sustainability

MDPI

*Article*

# Implementation of Efficient 5G AKA Protocol for Light-Weight Environment

**Seonghwan Cho [1], Cheolmin Yeom [2] and Yoojae Won [1,*]**

[1] Cyber Security Laboratory, Department of Computer Engineering, Chungnam National University, Daejeon 34134, Korea; 202050243@o.cnu.ac.kr
[2] National Forestry Cooperative Federation, 166, Seokchonhosu-ro, Songpa-gu, Seoul 05601, Korea; cjfals@nfcf.or.kr
* Correspondence: yjwon@cnu.ac.kr; Tel.: +82-042-821-6294

**Abstract:** There is a critical vulnerability in the 5G AKA protocol of the 3GPP. We propose a protocol that compensates for these vulnerabilities and speed up the existing protocol; and the study implements this protocol to experiment with the vulnerabilities. The protocol uses the Light-weight Encryption Algorithm and is expected to be $\frac{n}{3}$ times faster in than with the application of Advanced Encryption Standard algorithms. By using this algorithm, we improve the protocol efficiently, reduced the power consumed.

**Keywords:** 5G; 5G privacy; security; authentication protocol; 5G AKA protocol; authentication and key agreement; AKA vulnerability; LEA; 3GPP

check for updates

**Citation:** Cho, S.; Yeom, C.; Won, Y. Implementation of Efficient 5G AKA Protocol for Light-Weight Environment. *Sustainability* **2021**, *13*, 8982. https://doi.org/10.3390/su13168982

Academic Editors: Manuel Fernandez-Veiga and Jaime Lloret

Received: 31 May 2021
Accepted: 5 August 2021
Published: 11 August 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

According to the investigation report published by the Global System for Mobile Communication Association (GSMA) [1], over 5.2 billion people subscribed to mobile services by the end of 2019, accounting for 67% of the global population.

As shown in Figure 1, GSMA also predicted that by 2025, approximately 20% of all mobile service subscribers would use 5G. 5G mobile communication is a next-generation wireless technology that represents a complete transition of communication through innovative changes in data transmission speed, delay rate and number of device connections compared to conventional mobile communication.

5G users are estimated to increase because of the use case scenarios that 5G can provide. According to a report by the International Telecommunication Union (ITU), the use case requirement comprises three things: enhanced mobile broadband, massive machine-type communications and ultra-reliable and low latency [2]. If these use case requirements are satisfied, 5G can implement a number of use cases, including voice calls and simple data communication, as well as self-driving vehicles, augmented reality and smart cities connected with the Internet of Things (IoT). If they are implemented, it is expected that more IoT devices will be connected and a large amount of data will be delivered.

The provision of various services can increase user convenience, but it can also simultaneously put them at risk because of increased attack surfaces [3]. IoT devices are often unable to provide the computing power to ensure sufficient security protocols. Since attackers can exploit these vulnerabilities to carry out Denial-of-Service attacks, network operators and standards organizations must operate the network considering low-level IoT devices.

The 3rd Generation Partnership Project(3GPP), which is a standardization body, emphasizes that sensitive information such as subscribers' identifiers, locations and data is stored and transferred over network, and that presenting standards that protect users' privacy in various ways is important for the operation of secure 5G networks [4]. Currently, studies on the 6G standard are underway. The 6G white paper emphasizes the importance

of developing technologies that not only meet users' requirements as they shift to the next generation, but also provide protection. In 6G, the network is more expansive than the previous generation, and privacy protection technologies must be developed accordingly. For the development of standards for these privacy protection technologies and regulations, it is critical to consider how 5G technologies can affect the next generation. If a technology with a vulnerability is inherited, it could lead to more serious consequences in the future [5]. Furthermore, technologies that consider IoT environments should be considered for network expansion. Mitigating IoT-related security threats in 5G is one of the main goals in 6G standard studies, as recommended by the existing standards, because previous generations of vulnerability continue to threaten the network [6]. To avoid this risk, studies have been conducted to improve security in various ways, which include creating secure, efficient connections and trust models from an IoT perspective, improved related algorithms and secure architectures [7–9]. Improving standards in consideration of low-level devices helps develop sustainable technologies. Increasing the performance of the standard, considering only security, may prevent devices from operating, in the worst case scenario. To prevent this, standards should be improved in an efficient way that is fast and low consumable [10]. These improvements can be driving the development of sustainable technologies.

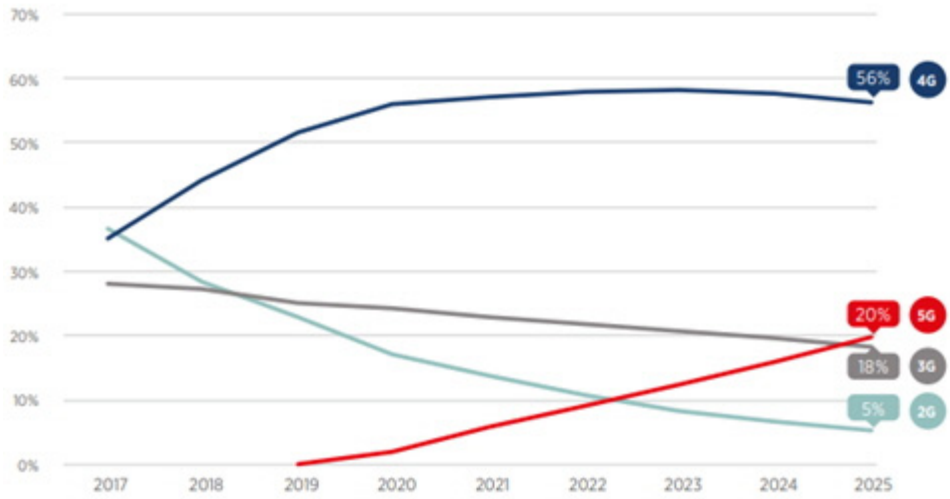


**Figure 1.** GSMA prediction regarding 5G users (Adapted from [1]).

Vulnerabilities had been identified in the 5G key agreement protocol, which is an important element of the 3GPP's standards [11–13]. The protocol had several problems in the previous generation of 5G but was not properly modified, inheriting vulnerabilities and affecting the 5G. Borgaonkar et al. found structural weaknesses in 5G authentication and key agreement (AKA) protocol, one of the 5G key agreement protocols [14]. Vulnerability is the exposure of users' information in plaintext when authentication fails under certain conditions; this study explores passive attacks that exploit this simply to eavesdrop and active attacks that exploit structural protocols with inexpensive equipment to manipulate data. This situation contradicts the protection of users' information emphasized by 3GPP, and several studies have been conducted in an attempt to solve it. Jung et al. supplemented the effort by introducing hardware equipment, but that the requirement of additional equipment is a disadvantage [15]. Braeken et al. proposed the random replacement of random numbers and sequence numbers proposed to complement the protocol, but this burdens the user equipment (UE) and home network (HN), and it necessitates changes in the protocol structure [16].

Therefore, we propose a method to address this problem and increase the speed of the protocol. The improved protocol encrypts plain text to compensate for vulnerability and applies a Lightweight Encryption Algorithm (LEA) [17] for a lightweight environment. It

replaces the parts encrypted with an Advanced Encryption Standard (AES) algorithm [18] used in the existing protocol and assumes that it can increase the speed of the protocol.

By introducing LEAs that are lighter and faster than an AES algorithm, in future, even low-level devices will be able to successfully carry out authentications in IoT environments. In addition, more efficient and sustainable authentication can be performed because LEA reduce power consumption during encryption and decoding. The implemented improved protocol will be introduced into the core network that will be implemented in our lab in future to experiment with the vulnerabilities of the AKA protocol in a real-world environment.

### *1.1. Background*

Before explaining the methodology of this study, we expand upon two major pieces of background knowledge: the vulnerability of the 5G AKA protocol and the algorithms used in the protocol. In the 5G AKA protocol section, we provide an overview of both the protocol and attacks that exploit the vulnerability. In the algorithm section, we describe the MILENAGE algorithm [19] introduced to apply the LEA.

#### 1.1.1. Vulnerability of the 5G AKA protocol

5G AKA protocol is a third-party authentication protocol [20]. The four objects participating in the AKA protocol can be described as follows:

- UE: The UE is a device used to access a network. Typically, these are mobile phones, and they can be identified in HNs through a unique identifier called subscription permanent identifier (SUPI), which plays the role of 'IMSI' in pre-5G generations.
- SEAF: The security anchor functionality (SEAF) serves in the serving network (SN) as the anchor for security in 5G. The SN controls the base station to communicate wirelessly with the UE. The SEAF provides security anchor functions.
- AUSF: The authentication server function (AUSF) serves the HN, which is the service provider of the subscriber. The AUSF handles authentication requests for both 3GPP and non-3GPP access.
- ARPF: The authentication credential repository and processing function (ARPF) serves in the HN. It provides authentication and key agreement credentials.

Before describing the 5G AKA protocol sequence, we provide information about the protocol. The USIM of the subscriber contains important information needed for authentication. The SUPI and HN's public keys (pkHN) and long-term keys (K) are used as shared private keys by the HN. The pkHN can encrypt the SUPI and convert it to a subscription-concealed identifier (SUCI). K is used to calculate the various key materials required to calculate the anchor key ($K_{SEAF}$) and $K_{AUSF}$ [20]. The function used in the calculations is called the key derivation function (KDF), which is further described in the algorithm section.

The protocols authenticate each other via a challenge-response method by calculating the response value (RES) and message authentication code (MAC). The RES is computed in the UE; and the XRES is computed in the HN. The SN authenticates the UE and HN by comparing the XRES and RES. The MAC is an authentication code generated by the HN, and XMAC is the value calculated by the UE. The UE authenticates the HN by comparing the MAC and the XMAC. After each is authenticated, they store the $K_{SEAF}$ computed in the previous process, allowing efficient access to the network in the future. The primary values used in this calculation are as follows:

- RAND: The 128-bit random value.
- AUTN: The authentication token consists of the sequence number (SQN), authentication management field (AMF) and the MAC. These materials require key calculation and synchronization verification for freshness of the authentication(synchronization).
- (X)RES*, H(X)RES*: The (X)RES* is a calculated value using RES by KDF to authenticate the UE and the HN. The H(X)RES* is the hash-based authentication code (HMAC) calculated with the RES* using the sha-256 hashing algorithm. Comparing (X)RES*

is at HN side and comparing H(X)RES* is at SN side. More details about calculating *-labelled value and HMAC are at Section 1.1.2.

- $K_{AUSF}$: The top-level key required to compute $K_{SEAF}$. $K_{AUSF}$ and $K_{SEAF}$ are calculated using the KDF.
- Cypher key (CK), Integrity key (IK) and Anonymity key (AK): These are the keys for use in KDF.
- AV: Authentication vector (AV) is the vector flow in HN and SN. HE AV is the AV flow between the ARPF and AUSF, and SE AV is the AV flow between the SEAF and AUSF. HE AV consists of RAND, XRES*, $K_{AUSF}$ and AUTN. SE AV consists of RAND, HXRES* and AUTN.

There are two cases in which the protocol fails. The first is when the MAC and XMAC do not match. If these two values do not match, the protocol is interrupted by the MAC failure message. The other is when the sequence number (SQN) is not within the allowed increment range. The SQN is a number that sequentially increases during the protocol. Its value is used during the authentication process. If the difference between the SQN calculated by HN and that calculated by the UE is large, the protocol is interrupted with a synchronization failure message. In such a situation, the UE sends a failure message to the SEAF, and the message indicates that there is an error in the authentication process.

Figure 2 shows the sequence diagram for the operation of the 5G AKA protocol. The red box in the figure represents a vulnerable area to be addressed intensively in this study.

(1) The UE sends the SUCI to the SEAF. The SUCI is the encrypted SUPI value. The UE encrypts/decrypts SUCI using pkHN.
(2) The SEAF sends SUCI of the UE and SN name values to the AUSF. The SN name value consists of '5G' and 'SN id'.
(3) The AUSF checks the SN list to authenticate the SN. After authentication, the AUSF sends the SUCI and SN names to the ARPF.
(4) The ARPF authenticates the UE and verifies the authentication material and the AV. The UE is authenticated by checking the UE list. The SQN, MAC and XRES are calculated for use in authentication. The MAC, XRES, CK, IK and AK were calculated using f1-f5. XRES is one of the challenge-response values calculated in HN. The 5G HE AV is the AV sent to the AUSF. The 5G HE AV comprises the authentication material of the protocol.
(5) ARPF sends the 5G HE AV to the AUSF.
(6) The AUSF can calculate HXRES*, $K_{AUSF}$ and SUPI using 5G HE AV. The $K_{AUSF}$ is the key for calculating $K_{SEAF}$. The $K_{SEAF}$ is a very important key, so it should not be leaked outside the HN until authentication is completed. The HN does not expose the $K_{SEAF}$ directly, thus inducing the UE to compute the $K_{SEAF}$ with information that only the UE and HN know: KEY, and RAND. Therefore, the AUSF sends only the $K_{AUSF}$ to the SEAF.
(7) The AUSF sends 5G SE AV to the SEAF. The SEAF calculates HXRES* and stores the value to authenticate the UE and HN.
(8) The SEAF sends RAND and AUTN to the UE.
(9) The UE calculates and compares the MAC to the XMAC and SQN. To calculate the XMAC, the UE calculates the key material using f1-f5. In this step, if the authentication fails, the UE sends failure message to the SN. Once the UE has successfully authenticated, it calculates RES* and $K_{SEAF}$ using materials.
(10) The UE sends the RES* to the SEAF.
(11) The SEAF authenticates the UE and HN by comparing the HRES* to HXRES*. HRES* and HXRES* are hashed values.
(12) The SN sends the RES* value to the HN.
(13) Finally, the AUSF authenticates the UE by comparing the RES* and XRES* values. If authentication is successful, the AUSF sends the authentication result to the SN and ARPF.
(14) The AUSF sends the authentication result to the ARPF and SN, and the $K_{SEAF}$ to SN.

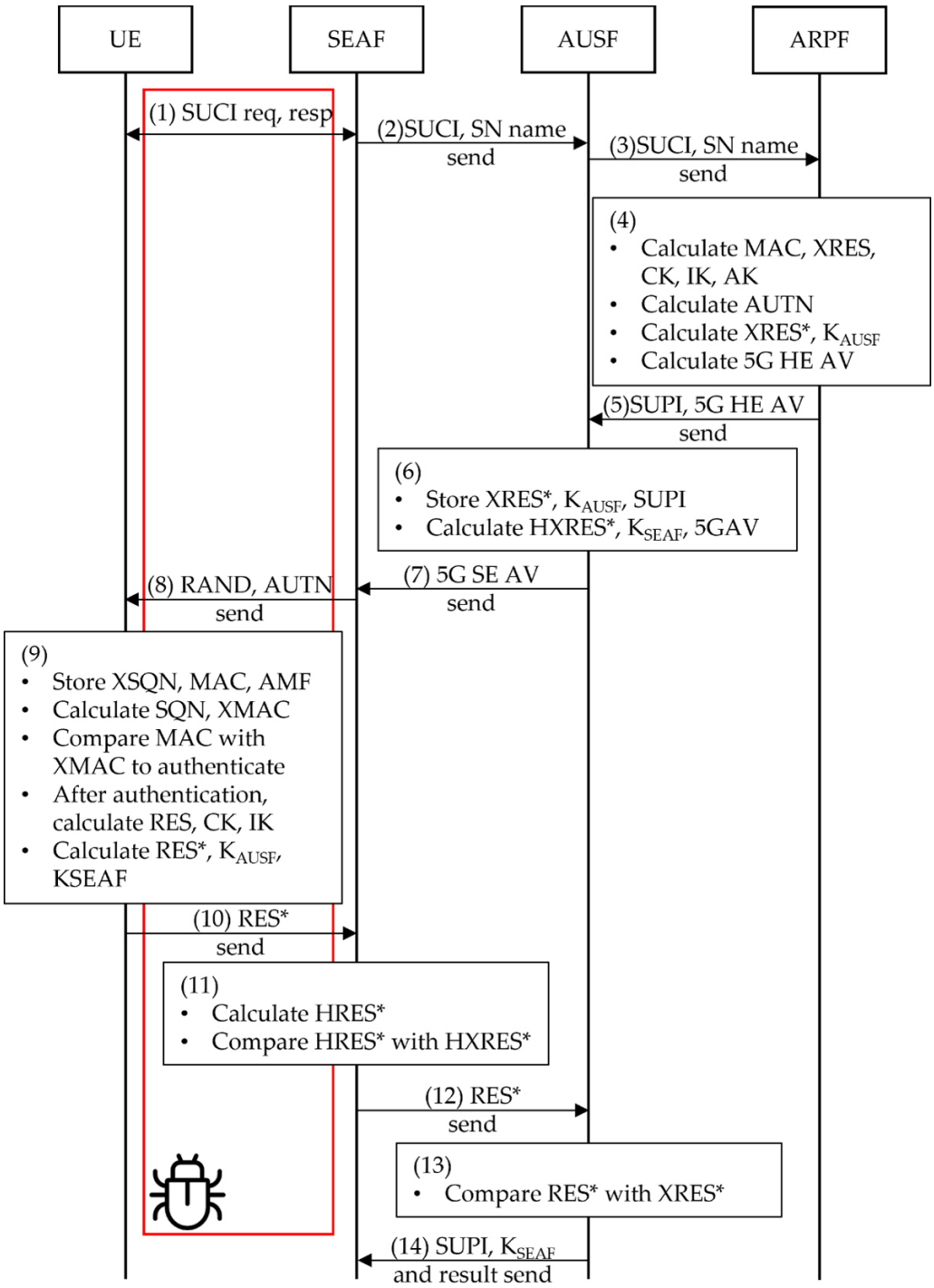


**Figure 2.** 5G AKA protocol sequence.

The UE and SN communicate in a wireless environment, making it easy to eavesdrop messages using simple equipment [14]. The SUCI value is disclosed because the 5G AKA protocol proposed in the standard sends an authentication failure message in plaintext. This implies that if an attacker injects a packet or uses the Man-in-the-Middle attack (MitM) to cause an authentication failure, the UE sends a plaintext message containing the SUCI value to the SEAF. These types of attackers are called active attackers; they have fake SEAF and are familiar with the protocol, so they can use MitM attack. This attack can be established by the race condition, allowing an active attacker to generate a failure message for the target and a passive attacker, who only eavesdrop failure message using simple

equipment. These attackers compromise the security of the protocol. If an attacker who bears the target's USIM wishes to proceed with additional attacks using SUCI information, they can easily do so.

There are two kinds of attacks: (1) unauthorized access to the network using the USIM of the target, and (2) intentionally making an authentication failure message to the target. In addition, attacker receive message can locate the target. When an authentication request is sent via broadcast, other UEs within the base station that do not correspond to the MAC provide MAC failure messages, but they send synchronization failure messages if the target exists in that base station range. In other words, an attacker can locate the target UE within the range of the attacker's base station if the authentication fails, because a synchronization failure message is received.

#### 1.1.2. Algorithm

This section describes the algorithms used in the protocol. We describe only the purpose of using this algorithm and its core principles.

MILENAGE is a set of algorithms developed by the 3GPP security working group for authentication and key generation. Even though it is not a standard because it is a set presented in 3GPP, we implemented the protocol using MILENAGE. It has seven cryptographic functions, f1, f1*, f2, f3, f4, f5 and f5* [19]. The *-labelled functions are re-synchronization functions used to verify whether the UE and HN have calculated the same values. The way of these functions is same with original function. The first three functions (f1, f1* and f2) are functions for authentication, while the remaining four functions (f3, f4, f5 and f5*) are functions used for key generation.

As shown in Figures 3 and 4, we describe how f1–f5 functions work at the HN and UE side [21].

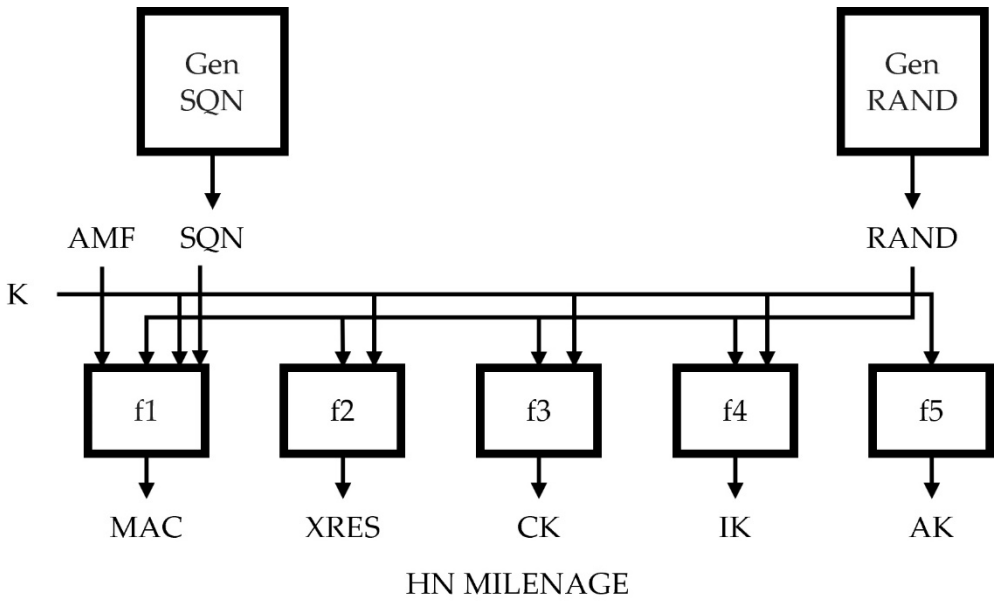


**Figure 3.** MILENAGE algorithm set.

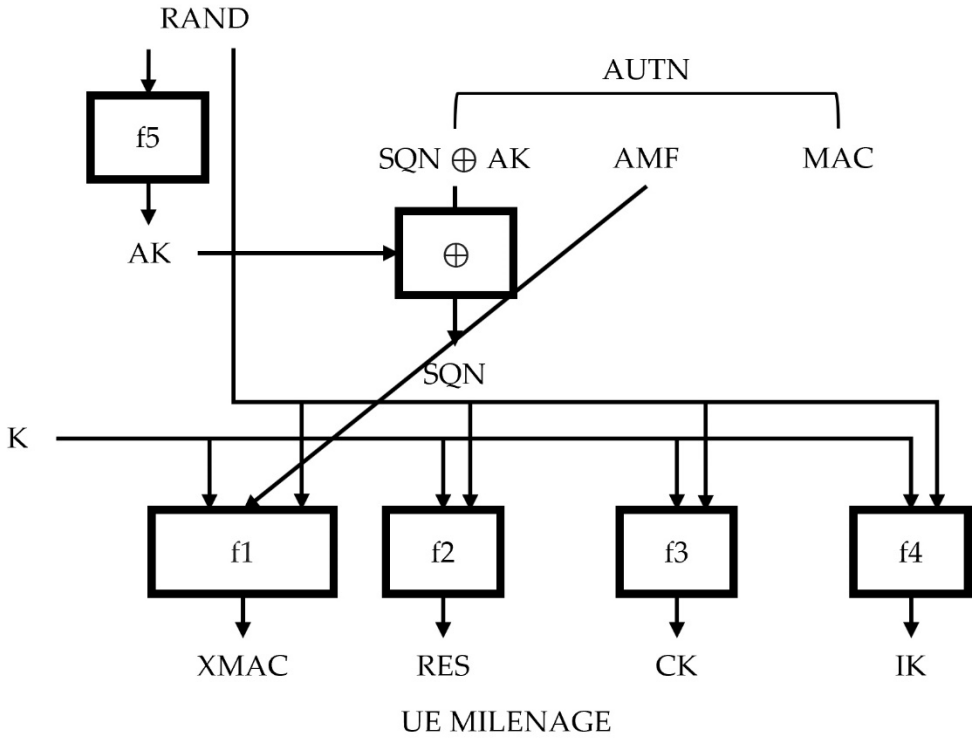


**Figure 4.** MILENAGE algorithm set at UE side.

The KDF is used to derive keys for secure communication. Table 1 shows the KDF used in the 5G AKA protocol [20].

**Table 1.** KDF function.

| Name | Input Key | Input |
|---|---|---|
| $K_{AUSF}$ | CK\|\|IK | 0x6A\|\|SN name\|\|length of the SN name \|\|SQN ⊕ AK\|\| length of SQN ⊕ AK |
| (X)RES* | CK\|\|IK | 0x6B\|\| SN name\|\|length of the SN name\|\|RAND\|\|length of RAND\|\|(X)RES\|\|length (X)RES |
| $K_{SEAF}$ | $K_{AUSF}$ | 0x6C\|\|SN name\|\|length of the SN name |
| (X)HRES* | RAND\|\|(X)RES* | RAND\|\|(X)RES* |

Ultimately, keys are required to generate $K_{AUSF}$ and $K_{SEAF}$. 3GPP recommends using the AES algorithm when implementing MILENAGE.

LEA is an international standard encryption algorithm developed by the National Security Institute of Korea [17]. It is a 128-bit block password that provides confidentiality in high-speed environments such as big data and cloud computing as well as in light-weight environments such as in mobile devices. Advances in the IT environment have required high-speed, lightweight and low-power cryptosystems on the IoT, mobile and cloud, resulting in the proposed efficiently powered LEA on a 32bit platform [21]. When built with software, both ROM and RAM display a higher efficiency based on the ARM platform and a 1.5 times faster performance than AES. When built with hardware, it can also be highly efficient because bit XOR, rotations, and 32-bit adders without complex operations are possible. For security, it provides a 24/28/32 round count depending on the key length (128/192/256 bits). It provides 30% security margin performance for partial round attacks.

### *1.2. Contributions*

Contributions to this study include:

(1) We implemented the improved 5G AKA protocol for use in our lab, when the 5G testbed was implemented in our lab.

(2) To implement faster protocols, we introduced LEA from the existing AES algorithm and tested the degree of improvement.

### *1.3. Organization*

The remainder of this paper is organized as follows. Section 2 introduces the structure of the complemented protocol and the evaluation process. Section 3 presents the results of the protocol, experiment and analysis the evaluation results. Finally, the results are presented in Section 4.

## 2. Materials and Methods

In this study, we propose an improved protocol that applies LEA.

Figure 5 shows the improved sections of the 5G AKA protocol. The sections marked by a gear increase the encryption speed using LEA, while those marked with a lock are made less vulnerable, thus complementing safety using LEA.

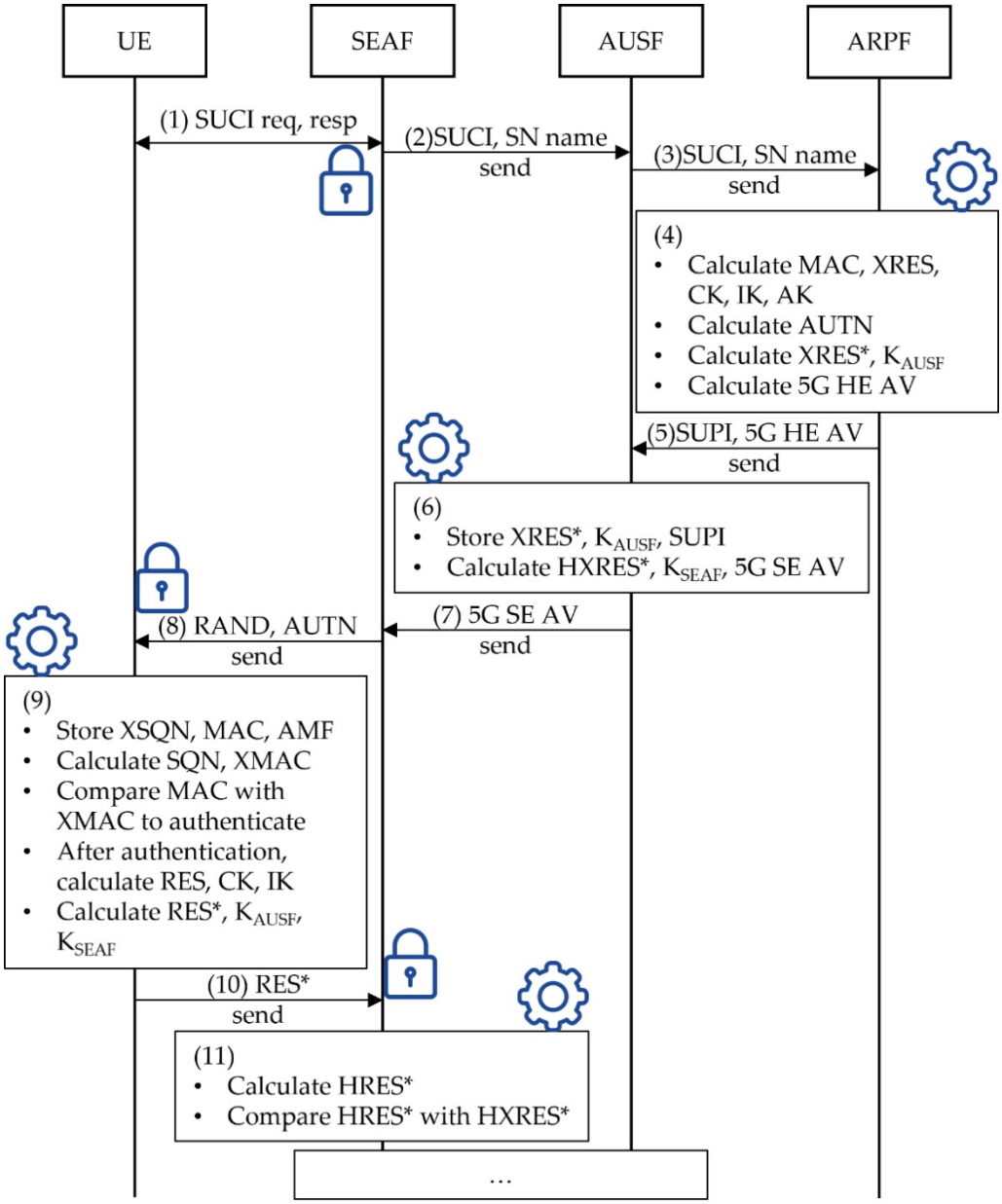


**Figure 5.** Improved 5G AKA protocol procedure.

To implement the f1-f5 function we introduced the MILENAGE algorithm set. We used the CryptoMobile open-source and custom MILENAGE algorithms. As shown in the Figure 6, we changed all AES encryption algorithms to LEA. Between the UE and the SN, we implemented the LEA encryption and decryption function to send an encrypted message and containing information on the UE (SUCI). Using LEA, the speed of the protocol was increased, and in the future, network can provide the environment where many IoT devices are connected.

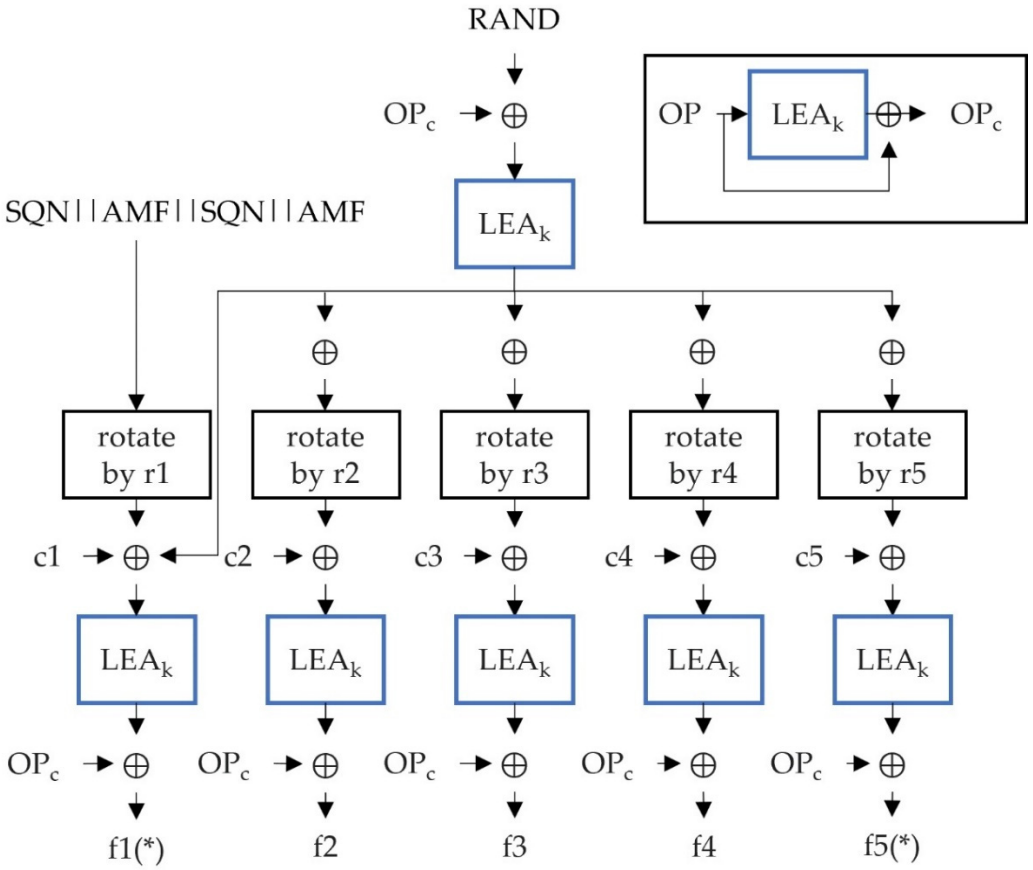


**Figure 6.** MILENAGE using LEA.

In summary, the vulnerable plain text message was encrypted using LEA, and the speed was improved by applying LEA to all parts of the AES algorithm.

### *2.1. Environment*

The protocol was implemented in the Ubuntu 20.04 version of the Linux OS using Python (version 3.8.5) and communicated using the UDP protocol. Since we do not currently have a 5G UE and could not implement a testbed, the protocol was implemented as a software. We used socket communication on a server. The experimental environment is shown in Figure 7, where SEAF is a regular server and not a base station.

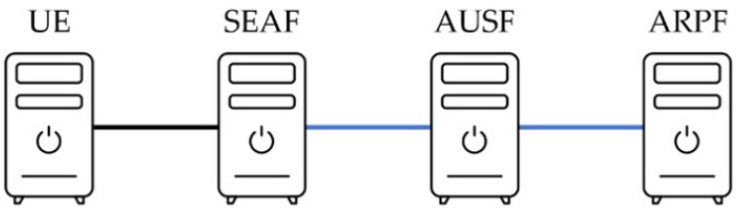


**Figure 7.** Environment overview.

For encryption, an open-source called a CryptoMobile was used. The open-source is a Python implementation of MILENAGE, which includes the functions f1–f5*. However, it was necessary to apply LEA because it is originally implemented as an AES algorithm. The Korea Internet and Security Agency (KISA) provides LEA for free within Python source code. Therefore, we changed the AES encryption algorithm to LEA in MILENAGE.

### *2.2. Experiment*

The experiment was conducted in two stages: checking how well the vulnerabilities were overcome and measuring the speed of the protocol.

First, we checked how SUCI and failure messages are delivered to flat statements in the existing protocols. Subsequently, we confirmed that the vulnerability was remedied by verifying the SUCI that had been encrypted and transmitted by the improved protocol. In addition, the SUCI, K and SN names were tested with fixed values because no real 5G UE and testbed were available for the experiment.

Next, to verify the speed-up, we implemented the AES encryption 5G AKA protocol and LEA 5G AKA protocol and compared their processing times. After the experiment was conducted a sufficiently large number of times, we compared the protocol processing times.

### *2.3. Formula*

Theoretically, the degree of speed improvement when applying LEA is obtained by formula.

Assuming that the speed of the LEA is 1.5 times faster than that of the AES, an expression was derived to compare the speed [17]. we assumed that $T_{AKA}$ is the performance time of the existing protocol, $T_{AES}$ is the encryption performance time and $T_{LEA}$ is the LEA encryption performance time. The derived expression is shown below.

$$T_{AES} = 1.5 \times T_{LEA} \quad (1)$$

All AES encryption algorithms were improved by applying LEA, which can improve the time required for AES encryption. The speed of the improved 5G AKA protocol is called $T_{LEA_AKA}$.I If AES encryption is applied "n" times, expression (2) is established.

$$T_{LEA_AKA} = T_{AKA} - n \times T_{AES} + n \times T_{LEA} \quad (2)$$

The application of expression (1) in expression (2) yields the expression shown in (3) and shows that the improved protocol has a speed improvement of $\frac{n}{3}$ $T_{AES}$.

$$T_{LEA_AKA} = T_{AKA} - n \times \frac{1}{3} T_{AES} \quad (3)$$

## 3. Results

*The Result of the Protocol*

1. Protocol result

Figure 8 shows the process of the UE encrypting authentication message (9). We used random pkHN in the experiment; in this case, we used the shared key, and the UE encrypted the RES* value with the shared key to send to the SEAF. In addition, we sent encrypted failure message to the SEAF, and the passive attacker could eavesdrop on the message but could not calculate the SUCI and MAC values.

```
UE start
> dh_sharedkey :  5c6a8c088a371dd717cc8bb664728afa
SEAF
Authentication Response recv
[dec]ct  :  b"Z\xeb\xca\xd9{\x11\x1a P\x9a\xc5\xc0/\xe6\x11)\xc5\xaen\xcc\x0cj\xa3$\x82\xa5j;qy:\x8f\xc2\x94
```

```
def LEA_encrypt(pla_data_):
    pt = bytearray(pla_data_, 'utf8')
    print("[enc]pt : " + str(pt))
    leaECB = LEA.ECB(True, hdh_sharedkey, True)
    ct = leaECB.update(pt)
    ct += leaECB.final()
    return str(ct)
```

```
def LEA_decrypt(cry_data_):
    ct = bytearray(ct_preprocessing(cry_data_), 'utf8')
    ct = ct.decode('unicode-escape').encode('ISO-8859-1')
    print('[dec]ct  : ', ct)

    leaECB = LEA.ECB(False, dh_sharedkey, False)

    pt = leaECB.update(ct)
    pt = remove_padding(pt)
    print('[dec]pt  : ', pt)
    decrypt_output = pt.decode('utf8')
    print("[dec]result : ", decrypt_output)
    return str(decrypt_output)
```

**Figure 8.** Encrypted authentication message and algorithm.

2. Speed analysis

We implemented the 5G AKA protocol using the AES encryption algorithm and LEA. To compare the processing time, we experimented 500 times and measured the protocol processing time. We used the time it module in Python to measure the process time.

As detailed in Section 2.3, we calculated the time taken by the existing protocol. The 5G AKA protocol using LEA was used to complement the vulnerability in the encryption and decryption functions. Since this function does not exist in the existing protocol, we measured the processing time excluding encryption and decryption function time between the UE and the SN in the improved protocol. We present the time graph in Section 4.

## 4. Discussion

Experiments show that existing vulnerabilities are supplemented through encrypted protocols. The measurement protocol process time was used to analyze the speed of the improved protocol.

Figure 9 shows the time graph of the process time.

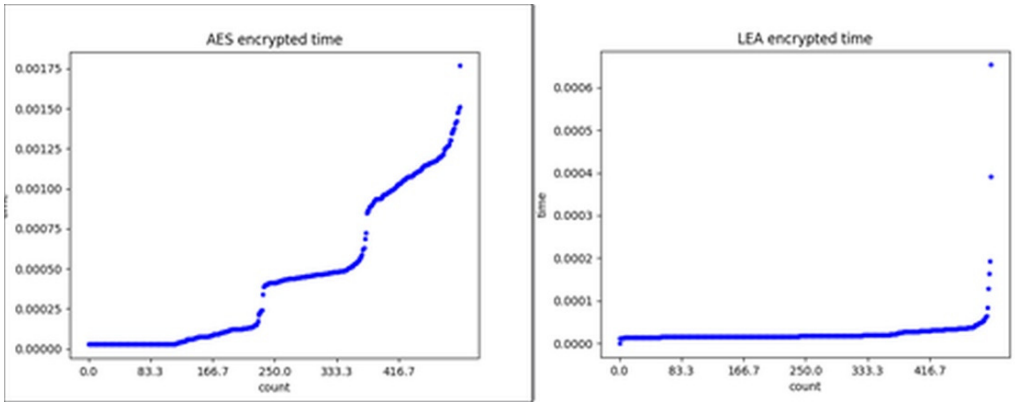


**Figure 9.** Measurements time graph.

We performed the experiment 500 times and sorted the measurements. In the case of AES encryption, the measured average time was 0.000442 s and using LEA encryption, the measured average time was 0.000024 s. The measured maximum time using the AES encryption was 0.00176 s and using LEA encryption was 0.00065 s. When the time was sufficiently large, the performance of LEA was 1.5~2 times faster than that of the AES encryption.

A discussion is needed for the degree of increase in speed. In Section 2.3, we expected that $T_{LEA_AKA}$ would be faster than $T_{AKA}$ only in encryption time. However, in total time, $T_{LEA_AKA}$ is more than twice as fast as $T_{AKA}$. Therefore, we must redesign experiments more carefully to accurately measure time.

In the processing time using LEA, because the encryption and decryption functions were added, the time is bigger than the $T_{AKA_LEA}$ in Figure 8. Therefore, we need to improve encryption and decryption functions.

## 5. Conclusions

The standardized 5G AKA protocol is vulnerable during authentication because the messages sent between the UE and SEAF are transmitted unencrypted. While 3GPP places considerable emphasis on the privacy of subscribers, it was necessary to modify the 5G AKA protocol to compensate for the structural vulnerabilities that have been inherited. We also expect that if the protocol is inherited as it is in future 6G standard studies, it will have serious consequences as it inherits vulnerabilities arising from 5G. These standards should be improved in a way that can be operated in a fast and light-weight environment in consideration of the low-level device.

To complement this vulnerability, we proposed an improved protocol that uses LEA for encryption the UE and the SN. In addition, instead of using AES algorithms recommended by existing standards, we speed up the protocol using LEAs that can be implemented in IoT in the future and can provide high speed in cloud environments [22]. In addition, LEA reduce the power consumed, it is efficient to use LEA in encryption and decryption section.

Experiments demonstrate that using pkHN during encryption/decryption provides higher performance than protocols using conventional AES algorithms. We find the improved protocol valuable because it encrypts the information sent by plain-text and is increased the speed of the protocol and we can experiment with them directly using existing structures. In the experiment, we demonstrate an improvement in the speed of the encryption and decryption.

The improved AKA protocol has been complemented the vulnerability and sped up the processing time. Using the LEA we speed up the processing time of the protocol and reduce consumed power. In addition, we don't change the framework or append the hardware equipment or logical structure, there is no burden on revising the standards. Since sustainable technology is important in the environment of the 5G and the next generation, we expect this protocol is effective.

If we can continue our research, we want to design the protocol safe to address the MitM attack. In addition, we improve the encryption and decryption function to reduce the processing time. Further, we measure safety and speed even in multi-UE environments, and see how much higher performance, when implemented in cloud environments.

**Author Contributions:** C.Y. research paper for the 5G AKA protocol, S.C. design and experiment. Y.W. supervised the study. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** This work was supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-01343, Training Key Talents in Industrial Convergence Security).

**Conflicts of Interest:** The funders had no role the study design; in the collection, analysis or interpretation of data; in the writing of the manuscript or in the decision to publish the results.

## References


1. GSMA. The Mobile Economy. 2020. Available online: https://www.gsma.com/mobileeconomy/wp-content/uploads/2020/03/GSMA_MobileEconomy2020_Global.pdf (accessed on 21 May 2021).
2. ITU. *IMT Vision–Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond*; Recommendation ITU: Geneva, Switzerland, 2015; Available online: https://www.itu.int/rec/R-REC-M.2083 (accessed on 3 May 2021).
3. Nieto, A.; Acien, A.; Fernandez, G. Crowdsourcing Analysis in 5G IoT Cybersecurity Threats and Mitigation. *Mob. Net. Appl.* **2019**, *24*, 881–889. [CrossRef]
4. 3GPP-Technical Report; Study on the Security Aspects of the Next Generation System. (Release 14). 2017. Available online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3045 (accessed on 21 May 2021).
5. Ylianttila, M.; Kantola, R.; Gurtov, A.; Mucchi, L.; Oppermann, I.; Yan, Z.; Nguyen, T.H.; Liu, F.; Hewa, T.; Liyanage, M.; et al. 6G white paper: Research challenges for trust, security and privacy. *arXiv* **2020**, arXiv:2004.11665.
6. Akhtar, M.W.; Hassan, S.A.; Ghaffar, R.; Jung, H.; Hossain, M.S. The Shift to 6G Communications: Vision and Requirements. *Hum. Cent. Comput. Inf. Sci.* **2020**, *10*. [CrossRef]
7. Jeong, Y.S.; Park, J.H. Security, Privacy, and Efficiency of Sustainable Computing for Future Smart Cities. *J. Inf. Process. Syst.* **2020**, *16*, 1–5.
8. Park, J.S.; Park, J.H. Future Trends of IoT, 5G Mobile Networks, and AI: Challenges, Opportunities, and Solutions. *J. Inf. Process. Syst.* **2020**, *16*, 743–749.
9. Rad, M.M.; Rahmani, A.M.; Sahafi, A.; Qader, N.N. Social Internet of Things: Vision, challenges, and trends. *Hum. Cent. Comput. Inf. Sci.* **2020**, *10*. [CrossRef]
10. Pan, F.; Wen, H.; Song, H.; Jie, T.; Wang, L. 5G security architecture and light weight security authentication. In Proceedings of the 2015 IEEE/CIC International Conference on Communications in China-Workshops (CIC/ICCC), Shenzhen, China, 2–4 November 2015; pp. 94–98.
11. Basin, D.; Dreier, J.; Hirschi, L.; Radomirović, S.; Sasse, R.; Stettler, V. A Formal Analysis of 5G Authentication. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18), Toronto, ON, Canada, 15–18 October 2018. [CrossRef]
12. Dehnel-Wild, M.; Cremers, C. *Security Vulnerability in 5G-AKA Draft*; Department of Computer Science, University of Oxford: Oxford, UK, 2018.
13. Koutsos, A. The 5G-AKA Authentication Protocol Privacy. *IEEE EuroSP* **2019**, 465–469. [CrossRef]
14. Borgaonkar, R.; Hirschi, L.; Park, S.; Shaik, A. New Privacy Threat on 3G, 4G, and Upcoming 5G AKA Protocols. *Proc. Privacy Enhancing Technol.* **2019**, *2019*, 108–127. [CrossRef]
15. Jung, J.W.; Lee, S.J. The Security Vulnerabilities of 5G-AKA and PUF-Based Security Improvement. *Converg. Sec. J.* **2019**, *19*, 3–10. [CrossRef]
16. Braeken, A.; Liyanage, M.; Kumar, P.; Murphy, J. Novel 5G Authentication Protocol to Improve the Resistance Against Active Attacks and Malicious Serving Networks. *IEEE Digit. Object Identifier* **2019**, *7*, 64040–64052. [CrossRef]
17. Hong, D.; Lee, J.K.; Kim, D.C.; Kwon, D.; Ryu, K.H.; Lee, D.G. LEA: A 128-Bit Block Cipher for Fast Encryption on Common Processors. In *Information Security Applications. WISA 2013*; Kim, Y., Lee, H., Perrig, A., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2013; Volume 8267. [CrossRef]
18. Heron, S. Advanced Encryption Standard (AES). *Netw. Secur.* **2009**, *2009*, 8–12. [CrossRef]
19. 3GPP-Technical Specification Group Services and System Aspects; 3G Security; Specification of the MILENAGE Algorithm Set: An Example Algorithm Set for the 3GPP Authentication and Key Generation Functions f1, f1*, f2, f3, f4, f5 and f5*. In Document 2: Algorithm Specification. (Release 16). 2020. Available online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2391 (accessed on 21 May 2021).
20. 3GPP-Technical Specification Group Services and System Aspects; Security architecture and procedures for 5G system; (Release 16). 2020. Available online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3169 (accessed on 21 May 2021).
21. 3GPP-Technical Specification Group Services and System Aspects; 3G Security; Security Architecture; (Release 16). 2020. Available online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2262 (accessed on 21 May 2021).
22. Concone, F.; Lo Re, G.; Morana, M. SMCP: A Secure Mobile Crowdsensing Protocol for Fog-Based Applications. *Hum. Cent. Comput. Inf. Sci.* **2020**, *10*, 28. [CrossRef]