Discovering Emergency Call Pitfalls for Cellular Networks with Formal Methods

Kaiyu Hou*
Northwestern University
Illinois, USA
kyhou@u.northwestern.edu

You Li*
Northwestern University
Illinois, USA
youl@u.northwestern.edu

Yinbo Yu
Northwestern Polytechnical University
Shaanxi, China
yinboyu@nwpu.edu.cn

Yan Chen
Northwestern University
Illinois, USA
ychen@northwestern.edu

Hai Zhou
Northwestern University
Illinois, USA
haizhou@northwestern.edu

ABSTRACT

Availability and security problems in cellular emergency call systems can cost people their lives, yet this topic has not been thoroughly researched. Based on our proposed *Seed-Assisted Specification* method, we start to investigate this topic by looking closely into one emergency call failure case in China. Using what we learned from the case as prior knowledge, we build a formal model of emergency call systems with proper granularity. By running model checking, four public-unaware scenarios where emergency calls cannot be correctly routed are discovered. Additionally, we extract configurations of two major U.S. carriers and incorporate them as model constraints into the model. Based on the augmented model, we find two new attacks leveraging the privileges of emergency calls. Finally, we present a solution with marginal overhead to resolve issues we can foresee.

CCS CONCEPTS

• **Networks** → **Formal specifications**; **Protocol testing and verification**; *Mobile networks*; • **Security and privacy** → Mobile and wireless security.

KEYWORDS

Cellular Network Protocol, Emergency Call, Protocol Specification, Protocol Formal Verification, Formal Methods

ACM Reference Format:
Kaiyu Hou, You Li, Yinbo Yu, Yan Chen, and Hai Zhou. 2021. Discovering Emergency Call Pitfalls for Cellular Networks with Formal Methods. In *The 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '21), June 24–July 2, 2021, Virtual, WI, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3458864.3466625

*Both authors contributed equally to this research.

MobiSys '21, June 24–July 2, 2021, Virtual, WI, USA

ACM ISBN 978-1-4503-8443-8/21/06.
https://doi.org/10.1145/3458864.3466625

1 INTRODUCTION

Public Safety Answer Points (PSAPs), such as the 911 call center in the U.S. and the 112 emergency service center in Europe, receive millions of emergency calls each year. Those calls are then dispatched to entities like police stations, fire stations, and ambulance centers. Cellular network systems play critical roles in this service. For example, in the U.S., there are an estimated 240 million emergency calls made to 911 each year, and more than 80% of those calls are from cellular devices [44]. Considering the huge number of emergency calls made through cellular networks and the potential impact of missing any of them, the correctness of cellular emergency call services deserves a close investigation by researchers. Unfortunately, as far as we know, there is no thorough research on this topic.

In 2019, a news report was widely spreading and had terrified the public in China. During an emergency medical situation, a cellphone (referred to as UE, user equipment) could not dial 120, the ambulance emergency number in China, even though a valid domestic SIM card was inserted (§3.1). Similar complaints about the availability of the cellular emergency call services can be found by skimming over public websites. In fact, a cellular emergency call system can be unreliable and is subject to attacks. First, it is difficult to unify the system design, as different countries have different emergency numbers and different settings for historical reasons. Second, there exists no economic incentive for the cellular network carriers or the UE device manufacturers to optimize system implementation or correct existing errors.

We aim to systematically address the availability and security issues in cellular emergency call systems and explain their underlying causal mechanisms in depth.

Researchers have widely adopted formal methods, including model checking [32, 34, 60] and symbolic analysis [1, 12, 14, 20], into network protocol studies. Given a set of security properties and a specified model, formal verification tools can either verify that the model upholds these properties or shed light on how the model violates them. However, most works suffer from two major problems in *formal specification*: modeling granularity and misrepresentation. First, the inappropriate level of granularity either requires enormous efforts in building models or can lead to false positives. Second, models specified from the protocols may not

precisely represent the deployed systems, as a real-world system is just an instance from generally broad protocols.

To solve these difficulties, we propose a *seed-assisted specification* method. It combines prior knowledge and adaptive model construction in addition to protocol-based formal specifications. Using the above-mentioned case in China as a starting point, we first manually investigate its underlying causes. Instead of building a model of the whole system and aimlessly checking it for possible vulnerabilities, we limit the scope to the problem we want to explore: the routing failures of emergency calls. Therefore, we can include any related details into our model, slice away unrelated ones, and, most importantly, control the right level of granularity. Existing works aim to discover vulnerabilities in protocol designs. We believe that discovering vulnerabilities in deployed systems is equally or even more critical. In that sense, we augment our model by configurations measured from deployed systems. As a benefit, we can ensure the counterexamples we found are practical in the real world.

Following the proposed method, we conduct the first research to systematically study the availability and security issues in cellular emergency call systems. Our thorough investigation of cellular emergency call systems is another major contribution of this paper. We have released our formal specifications as open source. (§3)

With this model, we systematically find 4 emergency call failure scenarios in China. In any of the scenarios, emergency calls cannot be routed to PSAPs. One of the major carriers has confirmed our findings. The public is unaware of all these scenarios, while 2 of them are unknown to carriers. (§4)

Moreover, we allow the flexibility to incorporate other configurations as model constraints to check other systems. We measure the emergency call configurations of two major U.S. carriers and find 2 new attacks abusing privileges of emergency calls. One attack is the first reported attack, which can bypass the screen password of UEs and make any calls. Another attack can block calls to the targeted numbers. It does not affect calls to other numbers and keeps effective longer than similar known attacks. We have reported them, and they are acknowledged by corresponding carriers. (§5)

We recommend a solution addressing all those failures and attacks, and show its correctness. We argue that lacking regulations or financial stimuli is another factor for the prevalent weaknesses in emergency call systems. (§6)

Finally, we summarize step-by-step procedures for the proposed *seed-assisted specification* method. This method can be easily extended to other systems which are described by protocols in general. (§7)

2 APPLYING FORMAL METHODS TO CELLULAR NETWORKS: CHALLENGES AND OUR SOLUTIONS

As opposed to manual investigation [33, 39, 52], the use of formal methods brings a systematic and solid approach to cellular network research. In that sense, formal methods are introduced to protocol verification and have succeeded in cryptography-related analysis, such as authentication and key agreement (AKA) protocols [1, 19].

Nevertheless, problems emerge when applying formal methods to general cellular network protocols. Formal verification cannot be performed directly on human-language-based protocols. These protocols need to be translated into formal models first (*e.g.*, finite state machines). Meanwhile, a set of security properties are extracted from the protocol requirements. The process of constructing formal models and properties is called *specification*. The verification problem is to check whether models always follow their corresponding properties. However, meaningful attacks and vulnerabilities are unlikely to be found on an arbitrarily constructed specification without the guidance of strong prior knowledge.

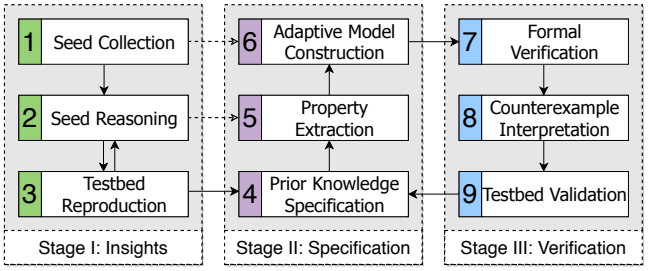


Figure 1: Framework of Seed-Assisted Specification

Formal verification then executes in iterations. Each time the model checker finds a violation to the properties and returns a counterexample, researchers need to validate the counterexample trace and determine whether it is feasible on a real-world system. Because there are always gaps between protocol definition, formal specification, and system implementation, a violation to a model built on protocols alone may not reflect a real-world problem. Researchers then have to exclude such an infeasible counterexample from the model and proceed to the next iteration. Without incorporating the information of real systems, the verification process can iterate forever until finding a non-trivial issue.

Therefore, there is still much room for improvement in the current approaches for verifying cellular network protocols. Several works [32, 59, 60] lie in this field, yet they all suffer from two major problems in specification: *modeling granularity* and *misrepresentation*.

- *Modeling Granularity*: In general, there is no golden rule to guide the granularity of modeling. Coarse-grained specification leads to false counterexamples because abstractions over-approximate the possible behaviours of a model. In contrast, the uniform fine-grained specification requires enormous efforts to build a model and has prohibitively large state space for model checkers.
- *Misrepresentation*: Protocols are generally broad and include many implementation options. Meanwhile, a real-world system is just an instance of protocols, and every possible option in protocols becomes a fixed assignment on the system. Therefore, models specified from protocols may not accurately represent the deployed systems.

From our observation, security analysis for cellular network systems has the following characteristics: *i)* an exposed critical security issue can attract widespread attention and is worthy of efforts to investigate the underlying causes; *ii)* a systematic search of potential vulnerabilities under similar causes is important; and *iii)* some configurations of deployed systems can be measured on-air

or at the UE side. We believe that capturing these characteristics is the key of conducting formal analysis on cellular network systems.

Thus, we propose the *seed-assisted specification* method to solve both the modeling granularity and misrepresentation problems. Different from general protocol formal analysis, we leverage the prior knowledge, which is gained from investigating existing security issues and measuring real-world systems, to augment protocol-based specification. To this end, we use an exposed critical security issue as the "seed". The scope of verification, *i.e.*, what procedures to keep and their corresponding granularities, are decided by investigation and reasoning on the seed. When constructing a model from protocols, the implementation configurations and other flexibilities are distinguished and collected. In that way, real-world situations can later be incorporated into the model to reflect real-world systems.

Figure 1 shows the framework of the proposed method. It consists of 3 major stages in 9 steps. Prior knowledge is gained from and verified in the *Insights* stage. In the *Specification* stage, we specify the model, design security properties the system should satisfy, and adapt real-world configurations as model constraints. Counterexamples are generated in the *Verification* stage by model checkers. With our prior knowledge, we can interpret counterexamples as availability issues or attacks on real-world systems and test their validities.

In the rest of this paper, we first provide a case study, specifying cellular emergency call systems and discovering their pitfalls, to illustrate each of the steps by examples (§3, §4, and §5). We then formally explain each of the steps in §7. We believe it would be easier than before for researchers and engineers to secure their systems from vulnerabilities by following these steps.

3 SPECIFICATION AND VERIFICATION OF EMERGENCY CALL SYSTEMS

For our study, we take a security issue exposed in 2019 as the seed and aim to systematically address the availability and security issues in cellular emergency call systems. We give an in-depth explanation of their underlying mechanisms and provide a solution to these vulnerability issues. In this section, we first manually investigate all underlying causes of the seed, then build a formal model for cellular emergency call systems. Based on this model, we systematically find 4 emergency call failure scenarios similar to the seed. Moreover, we measure the emergency call system configurations from two major carriers in the U.S. and incorporate them as model constraints. As a result, we find 2 new practical attacks.

We verified all of the 4 failures and 2 attacks by real-world experiments on commercial UEs and public cellular networks. One of the major Chinese carriers has confirmed these 4 failures. We also reported to the corresponding U.S. carriers about the 2 attacks, and the carriers have acknowledged them.

3.1 The Seed: A Piece of Shocking News

In 2019, news about the failure of emergency calls attracted public attention [58]. As reported by major Chinese media outlets, in an emergency medical situation, the victim tried to dial 120, the ambulance emergency number, using her Meizu MX6 UE. A SIM card with a valid subscription from carrier China Mobile was inserted in the UE. However, all 120 phone calls made through that UE, no matter the ones from the locked screen emergency panel or the unlocked normal panel, could not reach the ambulance emergency center. What she heard was only a repeated dialing instruction, as shown in Figure 2.

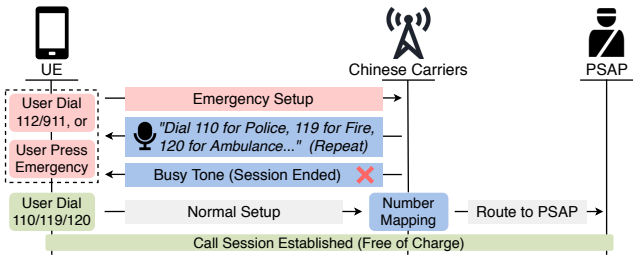


Figure 2: Calls initiated by the `Emergency Setup` signaling can only receive a recorded instruction. Carriers only route calls initiated by the normal `Setup` signaling.

3.2 Stage I: Insights

Step 1: Seed Collection. We performed the seed collection from three aspects: the cellular network protocols, the implementation details of UEs, and the configurations of carriers.

The cellular network standard is developed by the 3GPP (3rd Generation Partnership Project), which consists of more than one thousand documents. By leveraging the seed, we do not need to investigate all of them. Instead, we narrow our reasoning and specification to call setup protocols [7–9] and emergency call-related protocols [6, 10].

From news interviews with the victim, we were able to collect the UE technical specification and system image (referred to as ROM) version. We were also able to record the procedure that she executed during the failure. Additionally, after this security issue was exposed, the Meizu corporation announced that they had solved this problem by releasing a ROM update. This ROM was also collected for investigation. We present more details about collecting information of UEs and carriers in *Step 6: Adaptive Model Construction*.

Step 2: Seed Reasoning. The 2G/GSM (Global System for Mobile Communications) emergency call service is inherited from the traditional landline system for public safety. In GSM, the normal `Setup` signaling is used to initiate normal phone calls, while the `Emergency Setup` signaling initiates emergency calls [7]. Two widely used emergency numbers, 112 (Europe) and 911 (North America), are listed as fixed emergency numbers [10]. When users dial them, UEs always send out emergency signaling.

When GSM was introduced in China in 1994, three separate emergency numbers, 110 (police), 119 (fire), and 120 (ambulance), had already been standardized and used for the landline system. All emergency calls to them were directly routed to their corresponding callees. The landline system uses the ITU (International Telecommunication Union) signaling system. It does not support `Emergency Setup` [35].

To deploy GSM in China, the following compromise was accepted (Figure 2). When UEs initiate the `Emergency Setup` signaling, (most likely when the users dial 112/911 or press the emergency button),

carriers will not route these calls. Instead, carriers will loop sound recording of instructions to notify users of the correct emergency numbers in China. In contrast, call requests to 110/119/120 will be successfully routed if normal `Setup` initiates them. At a glance, such a scheme is backward compatible with the legacy emergency processing system, while it does also "respond to" the `Emergency Setup` signaling to some extent. This setting has been inherited by 3G/4G networks and is still active today.

We therefore speculated that when the victim dialed 120, the Meizu MX6 falsely initiated the call with the `Emergency Setup` signaling and was thus rejected by Chinese carriers.

Step 3: Testbed Reproduction. We reproduced the seed by using the same model UE with the same ROM. MX6 supports two major Chinese carriers. We dialed all emergency numbers with their SIMs under both the normal and the emergency panels. Calls from MX6 could not be successfully routed to PSAPs under any of these situations. Packet sniffer tools showed that MX6 used `Emergency Setup` to initiate these dials. We also used other UEs to initiate calls to 110/119/120 with `Emergency Setup`. All of them failed. These experiments proved that our seed reasoning is correct.

3.3 Stage II: Specification

Step 4: Prior Knowledge Specification. Following the idea of the *seed-assisted specification*, we exploited the prior knowledge to decide whether to keep, drop, or abstract a procedure or sub-procedure. Specifically, we are focusing on availability and security issues surrounding emergency call systems. From the seed, we learned that the most critical problem is related to routing: whether an emergency call can be routed and how exactly it is routed. The whole routing process is decided by the NAS (non-access stratum) layer protocol, which manages the communication session between the UE and the network [8, 9]. Procedures on the NAS layer depend on the bearers established on the RRC (radio resource control) layer [5]. However, we were not interested in modeling the RRC layer as any call will fail if the RRC layer fails.

Among all procedures on the NAS layer, *call control* procedures are most important. Meanwhile, the *attach* procedures are closely related to call control procedures. For instance, attach status and session contexts will afterward impact call setup and connection. The other procedures, such as *handover* procedures, *detach* procedures, and *identification* procedures, do not have a direct impact on routing an emergency call. Therefore, the details of those procedures were abstracted away. We kept the skeletons of those procedures to ensure our model can still depict the whole call process.

We further analyzed in detail the call control and the attach procedures. The seed suggests that problems are likely to occur on occasions that normal procedures and emergency procedures are different. So, we distinguished the details that make normal and emergency calls different. For example, the available services are different depending on if the UE is attached to the network in emergency mode; the routing process and the response of the network are different depending on if the UE sends the call request in emergency mode.

We built our formal model in TLA+ [37]. The model has two major components: the UE and the network. Both components are flattened to avoid the hierarchical network structure between layers. A message channel synchronizes their interactions. A total of 36 configurable variables are included in our specification. Behaviors of the model are characterized by 20 TLA+ procedures. The original model with no constraints yields 10.59 billion distinct states and has a maximal diameter of 26 transitions from the initial states.

The open-source model, as well as corresponding model checking and counterexample interpretation utilities, are available online.[1]

Step 5: Property Extraction. There are two major categories of properties: *safety* and *liveness*. Safety checking can guarantee the system never enters designated bad states, while liveness checking is typically used to check availability.

We elaborated the requirement that emergency calls *should* be routed to correct PSAPs with a *Liveness Property* ϕ_1: *If a user dials a local emergency number in China, the call should eventually be routed to the corresponding PSAP.* It states the basic availability requirement for emergency call systems in both 3GPP protocols [6, 10] and telecommunication regulations of China [42, 43].

We also detailed a *Safety Property* ϕ_2: *Any call should not be routed to a non-corresponding callee.* It has two implications. First, a call initiated by `Emergency Setup` shall not be routed to non-PSAP destinations. It eliminates the chance of adversaries leveraging emergency call privileges in normal dials. Second, a call made to a normal number shall not be routed to PSAPs. It prevents the possibility that emergency call systems interfere with normal calls.

Step 6: Adaptive Model Construction. We have found that using only the protocols is insufficient to discover or reproduce vulnerabilities in real, deployed systems. In many situations, a pitfall can only be reproduced on certain UEs and carriers. Their specific configurations should be modeled as *model constraints*. A formal model built on protocols is usually broad and lacking these details. Therefore, it is important to augment information from other sources to a general model.

First, it is necessary to locate the key configurations which can affect the "seed" problem. After *Seed Collection* and *Seed Reasoning*, we can locate a couple of key factors. Their assignments are determined by the literature survey, code analysis, or measurement. Next, if the model checking result is non-deterministic on the model, it usually indicates some key variables are missing. We should refine the model further.

Specifically, a UE can be considered to be one instance of the protocols. Following this idea, we analyzed the source files related to the *telephony* functionality from the Android Open Source Project (AOSP) [28] and Meizu MX6 ROMs.

Emergency calls have many privileges defined by the protocols, such as authentication-free registration and toll-free. Nevertheless, all of them rely on the configurations of carriers. Some detailed configurations of carriers were acquired indirectly from packet sniffing. We used QxDM [49], MTK Catcher [41], and MobileInsight [40] to sniff packets going between UEs and carriers. For directly testing a particular response, we programmed our UEs to send corresponding requests. For example, we programmed a UE without a valid subscription to test how carriers respond to an emergency attach request. Other configurations were partially inferred from the publicly available documentation by solution providers, *i.e.*, Cisco and Huawei [17, 18, 31]. In this paper, we denote the original model

[1] https://github.com/FormalCellular/EmergencyCall

as $\mathcal{M}$, and the adapted model as $\mathcal{M}^*$. More details about model construction are elaborated in §4 and §5.

3.4 Stage III: Verification

Step 7: Formal Verification. Model checking was executed with TLC [64] on an 8x3.6GHz machine with 64GB of RAM. Verifying the 4 failures took 195, 248, 694, and 328 seconds, respectively, while finding the 2 attacks took 508 and 309 seconds, respectively.

Step 8: Counterexample Interpretation. We took different approaches to interpret the counterexamples found by availability analysis (§4) and security analysis (§5). For availability analysis, we kept asserting that ϕ_1 fails. Assignments to configuration and condition variables were refined until we found the root cause of one issue. For security analysis, any violation of ϕ_2 could constitute a potential attack.

Step 9: Testbed Validation. For availability issues, we simply verified them with off-the-shelf UEs and real-world carriers. For potential attacks, we constructed a threat model and evaluated their feasibility using our hardware testbed.

§4 and §5 provide more details about the verification stage. In §4, we use our formal model to systematically study the emergency call availability issues in China. In §5, we augment system configurations of U.S. carriers as model constraints and discover security vulnerabilities on them.

4 AVAILABILITY STUDY: ROUTING FAILURES OF EMERGENCY CALLS

We explain our methodology for failure discovery in §4.1. We found 4 emergency call failure scenarios in China. These scenarios are elaborated and discussed in §4.2.

4.1 Failure Discovery

The purpose of availability checking is not just to point out that there exist failures in some scenarios. Rather, it attempts to undermine the essential causes of the failures.

Initially, our model is augmented by the system constraints of carriers in China (*Step 6*). Here is how it looks like:

$$o \triangleq \wedge \texttt{network\_route\_with\_number\_or\_type} = \texttt{number} \\ \wedge \texttt{network\_emergency\_numbers} = \{110, 119, 120\} \\ \wedge \ldots$$

which says the network routes calls based on the callee number instead of the `Setup` message type; the network only recognizes `110,119,120` as emergency numbers.

Besides, the behavior of the model also depends on a set of condition variables, c, which is the abstraction of a scenario. For example,

$$\begin{aligned} c \triangleq & \wedge \texttt{ue\_sim\_present} = \texttt{False} && (c_1) \\ & \wedge \texttt{ue\_screen\_locked} = \texttt{False} && (c_2) \\ & \wedge \texttt{user\_dial\_panel} = \texttt{normal\_panel} && (c_3) \\ & \wedge \ldots \end{aligned}$$

The values of condition variables keep unchanged after model initialization. These values contain the root cause of a failure when the model violates properties.

Our initial liveness property, ϕ_1, states that: *If a user dials a local emergency number in China, the call should eventually be routed to the corresponding PSAP.* The strengthened negation of it, ϕ_1^*, becomes: *If a user dials a local emergency number in China, the call should never be routed to the corresponding PSAP.* If ϕ_1^* is true, ϕ_1 should definitely be false. Checking the correctness of ϕ_1^* has several benefits. *i)* ϕ_1^* is now a safety property, which significantly benefits the execution time of the model checker. *ii)* By checking the safety property, we can avoid finding infinite loops in the model. Trivial loops in some local procedures can thwart liveness checking, *e.g.*, the case that users keep dialing and hanging up; safety checking can bypass such problems. *iii)* Most importantly, only then are we able to find the root cause, which always leads to emergency call failures.

We start by searching for a full assignment to all condition variables, $c = c_1 \wedge c_2 \wedge \cdots \wedge c_n$, and query the model checker on model $\mathcal{M}^*$ for ϕ_1^*. In practice, we can find such an assignment that satisfies ϕ_1^* from our insights. But the assignment is indeed the smallest cube, which leads to a very narrow real-world scenario. Then we attempt to remove a c_k from the current cube c. It can be removed if the cube after removal still satisfies ϕ_1^*. The process terminates when no more condition can be removed. The resulting cube, c^*, which describes an essential condition of a failure, is called *condition core*.

Then the condition core is ruled out from the model: $\mathcal{M}^* \leftarrow \mathcal{M}^* \wedge \neg c^*$. We iterate on the process in the last paragraph to extract the next condition core.

The order of removing c_k can decide the result of the current condition core. However, will the order of removal impact the set of found condition cores? No, because the other condition cores can still be found later, as any state $s \in \neg c^*$ is guaranteed to satisfy ϕ_1^* and violate ϕ_1.

4.2 Failure Scenarios

We found 4 meaningful scenarios that an emergency call cannot be routed to a PSAP, denoted as ***F-1***, ***F-2***, ***F-3***, and ***F-4***. We have provided a summary of condition cores and their real-world interpretations in Table 1. All these scenarios are public unawareness, while ***F-3*** and ***F-4*** are unknown to carriers. Note that, we do not claim the search for failure scenarios is exhaustive.

F-1: *A call made in China cannot be routed to a PSAP if no SIM card is present.*

Explanation: Chinese cellular network carriers refuse to route a call with the `Emergency Setup` signaling. In the case that no SIM card is present, UEs will stay in *limited service state* [5] and can only provide "emergency calls only" service. Those calls initiated by `Emergency Setup` cannot be successfully routed to PSAPs in China.

Our experiment shows that all GSM/3GPP UEs we have tested fall into ***F-1*** (Table 2). We are not able to control the exact carrier a UE attaches to, as no SIM card is present. Therefore, we perform our testing in multiple locations in three different cities.

Public Unawareness: We initially thought ***F-1*** is common knowledge to the public. After seeing discussions online and surveys

Table 1: Four scenarios of emergency call routing failures in China. (found via TLC, verified in the real world)

Failure Scenario	SIM Inserted	Roaming	Localization	Voice Subscripted	Dialed from Normal Panel	When an emergency number is dialed in China: This call would **fail to be routed** to a PSAP if ...	Affected UEs
F-1	✗	-	-	-	-	*No SIM is inserted in the UE.*	All
F-2	✔	✗	✗	-	-	*The UE is not localized correctly.*	Partial
F-3	✔	(✔)	-	✗	-	*The SIM does not have a valid subscription.*	All
F-4	✔	✔	-	(✔)	✗	*The User dials from the emergency panel.*	Partial

"✔", "✗", "-" indicate True, False, no restriction, respectively. "(✔)" indicates no restriction but only a True value makes the case non-trivial.

Table 2: *F-1*: Availabilities for GSM/3GPP UEs to dial emergency numbers when no SIM card is present.

Number	110/119/120			112/911		
City	Beijing	Hangzhou	Wuhan	Beijing	Hangzhou	Wuhan
Available	✗	✗	✗	✗	✗	✗

offline, we believe a vast majority of the public holds the opposite opinion.

This wrong impression might be due to the following reasons: *i)* most people do not have a real experience of dialing an emergency number; *ii)* UEs will show "emergency calls only" on screens when users remove their SIMs, which misleads people that "emergency call is available" without SIMs.

F-2 *A call made in China cannot be routed to a PSAP if the UE does not have correct localized configurations.*

Explanation: Initially, UEs could not identify local emergency numbers, *e.g.*, 110/119/120. Hence, when the UE was in a geographic region where the subscribed carrier provides no coverage, the UE could not utilize emergency channels from other carriers. To solve this problem, starting from the era of 3G, 3GPP requires carriers around the world to store emergency numbers of home countries in SIMs [4]. Since then, UEs can recognize local emergency numbers when local SIMs are inserted.

This requirement negatively impacted the cellular emergency call system in China. Before that requirement took effect, a call to emergency numbers of China was made through normal `Setup` and could then be routed to PSAPs. However, after UEs identify a dialed number as the emergency number, this call is initiated by `Emergency Setup` and then fails in China. Consequently, UE device manufacturers have to take some compromised solutions, called *localization* in this paper.

Localization: We unpacked several ROMs of Android UEs that did not fall in ***F-2*** and investigated their source codes. We will summarize our findings in the following. When the UE identifies that the user is dialing an emergency number of China and the UE is attached to a Chinese carrier, the Android operating system (OS) will display "emergency dialing" on the screen. Meanwhile, the OS will command the hardware to make a call through normal `Setup`.

Different UE device manufacturers have their own implementations of this idea. Code 1 is a segment from the *ecc_list.xml* file of the Xiaomi Redmi 6A. The localized Android OS queries this file after the emergency number identification process. As shown in

```
1  <!--Condition: there are following values:
2    - 0: ecc only when no sim
3    - 1: ecc always
4    - 2: show ecc but send as normal   -->
5  <!-- Add for China CTA -->
6  <Ecc="110" Condition="2" Plmn="460FFF"/>
7  <Ecc="119" Condition="2" Plmn="460FFF"/>
8  <Ecc="120" Condition="2" Plmn="460FFF"/>
9  <!-- 3GPP 22.101 -->
10 <Ecc="112" Category="0" Condition="1" />
11 <Ecc="911" Category="0" Condition="1" />
```

Code 1: *F-2*: Excerpt of *ecc_list.xml* from Xiaomi Redmi 6A. Condition 2 is enforced when attached to a Chinese carrier (MCC 460) and dialed number 110/119/120 (L6-L8).

this segment, if the UE is attached to a PLMN[2] in China (MCC 460) and the dialing number matches any of the three entries (L6-L8), the system will enforce *condition 2*, sending normal `Setup` for this call. However, this solution has nothing to do with ***F-1***, because normal `Setup` is disabled when the SIM card is not present.

The default AOSP source code does not provide this special modification. Therefore, UEs using default AOSP or making mistakes in implementation (the seed case) are not correctly localized. Emergency calls from them cannot get routed in China. Public news shows that almost all major UE device manufacturers have made mistakes similar to the seed case one after another in the past decade.

F-3 *A call made in China from a roaming UE cannot be routed to a PSAP, if the UE does not have a valid subscription.*

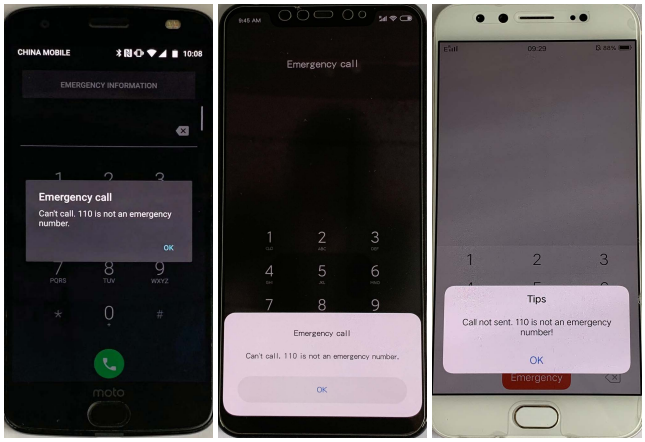
Explanation: If a foreign SIM card is present, it is possible for the roaming UE to pass the authentication and then attach to the network. An emergency call can still not be made as normal call service is unavailable without a valid subscription. It means any users who have not activated their roaming services beforehand have no access to emergency call service in China. This scenario also applies to roaming UEs with the data-only roaming plan [51]. As opposed to ***F-1***, roaming UEs fell in ***F-3*** can use keys stored in the SIMs to authenticate partnered carriers in China.

We tested SIMs from four major U.S. carriers[3] (Table 3). Among them, UEs with SIMs from carrier US-V can attach to carrier CN-T's network without activating roaming services. Emergency calls made from none of them can be routed.

[2]PLMN (public land mobile network) consists of an MCC (mobile country code) and an MNC (mobile network code), corresponding to a carrier.
[3]The four major carriers in the U.S. are AT&T (US-A), Verizon (US-V), T-Mobile (US-T), and Sprint (US-S). The three major carriers in China are China Mobile (CN-M), China Unicom (CN-U), and China Telecom (CN-T).

Table 3: *F-3*: Availabilities for roaming UEs to dial emergency numbers on the normal panel.

Subscription	No				Yes	
SIM issued	US-A	US-T	US-S	US-V	Either US	
Attach To	*Emergency Call Only*			CN-T	CN-M	CN-U
Available	✗	✗	✗	✗	✔	✔



(a) Moto Z2 **(b) Xiaomi 8** **(c) Vivo x9s**

Figure 3: *F-4*: UEs which cannot identify emergency numbers of China (*e.g.*, 110) when foreign SIMs are inserted.

F-4 *A roaming UE cannot initiate an emergency call in China by the emergency panel, even with a valid subscription.*

Explanation: In this scenario, a visitor, whose SIM is issued by a country which uses different emergency numbers from China, cannot make emergency calls on the emergency panel. Even though she has a valid SIM card inserted to avoid ***F-1***; is using a UE with localization to avoid ***F-2***; and has subscribed to roaming service to avoid ***F-3***, the emergency service is still unavailable when she dials without unlocking the screen. ***F-4*** best demonstrates the power of formal methods. It is hard to discover without systematic formal analysis. Yet it is easy to reproduce once found (Figure 3).

Reasoning: The purpose of the emergency panel is to allow users to dial emergency numbers without unlocking the screen. Nevertheless, emergency numbers differ from country to country. There are four sources for UEs to determine emergency numbers (Figure 4) [10]. First, two *fixed emergency numbers*, 112 and 911, are always identified as emergency numbers by UEs. Second, six common emergency numbers are stored in the *UE default emergency number set*. As shown in *Case 1*, if no SIM is present, those numbers are identified as emergency numbers. Third, *local emergency numbers* are stored on SIMs issued by local carriers. When a SIM is inserted, the *UE default emergency number set* will no longer be effective. Fourth, to notify a roaming UE with local emergency numbers, carriers can push the *local emergency number list* when UEs attach to them. Nevertheless, our measurement shows that none of the

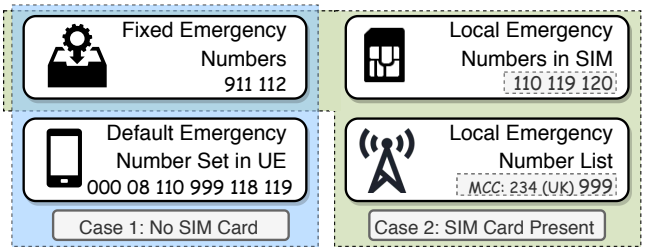


Figure 4: Four sources used by UEs to identify emergency numbers. Inserting a SIM card will invalidate the UE default emergency number set.

Chinese carriers push this list, which leads to the failure stated by ***F-4***. More details about the *local emergency number list* are discussed in §5.5.

Please notice this failure cannot be mitigated by simply pushing the *local emergency number list*. Otherwise, if roaming users call emergency numbers on the *normal* panel, these calls will be initiated by `Emergency Setup` and thus fail.

Outside China. Similar problems can happen beyond China. Any countries that have multiple emergency numbers or have an emergency number other than 112/911 can also suffer from this problem. For instance, a thread on a Japanese forum discusses such a case: one cannot dial 110, the police emergency number in Japan, through *SoftBank*'s network [36].

5 SECURITY STUDY: ABUSE OF EMERGENCY CALL PRIVILEGES

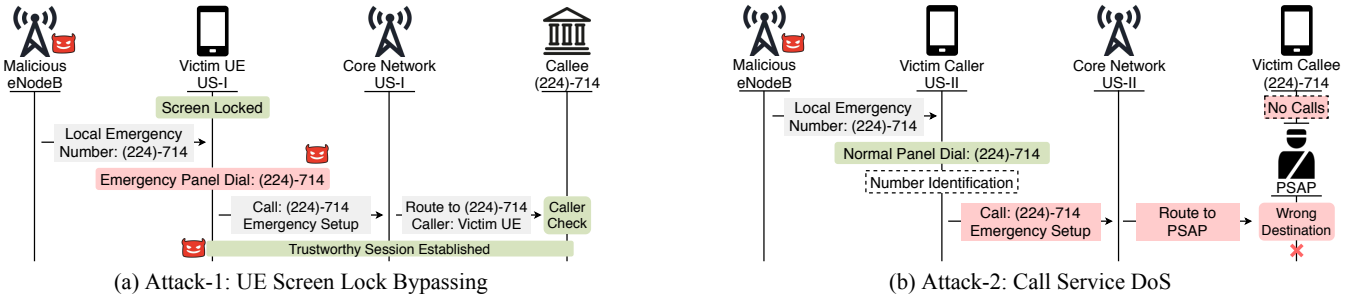
We leverage the formal model to discover potential attacks in §5.1 and define the threat model in §5.2. We find two new attacks. Both of them can have significant impacts on major U.S. carriers. They are introduced in §5.3 and §5.4. In §5.5, we demonstrate ways to deploy these attacks in the real world.

5.1 Attack Discovery

We augment the general model of emergency call systems, $\mathcal{M}$, with our measured configurations of two major carriers in the U.S. (denoted as US-I/US-II for anonymity). These two carriers use two distinct sets of configurations when routing calls with `Emergency Setup`. US-I determines the destination of a call only by the dialed number, while US-II routes a call to PSAPs provided `Emergency Setup` initiates it.

To find potential attacks, we assume an adversary who can impersonate one legitimate entity to inject messages within the channel between UEs and the network. Nevertheless, the adversary does not have any capability beyond a budget fake base station. For instance, the adversary cannot route calls from a victim UE to a real-world callee.

We use TLC to check whether those two augmented models satisfy the safety property ϕ_2. In fact, ϕ_2 is violated by both models. For carrier US-I, the adversary can bypass screen locking and SIM card locking to make phone calls. Nowadays, many individuals and companies use the incoming phone number to verify the identity of



(a) Attack-1: UE Screen Lock Bypassing (b) Attack-2: Call Service DoS

Figure 5: (a) *Attack-1*: UE Screen Lock Bypassing attack on carrier US-I. The adversary can bypass the password of the victim's UE and dial any number on behalf of the victim. (b) *Attack-2*: Call Service DoS attack on carrier US-II. The adversary can block phone calls to target phone numbers in a specific area. Calls to these numbers will be falsely routed to PSAPs.

the caller. The adversary can impersonate the victim by launching this attack. For carrier US-II, the adversary can block calls from US-II's subscribers to any phone numbers in a specific area. These calls will not be routed to correct destinations. This violation can be used to launch a denial-of-service (DoS) attack.

Both of the two sophisticated attacks leverage the *local emergency number list* feature. We can precisely specify this feature in fine-grain because we have prior knowledge coming from the reasoning of the seed case and measurement results on each of the two carriers. A general, abstract specification of protocols is unlikely to reveal these attacks.

5.2 Threat Model

We assume the adversary can set up a malicious base station, *i.e.*, eNodeB, to send sophisticated messages. We will discuss how to achieve this with existing techniques in §5.5. We also assume the adversary is geographically close to the victim's UE, where the adversary can impersonate the legitimate eNodeB of the targeted carrier by broadcasting messages with higher signal power. The message parameters related to carrier information can be learned by signal sniffing and analysis tools, such as QxDM and MobileInsight. In the UE screen lock bypassing attack, we assume the adversary can physically touch the victim's UE, while strong passwords protect both the UE and its SIM card.

5.3 Attack 1: UE Screen Lock Bypassing

Attack-1 *The adversary can dial any normal number on the emergency panel of the victim's UE and get routed to the callee, if the UE is a subscriber of carrier US-I.*

Objective of the Adversary: The adversary wants to initiate a normal call from the victim's UE to impersonate the owner of this UE. From the callee's viewpoint, the caller ID (phone number) belongs to the victim. However, when the screen is locked, the UE will block any phone calls it believes not to be an emergency number.

The adversary may not be able to simply put the victim's SIM into another compromised UE because either *i)* the victim uses a virtual SIM, or *ii)* a password protects this SIM.

Attack Description: Figure 5 (a) shows the attack process. ① The adversary puts the desired number in the *local emergency number list* and pushes the list to the victim's UE through a malicious eNodeB. Possible ways to push this list will be discussed in §5.5. ② The adversary dials the desired number on the emergency panel without unlocking the UE. Now the OS will accept the adversary dialed number as an emergency number and command the hardware to send out an emergency call. ③ As for carrier US-I, calls are routed base on the dialed numbers. Consequently, this call will be routed to the desired callee as if it is a normal call.

Attack Consequence: Many customer service centers today use incoming caller IDs as the identification of callers. Now the adversary can impersonate the victim on those calls. Besides, financial institutions, as well as other online companies, rely on caller IDs as an important source of two-phase authentication. Now the adversary is possible to get the temporary identification code by making a phone call.

Attack Novelty: This attack can achieve similar consequences as the already known *caller ID spoofing* attack [21, 26]. The latter forges the phone number of a trusted caller by falsifying the information transmitted to callees. Nevertheless, the found attack is indeed a different attack, whether in principle or in effect. As for ***Attack-1***, the caller ID shown on the callee side is not forged. Therefore, the found attack can bypass *any* state-of-the-art defense mechanisms for caller ID spoofing, *i.e.*, callee-end defense [21] and in-network defense [55]. Additionally, the adversary could also receive call-backs from the callee for confirmation. To launch this attack, the adversary will need to access the victim's UE physically. However, to the best of our knowledge, this is the *first* attack that can bypass the screen password and make phone calls.

5.4 Attack 2: Call Service DoS

Attack-2 *The adversary can block phone calls made to a set of any phone numbers in a specific area, if the caller UE is a subscriber of carrier US-II.*

Objective of the Adversary: The adversary wants to block phone calls to designated normal numbers within a region. She controls a malicious eNodeB and can broadcast messages with higher signal power. Thus, victims' UEs in that region will attach to her eNodeB. Additionally, she wants to *i)* avoid blocking other numbers, and *ii)* prevent always turning on the eNodeB to reduce the chance of being discovered.

Attack Description: Figure 5 (b) presents the attacking steps. ① Similar to *Attack-1*, the adversary pushes the fake *local emergency number list* to the victim's UE. Now the list stores the numbers which the adversary wants to block. ② Emergency number identification takes precedence over any other call-related processes [10]. When the victim dials these numbers on the normal panel, the OS will accept dialed numbers as emergency numbers and command the hardware to send them out with the `Emergency Setup` signaling. ③ Unlike carrier US-I, US-II disregards the dialed numbers. All phone calls through `Emergency Setup` will be routed to the PSAP. Thus, in effect, all calls made to these numbers are failed.

Attack Consequence: This attack can be used to obtain illegal economic benefits. For example, the adversary may want to disable phone calls from potential customers to business competitors. In addition, the adversary can get faster service by blocking others' competing calls to that service.

The adversary has an alternative way to leverage this vulnerability. She can broadcast a forged *local emergency number list* with popular numbers stored in it. All phone calls to those numbers from the subscribers of US-II will then be falsely routed to the local PSAP, which becomes a distributed denial-of-service (DDoS) attack to the local PSAP.

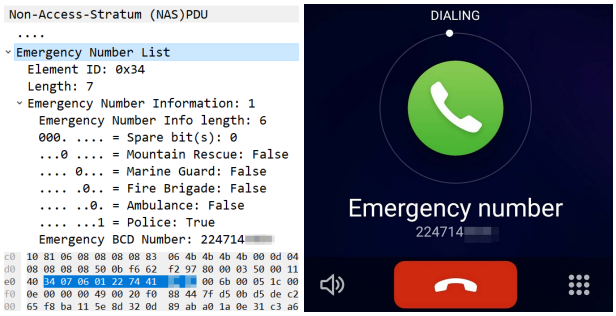
Attack Novelty: Comparing with existing call service DoS attacks [27, 32, 39, 54], the newly found attack has two different characteristics. First, it stays effective even after the malicious eNodeB is turned off. Second, the newly found attack only blocks the calls to a targeted set of phone numbers while keeps the calls to other phone numbers unaffected.

5.5 Deployment of Attacks

Background. NAS layer protocols are there to establish and maintain the communication session between the UE and the *core network*. Among those, the session establishment related procedures are called *attach* or *registration*. The *local emergency number list* is an optional Information Element (IE) in the `Attach Accept` message (for 2G-4G) and the `Registration Accept` message (for 5G). Within a *local emergency number list*, emergency numbers, together with their types and lengths, can occupy no more than 50 Bytes. The UE only stores the latest *local emergency number list* it receives from the network, meaning the previous list will be overwritten if a new list comes in [9].

Carrier US-I does not push the *local emergency number list*. Note the 911, the emergency number in the U.S., is always identified as an emergency number by UEs because it is a 3GPP fixed emergency number (Figure 4).

Implementation. We use USRP B210 [45] as the eNodeB hardware. It is driven by OpenAirInterface (OAI) [47], an open-source cellular network protocol stack emulator. The hardware and software suite supports the essential functionalities of the 4G core network and the eNodeB. The dedicated hardware costs about $1,200. The original OAI lacks implementation of the *local emergency number list*; only a stub interface is provided. We implemented this feature within the `Attach Accept` message. Figure 6 (a) shows the Wireshark decoded a NAS message we pushed to our UEs. This message contains the *local emergency number list* IE, which has one fake emergency number, (224)-714-*, in it.



(a) Wireshark Log (b) UE Screenshot

Figure 6: (a) Wireshark log of the fake *local emergency number list* we pushed. It contains (224)-714-*. (b) UE identifies the normal number (224)-714-* as an emergency number. We are dialing this number on the emergency panel without unlocking the UE.

Deployment. Both attacks are relying on the fake *local emergency number list* pushed by a malicious eNodeB. Although non-emergency `Attach Accept` messages are protected by encryption in 4G, ways to set up malicious cellular base stations and push fake messages are plenty [30, 38, 54]. However, enforcing any of them may violate the Federal Communications Commission's (FCC) regulation [25] and may block real-world phone services. For that reason, to show the proof of concept, we provide a new in-lab solution, called *dual-SIM leakage*, to launch these attacks. Only our controlled UEs will be affected by our malicious signal. Using existing ways of pushing fake messages, a real attacker can launch these attacks more easily. Next, we will discuss these ways. Please note, we do not claim the ways to push fake messages (except the dual-SIM leakage) are contributions of this paper.

The dual-SIM leakage of the *local emergency number list* is a vulnerability we find on the dual-SIM UEs. To be specific, the last received *local emergency number list* will overwrite the previous one, no matter which SIM we are using to make calls. Therefore, for our experiment, we utilize it by inserting two SIMs in a UE. One SIM is from the tested carrier (US-I or US-II), and the other is our customized SIM. The malicious *local emergency number list* is pushed on the experimental-licensed spectrum by our eNodeB. Only with our customized SIM can the UEs attach to our eNodeB. The broadcast signal will not affect other UEs. Then we make calls from the commercial SIM. Our experiment follows the setup mentioned in this section and validates the effectiveness of the proposed attacks. Figure 5 (b) shows that the UE identifies the normal number (224)-714-* as an emergency number and dials it out without unlocking the screen.

An attacker can use other exposed cellular network vulnerabilities to launch the proposed attacks in the real world. *i) Force 2G fall back:* Lin [30] first demonstrated how to practically force a UE redirecting from a 4G eNodeB to a 2G base station. 2G does not provide network side encryption. The attacker can then push the fake *local emergency number list* to the victim UE. After the UE reattach to a legitimate 4G eNodeB, the fake list is still effective, as carrier US-I does not send a new list to overwrite it. *ii) Force*

emergency attach: Yu *et al.* [63] proposed a method that can force a UE to set up *emergency attach* procedures. Emergency attach procedures have the privilege to skip authentication. In this case, the `Attach Accept` message is not encrypted, and the *local emergency number list* can be forged into it. *iii) Malicious Wi-Fi and fake DNS:* 3GPP has allowed interconnections with non-3GPP access networks. Specifically, local emergency numbers can be provided through DNS queries within non-3GPP access [3]. By setting up a malicious Wi-Fi AP and forging fake DNS responses [52], fake local emergency numbers are then pushed to the victim UE. In the emerging 5G, such non-3GPP interconnections can be more prevalent.

Limitations of the Attacks. We noticed a misimplementation on Qualcomm SoCs (System-on-Chips): they truncate an emergency number of more than 6 digits to its first 6 digits. The protocol does not limit the length of each emergency number in the *local emergency number list*. Although most common emergency numbers are 3 digits long, longer emergency numbers are widely used for some special *local* services, such as maintain rescue and marine guard. It is indeed a bug and a violation of clearly defined protocols. However, this bug does weaken the effect of our attacks on those UEs using Qualcomm SoCs. The misimplementation only happens to Qualcomm SoCs. Huawei and MediaTek SoCs do not truncate emergency numbers.

The carrier US-II has *local emergency number list* IE in the `Attach Accept` message with 911 in it; even 911 is a 3GPP fixed emergency number. Because of this setting, US-II can sometimes escape from the proposed attack depending on the message pushing frequency and mechanism.

6 RECOMMENDATIONS

We propose a solution addressing all failures and attacks in §6.1 and show its correctness in §6.2. In §6.3, we argue that lacking regulations or financial stimuli is another factor for the prevalent weaknesses in emergency call systems.

6.1 Proposed Technical Solution

We devise a solution consisting of 4 stages. It will be the carriers' responsibility to take these actions. The overhead of the solution is marginal.

① *Pushing Local Emergency Number List.* We suggest that all official local emergency numbers should be included in the *local emergency number list*. Pushing this list to serviced UEs shall be mandatory for carriers. A list containing 3 emergency numbers only takes 12 extra Bytes during *attach* procedures. Upon receiving the correct list, the malicious list pushed by the adversary will be overwritten.

② *Accepting Emergency Setup.* Following the discussion above, all emergency calls that distinguishable by UEs should always raise the `Emergency Setup` signaling instead of the normal `Setup` signaling, to indicate the case of emergency. Hence, it is the carriers' responsibility to handle `Emergency Setup` properly. On the other hand, it is favorable that carriers can route calls to local emergency numbers sent through normal `Setup` properly in cases UEs cannot detect them. Besides, carriers should allow *emergency attach* regardless of the subscription status of UEs.

③ *Emergency Numbers in SIMs.* A traveler just roamed to a new country may not know the local emergency numbers there. Instead, she may dial an emergency number in her home country. If the home emergency numbers are hardcoded into the SIM issued by her home carrier, the UE will deem them equivalently as other emergency numbers. Local carriers can handle these calls properly by following the routing indications from her home carrier [6].

④ *Filtering Non-emergency Numbers.* Making an emergency call is a privilege and can bypass authentications of users to UEs or of UEs to networks. Only emergency traffics should be allowed on the emergency channel. Network carriers should apply filters to block other traffics on the emergency channel. A possible way would be binding the filtering rules to the location: only calls made to the fixed, local, or home emergency numbers are allowed to be routed to the corresponding PSAPs, while other calls initiated by the `Emergency Setup` signaling should be rejected.

6.2 Correctness of the Proposed Solution

We show the correctness of the proposed technical solution in principle, by our formal model, and by the testbed.

F-1/***F-2***: Now that carriers correctly route `Emergency Setup`, users in these scenarios can access the emergency service. Such an improvement is also backward compatible with already localized UEs because carriers are still able to handle emergency requests initiated by normal `Setup`.

F-3/***F-4***: According to our solution, UEs download the *local emergency number list* when they attach to the network. As a result, the local emergency number identification is available. Users can now dial local emergency numbers no matter on the normal panel or the emergency panel. Besides, calls to emergency now can be routed to PSAPs, no matter roaming users dial home or local emergency numbers.

Attack-1: The adversary now cannot successfully dial any normal numbers from the emergency panel because of the added filter on the network side. Notice that there are no additional benefits for the adversary to dial an emergency number from the emergency panel.

Attack-2: Pushing the correct *local emergency number list* can overwrite the previously stored malicious list. In addition, the non-emergency filter rejects calls to PSAPs with normal numbers. It solves the potential DDoS threat to PSAPs.

We translated the solution into formal conditions. TLC proved that under these conditions, availabilities of emergency calls are now maintained in the 4 failure scenarios, and the 2 attacks are no longer possible. Please note, a formal specification cannot capture all information of real-world systems, so such a correctness proof is not complete. As a matter of fact, an interruption that happened to the physical layer can cause interruptions on any upper layers.

We also implemented a prototype on our testbed. Under the prototype, none of those availability issues and attacks can still affect the emergency call system. Nevertheless, as our testbed does not have the same capabilities to a real-world carrier, emergency calls on the testbed cannot really be routed to local PSAPs. We are collaborating with corresponding carriers regarding the deployment of the complete solution.

6.3 Social Economic Solutions

Cellular emergency call systems are technically complicated, yet this does not explain the extreme prevalence of attacks and reliability issues of these systems. We believe the root cause is the lack of motivation for carriers. Emergency call services are free of charge for end-users, which means the carriers may not put enough effort into testing and improving them. For those users who do not have a valid subscription or their subscribed carrier has no service in that region, it is even impossible for them to accuse other carriers.

We argue that cellular network features, which have high social impacts but make no profits, *e.g.*, emergency calls, shall be seriously considered and clearly defined by protocol designers. It is the social responsibility of the protocol committee to the public. Meanwhile, stronger regulations by authorities are also critical in solving this problem.

7 SEED-ASSISTED SPECIFICATION

We summarize all steps in the framework of the *seed-assisted specification* method in this section.

Stage I. Insights

Step 1: Seed Collection. A seed is an exposed issue on a security-critical system. In this step, all relevant information about this issue should be collected, such as the course of events and the circumstance when it happened. It can be collected from sources like official disclosures, news reports, and related protocols. In addition, specific modeling information is of great interest, including the system configurations, initial conditions, and execution procedures that lead to the issue.

Step 2: Seed Reasoning. The related parts in protocols should be looked through to find the execution path that raises this exposed issue. Although a protocol usually suggests a broad implementation and configuration space, the information in *Step 1* can help us to determine those configuration assignments. Sometimes, real-world measurements and investigations are also essential to portray the execution path. Reasoning can also help distinguish the essential procedures causing the issue and how they correlate to the whole system.

Step 3: Seed Reproduction. If the seed reasoning is correct, it would be possible to reproduce the security issue on the testbed. To simulate the real-world system, the test environment needs to be augmented with the configuration assignments. If the issue cannot be reproduced following the reasoning, the reasoning result in *Step 2* needs to be revised.

Stage II. Specification

Step 4: Prior Knowledge Specification. With the prior knowledge from *Stage I*, the security researcher can then specify the model, $\mathcal{M}$, in the appropriate level of granularity. Instead of building a model for the whole protocol with all possible details, we suggest limiting the scope to just explore the similar security issues and only expatiate related state transactions. Nevertheless, the specification should follow the protocols and provide flexibility to support all the possible options provided by the protocols.

Step 5: Property Extraction. Model checkers can verify whether $\mathcal{M}$ satisfies a given security property ϕ: $\mathcal{M} \models \phi$. ϕ can be extracted from either the protocols or regulations and should be able to reveal the execution path of the seed issue. In other words, the seed is a violation of ϕ. One can also extract other security properties to find other vulnerabilities.

Step 6: Adaptive Model Construction. We treat the real-world system configurations collected from *Step 1* as the observed model constraint, o. The adaptive model $\mathcal{M}^*$ is then the conjunction of $\mathcal{M}$ and o. This step assures the security issues reported by $\mathcal{M}^*$ to be practical for the real-world system. The general specification $\mathcal{M}$ can be reused on other verification tasks to the same protocol by re-applying model constraints with the configurations from other implementations. The benefits of adaptive model construction are more than being accurate and being universal for the model. The construction can also control the size of searching space for model checkers, reducing the execution time of verification.

Stage III. Verification

Step 7: Formal Verification. The verification problem is to check whether $\mathcal{M} \models \phi$ holds. If it does, the model checker returns with no counterexamples. Otherwise, it returns with a counterexample π, which is a trace of state transitions. The *initial condition*, c, can be extracted from the first state.

Step 8: Counterexample Interpretation. Not all counterexamples are feasible and meaningful. To interpret and reproduce a counterexample, it needs a decomposition of the counterexample into procedures, and then needs a close look into every procedure. If a π can be interpreted and reproduced without external intervention, we conclude it is a *failure*. If it is not a failure, but we can assume a reasonable attacker to practice the external intervention, we conclude it is an *attack*.

Step 9: Testbed Validation. We should try to reproduce each failure or attack that is potentially feasible on the testbed. If it is not reproducible, it means either we have mistakes or have over-approximation in our specification, leading to a false-positive counterexample. In both cases, we need to go back to the specification stage to revise the model and rerun verification. Finally, all failures and attacks reported by the model checker are valid in the real world.

Any systems characterized by human-language-based standards or protocols can benefit from our proposed method because inappropriate granularity and misrepresentation are inevitable in applying formal analysis. Therefore, the proposed method can be generalized to verify other security-critical systems and infrastructures [16], such as smart grids [53], intelligent transportation systems [62], and critical financial services [22]. In these systems, even small issues can have widespread consequences. In-depth investigations are always desired, including building formal models, running formal verification, and reasoning about deployed systems to reveal potential vulnerabilities.

8 DISCUSSIONS AND ETHICAL CONSIDERATIONS

Protocols. We limit our study to the GSM/3GPP series cellular network protocols. In reality, 3GPP protocols have become the *de facto* and are the only solution for the 4G and the emerging 5G. The CDMA/3GPP2 series protocols have been announced their ending in the 3G era [11]. In the era of 2G/3G, 3 major carriers in China and the U.S., namely CN-T, US-V, and US-S, support CDMA, while all others support GSM/3GPP. Nevertheless, those three have also

converged to GSM/3GPP series protocols in the era of 4G/5G and announced to terminate CDMA supports recently [56, 61].

UEs. CDMA based networks do not use the 3GPP `Emergency Setup` signaling. We noticed some UEs with only CDMA support can successfully connect to PSAPs in China without a SIM inserted. However, the problem remains for all UEs that are compatible with both GSM and CDMA networks (include a vast majority of UEs on the market). Details about how the emergency call works for CDMA are out of the scope of this paper.

Thanks to the AOSP project, for android UEs, we can investigate the source code to cross-validate the correctness of our findings from measurement and formal verification. The same methodology does not apply to Apple iPhones.

Ethics Concerns. Our work does not present ethical issues as we handle neither personal data nor human subjects. We run attack experiments in a responsive and controlled manner. All UEs and SIMs are under our control. Only UEs with our customized SIMs inserted can attach to our station.

9 RELATED WORK

9.1 Formal Methods on Cellular Networks

Various formal verification techniques have been applied to security research on cellular network protocols and systems. We classify them into three categories.

Model Checking verifies correctness properties by exhaustively traversing the state space. Several previous works [32, 59, 60] have examined the security issues in 4G protocols with modern model checkers. Tu *et al.* [59, 60] focused on the reliability problems in protocol interactions. Random sampling was performed over all scenarios to cover a full permutation of usage scenarios in interaction space. Hussain *et al.* [32] exploited vulnerabilities in the NAS procedures by abstracting and modeling NAS protocols. Their framework, *LTEInspector*, does not cover the emergency call systems with proper modeling granularity, and thus cannot find failures and attacks reported by this paper. Both of them rely on manual model construction, using lots of standard documents as references.

The whole state space may be prohibitively large, especially for those systems involving cryptographic algorithms. The *Symbolic Analysis* employs predefined reduction rules to save efforts in verification. A lot of works [1, 12, 14, 20] applied modern symbolic provers, like ProVerif [15] and Tamarin [13], on AKA protocols used in 3G, 4G, and 5G. Nevertheless, cryptography-related procedures constitute only a small portion of cellular network protocols, and these methods cannot be generalized to other procedures.

Software Analysis aims to directly verify the implementations, as that can save time and efforts of building a model manually. For instance, Pi *et al.* [48] extracted binary codes from a Qualcomm baseband and performed static analysis and debugging. Yu *et al.* [63] ran software model checking on open-source cellular protocol emulators. However, one implementation is only a single instance of the protocols, so it can not reflect other implementations. In comparison, our approach is based upon protocols. It targets problems on a higher level and can be adapted to many instances.

9.2 Security of Emergency Call Systems

Emergency call systems have many privileges; they also have large impacts on society. Nevertheless, to the best of our knowledge, there is currently no work that formally analyzes the correctness or finds vulnerabilities of emergency call systems on either their designs or implementations.

Authorities usually make orders and standards to enforce local carriers to provide emergency call services. For example, FCC, the communication authority of the U.S., has issued orders [23, 24] to specify the requirements of wireless 911 calls. Ministry of Industry and Information Technology, the communication authority of China, has also published industry standards [42, 43], requiring the connectivity of emergency calls under the no-SIM condition. These documents, however, are more concentrated on functionalities than the security aspects of the system. Besides, these documents may state at a very high level, becoming ambiguous and incomplete.

Not much research literature focuses on emergency call systems. RFC 5096 [57] summarized the security threats that cellular emergency call systems might encounter in a conceptual manner. However, no concrete attacks or defense approaches are discussed in it. The chance of the DDoS attack on 911 services by leveraging the anonymity privilege has been mentioned in [29, 46]. Based on the estimation in [29], with 6,000 bots, 911 emergency services in a U.S. state can be blocked for a whole day. Rebahi *et al.* [50] proposed an attack in the current 3GPP's scheme that an adversary can impersonate PSAPs.

The wireless emergency alert (WEA) system, also known as the public warning system (PWS) or the earthquake and tsunami warning system (ETWS), broadcasts alert to all UEs in a geographic area. This system is not within our research scope. It is worth to mention that the message authentication of WEA has been discussed for years [2]. However, this feature has not been fully settled in protocols even today, leading to multiple fake alert attacks [32, 38].

10 CONCLUSION

This work concentrates on how to use formal methods on cellular networks. In particular, we systematically explore availability and security pitfalls in cellular emergency call systems. We demonstrate in the paper a novel way of specification, called *seed-assisted specification*, which can be applied to systems described by protocols in general. We emphasize the importance of prior knowledge in building the model, and we explain how it helps determine the critical processes and the granularity of the model. Then we describe how to integrate measurement results with a generalized formal model, such that a variety of scenarios can all be verified on real systems. From formal verification, we find 4 scenarios in China that emergency calls cannot be routed to PSAPs. Meanwhile, we find 2 new attacks in the U.S. that abuse emergency call privileges. We propose a unified solution for carriers. It can address the problems we have discovered and any similar problems we can foresee.

ACKNOWLEDGMENTS

We thank all anonymous reviewers and our shepherd for their insightful feedback. We thank all anonymous carriers for their support for this research project.

REFERENCES

[1] 3GPP. 2001. *Formal Analysis of the 3G Authentication Protocol.* Technical Report (TR) 33.902. 3rd Generation Partnership Project (3GPP). Version 4.0.0.

[2] 3GPP. 2019. *Public Warning System (PWS) requirements.* Technical Specification (TS) 22.268. 3rd Generation Partnership Project (3GPP). Version 16.3.0.

[3] 3GPP. 2020. *Access to the 3GPP Evolved Packet Core (EPC) via non-3GPP access networks; Stage 3.* Technical Specification (TS) 24.302. 3rd Generation Partnership Project (3GPP). Version 16.4.0.

[4] 3GPP. 2020. *Characteristics of the Universal Subscriber Identity Module (USIM) application.* Technical Specification (TS) 31.102. 3rd Generation Partnership Project (3GPP). Version 16.4.0.

[5] 3GPP. 2020. *Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification.* Technical Specification (TS) 36.331. 3rd Generation Partnership Project (3GPP). Version 16.1.1.

[6] 3GPP. 2020. *IP Multimedia Subsystem (IMS) emergency sessions.* Technical Specification (TS) 23.167. 3rd Generation Partnership Project (3GPP). Version 16.2.0.

[7] 3GPP. 2020. *Mobile radio interface Layer 3 specification; Core network protocols; Stage 3.* Technical Specification (TS) 24.008. 3rd Generation Partnership Project (3GPP). Version 16.5.0.

[8] 3GPP. 2020. *Non-Access-Stratum (NAS) protocol for 5G System (5GS); Stage 3.* Technical Specification (TS) 24.501. 3rd Generation Partnership Project (3GPP). Version 16.5.1.

[9] 3GPP. 2020. *Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3.* Technical Specification (TS) 24.301. 3rd Generation Partnership Project (3GPP). Version 16.5.1.

[10] 3GPP. 2020. *Service aspects; Service principles.* Technical Specification (TS) 22.101. 3rd Generation Partnership Project (3GPP). Version 17.2.0.

[11] 3GPP2. 2008. 3rd Generation Partnership Project 2. http://www.3gpp2.org.

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

[13] David Basin, Cas Cremers, Jannik Dreier, and Ralf Sasse. 2017. Symbolically analyzing security protocols using tamarin. *ACM SIGLOG News* 4, 4 (2017), 19–30.

[14] David Basin, Jannik Dreier, Lucca Hirschi, Saša Radomirovic, Ralf Sasse, and Vincent Stettler. 2018. A formal analysis of 5G authentication. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 1383–1396.

[15] Bruno Blanchet. 2016. Modeling and verifying security protocols with the applied pi calculus and ProVerif. *Foundations and Trends in Privacy and Security* 1, 1-2 (2016), 1–135.

[16] Elgin M Brunner and Manuel Suter. 2008. *International CIIP handbook 2008/2009: An inventory of 25 national and 7 international critical information infrastructure protection policies.* Center for Security Studies (CSS), ETH Zurich.

[17] Cisco. 2016. MME Administration Guide, Emergency Bearer Services. https://www.cisco.com/c/en/us/td/docs/wireless/asr_5000/21-5_N5-8/MME/21-5-MME-Admin/21-5-MME-Admin_chapter_010010.html. Accessed on Jan. 5, 2021.

[18] Cisco. 2016. MME Administration Guide, Local Emergency Numbers List. https://www.cisco.com/c/en/us/td/docs/wireless/asr_5000/21/MME/b_21_MME_Admin/b_21_MME_Admin_chapter_011111.pdf. Accessed on Jan. 5, 2021.

[19] Piergiuseppe Bettassa Copet, Guido Marchetto, Riccardo Sisto, and Luciana Costa. 2017. Formal verification of LTE-UMTS and LTE–LTE handover procedures. *Computer Standards & Interfaces* 50 (2017), 92–106.

[20] Cas Cremers and Martin Dehnel-Wild. 2019. Component-Based Formal Analysis of 5G-AKA: Channel Assumptions and Session Confusion. In *Proceedings of the 26th Network and Distributed Systems Security (NDSS) Symposium.*

[21] Haotian Deng, Weicheng Wang, and Chunyi Peng. 2018. CEIVE: Combating Caller ID Spoofing on 4G Mobile Phones Via Callee-Only Inference and Verification. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom).*

[22] European Central Bank. 2004. *Standards for Securities Clearing and Settlement in the European Union.* Technical Report. CESR.

[23] Federal Communications Commission, USA. 2002. Revision of the Commission's Rules to Ensure Compatibility with Enhanced 911 Emergency Calling Ssystems. 17 FCC Rcd 19012 (25).

[24] Federal Communications Commission, USA. 2015. Wireless E911 Location Accuracy Requirements. 30 FCC Rcd 2990 (4).

[25] Federal Communications Commission, USA. 2019. 800 MHz Cellular Service. https://www.fcc.gov/wireless/bureau-divisions/mobility-division/800-mhz-cellular-service. Accessed on Jan. 5, 2021.

[26] Federal Communications Commission, USA. 2020. Caller ID Spoofing. https://www.fcc.gov/consumers/guides/spoofing-and-caller-id. Accessed on Jan. 5, 2021.

[27] Nico Golde, Kévin Redon, and Jean-Pierre Seifert. 2013. Let me answer that for you: Exploiting broadcast information in cellular networks. In *Presented as part of the 22nd USENIX Security Symposium (USENIX Security)*. 33–48.

[28] Google. 2020. Android Open Source Project. https://source.android.com. Accessed on Jan. 5, 2021.

[29] Mordechai Guri, Yisroel Mirsky, and Yuval Elovici. 2017. 9-1-1 DDoS: attacks, analysis and mitigation. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 218–232.

[30] Lin Huang. 2016. LTE REDIRECTION: Forcing Targeted LTE Cellphone into Unsafe Network. In *the 7th Annual HITB Security Conference (HITBSecConf)*. https://goo.gl/Y2FtG4

[31] Huawei. 2019. CloudEC V600R019C00 Feature Guide, Emergency Call. https://support.huawei.com/enterprise/en/doc/EDOC1100059085/e0804ebc/emergency-call. Accessed on Jan. 5, 2021.

[32] Syed Hussain, Omar Chowdhury, Shagufta Mehnaz, and Elisa Bertino. 2018. LTEInspector: A systematic approach for adversarial testing of 4G LTE. In *Proceedings of the 25th Network and Distributed Systems Security (NDSS) Symposium.*

[33] Syed Rafiul Hussain, Mitziu Echeverria, Omar Chowdhury, Ninghui Li, and Elisa Bertino. 2019. Privacy Attacks to the 4G and 5G Cellular Paging Protocols Using Side Channel Information. In *Proceedings of the 26th Network and Distributed Systems Security (NDSS) Symposium.*

[34] Syed Rafiul Hussain, Mitziu Echeverria, Imtiaz Karim, Omar Chowdhury, and Elisa Bertino. 2019. 5GReasoner: A Property-Directed Security and Privacy Analysis Framework for 5G Cellular Network Protocol. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 669–684.

[35] ITU-T. 1998. *ISDN user-network interface layer 3 specification for basic call control.* ITU-T Recommendation. International Telecommunication Union. Q.931.

[36] Kakaku BBS. 2017. I cannot make emergency calls such as 110 (in Japanese). https://bbs.kakaku.com/bbs/J0000024343/SortID=21105988/. Accessed on Jan. 5, 2021.

[37] Leslie Lamport. 2002. *Specifying systems: the TLA+ language and tools for hardware and software engineers.* Addison-Wesley Longman Publishing Co., Inc.

[38] Gyuhong Lee, Jihoon Lee, Jinsung Lee, Youngbin Im, Max Hollingsworth, Eric Wustrow, Dirk Grunwald, and Sangtae Ha. 2019. This is Your President Speaking: Spoofing Alerts in 4G LTE Networks. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*. ACM, 404–416.

[39] Chi-Yu Li, Guan-Hua Tu, Chunyi Peng, Zengwen Yuan, Yuanjie Li, Songwu Lu, and Xinbing Wang. 2015. Insecurity of voice solution VoLTE in LTE mobile networks. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 316–327.

[40] Yuanjie Li, Chunyi Peng, Zengwen Yuan, Jiayao Li, Haotian Deng, and Tao Wang. 2016. Mobileinsight: Extracting and analyzing cellular network information on smartphones. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom)*. 202–215.

[41] MediaTek. 2014. MTK Catcher. https://www.finetopix.com/showthread.php/40844-MTK-Catcher. Accessed on Jan. 5, 2021.

[42] Ministry of Industry and Information Technology, China. 2005. Technical requirement of routing and implementation for inter-network emergency call service. YD/T 1406-2005.

[43] Ministry of Industry and Information Technology, China. 2011. Technical requirement and testing methods for general function of mobile telecommunication terminal. YD/T 2307-2011.

[44] Nation Emergency Number Association. 2018. 9-1-1 Statistics. https://www.nena.org/page/911Statistics. Accessed on Jan. 5, 2021.

[45] National Instruments, Ettus Research. 2020. Universal Software Radio Peripheral (USRP) B210 SDR Kit (70 MHz - 6GHz). https://www.ettus.com/all-products/UB210-KIT/.

[46] Andreea Ancuta Onofrei, Yacine Rebahi, and Thomas Magedanz. 2010. Preventing Distributed Denial-of-Service Attacks on the IMS Emergency Services Support through Adaptive Firewall Pinholing. *The International Journal of Next Generation Network (IJNGN)* 2, 1 (2010).

[47] OpenAirInterface Software Alliance. 2020. Openairinterface 5G Wireless Implementation. https://www.openairinterface.org/.

[48] Peter Pi, XiLing Gong, and Gmxp. 2018. Exploring Qualcomm Baseband via ModKit. In *CanSecWest conference.*

[49] Qualcomm. 2020. QUALCOMM eXtensible Diagnostic Monitor (QxDM). https://www.qualcomm.com/documents/qxdm-professional-qualcomm-extensible-diagnostic-monitor.

[50] Yacine Rebahi, Andreea Ancuta Onofrei, and Thomas Magedanz. 2009. Security in the Emergency Services Support for the IP Multimedia Subsystem (IMS). *5th International Week on Management of Networks and Services, Venice, Italy* (2009).

[51] Red Pocket Mobile. 2020. Red Pocket Global Internet Data Plans. https://www.redpocket.com/global.

[52] David Rupprecht, Katharina Kohls, Thorsten Holz, and Christina Pöpper. 2019. Breaking LTE on layer two. In *Proceedings of the 2019 IEEE Symposium on Security & Privacy (S & P).*

[53] Takuro Sato, Daniel M Kammen, Bin Duan, Martin Macuha, Zhenyu Zhou, Jun Wu, Muhammad Tariq, and Solomon Abebe Asfaw. 2015. *Smart grid standards:*

specifications, requirements, and technologies. John Wiley & Sons.
[54] Altaf Shaik, Ravishankar Borgaonkar, N Asokan, Valtteri Niemi, and Jean-Pierre Seifert. 2016. Practical attacks against privacy and availability in 4G/LTE mobile communication systems. In *Proceedings of the 23rd Annual Network and Distributed System Security (NDSS) Symposium*.
[55] Jaeseung Song, Hyoungshick Kim, and Athanasios Gkelias. 2014. iVisher: Real-Time Detection of Caller ID Spoofing. *ETRI Journal* 36, 5 (2014), 865–875.
[56] Sprint. 2019. What this means to you after April 30, 2019. https://www.sprint.com/en/support/account/oma-slot.html. Accessed on Jan. 5, 2021.
[57] T. Taylor, H. Tschofenig, H. Schulzrinne, and M. Shanmugam. 2008. *Security Threats and Requirements for Emergency Call Marking and Mapping*. RFC 5069. IETF.
[58] THE PAPER. 2019. Father fell to the ground with cerebral haemorrhage, mother's mobile phone can not make emergency calls. Meizu said: possible a system problem (in Chinese). https://www.thepaper.cn/newsDetail_forward_3749664. Accessed on Jan. 5, 2021.
[59] Guan-Hua Tu, Yuanjie Li, Chunyi Peng, Chi-Yu Li, and Songwu Lu. 2016. Detecting problematic control-plane protocol interactions in mobile networks. *IEEE/ACM Transactions on Networking* 24, 2 (2016), 1209–1222.
[60] Guan-Hua Tu, Yuanjie Li, Chunyi Peng, Chi-Yu Li, Hongyi Wang, and Songwu Lu. 2014. Control-plane protocol interactions in cellular networks. In *Proceedings of the 2014 ACM Conference on SIGCOMM*. ACM, 223–234.
[61] Verizon Wireless. 2019. CDMA Network Retirement. https://www.verizonwireless.com/support/knowledge-base-218813/. Accessed on Jan. 5, 2021.
[62] Bob Williams. 2008. *Intelligent transport systems standards*. Artech House.
[63] Yinbo Yu, You Li, Kaiyu Hou, Yan Chen, Hai Zhou, and Jianfeng Yang. 2019. CellScope: Automatically Specifying and Verifying Cellular Network Protocols. In *Proceedings of the ACM SIGCOMM 2019 Conference Posters and Demos*. ACM, 21–23.
[64] Yuan Yu, Panagiotis Manolios, and Leslie Lamport. 1999. Model checking TLA+ specifications. In *Advanced Research Working Conference on Correct Hardware Design and Verification Methods*. Springer, 54–66.