A Formal Analysis of Emergency Communication System Based on Model Checking

2023 IEEE 13th International Conference on Electronics Information and Emergency Communication (ICEIEC) | 979-8-3503-3172-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICEIEC58029.2023.10200747

Meichen Liu
Beijing Institute of Technology
School of Information and Electronics
Beijing, China
LMCmeichen@163.com

Hai Li
Beijing Institute of Technology
School of Information and Electronics
Beijing, China
haili@bit.edu.cn

Abstract—**Emergency communication system is utilized widely to serve professional mobile communication for public security. Strict inspections must be conducted to ensure the system's high reliability, as inherent protocol complexities and design flaws may cause logical vulnerabilities in protocol implementation. In this paper, we investigate the security of the emergency communication, taking the TETRA system as an example. Specifically, we analyze two critical procedures (i.e., registration and call control) of the TETRA system. As a result, we uncover several potential flaws in the system. Furthermore, to expose vulnerabilities, we apply model checking to the automatic analysis of the TETRA system using the tool NuSMV. At last, we verified our findings in the real TETRA testbed environment. The results show that the attacks can induce service disruption, and the proposed method is feasible to TETRA system analysis.**

Keywords- emergency communication; TETRA system; model checking; protocol security; NuSMV

I. INTRODUCTION

Emergency communication, intended for departments with high communication security requirements in the public security network, is a unique communication mechanism with rapid response capabilities. It provides effective supports for responding to natural disasters, handling emergencies, as well as ensuring large-scale gatherings and significant events. With the growing demands for public security prevention, linkage scheduling and high-speed data transmission, the scale of emergency communication like digital trunked communication systems will continue to increase. In addition, deployment mode is becoming more complicated to accommodate this evolution, which may generate new challenges in the reliability and security of emergency communication. Therefore, the research on its security and reliability is quite prominent and necessary.

There exist many theoretical studies on security from aspects of requirements, risks, mechanisms, and vulnerabilities. A security scheme based on the optimized Byzantine Generals' Problem and trusted security analysis was proposed in [1], incorporating the ECDSA algorithm for security and integrity. The authors in [2] analyzed security threats during emergency communication, and categorized solutions into different emergency service life cycles, including preparation, deployment, maintenance, expansion and monitoring. For TETRA system, [3] demonstrated that TETRA's location privacy and dependability can be undermined and weakened by implementing a practical localization and fuzzing framework for TETRA. An authentication protocol that may cause a linkability attack was disclosed in [4]. Authors in [5] assessed the effectiveness of user authentication methods implemented in systems used in emergency communication centers. To improve its robustness, a test method for the radio link interference resistance of TETRA base station receiver was proposed in [6]. However, the previous work primarily concentrates on authentication mechanism by manually inspecting specification documents or simulating signal transmission, lacking formal analysis of protocol signaling interactions.

Inspired by the systematic analysis of the 4G/5G protocol using model checking in [7] and [8], in this paper, we employ the semi-automatic method to formally examine TETRA system, a widely used system for emergency communication. First, we model the communication subject and its behavior by describing protocol properties using temporal logic formula. Subsequently, we apply a model checking tool to verify whether the system model meets the property, and thus realize a formal analysis for detecting whether the execution logic of TETRA protocol meets the requirements.

II. THE RELATION BETWEEN MODEL CHECKING AND SYSTEM ANALYSIS

This part mainly discusses the principle of model checking, and then explains the relationship between model checking and protocol analysis.

A. Model Checking

Model checking is a state search method. It considers that each subject has a set of states related to it. The union of all subject state sets is the system state model. Model checking verifies the expected properties of the system by thoroughly checking all reachable states and their behaviors in a given system model. If it is detected that the design fails to meet an expected property, it always generates a counterexample to show how it is violated. This defect track helps to understand the real cause of the failure, and we can design attacks based on

979-8-3503-3172-1/23/$31.00©2023 IEEE

it. In model checking, Kripke structure usually represents the system model, and the expected properties are represented by temporal logic.

1) Kripke Structure

Kripke structure is a particular type of finite state machine. In Kripke structure, each state is labelled with a set of atomic propositions that are true in this state. Assuming AP is a set of atomic propositions, the Kripke structure on AP is a four-tuple:

$$M = (S, S_0, R, L) \tag{1}$$

Where S is a finite set of all states, $S_0 \subseteq S$ and S_0 is a finite set of initial states, R is a total state transition relation such that $R \subseteq S \times S$, and L is a function that labels each with the collection of atomic propositions that are true in that state such that $L : S \to 2^{AP}$.

2) Temporal Logic

Temporal logic can describe events in time order without introducing time details, so it can well characterize concurrent systems. In model checking, temporal logic is used to define the required specifications. The temporal logic used generally is Linear Time Temporal Logic (LTL) and Computational Tree Logic (CTL). LTL models time as a sequence of states, describing the logical relationship between them, while CTL models time as a tree structure where any current moment may bifurcate into multiple possible future moments, and it represents property only related to the current state. Frequently-used temporal operators in LTL and CTL are shown in Table I and Table II.

TABLE I. FREQUENTLY-USED LTL OPERATORS

LTL Operator	Description
$F\,p$	a certain condition p holds in one of the future time instant
$G\,p$	a certain condition p holds in all future time instants
$p\,U\,q$	condition p holds until a state is reached where q holds
$X\,p$	condition p is true in the next state
$O\,p$	a certain condition p holds in one of the past time instants
$Y\,p$	a certain condition p holds in the previous time instant

TABLE II. FREQUENTLY-USED CTL OPERATORS

CTL Operator	Description
$AX\,p$	condition p is true in all next reachable states
$EX\,p$	condition p is true in some of next reachable states
$AF\,p$	all the paths stating from now finally condition p hold
$EF\,p$	there exists some path that finally in the future satisfies p
$AG\,p$	requires that condition p is always true in all the states of all the possible paths
$EG\,p$	there is some path along which condition p is always true

B. The Relation Between Model Checking and Protocol Analysis

The model-checking algorithm is to give a Kripke structure M to represent a finite state system and then gives a temporal logic formula f to describe the property of the system. In this way, "whether the system meets the desired property" transformed into a mathematical question "whether the state transition diagram meets the formula," which is expressed as:

$$s \models f \tag{2}$$

Then find out the set of all states that meet the formula:

$$\{s \in S \mid M, s \models f\} \tag{3}$$

If all the initial states of the system are in this set, then it meets the property specification.

In addition, finite-state models are widely used in the design of communication protocols and protocol implementation. The protocol specifies the initial state set and all state transitions. Each step of the protocol is regarded as a state transition rule, and the execution process of the protocol is regarded as a state transition process. The protocol is modeled as a state transition system, and propositional temporal logic is used to represent the protocol specification property. Through model checking, traverse the entire state space to see if there is a transition process from the initial state to an unsafe state.

III. THE PROPOSED METHOD

This section will explain how we apply the semi-automatic method to TETRA system effective analysis. We first briefly describe an overview of the method. Then we elaborate on the main aspects and discuss our realization. At last, we verify the results in a real testbed environment. Figure 1 shows the main phases of using model checking to analyze protocol.

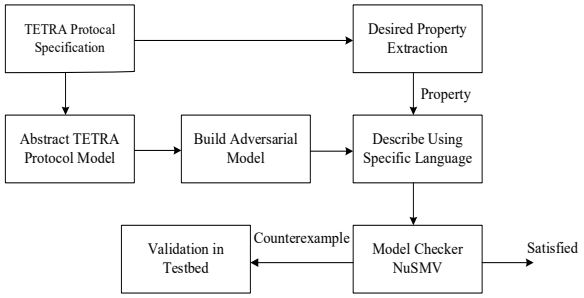

Figure 1. The main steps of protocol analysis using model checking

A. Overview

In our method, the system state transition model and properties are abstracted from TETRA specifications. Besides, we build a Dolev-YAO adversary model to mimic an attacker's behavior, which can replay, remove, inject, intercept and learn the content of any messages in the network. The entire communication network is under control of the adversary. Then we use NuSMV[6], a model checking tool, to verify the abstracted properties and discover some vulnerabilities. To verify the attributes, we use the language specified by the tool to give a formal description of the model and properties. If the tool shows that a counterexample violates the property, we think we have an attack path. Finally, we verify the attacks in a real environment.

A major challenge in model checking is scalability. If the model is too large, the model checker may be unable to terminate. We address the problem by scientifically reducing the number of auxiliary variables, message packets and state machines in the model design process to reduce the complexity of the model without affecting the accuracy of the verification results as much as possible.

NuSMV is a model checking tool, which can flexibly and efficiently model the software requirement model and has strong operability. It can use LTL and CTL to analyze the system formally. It has been applied to detecting logic design errors in system, such as operating systems, data

communication protocols and parallel algorithms. Therefore, we choose NuSMV as the research tool.

B. Abstraction of Protocol

In this subsection, we explained FSM model building and property extraction. At this step, we build a TETRA system model with a Dolev-YAO attacker and extract the required properties from the protocol specifications and system requirements.

1) FSM Building

TETRA protocol can be regarded as a family of protocols, including many sub-protocols, such as MM Protocol and CMCE Protocol. Each protocol specifies the syntax, semantics and timing of the respective service control process. Therefore, each procedure can be modeled as the communication between several state machines. An abstract finite state machine can be represented by a five-tuple:

$$\mathcal{M} = (\Sigma, \Gamma, S, S_0, T) \tag{4}$$

Where Σ and Γ are the non-empty sets of conditions and actions for the protocol, respectively, S is a finite set of states in which the protocol can reside, S_0 is the initial state of the protocol, and T is a finite set of transitions in S.

Although many network entities are participating in the procedure, to simplify the protocol model, we abstract the network side into an entity called SwMI(Switching and Management Infrastructure), which not only improves the efficiency of model checking but also does not affect the analysis results, because we mainly focus on the service control logic between user devices and network.

Specifically, in the process of abstracting the model, we focused on these three factors: message type, critical payloads of message packets, and Timers.

a) Message type

There are various types of messages between entities communicating with each other. In most cases, receiving different messages from the communication peer will cause different state transitions and the corresponding action.

b) Critical payload of message packets

Some payloads contained in protocol packets also affect the abstract modeling of TETRA system. Because the range of payloads is extensive, the generated model is unlikely to be suitable for model checking by directly representing all the grouped payloads, which may cause the state explosion problem. Therefore, we mark the concerned payload separately and express it as the system behavior equivalent to the message type.

c) Timers

Some timers also constrain state transition and behaviors of TETRA system. For example, when the caller sends a U-SETUP message in call control procedure, it will start a timer. The state transition will occur if no reply signaling is received before the timer ends. Therefore, timers should also be considered when building abstract models. In the same way as the payload, we do not model the specific time value of the timer. We only focus on whether the timer ends. So for timers, we use Boolean variables, *timer_end*, to express the behavior of the timer.

2) Property Extraction

In protocol verification, we mainly care about two types of properties: safety properties and liveness properties. We first extract properties from the protocol specification documents and system requirements by mainly carefully analyzing the file [9] [10], and then we convert them into formal formulas.

C. Implementation

In this subsection, we will use specific examples to illustrate our realization.

1) Registration

The registration procedure refers to the process that makes a UE become reachable and legal in the network side. It is the basis and premise for other services. On the contrary, deregistration is the process in which the UE information becomes invalid in the network. The FSM model, according to the registration/ deregistration procedure, is shown in Figure 2.

For the system requirement, we expect that once the UE is registered, its state from the perspective of network will never go to the deregistered state until the UE sends a U-ITSI-DETACH message. Then we extract the property "*the state of network cannot transition from registered state to deregistered state unless UE sends U-ITSI-DETACH message.*" We express it using LTL formula as follows:

$$\begin{aligned} &\mathrm{G}\ ((\mathrm{BS\_state} = \mathrm{bs\_registered}\ \&\ \mathrm{X}(\mathrm{BS\_state} = \mathrm{bs\_deregistered})) \\ &\quad \text{—>}\ \mathrm{Y}\ (\mathrm{UE\_action} = \mathrm{ue\_u\_itsi\_detach})) \end{aligned} \tag{5}$$

Using NuSMV, the checking result is shown in Figure 3. According to it, we can get a counterexample: after the target UE registers successfully, an injected U-ITSI-DETACH message could make the network think the UE deregistered by mistake.

2) Call Control

Call service is the core of TETRA system. Call control solves the problems of call establishment, disconnection and maintenance procedure. According to the call control protocol flow, we build the FSM model shown in Figure 4.

For call control, we want to verify the property "the call cannot end unless the participants hangs up or some timers end." We express it with LTL formula as follows:

$$\begin{aligned} &\mathrm{G}((\ \mathrm{bs\_action} = \mathrm{bs\_d\_release}\ ) \\ &\quad \text{—>}\ \mathrm{O}\ (\mathrm{ue\_action} = \mathrm{ue\_u\_disconnect} \mid \mathrm{timer\_out}\ )\ ) \end{aligned} \tag{6}$$

Using NuSMV, the result is shown in Figure 5. According to it, we get a counterexample: when a call is established and ongoing, an injected U-DISCONNECT message could induce the call to collapse.

D. Validation in Testbed

We verify the potential threats in a real TETRA testbed environment to ensure feasibility of detected attacks. The test environment is an implemented TETRA system developed by our team, Motorola TETRA interphones, and the experimental platform integrating the malicious terminal, malicious base station, and protocol analysis module. The environment is shown in Figure 6.

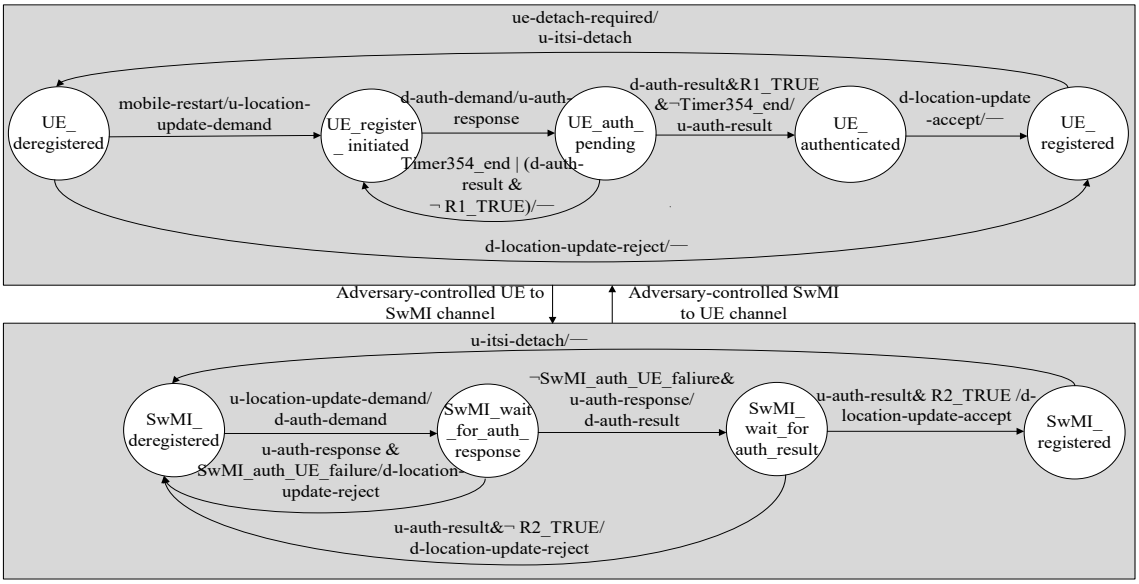

Figure 2. The FSM Model for the registration procedure

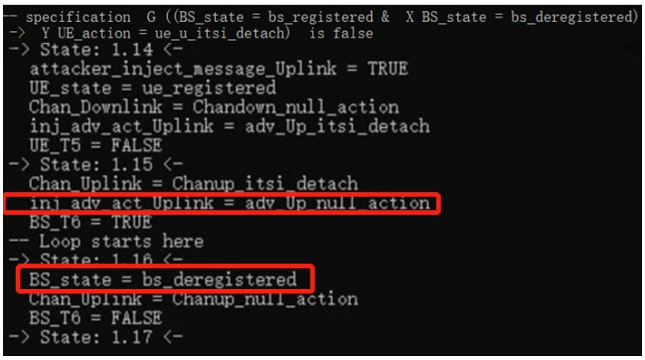

Figure 3. The model checking results of registration procedure

1) Forced-deregistered Attack by U-ITSI-DETACH message

Description. In our verification setup, the victim UE was in the normal registered state, and we make the malicious UE send a U-ITSI-DETACH containing the victim UE's identity. The process is shown in Figure 7.

Impact. The test verifies that the attack makes the SwMI mistakenly thinks that the legitimate UE on the line has completed the deregistration procedure. This attack can bypass the authentication protection mechanism of the target system because deregistration procedure lacks an authentication process. The attack could result in some service interruption. As shown in Figure 8, when other users call to the victim UE, SwMI sends a D-RELEASE message rejecting the call set-up.

2) Call Service Disruption

Description. When a call of the victim UE is established, the adversary could get the call identifier by sniffer or other clever methods. Then it can directly send a U-DISCONNECT message carrying the target call identifier, leading to an abnormal call end. The process is shown in Figure 9.

Impact. This attack can cause the UE to suffer a call service disruption, making communication and scheduling functions fail.

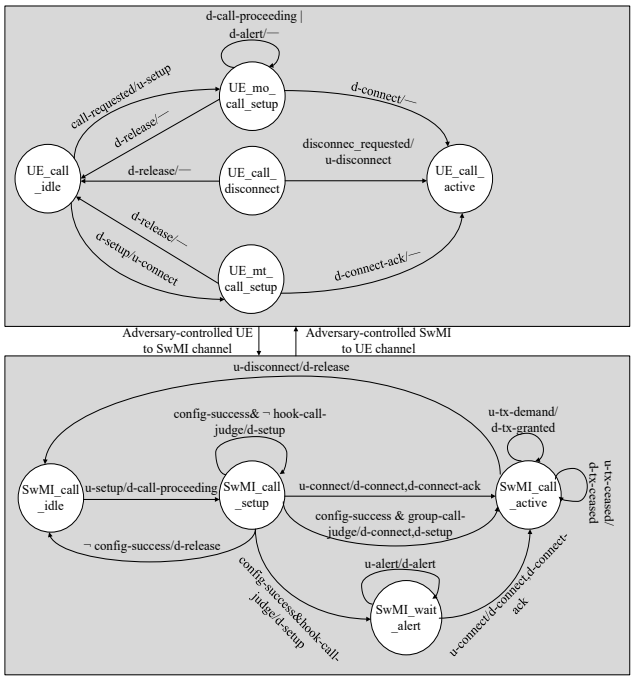

Figure 4. The FSM Model for the call procedure

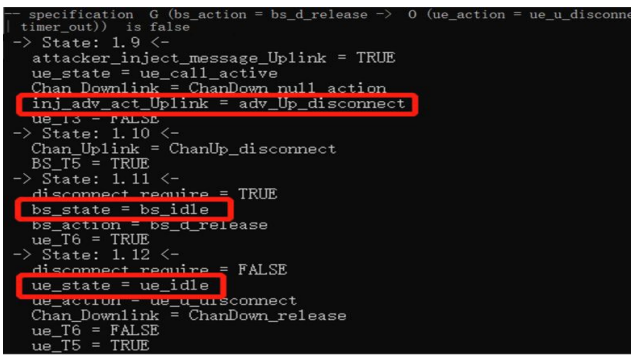

Figure 5. The model checking results of call procedure

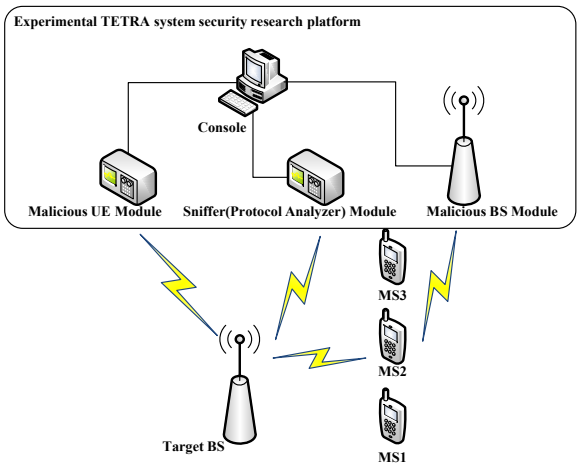

Figure 6. Testbed environment

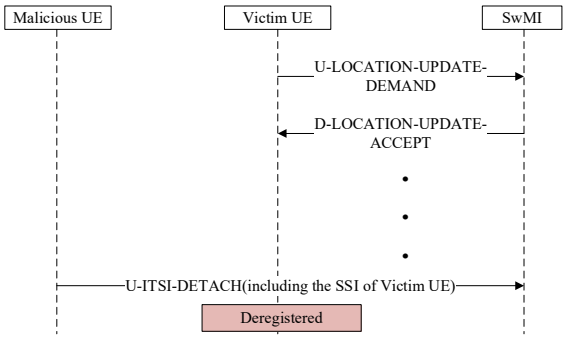

Figure 7. The process of the attack

3) *Other Interesting Results*

Numb Attack. During the registration procedure, the malicious base station sends a D-LOCATION-UPDATE-REJECT message carrying the cause "authentication-failure" to the target UE. Then the UE would be detached .The victim UE even remains in such a numb state until it reboots.

Impersonation attack. When a malicious UE pretends to be a legitimate UE invading the network, if the target legitimate UE is not registered or it is far away from the base station, the malicious UE can invade successfully. Conversely, if the target UE is attached to the network and close to the base station, which means that the power of signal transmission is relatively high, the malicious attach procedure will be interrupted, and the malicious UE cannot invade successfully.

IV. CONCLUSION

In this paper, we applied model checking to the automatic formal analysis of TETRA protocol, identify potential vulnerabilities, and verify the attacks in the implemented TETRA system. Specifically, we find two feasible attacks triggered by U-ITSI-DETACH and U-DISCONNECT messages. Our findings have significance in discovering protocol design and implementation flaws. Further, the experiment shows the effectiveness of model checking in the analysis of emergency communication and promotes its applicability. Combined with earlier related work, our study offers the universality of model-checking procedure in security and privacy analysis of mobile communication protocols.

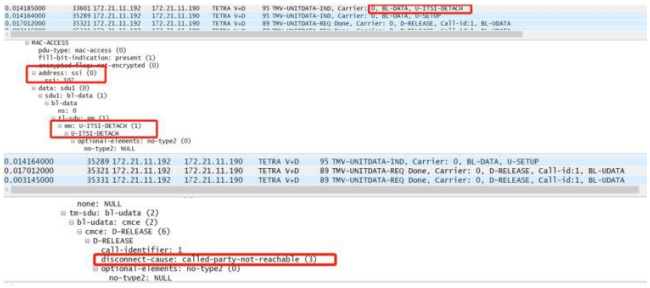

Figure 8. The results of the attack verification

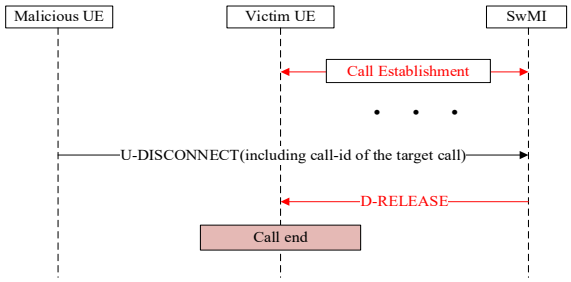

Figure 9. The process of the attack

In the next step, we aim to focus on the following directions: (i) enhancing the usability of model checking in mobile communication; (ii) improving the automation of model checking; and (iii) investigating defense techniques to strengthen the communication system.

REFERENCES

[1] Liu, Fugang, et al. "Lightweight trusted security for emergency communication networks of small groups." Tsinghua Science and Technology 23.2 (2018): 195-202.

[2] Seba, Abderazek, et al. "A review on security challenges of wireless communications in disaster emergency response and crisis management situations." Journal of Network and Computer Applications 126 (2019): 150-161.

[3] Pfeiffer, Martin, et al. "Analyzing TETRA Location Privacy and Network Availability." Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices. 2016.

[4] Zahednejad, Behnam, Mahdi Azizi, and Morteza Pournaghi. "A novel and efficient privacy preserving TETRA authentication protocol." 2017 14th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC). IEEE, 2017..

[5] Oluwafemi, Akintunde Jeremiah, and Jinjuan Heidi Feng. "Usability and security: a case study of emergency communication system authentication." HCI International 2019-Posters: 21st International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings, Part I 21. Springer International Publishing, 2019.

[6] van de Beek, Stefan, and Frank Leferink. "Robustness of a TETRA base station receiver against intentional EMI." IEEE transactions on electromagnetic compatibility 57.3 (2015): 461-469.

[7] Hussain, Syed, et al. "LTEInspector: A systematic approach for adversarial testing of 4G LTE." Network and Distributed Systems Security (NDSS) Symposium 2018. 2018.

[8] X. Hu, C. Liu, S. Liu, W. You, Y. Li and Y. Zhao, "A Systematic Analysis Method for 5G Non-Access Stratum Signalling Security," in IEEE Access, vol. 7, pp. 125424-125441, 2019, doi: 10.1109/ACCESS.2019.2937997.

[9] ETSI EN 300 392-2, Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D);Part 2: Air Interface (AI).

[10] ETSI EN 300 392-7: "Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D);Part 7".