5G SUCI Catcher: Attack and Detection

Lorens Barraud
Thales SIX GTS, France
lorens.barraud@thalesgroup.com

Francesco Caccavale
Montimage, France
francesco.caccavale@montimage.com

Jean-Baptiste Peyrat
Thales SIX GTS, France

Wissam Malouli
Montimage, France
wissam.mallouli@montimage.com

Véronique Capdevielle
Thales SIX GTS, France
veronique.capdevielle@thalesgroup.com

Hicham Khalife
Ericsson, France
hicham.khalife@ericsson.com

Ana Rosa Cavalli
Montimage, France
ana.cavalli@montimage.com

***Abstract*— The deployment of 5G networks opens up new possibilities for communication and connectivity. However, it also introduces new security threats. This paper explores one cyber threat: 5G SUCI Catcher attack. The 5G SUCI Catcher attack is a proof of concept involving monitoring from nearby mobile devices in the 5G paradigm. SUCI Catchers act as fake base stations by exploiting weaknesses of the 5G authentication and encryption protocol. In this paper, SUCI Catcher attack and detection rules are implemented in a 5G experimental environment. The detection solution demonstrates practically the capability to efficiently mitigate the risks associated with this 5G attack.**

Keywords—5G, SUCI Catcher, detection, MitM

I. Introduction

The advent of 5G technology brings unprecedented speed, connectivity and technological advancements. However, it also poses some risks in terms of security. Cyber-attacks are getting more sophisticated with evolving technology and 5G is no exception. In this paper, we are interested in the development of a SUCI Catcher which is a tool that will allow us the monitoring of nearby devices. The purpose of this kind of tool is similar to an IMSI Catchers for 4G networks which are surveillance equipment's that mimic base stations operation in order to intercept and monitor communications from nearby devices.

IMSI-Catchers act as Fake Base Station, as MitM-attacker (Man in the Middle) to intercept the user's permanent identity IMSI (International Mobile Subscriber Identity). They deceive nearby mobile devices that they are connecting to a legitimate base station. The implications of IMSI Catchers are far-reaching. They can be used for various purposes including law enforcement agencies or malicious hackers.

The "SUCI-Catcher" will exploits weaknesses of the 5G Authentication procedure from the network side to intercept nearby devices.

The contribution of this paper is to demonstrate the feasibility of generating and detecting such attack. The paper is organized as follows: a first section will cover the description of the SUCI-Catcher attack, then we will present the implementation of such attack in our local environment and the result we had to then describe the detection methodology. Finally, the last section will conclude on on the performance of the detection.

Attention: the proof of concept developed in this paper is far from what an actual "SUCI Catcher" should correspond. To simplify the implementation in our testbed we have virtualized each significant part of the 5G network from the User Equipment (UE) to the core with the use of opensource software as explained in part III. C. This mean in the testbed, MitM-attacker is placed on a network interface thus directly exploiting the NGAP layer and not the radio layer which should be the case with tools such as these. The sole goal of this paper is to show how a potential "SUCI Catcher" could algorithmically speaking work taking in account the new security 5G measures.

II. Description of SUCI Catcher

In 4G, commercial IMSI-Catchers act as a fake base station by copying the identity of the real network and actively request the user's permanent identity (IMSI). Any user within range eventually connects to the IMSI-Catcher and thus unwillingly exposes his or her identity. This attack as it stands in 4G is no longer feasible in 5G SA due to the encryption of the user's identity (SUCI).

SUCI Catcher attack then builds upon weaknesses of the AKA procedure (Authentication and Key Agreement) where pre-authentication messages are not protected (with use of cryptographic keys).

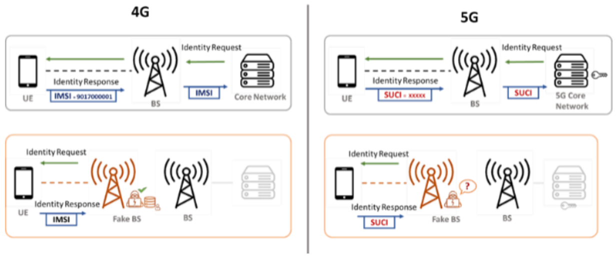


Figure 1: 5G Countermeasure against IMSI Catcher

At the beginning of the procedure the user establishes the connection with an initial Registration Request message containing the user's identity. The network proceeds with an Authentication Request and the UE will accept this request if the authentication code check and sequence number check is verified. The SUCI-Catcher attack exploits this.

The attack consists of two main phases: the Discovery Phase and the Attack Phase.

In the **Discovery Phase**, the attacker aims to gather the identifier(s) of the person(s) of interest (PoI) through any available means. This could involve reconnaissance techniques to gather information about potential targets.

The **Attack Phase** consists of two steps: the *Probing Phase* and the *Reset & Sync*. Phase. In the Probing Phase, the attacker generates SUCIs from a list of SUPIs (Subscription Permanent Identifiers). The attacker sends registration requests containing the generated SUCIs to the core network and relays them to all connecting UEs. Only the UE that successfully accepts the request indicates the presence of the targeted subscriber. In the Reset & Sync. Phase, the attacker sends a registration request with the SUCI of the UE the first time the UE connected to the network. This allows the attacker to continue the Probing Phase and refine the identification process.

The ultimate target of the SUCI Catcher attack is to determine if a specific known subscriber is present in the proximity of the SUCI Catcher, bypassing the encryption of the permanent identity in 5G-SA (Standalone) networks.

III. Implementation of the SUCI Catcher attack

A. *Description*

The setup of this attack was based on the paper [1]. This paper investigates to which extent the new user's identity encryption scheme keeps its privacy promises in practice. They built upon weaknesses in the AKA procedure (authentication and key agreement) that enable user link ability. They extent the existing weakness to the 5G SUCI scheme and conceptualize a SUCI-Catcher attack. As a result, the SUCI-Catcher can verify if a specific, known subscriber is present in proximity of the SUCI-Catcher, despite the encryption of the permanent identity in 5G-SA networks. Thus, the attack is more about IMSI-probing, but they keep the term "Catcher" to not hinder the general discussion on practical and effective surveillance.

1) *The user's identity in 5G SA network*

In 5G, the permanent SUPI identity (subscription permanent identity) - formerly known as IMSI - is encrypted using the public key of the operator before transmission. The encrypted SUPI is called SUCI (subscription concealed identity).

Only the operator - but no attacker - can read the user's identity. The SUCI is re-generated before every usage to prevent linking of SUCIs such that an observer cannot distinguish if the same user connects twice, or if this represents two distinct users. Yet, all the SUCIs generated from the same SUPI remain valid as they are equally processed by the network with no guarantee of freshness or authenticity.

SUPI concealing is an optional feature, configurable by operators [2]. Without SUCI encryption, the permanent identity is directly transmitted with the so-called null scheme (5G-EA0), which offers no protection – which is tantamount to IMSI in terms of security. In our testbed we are using the null scheme in the free5GC core network which is defined like that by default and making easier to exploit the NGAP protocol.

2) *The Authentication and Key Agreement (AKA) procedure*

AKA involves mutual authentication between User Device and the network and derive cryptographic keys to protect the User plane and Control plane data. As a result, User Equipment and network can only activate message encryption after the AKA procedure is performed. Pre-authenticated messages are thus unprotected.

3) *Authentication and Key Agreement link ability*

At the beginning of the procedure the user establishes the connection with an initial Registration Request message containing the user's identity. The network proceeds with an Authentication Request and the UE will accept this request if the authentication code check and sequence number check is verified. The SUCI-Catcher attack exploits this: it fetches an authentication challenge associated with the searched-for subscriber's identity (❶ in Figure 2). In this manner the authentication request is tailored to this subscriber (for X in) and it's valid since it is coming from the Core Network. Then, it sends this Authentication Request to all connecting UEs (❷ in Figure 2). Only the UE that accepts the request is the wanted subscriber.

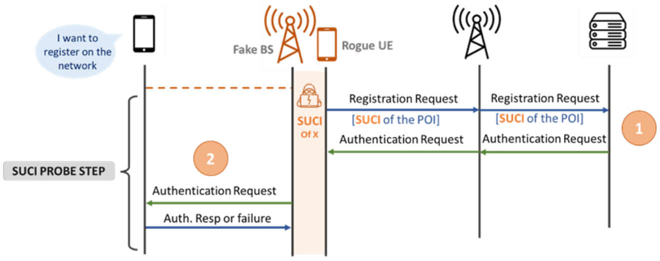


Figure 2. SUCI-Probe Step: Is the user that who wants to connect the person we are looking for? POI: person of interest (the user we are looking for)

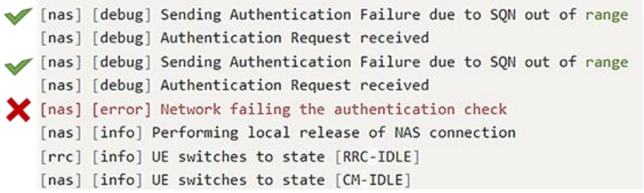
The attack is divided into two phases:

First, a discovery phase (some methods are described in the paper) identifies subscribers of interest (X in Figure 2) and associated SUCIs (SUCI in Figure 2). At this stage we know the identity of the subscribers behind the collected SUCIs. Second, when an unknown UE connects, the SUCI-Catcher can use these previously captured SUCIs – since they remain valid - to confirm whether the unknown UE belongs to the searched-for subscriber. This phase is called SUCI-Probe. As an example, in the above figure, the attacker wants to know if the UE that intends to register on the network is X – the person of interest.

Note: As already explained previously SUPI concealing is an optional feature, configurable by operators. This advantage is used for the testbed (More detail below in part: Testbed) as the concealing of the SUPIs is not being done automatically in the implemented core network. The idea is that we will collect the SUPIs of UEs trying to register to the core network and then use them to generate SUCIs.

4) Scalability: searching multiple subscribers

IMSI-Catchers scale well since each connecting UE only requires a single message to determine the identity, which becomes impossible with SUCI encryption. The basic SUCI-Probe supports testing for a single identity stage since the user **gets disconnected from the radio layer** in case of too many consecutive authentication failures as shown on the wireshark logs below:



```
✓ [nas] [debug] Sending Authentication Failure due to SQN out of range
  [nas] [debug] Authentication Request received
✓ [nas] [debug] Sending Authentication Failure due to SQN out of range
  [nas] [debug] Authentication Request received
✗ [nas] [error] Network failing the authentication check
  [nas] [info] Performing local release of NAS connection
  [rrc] [info] UE switches to state [RRC-IDLE]
  [nas] [info] UE switches to state [CM-IDLE]
```

Figure 3. UE logs from testbed

In the reference paper [1], they extended the scheme with an additional reset stage – called RESET&SYNC - that allows scaling the SUCI-Catcher attack and search for multiple subscribers among connecting UEs **without disconnecting them from the radio layer**: each smartphone entering the cell is tested for a series of subscriber identities.

The RESET&SYNC step performs a successful AKA between UE and network before the actual SUCI-Probe.

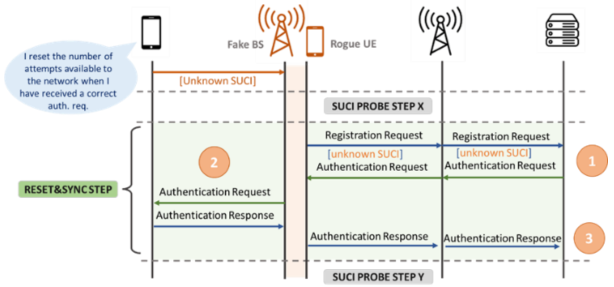
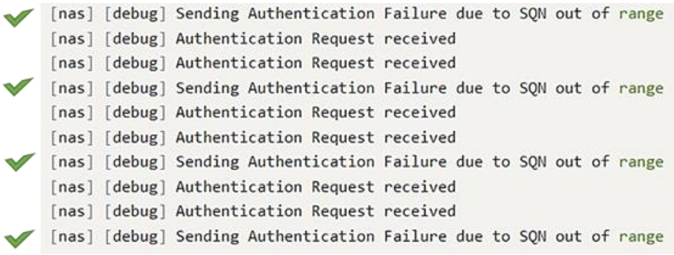


Figure 4. RESET&SYNC Step



```
✓ [nas] [debug] Sending Authentication Failure due to SQN out of range
  [nas] [debug] Authentication Request received
  [nas] [debug] Authentication Request received
✓ [nas] [debug] Sending Authentication Failure due to SQN out of range
  [nas] [debug] Authentication Request received
  [nas] [debug] Authentication Request received
✓ [nas] [debug] Sending Authentication Failure due to SQN out of range
  [nas] [debug] Authentication Request received
  [nas] [debug] Authentication Request received
✓ [nas] [debug] Sending Authentication Failure due to SQN out of range
```

Figure 5. UE logs with RESET&SYNC step.

B. Analysis of SUCI CATCHER attack

The attack aligns with the relevant tactics of Reconnaissance. The techniques used include Active Scanning and Gather Victim Identity Information, with the sub-technique of Determining Physical Location. Detecting the SUCI Catcher attack can involve employing network intrusion prevention measures to detect suspicious activities and mitigate the attack. Mitigation techniques include Adversary in the Middle, Data Encoding, and using Encrypted Channels to enhance security and protect against such attacks. The attack is associated with several CAPEC identifiers, including Interception, Sniffing Attacks, and Eavesdropping, highlighting the different aspects of information gathering and unauthorized access.

The section provides a detailed analysis of this attack, including tactics, techniques and sub-techniques based on MITRE Fight.

Moreover, the relevant CAPEC codes and potential detection and mitigation activities are provided. Finally, based on the above information, the CVSS score is calculated.

Table 1: CAPEC codes from MITRE database

Relevant Tactics	TA0043 – Reconnaissance
Sub-Techniques	T1591.001 – Determine Physical Location
Detection	M1031 – Network Intrusion Prevention
Mitigation	T1557 – Adversary in the Middle T1132 – Data Encoding T1573 – Encrypted Channel
CAPEC	CAPEC-117: Interception CAPEC-157: Sniffing Attacks CAPEC-651: Eavesdropping
CVSS Score	5.5

C. Implementation in the testbed

In the testbed, MitM-attacker is placed on the interface between the 5G gNB and the AMF [2] supporting NGAP protocol. Since it only concerns pre-authentication NAS-layer messaging where both NGAP and radio-layer are merely the transport, it does not affect the experiment.

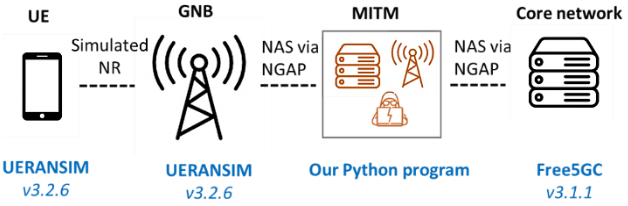


Figure 6: Testbed

In our testbed, 5 different UEs are connected to a gNB. For the simulation of the UEs and the gNB we used UERANSIM. The gNB is connected to another machine where the MITM program is running. All the traffic passing through the MITM is transferred to the free5gc core network and vice versa.

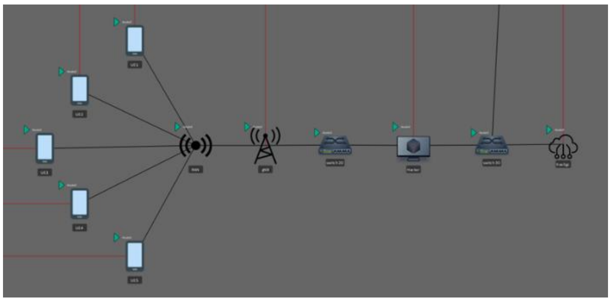


Figure 7. SUCI-Catcher in testbed

The MitM machine contains a docker image that will allow us to run two different scripts:

- A first script, "suci-catcher.py", which will listen passively the network on the NGAP layer that will simply collect SUPIs and SUCIs when a UE will try to connect register on to the core network. The collected SUPIs and SUCIs will be stored in a file that can be accessed through a docker mounting point.
- A second script, "suci-probe.py", which will launch the SUCI-Catcher attack. The script waits for a UE to fully register to the core network then start the probing attack by generating different SUCIs from the collected SUPIs.

IV. Detection of SUCI Catcher attack

The detection of SUCI catcher attack can rely either on a rule-based detection where an alarm is triggered when the number of registration requests exceeds a pre-configurable threshold or by applying a more advanced ML/AI-based detection where we first learn in a specific environment the normal usage of the network and detect drifts.

Both methodologies can be implemented by using the open source monitoring solution of Montimage called MMT[1]. The architecture of the tool [3] is presented in the Figure 8.

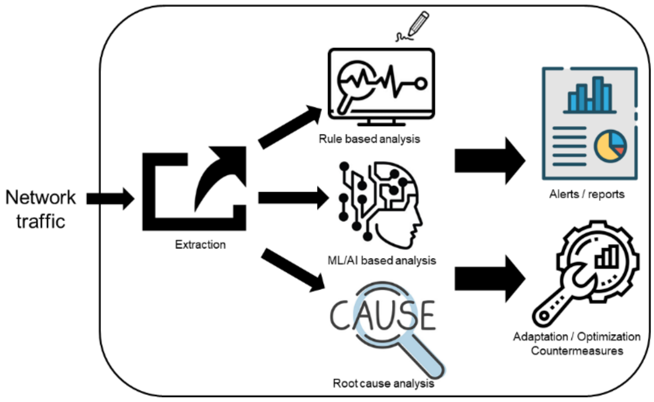


Figure 8. SUCI-Catcher in testbed

The tool is composed by a set of modules that are presented in the following subsections.

A. Feature Extraction library

MMT-Extract is a C library that analyses network traffic and application logs, in order to identify network and application-based events, such as protocol field values. Furthermore, the library uses Deep Packet Inspection (DPI) techniques to examine the payload of a large number of application protocols.

MMT-Extract permits parsing an extensive variety of network protocols (e.g., TCP, UDP, ARP, HTTP, etc.), and is able to calculate some performance indicators. The extraction is enabled by a plugin architecture that admits the addition of new network protocols or application messages to be processed. MMT-Extract uses DPI techniques for application identification and classification. This is crucial when examining applications that do not have a standard port number (e.g., P2P, Skype).

MMT has been extended in the context of H2020 SANCUS project [4] to be able to parse 5G protocols like SCTP, NAS-5G and NGAP and several analysis rules has been defined to build MMT-5G as a monitoring solution for 5G network.

B. Rule based detection

MMT-Security is a monitoring tool, that could perform as a HIDS (Host-based Intrusion Detection System), and be installed in a host, or as a NIDS (Network-based Intrusion Detection System) and be used for network inspection. In the case of SANCUS project, the library is considered to be used as a NIDS, that enables the inspection of network traffic according to a set of security properties denoted as MMT-Security properties. The main goal of these properties is to formally state security objectives to be achieved or malicious activity to be avoided, in relation to the application or protocol that is under monitoring.

MMT-Security properties are written in XML language, due to its simplicity and straightforward structure verification. Each property begins with a <property> tag and ends with </property>. A property is a "general ordered tree" as shown in Figure 9, where there are property nodes that are compulsory, operator nodes that are non-compulsory, and event nodes that are compulsory. The property is necessarily the root node, while the event nodes must be leaf nodes. Each property is constituted by a context in the left branch, and a trigger in the right branch. A property is valid when the trigger is valid, then the trigger is verified only if the context is valid.

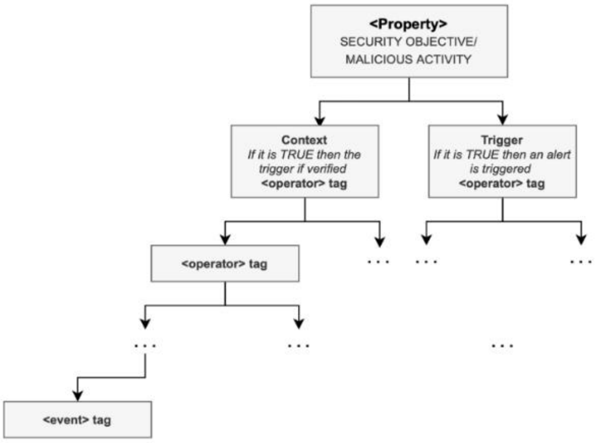

<Property>
SECURITY OBJECTIVE/ MALICIOUS ACTIVITY
Context
If it is TRUE then the trigger if verified
<operator> tag
Trigger
If it is TRUE then an alert is triggered
<operator> tag
<operator> tag
<event> tag

Figure 9. Tree representation of an MMT detection rule

The rule specified for SUCI catcher attack relies on only one event that is based on detecting the number of registration requests. An alarm is triggered when this number exceeds a configurable threshold.

C. ML/AI based analysis

MMT-ML is a classical anomaly-based Intrusion Detection Systems (AIDS) that works by comparing the actual comportment of the network with a previously-established "normal" model of its behavior. Any substantial deviance between the observed behavior and the model is considered as an anomaly, which can be translated as an intrusion or attack

[1] https://github.com/Montimage/mmt-probe

into the system. MMT-ML has drawn interest from a lot of scholars due to its capacity to overcome the limitation of the rule-based intrusion detection systems. Several ML/AI algorithms are available in the tool, like Stacked Auto Encoders (SAE) [5] and Convolutional Neural Network (CNN) [6].

D. *Root-cause analysis*

MMT-RCA relies on machine learning algorithms to identify the most probable cause(s) of detected anomalies based on the knowledge of similar observed ones. It enables systematizing the experience in dealing with incidents to build a historical database and verify whether a newly detected incident is similar enough to an observed one with known causes. Thanks to MMT-RCA's suggestions, remediation actions could be timely and wisely taken to prevent or mitigate the damage of the recurrence of problems.

E. *Adaptations and countermeasures*

The detected anomalies and attacks can be linked to countermeasures and adaptations that allow to mitigate the risk and increase the network resilience. In the context of SUCI catcher attack a simple blocking of the source of the requests can be performed.

F. *Reports and dashboard*

The network statistics as well as the attacks/anomalies detection are reported to the network operator in order to monitor the network status. MMT-Operator dashboard collects and aggregates extracted data, generates network and application statistics, and presents them via a graphical user interface. MMT-Operator is customizable; the user can define new statistics to be collected and configure new views or customize a large list of predefined ones.

V. Performances of SUCI CATCHER detection

The aim of this section is to assess the detection capabilities of the SUCI CATCHER attack using MMT-5G by relying on 5 main metrics.

- The detection delay: It is the difference of time between the attack generation time and the alert generation.
- The false positive rate: it is the percentage of false alerts generated by the detection tool.
- The false negative rate: it is the percentage of non-detected attacks by the detection tool.
- CPU usage : an average of CPU usage for the 12th Generation of Intel® Core™ i9-12900 HK
- Memory usage : an average of Memory usage for 31,7 Go available.

Two techniques has been used in MMT-5G: "rule-based detection" and "AI-based detection" and to ensure that the reported results are sound, we simulated normal behavior and SUCI CATCHER attacks in a random way during 24 hours. The attack has been generated 100 times for a duration that varies between 10 seconds and 30 seconds.

The results are reported in the following table 2 by aggregating all the results and presenting an average.

	Rule-based detection	**AI-based detection**
Detection delay	259 ms	718 ms
False positive rate	1 %	2 %
False negative rate	7 %	3 %
CPU usage	47 %	49%
Memory usage	56 % (/31,7 Go)	58 % (/31,7 Go)

Several conclusions can be drawn by relying on these results.

- The detection delays are less than 1 second. The only difficulty for AI-based detection is to have a learning phase with several datasets (in our case 2, nominal traffic and SUCI Catcher traffic).
- The false positive and false negative rates are comparable. This is because it is very different to set a good threshold for the rule based detection but even if other features are used for the CNN model, the detection efficiency is not optimal.
- Resources consumption: The two detection techniques are comparable in terms of CPU/Memory usage. This is because only one detection rule is applied. The AI model is simple enough to not consume a lot of resources. With more attack this can different.

VI. Conclusion

In conclusion, this paper has investigated into the emerging security threat posed by the 5G SUCI Catcher attack and has provided a comprehensive exploration of its attack mechanisms, implementation, and detection methods. The SUCI Catcher attack, an evolution of the IMSI Catcher, exploits vulnerabilities in 5G's authentication and encryption protocols to intercept user identities and harm their privacy.

The paper has not only described the attack in detail but has also implemented it in a controlled experimental environment. By doing so, the authors have demonstrated the practical feasibility of generating and detecting this type of attack. Two primary detection methods were discussed: rule-based detection and AI-based detection. Both methods showed promising results with minimal detection delays, manageable false positive and false negative rates, and reasonable resource consumption.

Funding

The research leading to these results has been partially supported by the European Commission under the Horizon 2020 Program, through funding of the SANCUS project (G.A. n 952672).

References

[1] Merlin Chlosta, David Rupprecht, Christina Pöpper, Thorsten Holz , "5G SUCI-Catchers: Still catching them all?", Proceeding of the 14th ACM Conference on security and privacy in Wireless and Mobile networks.

[2] 3GPP. 2018. Security architecture and procedures for 5G System. TS 33.501. http://www.3gpp.org/ftp/Specs/html-info/33501.htm

[3] Wissam Mallouli, Bachar Wehbi, Edgardo Montes de Oca, Michel Bourdelles, Online Network Traffic Security Inspection Using MMT Tool. In the 9th workshop on system testing and validation (STV). Paris, France, October 2012.

[4] H2020 SANCUS project : https://sancus-project.eu/ accessed on October 7th, 2023.

[5] Silhan, T., Oehmcke, S., Kramer, O. “Evolution of stacked autoencoders.” In the IEEE Congress on Evolutionary Computation, CEC 2019, Wellington, New Zealand, June 10-13, 2019. pp. 823–830. IEEE (2019). https://doi.org/10.1109/CEC.2019.8790182

[6] O’Shea, K., Nash, R. “An introduction to convolutional neural networks.” CoRR abs/1511.08458 (2015)