IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 24, 2019, accepted April 25, 2019, date of publication May 6, 2019, date of current version May 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2914941

Novel 5G Authentication Protocol to Improve the Resistance Against Active Attacks and Malicious Serving Networks

AN BRAEKEN[1], MADHUSANKA LIYANAGE[2,3], (Member, IEEE), PARDEEP KUMAR[4], (Member, IEEE), AND JOHN MURPHY[2], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, Vrije Universiteit Brussel, 1050 Brussel, Belgium
[2]School of Computer Science, University College Dublin, Dublin 4, D04 V1W8, Ireland
[3]Centre for Wireless Communications, University of Oulu, 90014 Oulu, Finland
[4]Department of Computer Science, Swansea University, Swansea SA1 8EN, U.K.

Corresponding author: Madhusanka Liyanage (madhusanka@ucd.ie)

This work was supported in part by the School of Computer Science, University College Dublin, in part by the European Union in RESPONSE 5G under Grant 789658, in part by the Project and Academy of Finland in 6Genesis Flagship under Grant 318927 Project, and in part by the COST Action CA15127 (RECODIS) and CA16226 (SHELD-ON) Projects.

ABSTRACT The security of mobile communication largely depends on the strength of the authentication key exchange protocol. The 3rd Generation Partnership Project (3GPP) Group has standardized the 5G AKA (Authentication and Key Agreement) protocol for the next generation of mobile communications. It has been recently shown that the current version of this protocol still contains several weaknesses regarding user localization, leakage of activity, active attackers, and in the presence of malicious serving networks, leading to potentially major security leaks. We propose a new version of the 5G AKA protocol to overcome all the currently identified weaknesses in the protocol. In the new protocol, we replace the sequence numbers with random numbers, making it possible to drastically reduce the number of required communication phases and steps in the protocol. The usage of random numbers for the 5G AKA protocol is possible since the current Universal Subscriber Identity Modules (USIMs) are now capable of performing randomized asymmetric encryption operations. Moreover, the proposed protocol provides two additional security features, i.e., post-compromise security and forward security, not present in the current 5G AKA protocol. Finally, we evaluate the performance, both computation and communication efficiency, of the proposed AKA protocol and show its improvements compared to the current 5G AKA protocol.

INDEX TERMS 5G, authentication, key agreement, security, mobile communication, formal verification.

I. INTRODUCTION

Due to the anytime anywhere connectivity feature and popularity of smart phones, mobile communication is becoming extremely popular among many users. Telecommunication systems have already evolved from 2G to 3G and then to 4G to facilitate the continuously increasing demand in voice and data traffic. Simultaneously, the variety and requirements of supported mobile network services are ever increasing with each generation. Today the telecommunications network support a huge set of network services such as HD video streaming, online gaming, mobile health care, mobile banking, mobile cloud services and many more. Moreover, booming vertical industries such as vehicle network, Internet of Things (IoT), Augmented and Virtual Reality (AR/VR), Industrial Internet, e-health, and smart grid are demanding fast yet ubiquitous network access via telecommunication to gain a new momentum [1]. As a result, mobile networks will have to support more than 1,000 times today's traffic volume by the year 2020 [2], [3]. Therefore 5G, which is the next generation of the mobile telecommunication networks, needs to offer greater capacity, higher-speed, more dynamicity and more cost-efficiency than any generation has provided before [4].

The 3rd Generation Partnership Project (3GPP) group, responsible for the standardization of 3G, 4G, and 5G

The associate editor coordinating the review of this manuscript and approving it for publication was Adnan M. Abu-Mahfouz.

technologies, designed the Authentication and Key Agreement (AKA) protocol that aims at mutually authenticating a device equipped with a Universal Subscriber Identity Module (USIM) card with networks, and establishing keys to protect subsequent communications. This protocol is notably implemented in all 3G and 4G USIM cards and cellular networks worldwide [5].

The 5G specification from 3GPP has put a lot of emphasis on the privacy issues, relating to subscribers and their data, in accordance with the new privacy regulations such as EU (European Union)'s General Data Protection Regulation (GDPR) framework [6]–[8]. As a consequence, an updated version of the AKA protocol has been defined by 3GPP for 5G communication. The latest version v15.1.0 of Release 15 of the Technical Specification (TS) has been released in June 2018 [9]. This version has been thoroughly reviewed by means of formal verification in [10] and several major issues have been revealed. These security concerns are related to user localization, leakage of activity, impact of active attackers and also the presence of malicious Serving Networks (SN) while roaming.

Especially, roaming can happen very frequently in the 5G network because of the popularity of local 5G networks or micro 5G operators [11]–[13]. Due to this new trend, the possibility of encountering a malicious serving network is quite high in 5G. Most of these local 5G operators or micro operators do not have a high level of security similar to the main Mobile Network Operators (MNOs). Therefore, it is relatively easy to attack local 5G networks or micro 5G operators.

Our Contribution: In this paper, we propose a novel version of the 5G AKA protocol to overcome all the currently identified weaknesses in the protocol [10]. We first summarize all the currently known weaknesses and instead of giving separate ad hoc solutions for each of them, we propose a new version of the AKA protocol, able to address all the issues in a combined approach. Moreover, the new protocol not only offers resistance against these identified weaknesses, but also provides a more efficient and secure solution than the current 5G AKA protocol. In the new protocol, we replace the sequence numbers with random numbers, making it possible to drastically reduce the number of required communication phases and steps in the protocol. The usage of random numbers for the 5G AKA protocol is possible since the current USIMs are now capable of performing randomized asymmetric encryption operations [10]. Moreover, the proposed protocol provides two additional security features, i.e. post-compromise security and forward security, not present in the current 5G AKA protocol. We presents a non-monotonic logic-based verification proof for verifying the proposed scheme. Finally, we evaluate the performance of the proposed AKA protocol and compare it with the current 5G AKA protocol. Since at each step, the proposed protocol needs to perform the same type and approximately the same amount of operations as the latest 5G AKA protocol, the impact of replacing the current 5G AKA protocol with our protocol from an implementation point of view will be minimal.

The paper's outline is as follows. Section II presents an overview of relevant related work. Section III deals with preliminaries on the 5G AKA protocol. In Section IV, we present our protocol, followed by a formal security proof in Section V. An extended discussion on the design choices in Section VI is provided. The comparison in performance between our proposed protocol and the current AKA protocol is discussed in Section VII. Finally, we end the paper with some conclusions in Section VIII.

II. RELATED WORK

Several survey papers on the security attacks related to 5G networks are available in literature [14]–[18]. However, this paper focuses on the latest version of the AKA protocol, proposed for 5G. For this protocol, some of the older identified attacks still hold. For instance, there are the attacks of [19]–[21] that exploit the usage of the different types of failures, MAC based or synchronization based, to track a specific subscriber. In the attack, an old authentication challenge already received by the subscriber needs to be replayed and identification of the same subscriber is obtained in case a synchronization failure is replied.

For the more recent AKA protocol, the authors in [22] identified a new attack where the complete activity pattern of a user can be revealed. These type of attacks are also called the activity monitoring attacks. The attack exploits the lack of randomness and the use of XOR in the concealed sequence number. By cleverly choosing several time-stamps to collect data, the confidentiality of the sequence numbers is broken. As these sequence numbers are linked with the activity of the user, the attacker is able to learn the typical service consumption of targeted subscribers.

In [10], the first formal evaluation has been performed on the AKA protocol using the verification tool Tamarin [23]. This allowed a thorough security check and revealed some additional weaknesses of the newly proposed protocol. In particular, they raised the fact that several security assumptions are missing in order to meet some critical security goals as specified in the standard. For instance, the addition of a successful key confirmation round is required in order to guarantee that the required security features are obtained and that the different communication links are bound with each other.

In this paper, we propose a novel 5G authentication protocol or 5G AKA protocol to improve the resistance against all the identity attacks mentioned in above literature.

III. PRELIMINARIES

A. AKA PROTOCOL

We provide a simplified description of the AKA protocol, proposed for the latest version v15 to be used in the 5G networks [9]. We focus on the message flow and corresponding content of the core protocol, which is common for all variants of the AKA protocol.

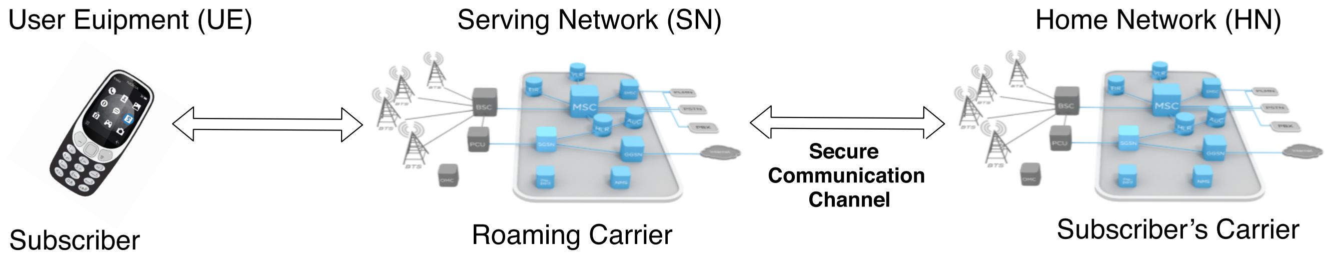

FIGURE 1. Overall communication architecture for a roaming subscriber.

The cellular network architecture is built up of the following three main components, as demonstrated in Figure 1.

- The User Equipment (UE), typically smart phones or IoT devices, is carried by the subscriber. The UE is uniquely identified by its Subscription Permanent Identifier (SUPI), stored in the Universal Subscriber Identity Module (USIM). Note that the SUPI in 5G plays the same role as the International Mobile Subscriber Identity (IMSI) in pre-5G standards. Besides the SUPI, the USIM also stores subscriber-related information and implements security functions required to run the AKA protocol. The term subscriber is used for referring to the combination of UE and USIM.
- A Home Network (HN) is responsible for the registration of their subscribers and their authentication.
- The Serving Network (SN) refers to the network to which the UE may be attached, different from its HN. This is for instance the case when the subscriber travels to locations where no base station of the HN is available, e.g. roaming.

We want to note that SNs and HNs are composed of several sub-entities (e.g. separate database and authentication server at the HN). However, this level of detail is not required to explain the main structure of the AKA protocol providing the different security features. In addition, it is also important to take into account that in the description of the protocol the UE and SN communicate over an insecure channel, while the SN and HN communicate over an authenticated channel, providing confidentiality, integrity, authenticity, and replay protection, as mentioned in the standard [9]-[TS 33.501, Sec. 5.9.3]. Pre-5G deployments such as 4G-LTE (Long Term Evolution) had also used secure communication channels, such as IPSec [24], [25] between HN and SNs. Therefore, it is recommended to use similar secure channels in 5G as well [9].

The USIM of the UE possesses several cryptographic capabilities including symmetric key encryption, MAC (Message Authentication Code) operation, EC (Elliptic Curve) multiplication and addition. In addition, it also stores the following parameters, which are integrity and confidentiality protected within the UE using a tamper resistant secure hardware component [9]-[TS 33.501, Sec. 5.2.4].

- The public key pk_{HN} of the HN.
- Subscription Permanent Identifier (SUPI). This is a unique and permanent subscriber identity. Note that in 5G, the UE never reveals this parameter in plaintext because of privacy reasons. Instead, it encrypts the SUPI together with a random number, using the public key of the HN. This encrypted value together with the identity of the HN forms the so-called Subscription Concealed Identifier (SUCI).
- K. This parameter represents a unique (linked with the SUPI) and permanent secret symmetric key between the UE and its corresponding HN.
- Sequence Number (SQN_{UE}). The sequence number is used to protect against replay attacks and is updated after each successfully received request of the SN.

On the other side, the corresponding HN of the UE also securely stores for each UE the symmetric key K and the associated sequence number SQN_{HN}. Note that SQN_{UE} and SQN_{HN} are not necessarily the same due to some potential desynchronization. In addition, the private key of the HN is also stored in an integrity and confidentiality protected environment. The functions f_1, f_1^*, f_5, f_5^* used in the authentication process are one-way keyed cryptographic functions, which are completely unrelated and or providing protection against integrity and confidentiality [10]. The exclusive-or (XOR) operation is denoted by $\oplus$. Challenge() and KeySeed() are complex Key Derivation Functions (KDFs) as specified in [9]. Note the difference in notation between xMAC and MAC in order to make a difference between the result of the computation from the user and the expected result (called xMAC). The user calculates the MAC operation and afterwards compares it with the received xMAC. The same principle holds for xRes and hxRes, SQN and xSQN. The main goal of the protocol is to establish mutual authentication between UE and SN, together with the derivation of a session key determined by the HN, called K_{SEAF}.

Figure 2 presents a high level description of the different steps in the basic AKA protocol. The secure channel between HN and SN is explicitly highlighted on the figure in order to emphasize that the sensitive parameters ($SUCI, K_{SEAF}, hxRes, \ldots$) are securely sent over this channel.

B. ASSUMED ATTACK MODELS

As mentioned before, the attack model is limited to the channel between subscribers and SNs. This channel is subject to passive attackers, able to eavesdrop, monitor, and collect data sent over the channel. However, also active attackers are

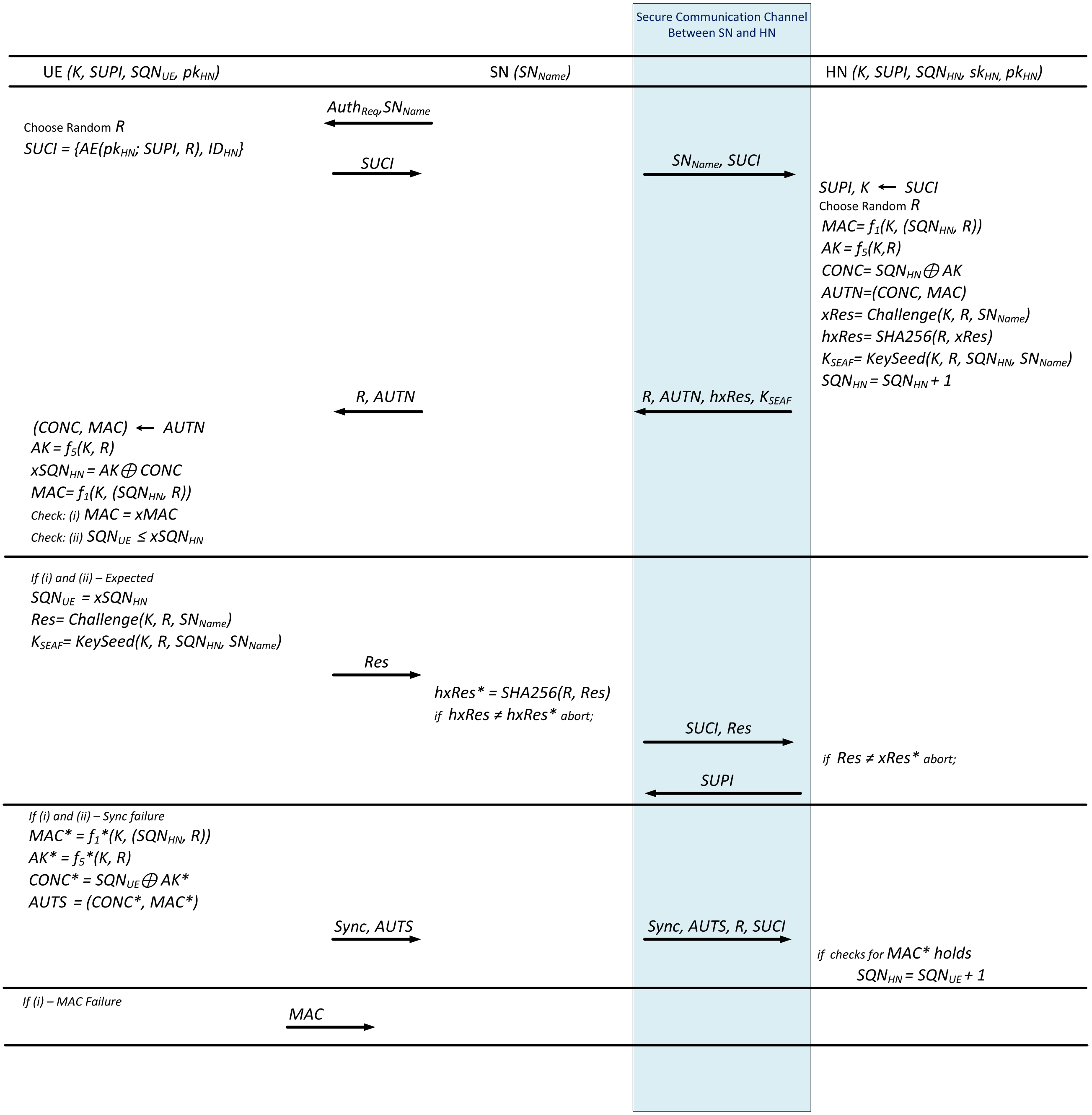
Secure Communication Channel Between SN and HN
UE $(K, SUPI, SQN_{UE}, pk_{HN})$
SN (SN_{Name})
HN $(K, SUPI, SQN_{HN}, sk_{HN}, pk_{HN})$
$Auth_{Req}, SN_{Name}$
Choose Random R
$SUCI = \{AE(pk_{HN}; SUPI, R), ID_{HN}\}$
SUCI
SN_{Name}, SUCI
$SUPI, K \leftarrow SUCI$
Choose Random R
$MAC = f_1(K, (SQN_{HN}, R))$
$AK = f_5(K,R)$
$CONC = SQN_{HN} \oplus AK$
$AUTN=(CONC, MAC)$
$xRes = Challenge(K, R, SN_{Name})$
$hxRes = SHA256(R, xRes)$
$K_{SEAF} = KeySeed(K, R, SQN_{HN}, SN_{Name})$
$SQN_{HN} = SQN_{HN} + 1$
R, AUTN
R, AUTN, hxRes, K_{SEAF}
$(CONC, MAC) \leftarrow AUTN$
$AK = f_5(K, R)$
$xSQN_{HN} = AK \oplus CONC$
$MAC = f_1(K, (SQN_{HN}, R))$
Check: (i) $MAC = xMAC$
Check: (ii) $SQN_{UE} \leq xSQN_{HN}$
If (i) and (ii) – Expected
$SQN_{UE} = xSQN_{HN}$
$Res = Challenge(K, R, SN_{Name})$
$K_{SEAF} = KeySeed(K, R, SQN_{HN}, SN_{Name})$
Res
$hxRes^* = SHA256(R, Res)$
if $hxRes \neq hxRes^*$ abort;
SUCI, Res
if $Res \neq xRes^*$ abort;
SUPI
If (i) and (ii) – Sync failure
$MAC^* = f_1^*(K, (SQN_{HN}, R))$
$AK^* = f_5^*(K, R)$
$CONC^* = SQN_{UE} \oplus AK^*$
$AUTS = (CONC^*, MAC^*)$
Sync, AUTS
Sync, AUTS, R, SUCI
if checks for MAC^* holds
$SQN_{HN} = SQN_{UE} + 1$
If (i) – MAC Failure
MAC

FIGURE 2. Steps in the 5G AKA protocol.

assumed, who can in addition intercept, manipulate, modify, replay, or even destroy data sent over the channel. While a passive attacker only listens over the channel, an active attacker needs to set up a fake base station to send and receive messages. Although, at the moment, there is no 5G-specific hardware publicly available yet, it is expected to become also easily feasible in the near future as has been the case with 4G, where open-source software and hardware became freely available [26], [27]. Therefore, the presence of active attackers needs to be considered.

Furthermore, with respect to compromised parties, we assume both malicious subscribers and malicious SNs. The last one is possible as mentioned before. Here, the attacker is able to establish and get access to an authenticated channel with the HN in order to authenticate some subscribers. Such situation can happen in case of roaming for instance. In fact the scenario of an active attacker and malicious SN is very similar. The main difference between both is that a malicious SN can be in the possession of a list of legitimate USIMs and corresponding session keys. Another possibility is that the attacker is in the possession of genuine USIMs and compromised USIMs. For those compromised subscribers, the attacker has access to all secret values stored in the USIMs, i.e., SUPI, K, and SQN.

C. SECURITY REQUIREMENTS

We here summarize the most important security features offered by 5G AKA, as explicitly identified in the formal evaluation of the scheme [9], with corresponding references to the standard.

1) Authentication between subscribers and HNs [TS 33.501, Sec. 5.1.2]. Assurance should be provided to the UE that it is connected to a SN authorized by the HN to provide services to the UE. This is an implicit authentication, i.e. automatically implied by a successful authentication.
2) Authentication between subscribers and SNs [TS 33.501, Sec. 5.1.2]. The SN is able to authenticate the SUPI and the subscribers shall be able to authenticate the SNs with the help of its HN. Note that once the SN is bound to the UE, it cannot claim to be a different SN [TS 33.501, Sec. 6.1.1.3].
3) Authentication between SNs and HNs [TS 33.501, Sec. 5.1.2]. The SN is able to authorize the UE (based on authenticated SUPU) through the subscription profile obtained from the HN.
4) Confidentiality on K_{SEAF} [TS 33.501, Sec. 3]. Knowledge of the session key does not leak information on previous or future versions of the session key.
5) Confidentiality of SUPI under passive attacks [TS 133.102, Sec. 5.1.1]. Otherwise, user location attacks would become possible.
6) Confidentiality of SQN under passive attacks [TS 133.102, Sec. C.3.2]. Otherwise, activity pattern of a targeted user can be revealed.
7) Protection against anonymity and unlinkability under passive attacks [TS 33.501, Sec. C.2], [TR 33.899, Sec. 5.2.3.8.2].

D. WEAKNESSES IN THE CURRENT VERSION

We now summarize the main weaknesses in relation with the above described security requirements in order to overcome the existing attacks (cf. Section II), present in the current 5G AKA protocol.

- Exploiting the different failure reasons, MAC or Synchronisation, as identified by several authors [19]–[21], can lead to tracking of targeted subscribers and thus can represent a major privacy threat, cf. Requirement 7.
- The SQN concealment mechanism is not sufficiently protected [22], leading to leakage of the SQNs and thus allowing activity monitoring attacks, cf. Requirements 6,7.
- As noted in [10], a successful key confirmation round is required after the execution of the protocol. In particular, the user should be ensured that the SN possesses the correct key material. The standard does not specify this additional key confirmation round, nor does it specify that the subscribers have to wait for it. This vulnerability can have serious consequences in two situations specified in the standard. First, SNs are able to initiate key change on the fly and are able to switch security contexts, including keys and parameters. Consequently, if the authentication is not fully verified by the UE, it can be possible that a malicious SN will be able to impersonate a legitimate SN, cf. Requirement 2.
- In the scenario where an attacker is able to compromise an SN, user privacy is completely broken. The malicious SN just needs to collect SUCI authentication requests of the UEs and then capture at a later stage the Res parameter sent by the UE over an insecure channel. Combining both the data $(SUCI, Res)$ and sending it to the HN, results in the retrieval of the SUPI. Although, cf. Requirement 5, confidentiality of SUPI is only required under passive attacks, it is also important to consider active attacks, taking into account the realistic threat of it as mentioned before.

IV. PROPOSED PROTOCOL

We first describe the design rationale and go then into a more deep description of the newly proposed protocol.

A. DESIGN RATIONALE

The proposed protocol is designed in such a way that the changes with respect to the original version are minimal, but are able to address the four mentioned weaknesses above.

- In order to address the first two weaknesses, we propose a variant of the protocol where SQNs are replaced by random numbers. Earlier, strong random number generation was not possible on the USIM, but in the current generation this is not an issue anymore. Current USIMs can perform randomized asymmetric encryption operations. Random numbers are now also included in the initial authentication request of the SN and the corresponding response of the subscriber. Based on these random numbers, a new random number is generated at the HN and used for the derivation of the final authentication parameters. As a consequence, there is no different failure message to be sent regarding synchronization failure, avoiding abuse in either tracking possibilities or activity revelations.

  Another advantage of removing the SQNs is that de-synchronisation attacks between user and HN by sending several authentication requests either to user or to HN are now prohibited. Such de-synchronisation attacks decrease the availability and lower user friendliness.
- To avoid the last two weaknesses, we propose to include a mechanism at the UE allowing to verify the validity of the authentication response, coming both from SN and HN (not only HN as in current version). Moreover, the SN receives from the HN encrypted information of the SUPI and the keys, which can be decrypted thanks to

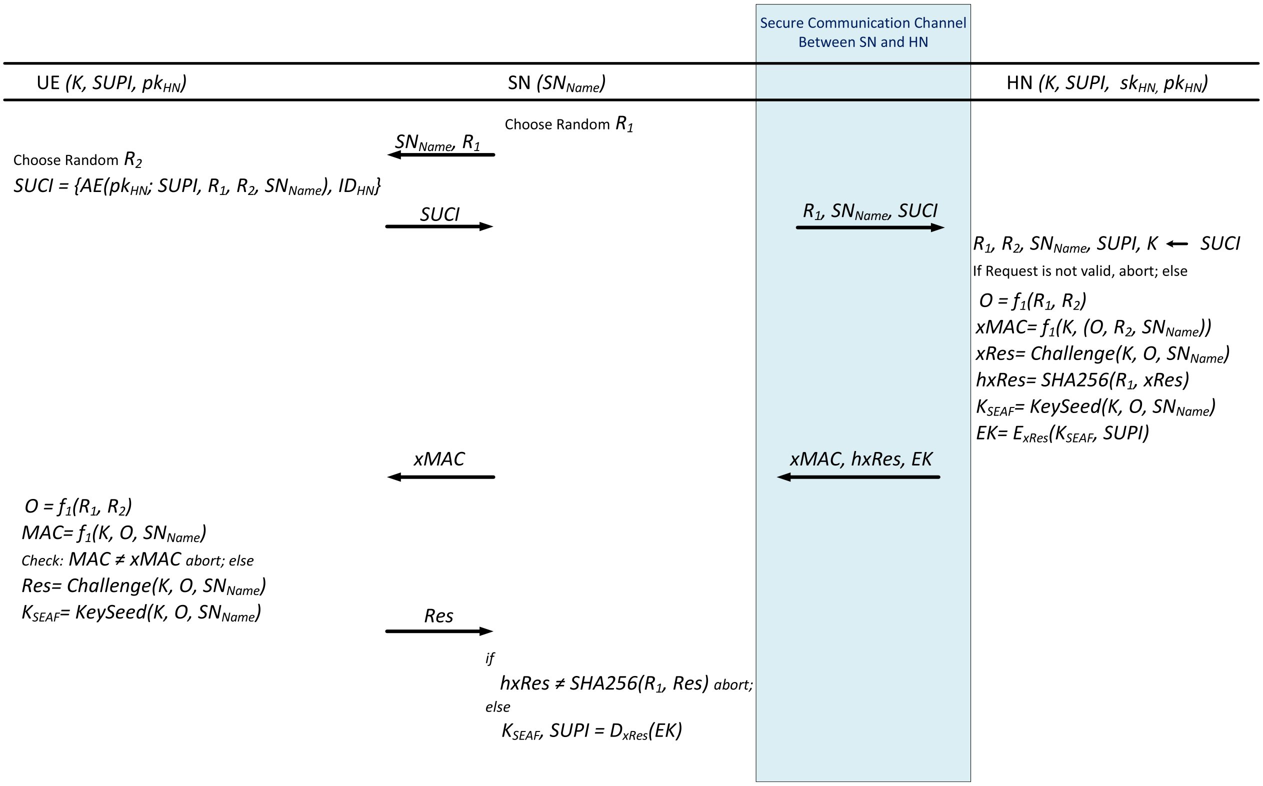
Secure Communication Channel Between SN and HN
UE (K, SUPI, pkHN)
SN (SNName)
HN (K, SUPI, skHN, pkHN)
Choose Random R1
SNName, R1
Choose Random R2
SUCI = {AE(pkHN; SUPI, R1, R2, SNName), IDHN}
SUCI
R1, SNName, SUCI
R1, R2, SNName, SUPI, K ← SUCI
If Request is not valid, abort; else
O = f1(R1, R2)
xMAC= f1(K, (O, R2, SNName))
xRes= Challenge(K, O, SNName)
hxRes= SHA256(R1, xRes)
KSEAF= KeySeed(K, O, SNName)
EK= ExRes(KSEAF, SUPI)
xMAC
xMAC, hxRes, EK
O = f1(R1, R2)
MAC= f1(K, O, SNName)
Check: MAC ≠ xMAC abort; else
Res= Challenge(K, O, SNName)
KSEAF= KeySeed(K, O, SNName)
Res
if
hxRes ≠ SHA256(R1, Res) abort;
else
KSEAF, SUPI = DxRes(EK)

FIGURE 3. Steps in the proposed protocol.

information sent from the UE after successfully approving the authentication mechanism by the UE. Consequently, this allows to shorten the current version of the protocol with additionally two phases (between HN and SN). Moreover, the key confirmation round is not required anymore, cf. weakness 4.

We now explain the process more into detail. There are two main phases, being the initialization phase and the actual authentication phase. In the authentication phase, we can distinguish another three steps. Figure 3 also depicts the different phases and steps.

B. INITIALISATION PHASE

In the initialization phase, the SN starts sending besides its name also a random number R_1. Note that this is different compared to the standard, where no random value is included in the initial request.

Upon arrival of this request, the UE chooses another random value R_2 and performs asymmetric encryption on the SUPI, along with the two random values R_1, R_2 and SN_{Name}. The combination of the output of this encryption, together with the identity of the HN now forms the new SUCI, which is sent to the SN as response on the authentication request. The SN forwards the SUCI together with its name and the random parameter R_1 to the HN.

Upon arrival of the message with the HN, after decryption of the SUCI, the HN can immediately verify the validity of the request, both with respect to session (R_1) as with the intended entities (UE by SUPI and SN by SN_{Name}).

We further note that in addition, the random value R_2, is not sent in clear text and unknown to the SN and any passive attacker. It takes over the role of the sequence number from the AKA protocol and will be used to construct the final key material.

C. AUTHENTICATION PHASE

This phase starts after verifying the validity of the request by the HN, as explained above. Also the corresponding symmetric key K of the user is retrieved. We now distinguish three major steps, according to the entity performing the computations

1) ACTIONS PERFORMED BY HN

After such positive verification, the HN derives a new parameter O, based on the input of the random parameters R_1 and R_2, defined by UE and SN respectively. In order to send the confirmation to the UE about the validity of the SN, the HN derives $xMAC = f_1(K, (O, SN_{Name}))$. Next, the HN derives two other parameters $xRes$, $hxRes$, which are later used in the protocol by the UE to prove to the SN that it is the legitimate user. This follows from the fact that the UE is the only user that is able to compute $xRes$ and by sharing $hxRes$ (a hash value on $xRes$) with the SN, it can also demonstrate that later on. Finally the session key K_{SEAF} is derived, based on input only known by the UE. The HN sends the session key and SUPI, which are encrypted in the message EK by means of the parameter $xRes$. Consequently, it is up to the UE to activate the session key and to release its identity to the SN later on in case of successful authentication. As a result the message $xMAC, hxRes, EK$ is sent to the SN, who forwards further $xMAC$ to the UE.

2) ACTIONS PERFORMED BY UE

Upon arrival of $xMAC$, the user starts computing O and $f_1(K, O, SN_{Name})$. If this result does not equal to the received $xMAC$ parameter, the session is immediately aborted. Else, the UE computes Res and K_{SEAF}. The value Res is sent to the SN.

3) ACTIONS PERFORMED BY SN

Upon arrival of the value Res, the SN verifies if $SHA256(R_1, Res)$ equals to the stored parameter $hxRes$. If so, it uses Res as decryption key to derive both the identity of the user and the session key.

V. NON-MONOTONIC LOGIC-BASED VERIFICATION

Following the literature, this section presents a non-monotonic logic-based verification proof for verifying the proposed scheme. The non-monotonic logic is also known as RUBIN logic. We chose RUBIN logic as this method is very close to actual implementation of the protocol. For more details, the reader may refer to [28], [29], [30].

A. BACKGROUND OF RUBIN LOGIC

Rubin logic integrates protocol analysis with specification and uses the notions of global sets, local sets, and actions, as follows.

1) GLOBAL SET

The Global Set comprises of various sets that are required to represent information in the protocol. The content of the global sets may change as the protocol run is progressed. Moreover, these sets are public to each principal in a protocol specification.

1) *Principal Set*: This set contains the principals who participate in a protocol.
2) *Rule Set*: This set contains inference rules for deriving new statements from existing assertions.
3) *Secret Set*: This set contains all the secrets that exist at any given time in the system.
4) *Observer set*: This set contains all principals who possibly know the secrets by listening to network traffic.

2) LOCAL SET

A Local Set comprises of various sets that are private to each participant. The local sets consist of the following:

1) *Possession set (P_i)*: This set holds all the data relevant to security, known or in possession of an entity, which includes encryption keys, public keys, and other secrets that are not publicly available. We represent the possession set by $POSS(P_i) = (poss_1, poss_2, \ldots, poss_n)$.
2) *Belief set*: This set holds all the beliefs held by a principal. For example, the beliefs about freshness, and the beliefs about the possessions of other involved principals. Denoted by $BEL(P_i) = bel_1, bel_2, \ldots bel_n$.
3) *Seen set (P_i)*: It holds plaintext message parts that P_i sees from messages sent across the network and it also contains a copy of the information as the Observer set.
4) *Behavior list (P_i)*: This is a list of elements, which are ordered. $BL = AL, bev_1, bev_2, \ldots bev_n$, here AL is an action list, which consists of zero or many actions executed by P_i and bev_k is a pair, i.e., (message, AL). Note that these messages represent basically two types: Send (P_i, message) and Receive (P_i, message). Furthermore, after every Send(.) operation, the Observer set has to be updated using the Update(.) operation. After each Update(.) operation the control passes to the next Receive(.) operation of principal, which is specified in the earlier Send(.) operation.
5) *Haskeys set (P_i)*: The Haskeys set holds keys that P_i sees, either as they are in the initial possession set, or as they appear in a message sent across the network and are added to the Seen set of P_i.

3) ACTIONS

Actions are important operations, which are utilized in the protocol specification. The actions are required to control the state of knowledge and possessions for evolved entities. For example, how a principal constructs messages, performs hash and concatenation operations, encryption and decryption, etc. Considering our requirements, following actions are defined as shown in Table 1.

4) INFERENCE RULES

We defined the inference rules that are required as per our requirements. Note that these rules are directly adopted from [28]. Moreover, in order to understand the inference rules, few notation are adopted: **X contain Y**: Y appears as a sub-message of X; **S:= F(S)** : S is replaced by the value of F(S); **X from P**: X is received from P; and **LINK(N)**: this formula basically links a response to a challenge (e.g., if a principal generates nonce) then **N** is added to belief set of the principal [28].

1) *Message-meaning rule*:

$$\frac{\{X\}k \text{ from } P_i \in POSS(P_i),\ \{P_i, P_j\} \subseteq POSS(P_i)}{BEL(P_i) = BEL(P_i) \cup \{X \in POSS(P_i)\}}$$

2) *Origin rule*:

$$\frac{X \in POSS(P_i), X \text{ contain } x1, P_j \in Observers(x1)}{x1 \text{ from} P_j \in POSS(P_i)}$$

3) *Sub-message origin rule*:

$$\frac{X \in POSS(P_i), X \text{ contain } \{x1, x2\} \text{ from } P_j}{x2 \text{ from } E_j \in POSS(P_i)}$$

TABLE 1. Actions in rubin logic.

Action	Condition	Result	Description
$Hash(h(.);X)$	$h(.), X \in POSS(P_i)$	$POSS(P_i) := POSS(P_i) \cup h(X)$	This is for hashing the data
$Encrypt$	$X, k \in POSS(P_i)$	$POSS(P_i) := POSS(P_i) \cup Xk$	The P_i encrypts own data.
$Decrypt(Xk, k)$	$Xk, k \in POSS(P_i)$	$POSS(P_i) := POSS(P_i) \cup X$	The P_i decrypts data
$Generate - Secrete.g.,(R2)$	–	$S := S \cup R2$, $Observers(R2) = P_i$, $POSS(P_i) := POSS(P_i) \cup R2, P_i$, $BEL(P_i) := BEL(P_i) \cup (R2)$	Generates a secret for an entity, when needed. Thereafter, a new secret R2, is added to secret S and the *Observers* and *Possession* sets are updated
$Check(X, Y)$	$X, Y \in POSS(P_i)$	Valid if $X = Y$, otherwise invalid	Verifies both values are correct
$Send(P_j, X)$	–	–	This action sends message X to P_j (i.e., P_i to P_j).
$Receive(P_j, X)$	–	–	This action receives message X from P_j
$Update(X)$	–	–	The purpose of update action is to update the Observer sets of all secrets.
$Abort$	–	–	Aborts the system, if protocol run is illegal

B. VERIFICATION USING RUBIN LOGIC

1) THE PROPOSED PROTOCOL SPECIFICATION

The protocol specifications are explained as follows.

Global Set:

1) Principal Set: UE,SN,HN. SN is the initiator of the protocol.
2) Rule Set: The inference rules are defined in Section V.A (refer to inference rules).
3) Secret Set: $\{K, SUPI, K_{SEAF}\}$
4) Observer Set:
   Observer(K): {UE, HN}
   Observer(R2,O): {UE, HN}
   Observer(SUPI): {UE,HN,SN}
   Observer(K_{SEAF}): {HE,UE,SN}
   Observer(xRes): {HE, UE, SN}
   Observer(Pr_{HN}): {HN}, Here Pr_{HN} is the private key of HN.

Local Set: Now, we define the local sets for each entity. Note that as the SN is initiating the communication, we start with SN as follows.

- Principal SN
  POSS(SN): $\{SN_{name}\}$
  BEL(SN): {#R1}
  BL(SN) =
  SN1: Generate-nonce($R1$)
  SN2: Send(UE, $\{SN_{name}, R1\}$)
  SN3: Update($SN_{name}, R1$)
  SN4: Receive(UE,$\{SUCI\}$)
  SN5: Send(HN, $\{R1, SN_{name}, SUCI\}$)
  SN6: Update($R1, SN_{name}, SUCI$)
  SN7: Receive(HN, $\{xMAC, hxRes, EK\}$)
  SN8: Send(UE,$\{xMAC\}$)
  SN9: Update($xMAC, hxRes, EK$)
  SN10: Receive(UE,$\{Res\}$)
  SN11: HxRex$'$ ← SHA256($R1, Res$)
  SN12: Decrypt(EK)
  SN13: Update(K_{SEAF},SUPI,Res)
- Principal UE
  POSS(UE): $\{K, R2, SUPI, PK_{HN}, ID_{HN}\}$
  BEL(UE): {K, R2}
  BL(UE) =
  UE1: Receive(SN, $\{SN_{name}, R1\}$)
  UE2: Generate-nonce($R2$)
  UE3: SUCI ← Concat($Encrypt_{PKHN}(SUPI, R1, R2, SN_{name}), ID_{HN}$)
  UE4: Send(SN, $\{SUCI\}$)
  UE5: Update($SN_{name}, R1, R2, SUCI$)
  UE6: Receive(SN, $\{xMAC\}$)
  UE7: O ←$f1(R1, R2)$
  UE8: MAC ←$f1(K1, O, SN_{name})$
  UE9: Check($MAC, xMAC$)
  UE10: Res ← $Challenge(K, O, SN_{name})$
  UE11: K_{SEAF} ← $KeySeed(K, O, SN_{name})$
  UE13: Send(SE, $\{Res\}$)
  UE12: Update(O, K_{SEAF})
- Principal HN
  POSS(HN): $\{K, R2, SUPI, Pr_{HN}, ID_{HN}\}$
  BEL(HN): {K, O}
  BL(HN) =
  HN1: Receive(HN, $\{R1, SN_{name}, SUCI\}$)
  HN2: Decrypt($SUCI$)
  HN3: Check($SN_{name}, SUPI$)
  HN4: O ←$f1(R1, R2)$
  HN5: xMAC ←$f1(K1, O, SN_{name})$
  HN6: xRes ← $Challenge(K, O, SN_{name})$
  HN7: HxRex ← $SHA256(R1, xRes)$
  HN8: K_{SEAF} ← $KeySeed(K, O, SN_{name})$
  HN9: EK ← $Encrypt(K_{SEAF}, SUPI)$
  HN10: Send(SN, $\{xMAC, hxRes, EK\}$)
  HN11: Update($xMAC, hxRes, EK$)

2) THE PROTOCOL VERIFICATION

In the proposed scheme, the SN initiates the communication, and then the action in BL(SN) are performed. Firstly, SN1 and SN2 actions in BL(SN) are performed. These two actions denote that SN generates R1 and sends SN_{name}, R1 to UE. Next, (UE1)-(UE4) actions in BL(UE) are executed to generate SUCI (=$AE_{PKHN}(SUPI, R1, R2, SN_{name})$) and to send SUCI to SN. However, the local sets of pricipal UE are changed as described below:

- POSS(UE) = $\{K, R1, R2, SUCI, SUPI, PK_{HN}, ID_{HN}, SN_{name}\}$
- BEL(UE) = $\{R1, R2, LINK(R1)\}$

Now the global sets are modified as follows:

- Secret set: $\{K, SUPI, K_{SEAF}, R1, R2\}$
- Observer sets:
  Observer(R1): {U, SN}
  Observer(R2): {U}

After the (UE1)–(UE4) actions are finished, SN starts the actions in BL(SN) with the received message (i.e., $SUCI$) from UE. It appends own $R1, SN_{name}$ to $SUCI$ and sends the message ($R1, SN_{name}, SUCI$) to HN, and updates its own table. The local sets of SN are changed as follows.

- POSS(SN) = $\{R1, SUCI, SN_{name}\}$
- BEL(SN) = $\{R1\}$

In this case, the global sets remain unchanged and so the secret set is the same.

Upon receiving the message from SN, HN performs the actions in BL(HN) on $R1, SN_{name}, SUCI$. The actions (HN1)-(HN3) are performed to check the correctness of $SN_{name}, SUCI$. If the condition succeeds, the (HN4)-(HN10) are computed to make the values $O, xMAC, xRes, K_{SKAF}, EK$, and to send the message $\{xMAC, hxRes, EK\}$ to SN. Finally, HN updates its own table in the action HN11. The local sets of HN are changed as follows.

- POSS(HN) = $\{K, R2, SUPI, Pr_{HN}, ID_{HN}, xMAC, hxRes, EK, K_{SKAF}\}$, $\{R1, SUCI, SN_{name}$ from SN $\}$
- BEL(HN) = $\{R1, R2, K_{SKAF}, LINK(O)\}$

Now the global sets are modified as follows:

- Secret set: $\{K, SUPI, K_{SEAF}, R1, R2, O\}$
- Observer sets:
  Observer(K): {HN}
  Observer(R2, O): {HN}
  Observer(K_{SEAF}): {HN}

Upon receiving the message from HN, SN performs the actions (SN8) and (SN9) in BL(SN). It sends $xMAC$ to UE and updates its own table. Now, the local sets of SN are changed as follows.

- POSS(SN) = $\{R1, SUCI, SN_{name}\}$, $\{xMAC, hxRes, EK\}$ from HN
- BEL(SN) = $\{R1, hxRes, EK\}$

In this case, the global sets remain unchanged and so the secret set is the same.

After the actions (SN8) and (SN9) are finished, the (UE6)-(UE10) actions are performed to verify MAC and challenge. If the conditions are not true then the system will be aborted. Otherwise, the (UE11) action computes K_{SEAF} and the (UE12) action sends Res to SN. Finally, UE updates its own table, and the local sets of UE are changed as follows.

- POSS(UE) = $\{K, R2, SUPI, Pr_{HN}, ID_{HN}, xMAC, hxRes, EK, K_{SKAF}\}$
- BEL(UE) = $\{K, K_{SKAF}, LINK(R2, R1)\}$

Now the global sets are modified as follows:

- Secret set: $\{K, SUPI, K_{SEAF}, R2, O\}$
- Observer sets:
  Observer(K): {UE}
  Observer(R1, R2, O): {UE}
  Observer(K_{SEAF}): {UE}

Upon finishing the action (UE12), the actions (SN10)-(SN12) are performed in BL(SN) with incoming message (Res). In action (SN11), if $hxRex$ is not verified then the system will be aborted. Otherwise, the action (SN12) and (SN13) are executed to obtain $SUPI, K_{SEAF}$ and to update its own table with ($SUPI, K_{SEAF}, Res$), respectively. Finally, the local sets of SN are changed as follows.

- POSS(SN) = $\{SUPI, xMAC, hxRes, EK, K_{SKAF}\}$
- BEL(SN) = $\{SUPI, K_{SKAF}\}$

Now the global sets are modified as follows:

- Secret set: $\{SUPI, K_{SEAF}\}$
- Observer sets:
  Observer(SUPI): {SN}
  Observer(K_{SEAF}): {SN}

This verification implies that:

- R2 and O are fresh for each session, and are known by the legitimate UE and HN.
- K is only known to UE and HN.
- UE, SN, and HN are mutually authenticated during the protocol execution.
- SUPI is only possessed by the UE, HN and SN.
- The session key K_{SEAF} is only possessed by the UE, HN and SN. Note that K_{SEAF} is an independent key for each session as it is computed over O (i.e., using random numbers). This also implies that if a long term key is compromised then the past session cannot be determined. Hence, it achieves forward secrecy. Likewise, it also achieves the post-compromised security.

This verifies the security claims for the proposed scheme.

VI. DISCUSSION ON PROPOSED PROTOCOL

We first discuss the differences between the proposed protocol and the current standard AKA protocol and why the identified weaknesses of AKA do not hold in our proposed protocol. Then we discuss the overall security strength, followed by some additional security features that our protocol is able to offer.

A. DIFFERENCES WITH AKA PROTOCOL

A key confirmation round is not needed in the proposed protocol because of the following reasons.

- The HN can verify the uniqueness of the session in time and bound to the entities SN and UE, thanks to the fact that the parameter SUCI also contains the random value R_1 and SN_{Name}.

In the current AKA protocol, in case of an impersonation attack on the SN, the UE and the HN are not aware of any problem unless a key confirmation round is performed [10]. Consequently, it is much more interesting to build such protection already from the beginning of the protocol.

- Due to the integration of SN_{Name}, R_1, R_2 in the MAC parameter, the UE can link the validation with the original session. Note that in the AKA protocol, the MAC parameter does not involve any info related to the SN, and again the UE and the HN are not aware of any problem unless a key confirmation round is performed [10].
- The SN can only activate the session key and the identity of the subscriber after receiving a positive response of the UE with parameter Res, who checked the validity of the session. As a consequence, the UE is sure that only the legitimate SN (the one verified by the HN) is able to derive its identity and session key. In the AKA protocol, the parameter Res was used to request the SUPI of the HN, without binding this request to one of the parameters used in the session. As a consequence, it is very easy to impersonate an SN in this way.

The usage of sequence numbers is not only impractical, requires more storage, but also leads to several vulnerabilities, like possibility of activity monitoring attacks or even traceability of a targeted subscriber. Therefore, we replaced these numbers by random numbers. However, the main reason to use sequence numbers in AKA was to avoid replay attacks. Consequently, we explicitly need to check resistance against replay attacks. Resistance is obtained because of the following reasons.

- The key material $EK, hxRes, xMAC$, established by the HN, is bound to a fixed session, determined by the initial parameters R_1 and R_2, independently defined by SN and UE respectively.
- The response Res of the UE is linked to the same session as it includes the parameters R_1 and R_2.

B. OVERALL SECURITY STRENGTH

We now discuss the strength of our protocol against the seven most important security requirements, defined in the standard and described in Section III-C.

- First of all, with respect to authentication (Requirements 1-3), thanks to the session binding of the UE with the SN in the SUCI message, the inclusion of the SN name in the MAC parameter, and the encrypted session key at the SN, mutual authentication is obtained in a direct way, i.e. without the need of an additional key confirmation round.
- Confidentiality on K_{SEAF} (Requirement 4) is obtained after a successful authentication of the UE as the UE shares at the end of the protocol the symmetric key to decrypt the previously received message of the HN containing the session key.
- Confidentiality of SUPI (Requirement 5) is now also achieved under active attacks (instead of only passive attacks as indicated in the standard) since it is treated in the same way as the session key (see Requirement 4).
- Confidentiality of SQN (Requirement 6) is no longer relevant as we do not use a parameter indicating the frequency of authentication requests.
- Protection against anonymity and unlinkability (Requirement 7) is now also achieved under active attacks (instead of only passive attacks as indicated in the standard) since it is related to the confidentiality of the SUPI, which is only released at the end of a successful authentication protocol by the authenticated SN. In case the authentication protocol is unsuccessful, the identity cannot be revealed.

C. ADDITIONAL SECURITY FEATURES

We distinguish, from a security point of view, at least two additional strengths, i.e. provide forward security and protection against exhaustive key search attack, of our proposed version, compared to the current 5G AKA protocol.

- Forward security means if the long-term secret (together with the party's current session key and all other secret states) is corrupted, then the past sessions are still secure as the previously established session keys cannot be revealed. Post-compromise security protects sessions against earlier compromise, i.e. if the long-term key (together with the party's current session key and all other secret states) is corrupted, the future messages can still be secure (if the previously corrupted party somehow becomes "clean" again) [31].

  It is clear that both features are not established by the current version of the AKA protocol in case the long-term key material of the UE is leaked, being the symmetric key K, SQN and $SUPI$, since K_{SEAF} is constructed based on these parameters together with the random value R sent over the insecure channel. These properties also hold if the database at the HN, storing all the secret key material of the different UEs is hacked.

  Instead, in our proposed scheme we are able to offer part of these features. There are two situations to distinguish:

  - The attacker is in possession of the key material of the UE. In this case, our proposed protocol is able to offer both forward and post compromise security as the session key is built using the ephemeral random variable R_2, which is encrypted by means of the public key of the HN. Note that this can be a realistic scenario if the attacker for instance is capable in breaking the tamper proof resistant storage capabilities of the device.
  - The attacker is in possession of the key material of the HN, including its long term secret key. In this case, forward security is broken. However, post-compromise security is still obtained when only changing the private-public key pair of the HN since the SUPI (including its key material) is not revealed, unless in the last step of the protocol only by the SN. However, this last scenario is very unlikely to

occur as both types of key material, private key and database with user data are stored and protected at different places.

- In an exhaustive key search attack, all possible values of the keys are tried. If an attacker collects in the current AKA scheme the variables (SN_{Name}, R, Res) by eavesdropping on the insecure channel, it can verify the validity of a guessed value K^* of K by checking if $Challenge(K^*, R, SN_{Name})$ equals to Res. If some part of the key is leaked somehow, for instance by a certain side channel attack, the attack can become feasible.

  In our proposed scheme, the parameters $xMAC, Res$ leaked in the insecure channel involve besides the secret key K, also the random variable R_2. Consequently, the complexity of an exhaustive search attack is in this case drastically higher and thus infeasible, especially as R_2 changes in each collected pair.

VII. PERFORMANCE ANALYSIS

Let us compare the 5G AKA protocol with our proposed protocol, both from the point of view of computation and communication.

A. COMPUTATIONAL COMPLEXITY

With respect to computational complexity, we notice that in both protocols the same type and approximately the same amount of operations are performed. The most important noticeable differences are at the HN and SN. At the HN, there is a one way function to obtain O instead of a random generation of R and an additional symmetric key encryption to encrypt the session key and identity of the subscriber. At the SN, there is an additional symmetric key decryption. As both HN and SN are powerful servers, we can conclude that the resulting impact of these differences is negligible.

B. COMMUNICATION COMPLEXITY

The advantage of our protocol is most visible with respect to the communication complexity. We consider three different scenarios to clarify this statement.

- **Successful authentication:** In this scenario, the AKA protocol requires an additional communication phase between SN and HN compared to our protocol, followed by a key confirmation round. Although this key confirmation round is not specifically described (not mentioned in the standard), it should involve at least one communication round between SN and UE and SN and HN in order to overcome the third and fourth identified weakness respectively, as described in Section III-D.
- **Invalid request of SN to HN:** As we assume the existence of active attackers, it is possible that a malicious SN intercepts a legitimate SUCI message and forwards this to the HN, using its own name.

  In the AKA protocol, the complete protocol will still be executed. The HN will construct key material using the incorrect SN name, forward the material to the malicious SN, from which part of it is again forwarded to the UE. The UE needs to perform all the computations and transmit its response to the SN. Only during the key confirmation round, this will be noticed by the UE.

  In our proposed protocol, it will even not be possible to construct key material by the HN, which is not linked with the SN to which the UE wants to connect as the SUCI message inherently contains the SN name.
- **Invalid request of SN to UE:** As we assume the existence of active attackers, it is possible that a malicious SN intercepts a legitimate $R, AUTN$ message in the current 5G AKA protocol and forwards this to the UE.

  In the AKA protocol, the UE still needs to perform all operations and will only notice the problem after the key confirmation round.

  With our proposed protocol the UE notices the problem and aborts the protocol in the MAC computation (second step).

VIII. CONCLUSION

This paper represents some fundamental changes to the current AKA protocol in order to address both well-known weaknesses and recently discovered weaknesses. An important change in the protocol is the replacement of the sequence numbers by random numbers. In addition, we give full control at the subscriber for releasing the identity and secret key session with the SN after a successful verification of data provided by the HN. As a consequence, this change provides also confidentiality of the SUPI and protection against anonymity and unlinkability, in the presence of active attackers, instead of only passive attacks as stated in the standard.

The proposed protocol possesses less communication phases and does not require an additional key confirmation phase in order to ensure full authentication of all entities on the key material. The same amount of operations is required at the side of the UE, thus no additional complexity is imported. Moreover, there is no secure storage required of the sequence number and simple de-synchronisation attacks are avoided.

Consequently, we think our proposed protocol will be a good candidate to replace the current AKA protocol. As the main structure and type of operations are similar as in AKA, replacement would not involve too many changes.

REFERENCES

[1] S. Li, L. Da Xu, and S. Zhao, "5G Internet of Things: A survey," *J. Ind. Inf. Integr.*, vol. 10, pp. 1–9, Jun. 2018.

[2] N. Panwar, S. Sharma, and A. K. Singh, "A survey on 5G: The next generation of mobile communication," *Phys. Commun.*, vol. 18, pp. 64–84, Mar. 2016.

[3] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for Internet of Things realization," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2961–2991, 4th Quart., 2018.

[4] M. Liyanage, I. Ahmad, A. B. Abro, A. Gurtov, and M. Ylianttila, *A Comprehensive Guide to 5G Security*. Hoboken, NJ, USA: Wiley, 2018.

[5] M. S. Bargh, R. J. Hulsebosch, E. H. Eertink, J. Laganier, A. R. Zugenmaier, and A. Prasad, "UMTS-AKA and EAP-AKA inter-working for fast handovers in all-IP networks," in *Proc. IEEE Globecom Workshops*, Nov. 2007, pp. 1–6.

[6] *Regulation (EU) 2016/679 of the European Parliament and of the Council*, European Union, REGULATION (EU), 2016, p. 679.

[7] R. Borgaonkar, L. Hirschi, S. Park, and A. Shaik, *New Privacy Threat on 3G, 4G, and Upcoming 5G AKA Protocols*. Warsaw, Poland: Sciendo, 2019, pp. 1–20.

[8] M. Liyanage, J. Salo, A. Braeken, T. Kumar, S. Seneviratne, and M. Ylianttila, "5G Privacy: Scenarios and Solutions," in *Proc. IEEE 5G World Forum (5GWF)*, Jul. 2018, pp. 197–203.

[9] *Security Architecture and Procedures for 5G System*, document 3GPP, TS 33.501. Accessed: Jan. 26, 2019. [Online]. Available: http://www.3gpp.org/DynaReport/33501.htm

[10] D. Basin, J. Dreier, L. Hirschi, S. Radomirovic, R. Sasse, and V. Stettler, "A Formal Analysis of 5G Authentication," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.* New York, NY, USA: ACM, 2018, pp. 1383–1396.

[11] A. Prasad, Z. Li, S. Holtmanns, and M. A. Uusitalo, "5G micro-operator networks—A key enabler for new verticals and markets," in *Proc. 25th Telecommun. Forum (TELFOR)*, Nov. 2017, pp. 1–4.

[12] P. Ahokangas *et al.*, "Future micro operators business models in 5G," *Bus. Manage. Rev.*, vol. 7, no. 5, p. 143, 2016.

[13] Y. Siriwardhana, P. Porambage, M. Liyanage, J. S. Walia, M. Matinmikko-Blue, and M. Ylianttila, "Micro-operator driven local 5G network architecture for industrial internet," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Nov. 2018, pp. 1–8.

[14] S. Behrad, E. Bertin, and N. Crespi, "Securing authentication for mobile networks, a survey on 4G issues and 5G answers," in *Proc. 21st Conf. Innovation Clouds, Internet Netw. Workshops (ICIN)*, Feb. 2018, pp. 1–8.

[15] D. Rupprecht, A. Dabrowski, T. Holz, E. Weippl, and C. Pöpper, "On security research towards future mobile network generations," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2518–2542, 3rd Quart., 2018.

[16] I. Ahmad, T. Kumar, M. Liyanage, J. Okwuibe, M. Ylianttila, and A. Gurtov, "Overview of 5G security challenges and solutions," *IEEE Commun. Standards Mag.*, vol. 2, no. 1, pp. 36–43, Mar. 2018.

[17] G. Mantas, N. Komninos, J. Rodriuez, E. Logota, and H. Marques, "Security for 5G communications," in *Fundamentals of 5G Mobile Networks*. London, U.K.: Wiley, 2015.

[18] I. Ahmad, T. Kumar, M. Liyanage, J. Okwuibe, M. Ylianttila, and A. Gurtov, "5G security: Analysis of threats and solutions," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Sep. 2017, pp. 193–199.

[19] M. Arapinis *et al.*, "New privacy issues in mobile telephony: Fix and verification," in *Proc. ACM Conf. Comput. Commun. Secur.* New York, NY, USA: ACM, 2012, pp. 205–216.

[20] C. Hahn, H. Kwon, D. Kim, K. Kang, and J. Hur, "A privacy threat in 4th generation mobile telephony and its countermeasure," in *Proc. Int. Conf. Wireless Algorithms, Syst., Appl.* Harbin, China: Springer, 2014, pp. 624–635.

[21] P.-A. Fouque, C. Onete, and B. Richard, "Achieving better privacy for the 3GPP AKA protocol," in *Proc. Privacy Enhancing Technol.*, vol. 4, pp. 255–275, Oct. 2016.

[22] R. Borgaonkar, L. Hirschi, S. Park, and A. Shaik. *New Privacy Threat on 3G, 4G, and Upcoming 5G AKA Protocols*. Accessed: Jan. 26, 2019. [Online]. Available: https://eprint.iacr.org/2018/1175.pdf

[23] S. Meier, B. Schmidt, C. Cremers, and D. Basin, "The TAMARIN prover for the symbolic analysis of security protocols," in *Proc. Int. Conf. Comput. Aided Verification*. Saint Petersburg, Russia: Springer, 2013, pp. 696–701.

[24] M. Liyanage, M. Ylianttila, and A. Gurtov, "A case study on security issues in LTE backhaul and core networks," in *Case Studies in Secure Computing: Achievements and Trends*, vol. 1. Boca Raton, FL, USA: CRC Press, 2014, p. 167.

[25] M. Liyanage, P. Kumar, M. Ylianttila, and A. Gurtov, "Novel secure VPN architectures for LTE backhaul networks," *Secur. Commun. Netw.*, vol. 9, no. 10, pp. 1198–1215, 2016.

[26] N. Golde, K. Redon, and J.-P. Seifert, "Let me answer that for you: Exploiting broadcast information in cellular networks," in *Proc. USENIX Secur. Symp.*, 2013, pp. 33–48.

[27] A. Shaik, R. Borgaonkar, N. Asokan, V. Niemi, and J.-P. Seifert, "Practical attacks against privacy and availability in 4G/LTE mobile communication systems," in *Proc. 23nd Annu. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2015, pp. 1–16.

[28] A. D. Rubin and P. Honeyman, "Nonmonotonic cryptographic protocols," in *Proc. Comput. Secur. Found. Workshop VII*, Jun. 1994, pp. 100–116.

[29] Y. Choi, D. Lee, J. Kim, J. Jung, J. Nam, and D. Won, "Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 14, no. 6, pp. 10081–10106, 2014.

[30] P. Kumar, A. J. Choudhury, M. Sain, S.-G. Lee, and H.-J. Lee, "RUASN: A robust user authentication framework for wireless sensor networks," *Sensors*, vol. 11, no. 5, pp. 5020–5046, 2011.

[31] K. Cohn-Gordon, C. Cremers, and L. Garratt, "On post-compromise security," in *Proc. IEEE 29th Comput. Secur. Found. Symp. (CSF)*, Jun./Jul. 2016, pp. 164–178.

AN BRAEKEN received the M.Sc. degree in mathematics from the University of Gent, in 2002, and the Ph.D. degree in engineering sciences from the Research Group Computer Security and Industrial Cryptography (COSIC), KU Leuven, in 2006. In 2007, she became a Professor with the Erasmushogeschool Brussel (currently since 2013, Vrije Universiteit Brussel) with the Industrial Sciences Department. Prior to joining the Erasmushogeschool Brussel, she was with the management consulting company called Boston Consulting Group (BCG), for almost two years. Her current interests include security and privacy protocols for the IoT, cloud and fog, blockchain, and 5G security. She is the coauthor of over 120 publications. She has been a member of the program committee for numerous conferences and workshops (IOP2018, EUC 2018, and ICNS 2018) and a member of the editorial board for the *Security and Communications* magazine. She has also been a member of the organizing committee for the IEEE Cloudtech 2018 Conference and the Blockchain in the IoT Workshop at Globecom 2018. In addition, since 2015, she has been a reviewer of several EU proposals and ongoing projects, submitted under the programs of H2020, Marie Curie, and ITN. She has cooperated and coordinated over 12 national and international projects. She has been the STSM Manager of the COST AAPELE Project (2014–2017) and is currently with the Management Committee of the COST RECODIS Project (2016–2019).

MADHUSANKA LIYANAGE (S'07–M'16) received the Ph.D. degree in communication engineering from the University of Oulu, Oulu, Finland. He is currently a Marie Curie Fellow of the School of Computer Science, University College Dublin, Ireland. He is also an Adjunct Professor with the University of Oulu. From 2011 to 2012, he was a Research Scientist at I3S Laboratory and Inria, Sophia Antipolis, France. Moreover, he was a Visiting Research Fellow of computer science and engineering with The University of Oxford; Data61; CSIRO; Sydney Australia; Infolabs21; Lancaster University, U.K., and computer science and engineering, The University of New South Wales (2016–2018). He is the coauthor of over 50 publications, including two edited books with Wiley. His research interests are SDN, the IoT, Block Chain, mobile, and virtual network security. He is a member of the ICT. He is also a Management Committee Member of the EU COST Action IC1301, IC1303, CA15107, CA15127, and CA 16226 projects. He has served as a Technical Program Committee Member at the EAI M3Apps 2016, 5GU 2017, EUCNC 2017, EUCNC 2018, MASS 2018, 5G-WF 2018, MCWN 2018, and IEEE WCNC 2018 conferences, and the Technical Program Co-Chair of the SecureEdge Workshop at the IEEE CIT 2017, MEC-IoT Workshop at 5GWF 2018, and Blockchain in IoT workshop at Globecom 2018 conferences. He has also served as the Session Chair in a number of other conferences, including the IEEE WCNC, EAI CROWNCOM, EAI 5GU, IEEE CIT, IEEE PIMRC, EAI BODYNET, and IEEE 5GWF. He is also the Demo Chair of the IEEE WCNC 2019.

PARDEEP KUMAR (M'13) received the B.E. degree in computer science from Maharishi Dayanand University, Haryana, India, in 2002, the M.Tech. degree in computer science from Chaudhary Devilal University, Haryana, in 2006, and the Ph.D. degree in ubiquitous computing from Dongseo University, Busan, South Korea, in 2012. From 2012 to 2018, he held postdoctoral positions at the Department of Computer Science, Oxford University, Oxford, U.K. (2016–2018), at the Department of Computer Science, The Arctic University of Norway, Tromso, Norway (2015–2016), and at the Centre for Wireless Communications and the Department of Communications Engineering, University of Oulu, Finland (2012–2015). He is currently a Lecturer/Assistant Professor with the Department of Computer Science, Swansea University, Swansea, U.K. His research interests include security in sensor networks, smart environments, cyber-physical systems, The Internet of Things, and 5G networks.

JOHN MURPHY received the Ph.D. degree from Dublin City University. He is currently a Professor of computer science with University College Dublin (UCD) and the Co-Director of the Performance Engineering Lab (PEL). He has over 25 years of expertise in performance engineering issues around networks and large distributed systems. He has published over 200 peer-reviewed journal and conference papers. He has graduated 24 Ph.D. students and has mentored 14 postdoctoral fellows. He has secured external competitive funding of over 6M granted, where he was the budget holder. He has 15 years of experience in successful industrial engagement with numerous companies, including IBM, Eircom, Ericsson, Telcordia, NEC, British Telecom, and Alcatel-Lucent, and this experience will be of a great benefit to him during his fellowship. His industry engagement has resulted in the securing of 2M in commercialization funding support, filing of six patents, and the successful spinout of four companies: Forkstream, Crovan, Pilot Photonics, and Logentries, which won the 2016 Mature Spin-Out Company Impact Award with UCD. Logentries was acquired for $68M by Rapid7 in 2015. He is an Editor of the *Communications Surveys and Tutorials* (since 2012) and the *Telecommunications Systems* (since 2008). He has served on the Editorial Board of the IEEE COMMUNICATIONS LETTERS (2008–2012) and the *IET Communications* (2006–2010).