Cryptographic Protocol Analysis and Verification

**The University of Texas at Dallas**

**Team Cookie Army:**

**Juhi Patel** - [jmp170130@utdallas.edu](mailto:jmp170130@utdallas.edu)

**Kunal Mangalorekar -** [kgm190005@utdallas.edu](mailto:kgm190005@utdallas.edu)

**Lizabeth Trevino -** [lct170130@utdallas.edu](mailto:lct170130@utdallas.edu)

**Nidhi Jawandhia -** [nxj190011@utdallas.edu](mailto:nxj190011@utdallas.edu)

# Table of Contents

[**Table of Contents**](#_edeq6mmlndky) **2**

[**Project Summary**](#_y3kc9qvlj5s3) **3**

[Description](#_swf3n9sg24k9) 3

[Motivation](#_lpwj0cxy64er) 3

[**Organization**](#_xfde47w35gv7) **4**

[Required Skills](#_q8dlaceug6p1) 4

[Total Budget](#_w7w1hjw3xb6a) 4

[Schedule](#_hihtpvh2vlrj) 4

[**Midterm Progress:**](#_s4gpiwsmc3kr) **4**

[**Technical Approach**](#_14sl8ceznc3r) **6**

[**Future Plans**](#_n0xn9pve92i8) **7**

[**Challenges**](#_erfnkdn8it7c) **7**

[**Total Progress:**](#_6ufyengcijfp) **7**

[Work Completed](#_wv9qhdjqbm1n) 7

[Formal Model](#_mlftitm34fmc) 7

[Analysis](#_3yw9n11cj4jy) 7

[Challenges](#_45qcp06a3e8e) 7

[Future Work](#_63tjtngmnd16) 8

[**Literature Review**](#_4pqmjr98tj8z) **8**

[**Biographical Sketches of Investigators**](#_3xt9c453zmhi) **10**

[**Bibliography**](#_vsm4prs8tyx9) **11**

# Project Summary

## Description

Cryptography forms the backbone of our digital society. It provides the basic security attributes of Confidentiality, Integrity, and Authenticity. It is well known that security is always as strong as its weakest link. Failing to secure this weakest link has dire consequences in today's data-driven world. Hands down it is very important that the effectiveness of cryptographic protocol is checked periodically especially in a time when the technology is getting updated at such a great pace.

We are interested in doing this project because not only we’ll be able to learn about various protocols used in real-life applications, but we will also be able to check loopholes if any, and mend them to the best of our abilities. It's a win-win project topic for us, with a lot to explore, learn and implement. Working on this project will give us a platform to practically implement what we have learned and expand our knowledge further.

## Motivation

During recent years there has been considerable interest and growth in computer networks and distributed systems. Computer networks employ encryption for several purposes, including private communication, message authentication, and digital signatures. The correctness and security of these applications depend not only on the strength of the cryptographic algorithms but also on the protocols for key management. With this being said, as with all the other things when a cryptography protocol was designed, it was written as per the resources available at that time. However, it is not necessary that these protocols will be as effective or will be effective at all with all the advancements in technology.

In this project, our team will be first analyzing the protocol and then work on the modification of the protocols so that it provides more security than before. The protocol that we will be analyzing is the 5G EAP-TLS cryptographic protocol.

Authentication and key management are fundamental to the security of cellular networks because they provide mutual authentication between users and the network and derive cryptographic keys to protect both signaling and user plane data. Each generation of cellular networks always defines at least one authentication method. For example, 4G defines 4G EPS-AKA, and 5G defines three authentication methods—5G-AKA, EAP-AKA’, and EAP-TLS. 5G differs from prior generations primarily in that it will not only provide faster speed, higher bandwidth, and lower delays, but also support more use cases such as enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable low-latency communications (uRLLC).

# Organization

## Required Skills

* Basics of Network Security
* Basics of Cryptographic Protocols

## Total Budget

Our team will spend around 8-10 hours a week collectively on this project. This includes the training we will be receiving on the CPSA software every week.

## Schedule

| **Task / Deliverable** | **Date** | **Person In Charge** |
| --- | --- | --- |
| CPSA Training | 7 Weeks Long | Enis/Everyone |
| Project Proposal | February 12th | Everyone |
| Midterm Report | March 25th | Everyone |
| Midterm Presentation | March 25th | Everyone |
| Final Report | May 6th | Everyone |
| Final Presentation | May 6th | Everyone |
| Completed 5g-EAP Model | May 8th | Everyone |

# Midterm Progress:

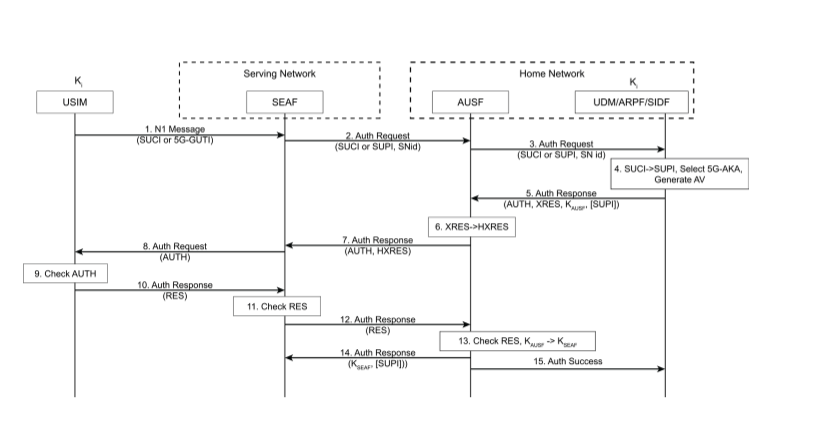
Our team has completed 3 weeks of the CPSA (Cryptographic Protocol Shapes Analyzer) seminar. From this seminar, we have learned the basics of the Dolev-Yao adversary and how it is applied. In addition, we have learned the basic syntax that CPSA uses as well as how to model basic protocols using the tool. Finally, we have also been introduced to the skeletons/shapes that the CPSA software produces. We are currently learning how to identify what the different structures of the shapes/skeleton mean in terms of the security of the cryptographic protocol.

We studied a number of papers and articles over the internet to understand the working of the protocols used in cellular networks over the years. Based on our understanding and readings, we analyzed some issues in the past technologies, like for example: Lack of network authentication in 2G, resulted in attacks such as network spoofing by faked base station, Lack of integrity protection for certain signaling messages, thus allowed signal spoofing and tampering(For example, an Identity Request (a non-access stratum signaling message in Long-Term Evolution , if not protected with authentication and integrity, can be sent by a faked base station to steal UE permanent identifiers, e.g., the international mobile subscriber identity). To help mitigate those issues, the 3GPP in 5G protocol defines an Authentication and Key Agreement (AKA) protocol and procedures that support entity authentication, message integrity, and message confidentiality, among other security properties. The 3GPP AKA protocol is a challenge-and-response authentication protocol based on a symmetric key shared between a subscriber and a home network, cryptographic keying materials are derived to protect subsequent communication between a subscriber and a serving network, including both signaling messages and user plane data (e.g., over radio channel).

Also we came across that a unified authentication framework has been defined to make 5G authentication both open (e.g., with the support of EAP) and access-network agnostic (e.g., supporting both 3GGP access networks and non-3GPP access networks such as Wi-Fi and cable networks.

In particular, EAP-TLS is defined in 5G for subscriber authentication in limited use cases such as private networks and IoT environments. When selected as the authentication method by Unified Data Management, EAP-TLS is performed between the UE and the Authentication Server Function(AUSF) through the Security anchor function(basically a middle man during authentication between UE and home network), which functions as a transparent EAP authenticator by forwarding EAP-TLS messages back and forth between the UE and the AUSF. To accomplish mutual authentication, both the UE and the AUSF can verify each other’s certificate or a pre-shared key (PSK) if it has been established in a prior Transport Layer Security (TLS) handshaking or out of band. At the end of EAP-TLS, an Extended Master Session Key(EMSK) is derived, and the first 256 bits of the EMSK is used as the KAUSF. As in 5G-AKA and EAP-AKA’, the KAUSF is used to derive the KSEAF (anchor key), which is further used to derive other keying materials (see Figure 5) needed to protect communication between the UE and the network. EAP-TLS fundamentally differs from 5G-AKA and EAP-AKA’ in its trust establishment between UE and the network, i.e., it uses a different trust model. In EAP-TLS, mutual authentication between a UE and a 5G network is obtained primarily based on the mutual trust of their public key certificates, acknowledging that TLS with a PSK is possible but is rarely used except for session resumption. In AKA-based methods, such trust is based solely on a symmetric key shared between a UE and the network. Such a fundamental difference is significant in that EAP-TLS removes the need to store a large number of long-term keys in the home network, thus reducing operational risks in the life cycle of symmetric key management. On the other hand, EAP-TLS introduces new overhead in certificate management, such as certificate issuance and revocation.

The following figure explains the 5G Authentication framework-



Briefly we analyzed that, Authentication in cellular networks has evolved over each generation: 5G authentication improves upon 4G authentication in a number of areas, including a unified authentication framework, better UE identity protection, enhanced home-network control, and more key separation in key derivation, however it still has its weakness- user trackability may still be possible in 5G.

# Technical Approach

To analyze the protocol we would be using the Cryptographic Protocol Shapes Analyzer (CPSA) analysis tool to search for vulnerabilities.We have followed the documentation provided to install it on our local systems and set up its requirements. CPSA uses the Dolev-Yao adversarial network model for running executions of the protocol. It works by taking a definition of the protocol and a skeleton for an execution from a user. Once these inputs are pre processed in a proper format, the output produced contains an xhtml page describing the executions and graph visualizations of the messages exchanged. These various possible execution/scenarios of the protocol which are called ‘shapes’ of the protocol. After gaining full knowledge of this tool by completing the training, we attempt to use it to analyze the 5G EAP-TLS protocol stated and document our findings.

# Total Progress:

## Work Completed

Throughout the semester our group has been able to accomplish several tasks related to our research. The first major accomplishment was our completion of the 7 week CPRA training program which required us to model several different cryptographic protocols like Neadham-Shroder, PAKE 0 & 1, Diffie-Helman, and SRP3. With the completion of this course we were able to gain sufficient practical knowledge on modeling protocols which aided in our modeling of the 5G EAP-TLS protocol.

Additionally, we were able to improve on the previous team’s model for the 5G EAP-TLS 1.2 model. While trying to model 1.3 we noticed that there were messages in the handshake that weren’t being modeled in the 1.2 model, which was the basis for the 1.3 model they tried to implement. Upon this revelation we chose to improve the 1.2 model since it would also be the basis for our 1.3 model. Apart from including the messages that were left out in the previous team’s model, we altered how the generation of the session key was being implemented from a concatenation of the client-nonce, the prekey, and the server-nonce to the hash of those values as it seemed a safer interpretation of what was described in the documentation for 5G EAP-TLS 1.2.

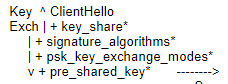
Lastly, we were also able to model and generate shapes for 5G EAP-TLS 1.3. We were able to take our model for 1.2 and use it as a basis for 1.3. At first we tried implementing 1.3 by incorporating the previous teams implementation of the TLS 1.3 messages in the place of the TLS portion of our improved 1.2 model, but we kept running into difficulties with the limit being exceeded and the adversary being able to obtain certain crucial information in those messages. We were able to overcome these difficulties by abstracting certain elements within the handshake messages which allowed for the program to finally generate shapes that showed proper execution of the protocol.

## Formal Model

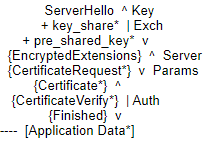
Our final model of 5G-EAP-TLS v1.3 illustrated the conversation between two roles: UE, or the user equipment, and AUSF, the authentication server function. We designed our model in CPSA which tests the security of the protocol using the Dolev-Yao adversary. The Dolev-Yao adversary has the power to overhear, intercept, and concatenate any message that passes through the network. The only thing that is able to stop this adversary is the cryptographic tools used in the protocol.

First, the UE role generates “rue”, a nonce that will be sent along with “SUPI”, the subscription permanent identifier. These two components are encrypted using the public key of AUSF. Once the AUSF role receives this message, it replies with the “TLS-START'' message that will indicate that the two roles will be initiating the protocol.

Once the protocol has been initiated, UE will send a concatenation of “ClientHello”, , and “mue”. This part of the protocol is meant to abstract the client hello section of the TLS 1.3 handshake. In our model, simulates the key that is shared and “mue” is the nonce.



AUSF then receives this message and replies with a concatenation of “ServerHello”, “h2”, and a series of encrypted messages. First, the name “ausf” is encrypted with the private key of the certificate authority which is assumed to be non-originating in the roles of both UE and AUSF. Next, the hash of and “mue” is encrypted using “as” an asymmetric key. Finally, that whole string of messages is encrypted using “”. This message simulates the server hello section of the TLS 1.3 handshake. The first encrypted message is the certificate.The second encrypted message is a hash of the handshake.



Once that is received, the UE then sends “nue” which is encrypted with “”. AUSF receives this message and replies with “nue” and “nausf” encrypted with “”. This simulates the finishing part of the handshake. Now that the handshake is completed, the 5G-EAP-TLS protocol communicates the success of the protocol by sending and receiving confirming messages.

## Analysis

Our CPSA models for v1.2 after implementing the missing messages from the previous teams model, terminated successfully producing a lot of trees which represent execution perspectives from UE or AUSF (Fig 1 and 2) . In each shape we noticed that the last 2 messages being exchanged and completed without alterations from the adversary indicating that there is no leak in the session key which is produced. The intermediate messages sent using nonces are being altered and the adversary can introduce new UE and AUSF instances to send these messages using newly generated nonces. The reason for this being CPSA tries to satisfy every node in different ways and one way it does so is by producing new roles. Our conclusion from this results that the session key intended for communication and securing application data is being generated and exchanged securely thus making the protocol secure for usage according to our model.



In the initial implementation of v1.3 (v1.3\_1) which was modeled **without** a high level abstraction produced shapes relatively faster that our v1.2 implementation. In this model we saw none of the



shapes were being completed from UE perspective which produced no fruitful result (shown in the figure 3 and 4). We then further tried to abstract the protocol on a high level by abstracting the handshake (v1.3\_2) and using the initial and final messages from our base v1.2 and noticed the protocol had shapes terminating successfully (Fig 5 and 6) . Although alongside this we also observed that from UE perspective the protocol would complete by itself. On further analysis we found that the private key used to sign the certificate had no origination assumptions in the code, which lead to the adversary having access to it and completing the protocol without the need for other messages.



In our final effort to model the protocol (v1.3\_final) we fixed the origination assumption and the shapes produced could complete safely and successfully with all the messages being exchanged well. We got two trees, the first tree representing UE perspective which could authenticate itself with the server using the handshake (Fig 7 and 8). The second tree shows the AUSF completing the protocol by itself, this is considered to be expected since the server can complete the protocol or authenticate itself. This concludes that the UE could securely authenticate and establish a connection with the server according to our implementation in CPSA.

## Challenges

One of the main difficulties we had while modelling is that we could not guarantee the correctness of our specifications by themselves. If we are implementing it wrong or we are missing something while executing, there is no way to verify that and this incorrectness will reflect in the result we generated, which will be wrong and the security property of TLS 1.3 cannot be proved. The second problem is that it is hard for us to interpret all the results. Though we have not found any vulnerabilities so far, we cannot prove that they do not exist in those shapes that we are not able to interpret. The third problem is that we can only specify a small number of message flows for TLS 1.3, while there are infinite possible message flows that we cannot specify them all. Fourth, a minor one, CPSA uses a lot of computing resources. There were instances when the program ran continuously for hours without terminating, and due to high cpu usage issues, it crashed.

## Future Work

Our implementation model of EAP TLS v1.3 could set the basis for further analysis on this protocol. Since v1.3 is the newest proposed version of TLS and is now being widely used as a backbone for communication security over networks including the Internet, it is desirable to ensure that it is fully secure. Our implementation of v1.2 spawned many roles which couldn't be restricted, this could be tried to be restricted in future implementations to check if the protocol could be modelled completely from UE perspective. TLS v1.3 has many other versions to be modelled like Failed Mutual Authentication, ClientHello, etc which in future, we need to verify the correctness of the results obtained by CPSA. All the versions of shapes of these models need to be explored in future to ensure that there is no vulnerability lying on those parts. In a normal scenario there are millions of messages transferred during a communication, while we tested for only a small set of messages, a test on how secure the protocol is for a large message set is needed to be done in future.

# Literature Review

**Formal Methods Analysis of the Secure Remote Password Protocol**

This article explains the exhaustive structural analysis of the Secure Remote Password Protocol (SRP) done using the Cryptographic Protocol Shapes Analyzer (CPSA) tool. The article first introduces the reader to the purpose for the analysis and gives a brief explanation as to the challenges of analyzing the SRP protocol due to the modular exponentiation and addition present in it’s mathematical equation.

The article then explains the formal methods used to analyse cryptographic protocols and give reasons why the team decided to use CPSA as the tool to analyse the protocol. They determined that the tool’s ability to enumerate the “minimal, essentially different executions of the protocol,” based on the assumptions made by the imputed model of the program, helped confirm the structural properties ensured by the protocol and could determine if there was any vulnerability to the protocol. Additionally, it was later stated in the paper that the scope of the model could be expanded which was what allowed the team to find the first potential attack on the SRP protocol.

Next, the article discussed the two part structure of the SRP protocol: the key establishment, and the key verification. This showed how both the client and the server would use the mathematical equation to establish the key for communication and how the server and client would then verify the key they exchanged.

Lastly, the article explained how the protocol and its mathematical equation was modeled in order to overcome the modular exponentiation and addition since CPSA doesn’t support arithmetic operations. The team modeled the equation, v+gb, as the encryption of gb using v as the encryption key. This adaptation of the equation and the creation of the service roles to provide function to the client and server allowed the team to run the model through CPSA and determine the shapes outputted by the tool had no indication of possible attack when both the server and client were honest in their interaction. The shapes generated also verified the properties claimed by the protocol, such as the SRP protocol does not leak the verifier to any potential attackers. The article then described the team’s expansion of the model to include the scenario of the server already compromised and launched an attack, which allowed for the discovery of a potential attack that can be performed against the SRP protocol.The article finishes with a discussion on the limitations of the analysis made on SRP due to the decisions made on the modeling of the mathematical equation and their use of CPSA’s point of view.

**Searching for Shapes in Cryptographic Protocols**

This article explains the methodology that finds every possible execution of a cryptographic protocol which are termed as the ‘shapes’ of the protocol. The paper asserts that most protocols only have a few shapes which allows their software, the Cryptographic Protocol Shapes Analyzer, to discover every shape and terminate for most cryptographic protocols. With the help of such shapes, observations can be made about the secrecy and authentication. The paper provides definitions and examples for important terminology that the CPSA software uses. The paper then applies this knowledge by analyzing Yaholm’s Protocol in order to demonstrate how the CPSA search methodology operates.

This paper is a good introduction into the CPSA software and various methods in identifying shapes with it. It introduces the reader to useful definitions that they will definitely encounter when using the software. In addition, the example that the authors provided was helpful in introducing the basic functions of the software as well as the expected outcomes.

**Formal Analysis of 5G EAP-TLS Authentication Protocol Using Proverif**

This paper explores the authenticity of the 5G EAP-TLS security protocol to see if it meets the standards needed for a communication network. The authors use the ProVerif model to construct a formal protocol model in order to identify flaws within the protocol. Their model uncovered a few flaws in the 5G EAP-TLS protocol such as the fact that the user cannot authenticate the identity of a home network. This means, an attacker could pretend to be the home network and compromise the user network. 5G EAP-TLS was also found to be susceptible to a man-in-the-middle attack, where the attacker can intercept messages from the user then forward them.

**An Analysis of the CAVES Attestation Protocol using CPSA**

This paper describes the CAVES attestation protocol and its analysis using the CPSA software. It starts by introducing an attestation protocol detailing its working and purposes along with the problem statement which explains the assumptions made during the protocol analysis with CPSA. Further, the paper gives an overview of CPSA, its language, syntax, parameters and various options and their functions.

The main goal of the protocol is the delivery of data while maintaining confidentiality, since the point of attestation is to ensure data goes only to a client with acceptable measurements. The analysis of this protocol is made under two scenarios: one is assuming the endpoints aren’t compromised & the channel is secure and the other assuming the adversary has access to the channel. Using CPSA these two cases are observed in detail and their findings are presented. Furthermore, analysis is done in studying if the protocol satisfies other high level attestation principles paraphrased during the introduction.

In conclusion the paper highlights the role that CPSA analysis played in justifying the protocol’s design and proving confidentiality and authentication properties. The paper also led to the formation of techniques within CPSA which model phenomena as private channels and hash functions.

**Security issues in the 5G standard and how formal methods came to rescue**

The paper described several serious security and privacy issues identified in the 5G standard and most of these issues were discovered with the help of security protocol verification tools based on formal methods. We learnt that 5G Authentication and Key Agreement (AKA) does not meet two critical security goals in particular which are: an attacker can either impersonate a serving network towards a mobile, or a mobile towards a network through formal security verification. The research also revealed that the specification, serving as deployment basis, of 5G networks does not meet some critical security goals. This led to a flawed deployment of 5G networks that may suffer from attacks. It is very likely that all 5G subscribers will be subject to the aforementioned attacks, a situation and this has already drawn the attention of news media. We came across study of automated tools which were based on mathematical principles, such as the Tamarin or Pro Verif verifiers for security protocol analysis. The resulting security guarantees were machine-checked and were based on mathematical foundation. The paper also disclosed that different lines of research in numerous verification tools are capable of analysing various security properties. This has led to numerous large-scale formal analyses of standard properties on real-life protocols recently, e.g., TLS 1.3 and the Signal instant messaging protocol. Some of those analyses have led designers and standardisation bodies to correct flawed protocols. It is safe to say that existing techniques for automated security verification have now reached maturity to guide design, analysis, and standardization.

# Biographical Sketches of Investigators

**Juhi Patel**

Juhi Patel is currently pursuing a masters in Information Assurance at the University of Texas at Dallas. This past summer, she has been working on designing and developing an app at UTSW that focuses on gathering data from young adults to see if they are at risk/already have depression. She has taken coursework for machine learning, database design, algorithm design, operating systems, systems security, and malicious code. She has also worked on the classification of malware binaries using extracted artifacts. She has a strong background with Java and C++ with some experience in MIPS, C, UNIX, x86, and ARM.

**Lizbeth Trevino**

Lizbeth Trevino is currently a masters in Information Assurance at the University of Texas at Dallas. This past fall semester she worked on front-end development using React for a start-up company called Amna. She has taken coursework for network security, database design, operating systems, algorithm analysis and design, and digital forensics. Additionally, she worked on creating a file retrieval program in a UNIX environment. Lastly, she has a strong background in Java and C++ with some experience in MIPS, Typescript, Javascript, UNIX, and Python.

**Kunal Mangalorekar**

Kunal Mangalorekar is currently a masters in Computer Science at the University of Texas at Dallas with Information Assurance as a concentration. He has done his Bachelors in Computer Science from National Institute of Technology India. He has been involved with a broad spectrum of technologies which include machine learning, cloud services, network security, and the Internet of Things. He also has industry experience with developing solutions in the cloud and is AWS certified associate solutions architect. Additionally, he has taken course work in network security, malicious code analysis, information security, and cryptography. He has experience with C++, Python, Unix, AWS, and IoT protocols like MQTT.

**Nidhi Jawandhia**

Nidhi is currently pursuing a masters in Computer Science with a concentration in Information Assurance at the University of Texas at Dallas. She has done her Bachelors in Mathematics and Computing from Birla Institute of Technology, India. This last year she has done multiple projects on security, one of the major ones include providing extra security on cloud-based applications. She has also worked on Image Impainting and then providing encryption using the missing pixel values as keys by using Elliptical Curve Cryptography. She has experience working in the Tata Institute of Fundamental Research Centre for Applicable Mathematics and Indian Statistical Institute. Also, she has taken coursework in data and application security, network security, algorithm design, advanced computer networks, advanced operating system and database design. She has a strong background in C, Python, and UNIX with some experience in Java and Javascript.

# Bibliography

[1] A.T. Sherman, E. Lanus, M. Liskov, E. Zieglar, R. Chang, E. Golaszewski, R. Wnuk-Fink, C.J. Bonyadi, M. Yaksetig, and I. Blumenfeld, “Formal Methods Analysis of the Secure Remote Password Protocol,” Cyber Defense Lab, University of Maryland, Baltimore, MD, USA, 2020. Accessed: Feb. 9, 2021. [Online]. Available: <https://arxiv.org/abs/2003.07421>

[2] Doghmi, Shaddin F., et al. “Searching for Shapes in Cryptographic Protocols.” *Tools and Algorithms for the Construction and Analysis of Systems*, pp. 523–537., doi:10.1007/978-3-540-71209-1\_41.

[3] J. Zhang, L. Yang, W. Cao and Q. Wang, "Formal Analysis of 5G EAP-TLS Authentication Protocol Using Proverif," in IEEE Access, vol. 8, pp. 23674-23688, 2020, doi: 10.1109/ACCESS.2020.2969474.

[4] Ramsdell, John D., Joshua D. Guttman, Jonathan K. Millen, and Brian O'Hanlon. "An analysis of the CAVES attestation protocol using CPSA." *arXiv preprint arXiv:1207.0418* (2012).

[5] Lucca Hirschi, Ralf Sasse, Jannik Dreier. Security Issues in the 5G Standard and How Formal MethodsCome to the Rescue. ERCIM News, ERCIM, 2019. Hal-02268822

[6] Zhenhua Li, Weiwei Wang, Christo Wilson, Jian Chen, Chen Qian, Taeho Jung, Lan Zhang, Kebin Liu, Xiangyang Li, and Yunhao Liu, “FBS-Radar: Uncovering Fake Base Stations at Scale in the Wild,” Proceedings of the Internet Society Symposium on Network and Distributed System Security (NDSS) (February 2017).

[7] 3GPP, “3GPP System Architecture Evolution (SAE)—Security Architecture” (Release 15), technical specification (TS) 33.401, v15.2.0 (September 2018).

[8] 3GPP, “Security Architecture and Procedures for 5G System” (Release 15), technical specification (TS) 33.501, v15.5.0 (September 2018).

[9] David Basin, Jannik Dreier, Lucca Hirschi, Saša Radomirovic, Ralf Sasse, and Vincent Stettler, “A Formal Analysis of 5G Authentication,” Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18) (October 2018).