING, H., TSUYOSHI, K., AND ALANÄRÄ, S. Enhancing Security and Privacy in 3GPP E-UTRAN Radio Interface. In *Indoor and Mobile Radio Communications (PIMRC)* (Sept 2007), IEEE Computer Society.

[15] GOOGLE ANDROID. Issue 5353: Ciphering Indicator. https://code.google.com/p/android/issues/detail?id=5353. [Online; accessed 12-May-2016].

[16] HAN, C.-K., AND CHOI, H.-K. Security Analysis of Handover Key Management in 4G LTE/SAE Networks. IEEE Computer Society, pp. 457–468.

[17] HARALD WELTE. Fuzzing your GSM phone using OpenBSC and scapy. https://events.ccc.de/congress/2009/Fahrplan/attachments/1503_openbsc_gsm_fuzzing.pdf. [Online; accessed 12-May-2016].

[18] HUAWEI. Security Advisory - Integrity Protection Vulnerability in Huawei E3276s Products. http://www.huawei.com/en/psirt/security-advisories/2016/huawei-sa-20160330-01-dongle-en. [Online; accessed 12-May-2016].

[19] JERMYN, J., JOVER, R. P., ISTOMIN, M., AND MURYNETS, I. Firecycle: A Scalable Test Bed for Large-scale LTE Security Research. In *International Conference on Communications (ICC)* (June 2014), ICC '14, pp. 907–913.

[20] JOVER, R. P. Security Attacks Against the Availability of LTE Mobility Networks: Overview and Research Directions. In *Wireless Personal Multimedia Communications* (June 2013), WPMC '13.

[21] LICHTMAN, M., REED, J. H., CLANCY, T. C., AND NORTON, M. Vulnerability of LTE to Hostile Interference. In *Signal and Information Processing* (Dec 2013), GlobalSIP '13, IEEE Computer Society, pp. 285–288.

[22] LTE ANBIETER.INFO. Stick: Speedstick III LTE von der Telekom. http://www.lte-anbieter.info/lte-hardware/all/stick-speedstick-iii-lte, 2013. [Online; accessed 12-May-2016].

[23] MULLINER, C., GOLDE, N., AND SEIFERT, J. SMS of Death: from analyzing to attacking mobile phones on a large scale. *USENIX Security* (2011).

[24] MULLINER, COLLIN AND MILLER, CHARLIE. Fuzzing the Phone in your Phone. https://www.blackhat.com/presentations/bh-usa-09/MILLER/BHUSA09-Miller-FuzzingPhone-PAPER.pdf. [Black Hat USA 2009; Online; accessed 12-May-2016].

[25] ORHANOU, G., AND EL-HAJJI, S. The New LTE Cryptographic Algorithms EEA3 and EIA3. In *Applied Mathematics & Information Sciences* (November 2013), vol. 7, pp. 2385–2390.

[26] OSMOCOM. Will my Phone Show An Unencrypted Connection? https://projects.osmocom.org/projects/security/wiki/WillMyPhoneShowAnUnencryptetConnection. [Online; accessed 12-May-2016].

[27] OSMOCOM PROJECT. SIM Trace. http://bb.osmocom.org/trac/wiki/SIMtrace. [Online; accessed 12-May-2016].

[28] SHAIK, A., BORGAONKAR, R., ASOKAN, N., NIEMI, V., AND SEIFERT, J.-P. Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems. In *Network and Distributed System Security Symposium* (2016), NDSS '16, The Internet Society.

[29] TELEGEOGRAPHY. Global LTE subscribers to quadruple by 2019. https://www.telegeography.com/products/commsupdate/articles/2015/06/16/global-lte-subscribers-to-quadruple-by-2019/, 2015. [Online; accessed 12-May-2016].

[30] VAN DEN BROEK, F., HOND, B., AND CEDILLO TORRES, A. Security Testing of GSM Implementations. In *Engineering Secure Software and Systems* (2014), ESSoS '14, Springer-Verlag, pp. 179–195.

[31] WEINMANN, R.-P. Baseband Attacks: Remote Exploitation of Memory Corruptions in Cellular Protocol Stacks. In *USENIX Conference on Offensive Technologies* (2012), WOOT '12, USENIX Association.

[32] XIAO, J., WANG, X., GUO, Q., LONG, H., AND JIN, S. Analysis and Evaluation of Jammer Interference in LTE. In *Innovative Computing and Cloud Computing* (2013), ICCC '13, ACM Press, pp. 46–50.

A Man-in-the-Middle Attack

We implemented a MitM attack where an active attacker impersonates a commercial LTE network to circumvent the network authentication. The schematic overview of our attack is depicted in Figure 7. The attack is based on the setup used in our fuzzing framework and is extended with a Telit EVK2 LE910 and SIMtrace [27] by OsmocomBB to communicate with a commercial network.

We assume that the victim UE tries to connect to our rogue LTE network. In a real-world scenario, we can force a new connection via jamming of known LTE frequencies and configuring a fake network to use a priority frequency, as described in the work of Shaik *et al.* [28]. During the AKA procedure, we need to use a commercial network as an oracle for an authentic pair of AUTN and RAND. SIMtrace allows us to sniff the communication between a programmable USIM card and the Telit board. The intercepted values are encapsulated in a new Authentication Request message and sent to the victim.

When our rogue network receives the Authentication Response, we send a NAS Security Mode Command with EIA0, EEA0, and a MAC set to 0. The victim UE answers with a NAS Security Mode Complete message signaling that our choice is accepted. We repeat this step for the RRC Security Mode Command, which is answered with an RRC Security Mode Complete message. Finally, an Attach Accept and an Attach Complete are exchanged, integrity protected with a MAC set to 0. Figure 8 illustrates the success of our attack.

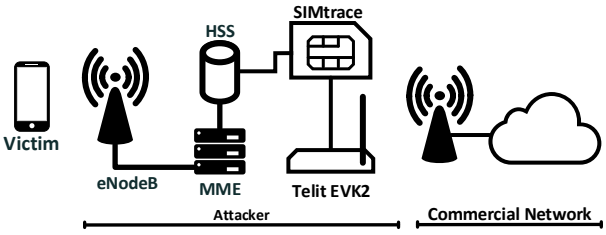

Figure 7: Schematic setup of our Man-in-the-Middle attack against the Huawei E3276 USB Dongle.

```
▾ Non-Access-Stratum (NAS)PDU
    0010 .... = Security header type: Integrity protected and ciphered (2)
    .... 0111 = Protocol discriminator: EPS mobility management messages (0x07)
  Message authentication code: 0x00000000
    Sequence number: 1
    0000 .... = Security header type: Plain NAS message, not security protected (0)
    .... 0111 = Protocol discriminator: EPS mobility management messages (0x07)
  NAS EPS Mobility Management Message Type: Attach accept (0x42)
    0000 .... = Spare half octet: 0
    .... 0... = Spare bit(s): 0x00
    .... .010 = Attach result: Combined EPS/IMSI attach (2)
```

Figure 8: Attach Accept message of a successful Man-in-the-Middle attack.

As a result, the Huawei E3276 USB Dongle is connected to our rogue LTE network circumventing the network authentication and using no data encryption. The victim UE can now establish arbitrary network connections via our fake network. We are able to eavesdrop the communication as well as actively intercept at any stage. There are only minor restrictions in the offered services since our setup implements essential parts of the LTE protocol stack. The lack of security is not indicated by any means hence the victim believes in a secure connection via a trusted service provider.

Mobile Communications Acronyms

AKA Authentication and Key Agreement

AS Access Stratum

AUTN Authentication Token

AV Authentication Vector

E-UTRAN Evolved UMTS Terrestrial Radio Access Network

EEA EPS Encryption Algorithm

EIA EPS Integrity Algorithm

eNodeB Evolved NodeB

EPC Evolved Packet Core

EPS Evolved Packet System

GSM Global System for Mobile Communications

HSS Home Subscriber Server

IMSI International Mobile Subscriber Identity

LTE Long Term Evolution

MAC Message Authentication Code

MitM Man-in-the-Middle

MME Mobility Management Entity

NAS Non-Access Stratum

RAND Random Number

RES Response

RRC Radio Resource Control

SDR Software Defined Radio

SQN Sequence Number

UE User Equipment

UMTS Universal Mobile Telecommunications System

USIM Universal Subscriber Identity Module

XRES Expected Response