OPEN ACCESS

UNIVERSITY OF DUNDEE

University of Dundee

A Formal Analysis of 5G Authentication

Basin, David; Dreier, Jannik ; Hirschi, Lucca ; Radomirovi, Saša; Sasse, Ralf; Stettler, Vincent

Formal Analysis of 5G Authentication

David Basin
ETH Zurich
basin@inf.ethz.ch

Jannik Dreier
Universite de Lorraine
jannik.dreier@loria.fr

Lucca Hirschi
ETH Zurich
lucca.hirschi@inf.ethz.ch

Saša Radomirović
University of Dundee
s.radomirovic@dundee.ac.uk

Ralf Sasse
ETH Zurich
ralf.sasse@inf.ethz.ch

Vincent Stettler
ETH Zurich
svincent@student.ethz.ch

ABSTRACT

Mobile communication networks connect much of the world's population. The security of every user's calls, SMSs, and mobile data, depends on the guarantees provided by the Authenticated Key Exchange protocols used. For the next-generation network (5G), the 3GPP group has standardized the 5G AKA protocol for this purpose.

We provide the first comprehensive formal model of a protocol from the AKA family: 5G AKA. We also extract precise requirements from the 3GPP standards defining 5G and we identify missing security goals. Using the security protocol verification tool Tamarin, we conduct a full, systematic, security evaluation of the model with respect to the 5G security goals. Our evaluation automatically identifies the minimal security assumptions required for each security goal and we find that some critical security goals are not met, except under additional assumptions missing from the standard. Finally, we make explicit recommendations with provably secure fixes for the attacks and weaknesses we found.

CCS CONCEPTS

• **Security and privacy** → **Formal security models**; **Logic and verification**; **Mobile and wireless security**;

KEYWORDS

5G standard, authentication protocols, AKA, symbolic verification, formal analysis

1 INTRODUCTION

Two thirds of the world's population, roughly 5 billion people, are mobile subscribers [24]. They are connected to the mobile network via their USIM cards and are protected by security mechanisms standardized by the 3rd Generation Partnership Project (3GPP) group. Both subscribers and carriers expect security guarantees from the mechanisms used, such as the confidentiality of user data (*e.g.,* voice and SMS) and that subscribers are billed only for the services they consume. Moreover, these properties should hold in an adversarial environment with malicious base stations and users.

One of the most important security mechanisms in place aims at mutually authenticating subscribers and their carriers and establishing a secure channel to protect subsequent communication. For network generations (3G and 4G) introduced since the year 2000, this is achieved using variants of the Authentication and Key Agreement (AKA) protocol, standardized by 3GPP. These protocols involve the subscribers, the Serving Networks (*SNs*) that have base stations in subscribers' vicinity, and Home Networks (*HNs*) that correspond to the subscribers' carriers. The protocols aim to enable the subscribers and the *HNs* to mutually authenticate each other and to let the subscribers and the *SNs* establish a session key.

Next-Generation (5G). Since 2016, the 3GPP group has been standardizing the next generation of mobile communication (5G), with the aim of increasing network throughput and offering an ambitious infrastructure encompassing numerous new use cases. The 5G standard will be deployed in two phases. The first phase (Release 15, expected by June 2018) will address the most critical requirements needed for commercial deployment and will be the basis for the first deployment. The second phase (Release 16, to be completed by March 2020) will address all remaining requirements.

In March 2018, 3GPP published version `1.0.0` of Release 15 of the Technical Specification (TS) defining the 5G security architecture and procedures [4]. The authentication in 5G Release 15 is based on new versions of the AKA protocols, notably the new *5G AKA* protocol, which enhances the AKA protocol currently used in 4G (EPS AKA) and which supposedly provides improved security guarantees. This raises the following question: *What are the security guarantees that 5G AKA provides and under which threat model and security assumptions?*

Formal Methods. In this paper, we give a precise answer to the above question. Namely, we apply formal methods and automated verification in the *symbolic model* to determine whether precise, fine-grained security guarantees are met by 5G AKA. Formal methods have already proved extremely valuable in assessing the security of large-scale, real-world security protocols such as TLS 1.3 [10, 16, 18], messaging protocols [26], and entity authentication protocols [7]. Symbolic approaches, in particular, allow one to automate reasoning using techniques including model-checking, resolution, and rewriting. Examples of mature verification tools along these lines are Tamarin [30], ProVerif [11], and DeepSec [14].

Unfortunately, the AKA protocols, and *a fortiori* 5G AKA, feature a combination of properties that are extremely challenging for state-of-the-art verification techniques and tools and, until very recently, a detailed formalization was outside of their scope. First, the flow and the state-machines of these protocols are large and complex. This is due in part to the use of sequence numbers (*SQN*) and the need for a re-synchronization mechanism should counters become out-of-sync. This complexity is problematic for tools that reason about a bounded number of sessions as they scale poorly here. It also eliminates the option of machine-checked manual proofs as the number of interactions is too large for humans to explore. Second, these protocols are stateful (the *SQN* counters are mutable and persist over multiple sessions) and have numerous loops. This makes inductive reasoning necessary and rules out fully

automated tools, which are not yet capable of automatically finding appropriate inductive invariants. Finally, the AKA protocols use the Exclusive-OR (XOR) primitive to conceal some values. This primitive is notoriously hard to reason about symbolically, due to its algebraic properties (*i.e.,* associativity, commutativity, cancellation, and neutral element). For this reason, prior works provided only limited models of the AKA protocols, which were insufficiently precise for a satisfactory analysis, see discussion on related work below. Given these features, we are left with just the verifier TAMARIN [30] as a suitable tool, and TAMARIN has only recently been extended to handle XOR [21].

Contributions. We describe next our three main contributions: our formalization, models, and analysis results.

Formalization of the 5G standard. We extract and formally interpret the standard's security assumptions and goals. In doing so, we identify key missing security goals and weaknesses in the stated goals. We target a wide range of properties: confidentiality, authentication, and privacy. As explained in Sections 2 and 3, this required considerable analysis and interpretation of the 3GPP Technical Specification (722 pages across 4 documents).

Formal Model of 5G AKA. We tackle the aforementioned challenges to provide the first faithful model of an AKA protocol that is detailed enough to serve as a basis for a precise security analysis and is still amenable to automation. As we explain in Section 4, the modeling choices for formalizing our interpretation of the standard are crucial. To support reasoning about our model, we develop dedicated proof techniques based on inductive lemmas and proof strategies that guide proof search.

Security Evaluation of 5G AKA. We carry out the first formal security evaluation of 5G authentication, providing a comprehensive analysis of the 5G AKA protocol. This includes:

- *a formal, systematic security evaluation*: we leverage our model of 5G AKA to automatically identify the minimal security assumptions required for each security goal to hold. We find that some critical security goals are not met, except under assumptions that are missing from the standard. We also show that some properties are violated in some situations. See the tables in Section 5.2 for details.
- *recommendations*: we make explicit recommendations and propose provably secure fixes for the attacks and weaknesses we identified. Most of our recommendations generalize to 5G Authentication as a whole, and not just 5G AKA.

We believe that our model of 5G AKA provides a valuable tool to accompany the 5G standard's evolution and assess the security of future update proposals and evolutions (*e.g.,* 5G phase 2). Our model can also serve as the basis for a comprehensive formal comparison between AKA protocols from all generations, providing precise answers to questions like "what guarantees does one obtain, or lose, when moving from 4G to 5G?"

Related Work. Formal methods have been applied to AKA protocols in the past, but prior work provided only weak guarantees due to the use of strong abstractions, protocol simplifications, and limitations in the analyzed properties.

The initial AKA protocol specified for 3G was manually verified by 3GPP using TLA and an enhanced BAN Logic [3]. The TLA analysis focused on functional properties, like the protocol recovers from de-synchronization. The short pen and paper proof in an enhanced BAN logic provides weak guarantees, *e.g.,* about key agreement and confidentiality, due to the logic's limitations. In particular, the logic does not account for, *e.g.,* compromised agents and type-flaws, and it has had soundness issues in the past [13]. Moreover, the proof considered a simplified protocol without *SQN* concealment or re-synchronization as *SQNs* were always assumed synchronized. This misses, for example, the privacy attack based on the desynchronization error message we observed.

Other prior works have used PROVERIF to formally check untraceability and basic authentication properties on simplified AKA protocols [6, 31]. These works acknowledged the challenges of formally verifying AKA protocols but only offered limited solutions. For instance, the *SQN* counters were abstracted away by nonces that are supposed to be initially shared by *HNs* and subscribers, thus reducing the protocol to a stateless protocol. The re-synchronization procedure was also omitted. The *SNs* and *HNs* were merged into a single entity. Furthermore, XOR was either not modeled or was replaced by a different construct with simpler algebraic properties. The resulting protocol was thus overly simplified and corresponding analyses would have missed the attacks we obtain in this paper (Table 1). Moreover, the only authentication property that was checked is mutual aliveness between subscribers and the network.

More recently, [25] proposed a model-based testing approach that used PROVERIF to carry out some analyses of EPS AKA from 4G. However, in addition to using the same aforementioned abstractions and simplifications, they only used PROVERIF to check if specific trace executions correspond to attack traces.

In summary, in stark contrast to previous work, we provide the first faithful formalization of an AKA protocol. Namely, we formalize the entire protocol logic including the full protocol state machine with all message flows and symbolic abstractions of all cryptographic operators. This allows for the first comprehensive formal analysis that characterizes the properties that are achieved in different adversarial settings.

Outline. We present in Section 2 the cellular network architecture and how authentication is achieved in the 5G ecosystem using the 5G AKA protocol. We carry out a systematic formalization of the security assumptions and goals of the standard in Section 3 and highlight shortcomings. Section 4 explains the basics of the verifier TAMARIN and our modeling and design choices. We present our comprehensive security analysis of 5G AKA and our recommendations in Section 5. We draw conclusions in Section 6.

2 5G AUTHENTICATION PROTOCOLS

We explain in this section how authentication and key establishment are achieved in the 5G ecosystem, following as closely as possible the specification 3GPP TS 33.501 [4], referred from here on as [TS 33.501]. We simplify terminology to improve readability and refer the knowledgeable reader to the correspondence table with the terminology from 3GPP given in Appendix A. We first present the general architecture and afterwards the authentication protocols.

2.1 Architecture

Three main entities are involved in the cellular network architecture (see Figure 1). First, *User Equipment* (*UE*), typically smartphones or IoT devices containing a *Universal Subscriber Identity Module* (USIM), are carried by *subscribers*. We shall call a *subscriber* the combination of a *UE* with its USIM. Second, *Home Networks* (*HNs*) contain a database of their subscribers and are responsible for their authentication. However, subscribers may be in locations where their corresponding *HN* has no *base station* (*i.e.,* antennas may connect *UEs* to the network); for example when roaming. Therefore, the architecture has a third entity: the *Serving Networks* (*SNs*) to which *UEs* may attach to. An *SN* provides services (*e.g.,* call or SMS) once both the *UE* and the *SN* have mutually authenticated each other (this supports billing) and have established a secure channel with the help of the subscriber's *HN*. The *UE* and *SN* communicate over the air while the *SN* and *HN* communicate on an authenticated channel (we list security assumptions later in this section).

As mentioned earlier, each subscriber has a USIM with cryptographic capabilities (*e.g.,* symmetric encryption, MAC). Relevant for our work is that the USIM stores:

- a unique and permanent subscriber *identity*, called *Subscription Permanent Identifier* (*SUPI*),
- the public asymmetric key pk_{HN} of its corresponding *HN*,
- a long-term *symmetric key* denoted as K (used as a shared secret between subscribers and their corresponding *HNs*),
- and a counter, called *Sequence Number* denoted as *SQN*.

The *HN*, associated to some subscriber, stores the same information in its database.

Simplifications. In the standard, *SNs* and *HNs* are composed of several sub-entities (*e.g., HNs* consist of a database, authentication server *etc.*). However, very few security properties require this level of granularity. We thus have chosen to consider these three larger logical entities (see Appendix A for more details).

2.2 Authentication Protocols

In order to let *SNs* and subscribers establish secure channels and authenticate each other, 3GPP has specified two authentication methods: *5G AKA* and *EAP-AKA'*. The choice between those two methods is left to the *HN*, once it has correctly identified the subscriber with the *Initialization Protocol*. We now describe these three security protocols. (All cryptographic messages are precisely described in Appendix A.)

Subscriber   Serving Network   Home Network

Phone (UE), USIM   Base station (antenna)   Subscriber's carrier
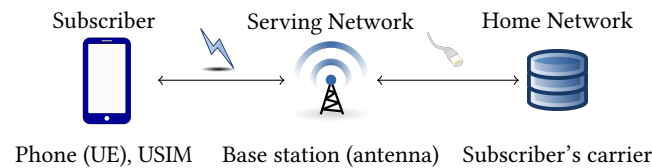
Figure 1: Overall architecture: The subscriber uses his phone (UE), equipped with a USIM, to communicate with a base station run by the SN over an insecure wireless channel. The SN communicates with the subscriber's carrier (HN) on the right over an authenticated (wired) channel.

2.2.1 Initialization Protocol [TS 33.501, Sec. 6.1.2]. Figure 2 depicts the sub-protocol responsible for the subscribers' identification and initializing the authentication. Once the *SN* has triggered an authentication with the subscriber, the latter sends a randomized encryption of the *SUPI* (for privacy reasons, as we explain in Section 3.2.3): $SUCI = \langle aenc(\langle SUPI, R_s \rangle, pk_{HN}), idHN \rangle$, where $aenc(\cdot)$ denotes asymmetric encryption, R_s is a random nonce, and *idHN* uniquely identifies an *HN*. The identifier *idHN* enables the *SN* to request authentication material from the appropriate *HN*. Upon reception of the *SUCI* along with the *SN*'s identity (referred to as *SNname*), the *HN* is able to retrieve the *SUPI*, the subscribers' identity, and choose an authentication method.

2.2.2 The 5G AKA Protocol [TS 33.501, Sec. 6.1.3.2]. As mentioned before, the key K is used as a long-term shared secret, and *SQN* is used as a replay protection[1] for the subscriber. While *SQN* is expected to be synchronized between the subscriber and the *HN*, it may happen that they become out-of-sync (*e.g.,* due to message loss). We thus use SQN_{UE} (respectively SQN_{HN}) to refer to the SQN value stored in the *UE* (respectively *HN*). The 5G-AKA protocol consists of two main phases: a *challenge-response* and an optional *re-synchronization procedure* (that updates the SQN on the *HN* side in case the *SQN* is out-of-sync). The entire 5G AKA protocol flow is depicted in Figure 3.

Challenge-Response. Upon receiving a request for authentication material, the *HN* computes an *authentication challenge* built from:

- a random nonce R (the challenge),
- *AUTN* (proving the challenge's freshness and authenticity),
- *HXRES** (response to the challenge that *SN* expects),
- K_{SEAF} (key seed for the secure channel that the subscriber and *SN* will eventually establish).

The functions f1 − f5, used to compute the authentication parameters, are one-way keyed cryptographic functions completely unrelated, and ⊕ denotes Exclusive-OR. Challenge(·) and KeySeed(·) are complex Key Derivation Functions (KDFs); see Appendix A for more details. *AUTN* contains a Message Authentication Code (MAC) of the concatenation of R with the corresponding sequence number SQN_{HN} stored for this subscriber. A new sequence number is generated by incrementing the counter. The sequence number SQN_{HN} allows the subscriber to verify the freshness of the authentication request to defend against replay attacks and the MAC proves the challenge's authenticity. The *HN* does not send the challenge's full response *RES** to the *SN* but only a hash therereof; the rationale being that *HNs* are willing to have assurance of the presence of its subscribers even with malicious *SNs*.

The *SN* stores K_{SEAF} and the expected response to the challenge, and then forwards the challenge to the subscriber. Upon receiving the challenge, the subscriber first checks its authenticity and freshness. To do this, the subscriber extracts $xSQN_{HN}$ and *MAC* from *AUTN* and checks that:

(i) *MAC* is a correct MAC value with respect to K, and replies `'Mac_failure'` if it is not the case,

---

[1]This design choice is for historical reasons: old USIM (*e.g.,* in 3G and 4G) have no capability to generate random nonces.
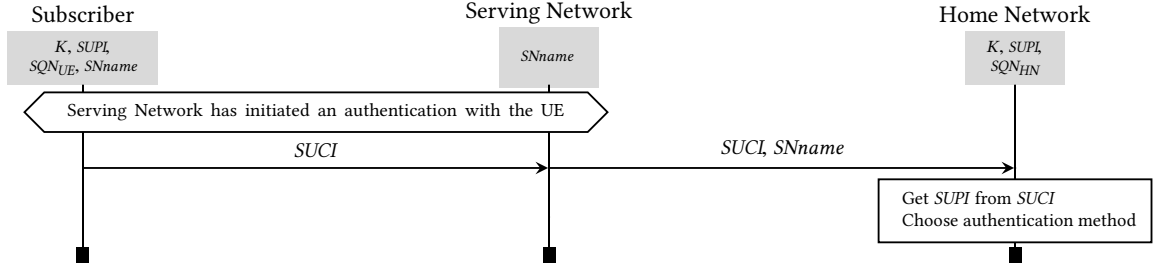
Figure 2: Initiation of Authentication

(ii) the authentication request is fresh[2], *i.e.*, $SQN_{UE} < xSQN_{HN}$ and replies ⟨'Sync_failure', *AUTS*⟩ otherwise (*AUTS* is explained in the re-synchronization procedure below).

If all checks hold then the subscriber computes the key seed K_{SEAF}, which is used to secure subsequent messages. It also computes the authentication response RES^* and sends it to the *SN*. The *SN* checks that this response is as expected and forwards it to the *HN*, who validates it. If this validation succeeds then the *HN* confirms to the *SN* that the authentication is successful and sends the *SUPI* to the *SN*. Subsequent communications between *SN* and the subscriber can be secured using the key seed K_{SEAF}.

Re-synchronization procedure [TS 33.501, Sec. 6.1.3.2.1]. In case of a synchronization failure (case ¬(ii)), the subscriber replies with ⟨'Sync_failure', *AUTS*⟩. The *AUTS* message's purpose is to allow the *HN* to re-synchronize with the subscriber by replacing its own SQN_{HN} by the sequence number of the subscriber SQN_{UE}; see [TS 33.102, Sec. 6.3.5, 6.3.3]. However, SQN_{UE} is not transmitted in clear text to avoid being eavesdropped on (it is privacy sensitive as explained in Section 3.2.3). Therefore, the specification requires *SQN* to be *concealed*; *i.e.*, XORed with a value that should remain private: $AK^* = f5^*(K, R)$. Formally, the concealed value is: $CONC^* = SQN_{UE} \oplus AK^*$ and allows the *HN* to extract SQN_{UE} by computing AK^*. Note that $f5^*$ and $f1^*$ are independent one-way keyed cryptographic functions completely unrelated to the functions $f1 - f5$. Finally, $AUTS = \langle CONC^*, MAC^* \rangle$, where $MAC^* = f1^*(K, \langle SQN_{UE}, R \rangle)$, allowing the *HN* to authenticate this message as coming from the intended subscriber.

2.2.3 The EAP-AKA' Protocol [TS 33.501, Sec. 6.1.3.1] and [RFC 5448]. EAP-AKA' is very similar to 5G AKA: it relies on the same mechanisms (challenge-response with *K* as a shared secret and *SQN* for replay protection) and uses very similar cryptographic messages. The main difference is the flow and some key derivation functions are subtly changed. Since we focus our analysis on the 5G AKA authentication method, we do not describe those differences in detail here and refer the curious reader to Appendix B.

3 THREAT MODEL AND SECURITY GOALS

In this section, we derive precise, formal security goals from the informal descriptions given in the Technical Specification (TS) and Technical Requirement (TR) documents issued by the 3GPP. Our formal definitions are our interpretations of these texts. We support them with quotes from and references to relevant excerpts of the

TS and TR documents. The full list of relevant excerpts along with explanations of our interpretations is given in Appendix D.

The extraction of precise properties from the standard's informally stated goals is an important prerequisite to applying a security protocol analysis tool (like TAMARIN). It is thus a crucial step in the security analysis of a complex protocol such as 5G AKA.

3.1 Security Assumptions and Threat Model

3.1.1 Assumptions on Channels. The channel between the *SN* and the *HN* provides confidentiality, integrity, authenticity, and replay protection [TS 33.501, Sec. 5.9.3].

The channel between the subscribers and *SNs* is subject to eavesdropping by passive attackers and manipulation, interception, and injection of messages by active attackers. A *passive attacker* listens to signaling messages (*i.e.*, messages sent on the physical layer) and can thus eavesdrop on all messages exchanged in its vicinity, but it never emits a signal. An *active attacker* sets up a fake base station to receive and send signaling messages, *e.g.*, to impersonate *SNs*. While no 5G-specific hardware is publicly available yet, we recall that 4G base stations have been built using open-source and freely available software and hardware [23, 35]. From now on, we shall consider active attackers, except when explicitly stated otherwise.

3.1.2 Assumptions on Cryptographic Primitives. The functions $f1, f1^*, f2$ are message authentication functions, and $f3, f4, f5, f5^*$ are key derivation functions [TS 33.102, Sec. 3.2, 6.3.2]. To our knowledge there is no comprehensive set of standardized security requirements for these functions. The requirements in [TS 33.105, Sec. 5] are insufficient, but we infer from the informal presentation in [TS 33.102, Sec. 3.2] and requirements in [TS 33.105, Sec. 5] that the former provide only integrity protection and the latter both integrity and confidentiality protection. However, since $f1$ and $f1^*$ are applied to data that should be secret, such as *SQN* (see Section 3.2.3), it is our understanding that they should also preserve the confidentiality of their inputs. We therefore assume in our analysis that all these functions protect both integrity and confidentiality, but we stress that this is either underspecified or subscribers' privacy is put at risk (see Section 3.2.3).

3.1.3 Assumptions on Parties. To provide strong, fine-grained guarantees, we consider different compromise scenarios. First, we consider an attacker who can compromise some *SNs*. This means that the attacker gets access to an authenticated channel between the compromised *SN* and *HNs*, which he can use to eavesdrop on and inject messages. This is a reasonable assumption in 5G, where authentication methods should provide security guarantees even in
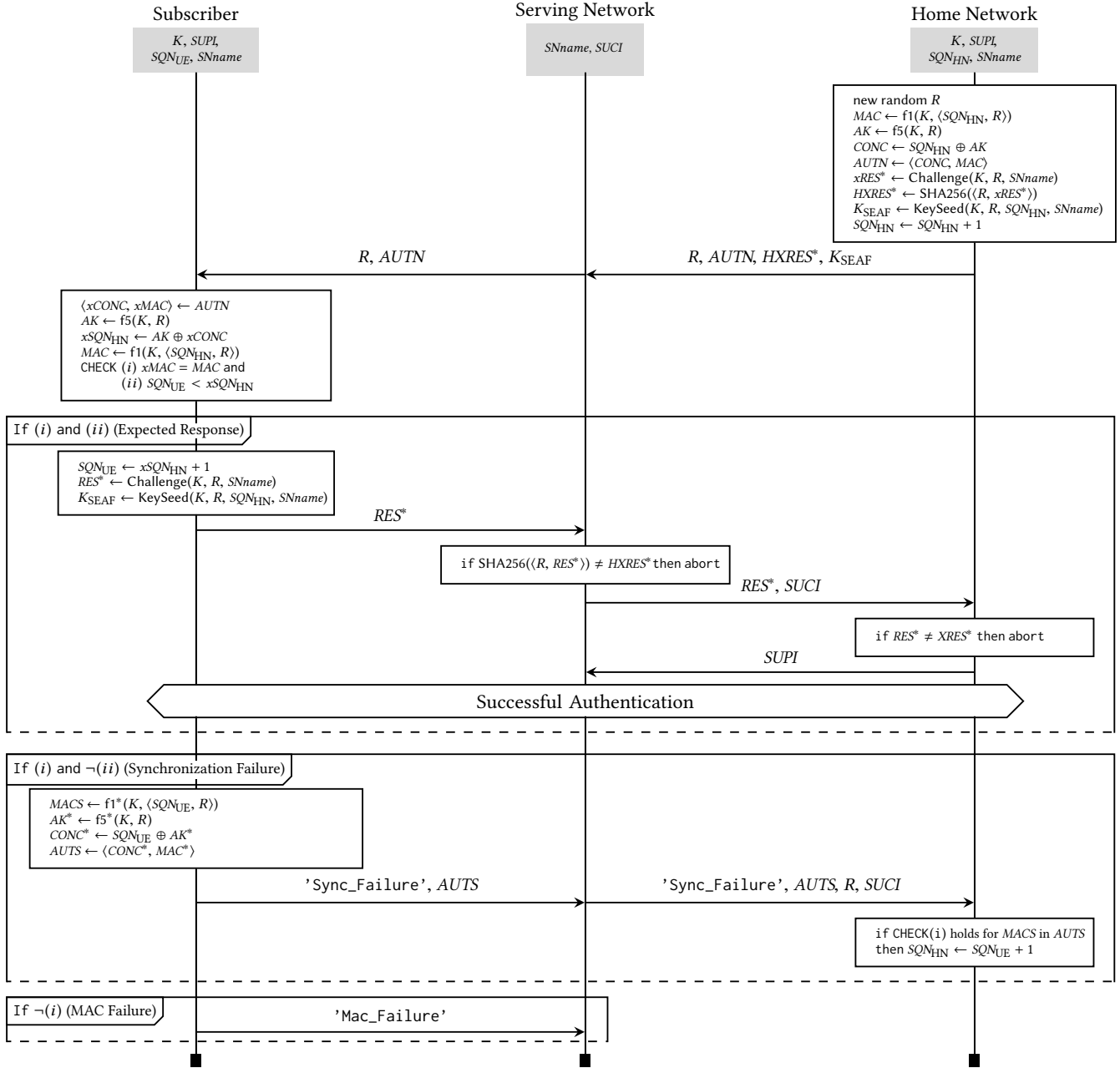
---

[2]The freshness check may additionally consider non-normative protection against wrapping around of SQN_{HN} that we do not describe here (see [TS 33.102, Sec. C]).

Subscriber

K, $SUPI$,
SQN_{UE}, $SNname$

Serving Network

$SNname$, $SUCI$

Home Network

K, $SUPI$,
SQN_{HN}, $SNname$

Home Network:

new random R
$MAC \leftarrow$ f1$(K, \langle SQN_{HN}, R \rangle)$
$AK \leftarrow$ f5(K, R)
$CONC \leftarrow SQN_{HN} \oplus AK$
$AUTN \leftarrow \langle CONC, MAC \rangle$
$xRES^* \leftarrow$ Challenge$(K, R, SNname)$
$HXRES^* \leftarrow$ SHA256$(\langle R, xRES^* \rangle)$
$K_{SEAF} \leftarrow$ KeySeed$(K, R, SQN_{HN}, SNname)$
$SQN_{HN} \leftarrow SQN_{HN} + 1$

Home Network → Serving Network: R, $AUTN$, $HXRES^*$, K_{SEAF}

Serving Network → Subscriber: R, $AUTN$

Subscriber:

$\langle xCONC, xMAC \rangle \leftarrow AUTN$
$AK \leftarrow$ f5(K, R)
$xSQN_{HN} \leftarrow AK \oplus xCONC$
$MAC \leftarrow$ f1$(K, \langle SQN_{HN}, R \rangle)$
CHECK (i) $xMAC = MAC$ and
$\quad\quad (ii)$ $SQN_{UE} < xSQN_{HN}$

If (i) and (ii) (Expected Response)

$SQN_{UE} \leftarrow xSQN_{HN} + 1$
$RES^* \leftarrow$ Challenge$(K, R, SNname)$
$K_{SEAF} \leftarrow$ KeySeed$(K, R, SQN_{HN}, SNname)$

Subscriber → Serving Network: RES^*

Serving Network: if SHA256$(\langle R, RES^* \rangle) \neq HXRES^*$ then abort

Serving Network → Home Network: RES^*, $SUCI$

Home Network: if $RES^* \neq XRES^*$ then abort

Home Network → Serving Network: $SUPI$

Successful Authentication

If (i) and $\neg(ii)$ (Synchronization Failure)

$MACS \leftarrow$ f1$^*(K, \langle SQN_{UE}, R \rangle)$
$AK^* \leftarrow$ f5$^*(K, R)$
$CONC^* \leftarrow SQN_{UE} \oplus AK^*$
$AUTS \leftarrow \langle CONC^*, MAC^* \rangle$

Subscriber → Serving Network: 'Sync_Failure', $AUTS$

Serving Network → Home Network: 'Sync_Failure', $AUTS$, R, $SUCI$

Home Network: if CHECK(i) holds for $MACS$ in $AUTS$
then $SQN_{HN} \leftarrow SQN_{UE} + 1$

If $\neg(i)$ (MAC Failure)

Subscriber → Serving Network: 'Mac_Failure'

Figure 3: The 5G AKA protocol (continuing Figure 2)

the presence of genuine but malicious *SNs* [TS 33.501, Sec. 6.1.4.1]. In such situations, the *HNs* may cooperate with such *SNs* to authenticate some subscriber. In practice, this may happen in roaming situations. Next, we consider that the attacker may have genuine USIMs and compromised USIMs under its control. For those compromised subscribers, the attacker can access all secret values stored in the USIMs; *i.e.*, *SUPI*, *K*, and *SQN*. Finally, the attacker can access all long-term secrets K, sk_{HN}, and *SUPI* from compromised *HNs*.

3.1.4 Assumptions on Data Protection. The subscriber credentials, notably the key K and the identifier *SUPI*, shared between

subscribers and *HNs*, should initially be secret, provided they belong to non-compromised agents [TS 33.501, Sec. 3.1].

The sequence number *SQN* is a 48-bit counter or a 43-bit counter [TS 33.102, Sec. 6.3.7, C.3.2] and therefore guessable with a very low probability. Note that an offline guessing attack on the sequence number counter is not possible, and online attacks on the *UE* first require a correct MAC (based on the shared secret K) before the *UE* responds whether the *SQN* was acceptable. We thus consider a reasonable threat model where the value of *SQN* is unknown to the attacker when the attack starts, but the attacker knows how it is incremented during the attack. This corresponds to an attacker who

(i) can monitor the activity of targeted subscribers in its vicinity during the attack but, (ii) can neither guess the initial value of *SQN*, (iii) nor can he monitor targeted subscribers all the time (*i.e.,* from the first use of the USIM up to the attack time).

While not explicitly stated in the specification, we shall assume that the private asymmetric key sk_{HN} is initially secret.

3.2 Security Requirements

We now extract and interpret from the 5G documents the security goals that 5G AKA should achieve according to the 5G standard.

3.2.1 Authentication Properties. 5G specifications make claims about authentication properties at different places in the documents. We have identified relevant claims and translate them into formal security goals, indicated in *purple, cursive text*. We use Lowe's taxonomy of authentication properties [28] to make the goals precise, prior to formalization. These properties are well established and understood avoiding ambiguity [8]. Moreover, there exists a formal relation between the taxonomy and mathematical definitions of security properties that can be directly modeled in TAMARIN [1].

We give an overview of Lowe's taxonomy and its relationship with formal definitions of authenticity in Appendix C. Intuitively, the taxonomy specifies, from an agent A's point of view, four levels of authentication between two agents A and B: (i) aliveness, which only ensures that B has been running the protocol previously, but not necessarily with A; (ii) weak agreement, which ensures that B has previously been running the protocol with A, but not necessarily with the same data; (iii) non-injective agreement, which ensures that B has been running the protocol with A and both agree on the data; and (iv) injective agreement, which additionally ensures that for each run of the protocol of an agent there is a unique matching run of the other agent, and prevents replay attacks.

Note that the 5G specification considers some authentication properties to be *implicit*. This means that the guarantee is provided only after an additional *key confirmation roundtrip* (with respect to K_{SEAF}) between the subscribers and the *SN*. We discuss the resulting problems and critique this design choice in Section 5.2.2. *Authentication between subscribers and HNs.* First, the subscribers must have the assurance that authentication can only be successful with *SNs* authorized by their *HNs*; see [TS 33.501, Sec. 6.1.1.3] and:

> **[TS 33.501, Sec. 5.1.2] Serving network authorization by the home network:** Assurance [that subscriber] is connected to a serving network that is authorized by the home network. [...] This authorization is 'implicit' in the sense that it is implied by a successful authentication and key agreement run.

Formally, a *subscriber must obtain non-injective agreement on SNname with its HN after key confirmation.*

In 5G, the trust assumptions are different than in previous standards, like 3G or 4G. Most notably, the level of trust the system needs to put into the *SNs* has been reduced. One important property provided by 5G is that an *SN* can no longer fake authentication requests with the *HNs* for subscribers not attached to one of its base stations [TS 33.501, Sec. 6.1.4.1]. Formally, *the HNs obtain the aliveness of its subscribers at that SN, which is non-injective agreement on SNname from the HNs' point of view with the subscribers.*

Authentication between subscribers and SNs. As expected, the *SNs* shall be able to authenticate the subscribers:

> **[TS 33.501, Sec. 5.1.2] Subscription authentication**: The serving network shall authenticate the Subscription Permanent Identifier (SUPI) in the process of authentication and key agreement between UE and network.

The SNs must obtain non-injective agreement on SUPI with the subscribers. As *SUPI* is the subscriber's identifier this is actually just *weak agreement for the SNs with the subscribers.*

Conversely, the subscribers shall be able to authenticate the *SNs*:

> **[TS 33.501, Sec. 5.1.2] Serving network authentication**: The UE shall authenticate the serving network identifier through implicit key authentication.

Since *SNname* is the *SN*'s identifier, *the subscribers must obtain weak agreement with the SNs after key confirmation.*
Authentication between SNs and HNs. The *SNs* shall be able to authenticate subscribers that are authorized by their corresponding *HN*:

> **[TS 33.501, Sec. 5.1.2] UE authorization**: The serving network shall authorize the UE through the subscription profile obtained from the home network. UE authorization is based on the authenticated SUPI.

The SNs must obtain non-injective agreement on SUPI with the HNs.

3.2.2 Confidentiality Properties. While it is not clearly specified, obviously 5G-AKA should ensure the *secrecy of* $K_{SEAF}, K, and sk_{HN}$ (see similar goals in 3G [TS 133.102, Sec. 5.1.3]).

5G-AKA should also ensure that knowing the session key K_{SEAF} established in one session is insufficient to deduce another session key K_{SEAF} established in either a previous session or a later session [TS 33.501, Sec. 3]. Formally, the key K_{SEAF} *established in a given session remains confidential even when the attacker learns the* K_{SEAF} *keys established in all other sessions.* Note that this is different from *forward secrecy* and *post-compromise secrecy [15]*, which fail to hold as we shall see in Section 5.1. Forward and post-compromise secrecy require session key secrecy even in the event that long-term key material is compromised. 5G-AKA does not meet these requirements as knowledge of the key K allows an attacker to derive all past and future keys.

Finally, the same key K_{SEAF} should never be established twice [TS 133.102, Sec. 6.2.3]. This will be analyzed as part of *Injective agreement properties on the established key* K_{SEAF} for different pairs of parties.

3.2.3 Privacy Properties. We first emphasize the importance given to privacy in 5G documentation:

> **[TR 33.899, Sec. 4.1,4.2]** Subscription privacy deals with various aspects related to the protection of subscribers' personal information, *e.g.,* identifiers, location, data, etc. [...] The security mechanisms defined in NextGen shall be able to be configured to protect subscriber's privacy.

This important role given to privacy can be explained by numerous, critical attacks that have breached privacy (*e.g.,* with IMSI-catchers [35, 36]) in previous generations; see the survey [32]. We also recall that privacy was already a concern in 3G:

Thus, 3G already had security requirements for user identity confidentiality, anonymity, and untraceability. However, these properties are required by the standard only against a *passive* attacker (we criticize this restriction in Section 5.2.3). We now list more precise requirements on how privacy should be protected in 5G.

In 5G, the *SUPI* is considered sensitive and must remain secret since it uniquely identifies users [TS 33.501, Sec. 5.2.5,6.12]. Indeed, an attacker who obtains this value can identify a subscriber, leading to classical user location attacks (see [TS 133.102, Sec. 5.1.1] above), much like IMSI-catcher attacks. Formally, *the SUPI shall remain secret in the presence of a passive attacker*.

Similarly, the *SQN* must remain secret [TS 33.102, Sec. 6.2.3, C.3.2]. An additional reason that is not explicitly stated is that the *SQN* leaks the number of successful authentications the corresponding USIM has performed since it was manufactured, which is strongly correlated to its age and activity. This is even more critical when the attacker learns the *SQN* at different times, as this allows activity estimation for that time-period. Formally, *the SQN shall remain secret in the presence of a passive attacker*.

Preventing the attacker from learning identifying data (*i.e., SUPI, SQN*) is insufficient protection against privacy attacks such as traceability attacks (we show an example in Section 5.2.3). While no formal and explicit statement is made on the necessity of ensuring untraceability for 5G, several claims in TR and TS documents (see Appendix D.2.3) and the fact that it was required for 3G ([TS 133.102, Sec. 5.1.1], see above), suggest that this property is relevant for 5G as well. Formally, 5G authentication methods are presumably required to provide *untraceability of the subscribers in the presence of a passive attacker*.

3.3 Security Goals are Underspecified

We now discuss the aforementioned standardized security goals and critique the lack of precision in the standard. We show that the requirements specified in the standard are not sufficient to provide the expected security guarantees in the context of mobile communication telephony use cases. This is completely independent of

whether the proposed protocols actually fulfill these properties or not (which we examine in Section 5).

First, given that the protocol is an *Authenticated Key Exchange* protocol, we expect at least mutual authentication requirements and agreement properties on the established key. It is thus surprising that the standard does not require any agreement on K_{SEAF}. The different pairs of roles, especially subscribers and *SNs* should at least obtain non-injective agreement on the shared key K_{SEAF}. Moreover, the key K_{SEAF} should be different for each session. This is a critical requirement, especially for typical use cases of these protocols. Indeed, if this property is not provided, an attacker could make *UEs* and *SNs* establish a secure channel based on a key that has been previously used, and could therefore replay user data. The crucial missing requirements are *injective agreements on K_{SEAF} between pairs of parties, especially between the SNs and subscribers.*

The standard specifies authentication properties as weak *authorization* properties that can be formalized as non-injective agreement on the roles' identifiers, or simply weak agreement properties (see Section 3.2.1). We discuss the standard's restriction to "implicit authentication" in Section 5.2.2. As explained earlier, 5G requires *HNs* to have the assurance that *UEs* are attached to *SNs* [TS 33.501, Sec. 6.1.4.1] currently. However, a non-injective agreement on *SNname* from an *HN* towards a subscriber is too weak since it suffices that the subscriber has attached to the corresponding *SN* in some session in the past to fulfill the property.

It is crucial in that scenario for the *HNs* to obtain assurance that the subscriber is attached to the *SN* during the present session. The derivation of K_{SEAF} includes *SNname* for the binding to *SN*. This derivation also includes a nonce R, due to which we obtain the desired assurance as a corollary of *injective agreement on K_{SEAF} from the HNs towards the subscribers*, which we consider instead.

Similarly, the subscribers should have the assurance that the *SNs* with which they establish secure channels are known and trusted by their *HNs* at the time of the authentication, not only in some past session. Therefore, they should obtain *injective agreement on K_{SEAF} (which is bound to SNname) with the HNs.* While less critical, other pairs of roles should also have stronger assurance. We describe how the standard can be improved in this regard in Section 5.3.

4 FORMAL MODELS

In this section, we give a basic introduction to the symbolic model of cryptographic protocols and the tool TAMARIN that automates reasoning in this model (Section 4.1). Afterwards, we give an overview on how security properties can be modeled in this tool (Section 4.2). Next, after describing our modeling choices (Section 4.3), we describe the challenges associated with modeling a large, complex protocol like 5G AKA and how we overcame them (Section 4.4).

4.1 The Tamarin Prover

To verify 5G AKA, we used the TAMARIN prover [33]. TAMARIN is a state-of-the-art protocol verification tool for the *symbolic model*, which supports stateful protocols, a high level of automation, and equivalence properties [9], which are necessary to model privacy properties such as unlinkability. It has previously been applied to real-world protocols with complex state machines, numerous messages, and complex security properties such as TLS 1.3 [17].

Moreover, it was recently extended with support for XOR [21], a key ingredient for faithfully analyzing 5G AKA. We chose TAMARIN as it is currently the only tool that combines all these features, which are essential for a detailed analysis of 5G AKA.

In the symbolic model and a fortiori in TAMARIN, messages are described as terms. For example, $enc(m, k)$ represents the message m encrypted using the key k. The algebraic properties of the cryptographic functions are then specified using equations over terms. For example the equation $dec(enc(m, k), k) = m$ specifies the expected semantics for symmetric encryption: the decryption using the encryption key yields the plaintext. As is common in the symbolic model, cryptographic messages do not satisfy other properties than those intended algebraic properties, yielding the so-called *black box cryptography assumption* (*e.g.,* one cannot exploit potential weaknesses in cryptographic primitives).

The protocol itself is described using multi-set rewrite rules. These rules manipulate multisets of *facts*, which model the current state of the system with *terms* as arguments.

Example 4.1. The following rules describe a simple protocol that sends an encrypted message. The first rule creates a new long-term shared key k (the fact !Ltk is persistent: it can be used as a premise multiple times). The second rule describes the agent A who sends a fresh message m together with its MAC with the shared key k to B. Finally, the third rule describes B who is expecting a message and a corresponding MAC with k as input. Note that the third rule can only be triggered if the input matches the premise, *i.e.,* if the input message is correctly MACed with k.

$Create\_Ltk$: $[Fr(k)]\!-\![]\!\rightarrow\![!Ltk(k)],$
$Send\_A$: $[!Ltk(k), Fr(m)]\!-\!\{Sent(m)\}\!\rightarrow\![Out(\langle m, mac(m, k)\rangle)],$
$Receive\_B$: $[!Ltk(k), In(\langle m, mac(m, k)\rangle)]\!-\!\{Received(x)\}\!\rightarrow\![]$ □

These rules yield a labeled transition system describing the possible protocol executions (see [1, 33] for the syntax and semantics). TAMARIN combines the protocol semantics with a Dolev-Yao [20] style attacker. This attacker controls the entire network and can thereby intercept, delete, modify, delay, inject, and build new messages. However, he is limited by the cryptography: he cannot forge signatures or decrypt messages without knowing the key (black box cryptography assumption). He nevertheless can apply any function (e.g., hashing, XOR, encryption, pairing, . . .) on messages he knows to compute new messages.

4.2 Formalizing Security Goals in Tamarin

In TAMARIN, security properties are specified in two different ways. First, trace properties, such as secrecy or variants of authentication, are specified using formulas in a first-order logic with timepoints.

Example 4.2. Consider the multiset rewrite rules given in Example 4.1. The following property specifies a form of non-injective agreement on the message, *i.e.,* that any message received by B was previously sent by A:

$$\forall i, m.Received(m)@i \Rightarrow (\exists j.Sent(m)@j \wedge j \lessdot i).$$

Since the 5G AKA protocol features multiple roles and multiple instantiations thereof, agreement properties additionally require that the views of the two partners (who is playing which role, and what is the identity of the partner) actually match; see Appendix C.

For each specified property, TAMARIN checks that the property holds for all possible protocol executions, and all possible adversary behaviors. To achieve this, TAMARIN explores all possible executions in a backward manner, searching for reachable attack states, which are counterexamples to the security properties.

Equivalence properties, such as unlinkability, are expressed by requiring that two instances of the protocol cannot be distinguished by the attacker. Such properties are specified using *diff*-terms (which take two arguments), essentially defining two different instances of the protocol that only differ in some terms. TAMARIN then checks observational equivalence (see [9]), i.e., it compares the two resulting systems and checks that the attacker cannot distinguish them for any execution and for any of its behaviors.

In fully automatic mode, TAMARIN either returns a proof that the property holds, or a counterexample/attack if the property is violated, or may not terminate as the underlying problem is undecidable. TAMARIN can also be used in interactive mode, where the user can guide the proof search. Moreover the user can supply heuristics called *oracles* to guide the proof search in a sound way. We heavily rely on heuristics in our analyses as they allow us to tame the complexity of the protocol, as explained below.

4.3 Modeling Choices

To better delimit the scope of our model and our analyses, we now describe some of our modeling choices.

Architecture. We consider three roles (subscribers, *SNs*, and *HNs*) and reason with respect to unboundedly many instances of each role. As expected, each subscriber credential is stored in at most one *HN*. We model communication channels between these parties that provide security properties as explained in Section 3.1. Additionally, the messages exchanged are *tagged* on the authenticated channel between the *SNs* and *HNs*. This models the implicit assumption that the authenticated channel between an *SN* and an *HN* role instance is protected from *type flaw attacks*.

Modeling Cryptographic Messages. We model and treat the subscribers' *SQNs* as natural numbers (using a standard encoding based on multisets [1, 34]). We assume the attacker cannot follow *UEs* from their creation so the *SQN* is not known (see Section 3.1) at first, and we thus start the sequence number with a random value. The freshness check (*i.e.,* (ii) from Figure 3) is faithfully modelled as a natural number comparison. Since the *SQN* may become out-of-sync during normal protocol execution, we also consider an attacker who can arbitrarily increase SQN_{UE} (*UE* does not allow decrease). Note that the attacker can already increase SQN_{HN} by repeatedly triggering authentication material requests. We fully model the re-synchronization mechanism and let the *HNs* update their SQN_{HN} accordingly. The concealment of the *SQN*, using Exclusive-OR (XOR), is faithfully modeled by relying on the recent extension of TAMARIN with equational theories including XOR [21].

Compromise Scenarios. We model various compromise scenarios: secret key reveals (of K or sk_{HN}), reveals of the *SUPI* or the initial value of *SQN*, and *SN* compromises (*i.e.,* the attacker gains access to an authenticated channel with the *HNs*). This is needed mainly for two reasons. First, the specification itself considers some of those scenarios and still requires some security guarantees to hold (cf. the

compromised *SNs* from Section 3.1). Second, this enables a comprehensive analysis to identify the minimal assumptions required for a property to hold. For instance, if some critical authentication property were violated when the attacker could access the initial value of the *SQN*, this would represent a potential vulnerability in the protocol since the *SQN* is not a strong secret and the search space of the *SQN* that the attacker needs to explore could be further reduced by exploiting the meaning of this counter.

Implicit Authentication. We equip the model with an optional key-confirmation roundtrip where the subscribers and *SNs* confirm their key K_{SEAF} by MACing different constants. Our security analysis is then parametric in this roundtrip, allowing us to derive which properties already hold without key confirmation, and what is gained by this key confirmation step.

Simplifications made. As usual in the symbolic model, we omit message bit lengths. Some key derivation functions also take the length of their arguments to prevent type-flaw attacks. This is covered in our model as such length-based misinterpretation cannot happen. The protocols under study feature some sub-messages that are publicly known constants, for example, fixed strings like AMF or 'MAC_Failure'. We mostly omit such sub-messages, unless they are useful as tags. We do not model the optional, non-normative protection against wrapping around the *SQN* [TS 33.102, Sec. C]. Note that this is in line with our modeling of the *SQN* as a natural number for which no wrapping can occur. The 5G AKA protocol establishes a session key, to which a key identifier is associated (the key set identifier ngKSI). Such identifiers are needed for subsequent procedures only and do not interact with the authentication methods and hence we omit them. An *SN* may create a pseudonym, called 5G-GUTI, associated with the *SUPI* of a subscriber who is visiting this *SN*, in order to recognize this subscriber in a subsequent session. We omit this optional mechanism. Authentication tokens do not expire in our model as is usual in symbolic models. However, since such mechanisms are never clearly specified in normative documents, we emphasize that critical security properties should not rely on them.

4.4 Tamarin Models of 5G AKA

We have built a TAMARIN model for the 5G AKA authentication method which enables automated security analyses. Our models and associated documentation are available [5], and use TAMARIN v1.4.0 [2], which includes XOR support.

Writing a formal model of such a substantial real-world protocol is challenging. However, the real difficulty is doing this in a way that enables effective reasoning about the models, *i.e.,* is amenable to automation. We now describe this modeling as well as the proof strategies we have developed, and argue why this can serve as a basis for future analyses of protocols in the AKA family.

4.4.1 Challenges. The 5G AKA protocol uses a combination of features that make reasoning about these models highly complex. First, 5G AKA is a stateful protocol, *i.e.,* it relies on internal states (the *SQNs*) that are persistent across sessions and that are mutable. In the symbolic model, the set of values these states can take — all natural numbers — is unbounded. This feature alone excludes most verification tools, leaving only the verifier TAMARIN. Verifiers for a bounded number of sessions are not a viable choice, simply due to

the size of a single session. Moreover, the sequence numbers are not only internal counters, they are also used for comparison on input. This requires the ability to compare two values (see Section 4.3) in the chosen representation of natural numbers. This is demanding in terms of proof efforts: to the best of our knowledge, this is the first time a complete, real-world protocol relying on natural numbers and comparisons is analyzed with a formal verifier. Previous examples are limited to the case of just an internal counter for a TPM [29] or small examples, like simplified Yubikey [27].

Second, 5G AKA heavily relies on XOR to conceal the value of *SQNs*. Reasoning about XOR in the symbolic model is challenging and its integration in TAMARIN is recent [21]. Intuitively, this is because there are numerous ways to obtain a given term when one can exploit the algebraic properties of XOR (*i.e.,* associativity, commutativity, cancellation, and neutral element). These different options must be explored separately and considerably increase the search space when proving properties. Again, in the symbolic model, we are not aware of any formal analysis of such a large-scale real-world protocol featuring XOR.

Finally, the state-machine of the 5G AKA protocol is large and complex. Role instantiations can be in 14 different states. Evolution between those states includes numerous loops, notably because of the persistent and mutable states' *SQNs*, *e.g.,* sessions can be repeated while using a given *SQN*.

4.4.2 Proof Strategies. The way *SQNs* are updated on the subscribers' and *HN*'s sides, notably with the re-synchronization procedure, induces complex state-change that must be tackled by our proof strategies. Manual proofs are not feasible due to the size of the search space one would have to explore. In contrast, TAMARIN's fully automatic mode fails to prove relevant security properties and even extremely weak properties such as the full executability of the protocols. Our work straddles this divide: we developed a proof structure based on intermediate lemmas (called *helping lemmas*) as well as proof strategies for proving these lemmas and the security properties. Proof strategies are implemented through *oracles* that offer a light-weight tactic language, implemented in Python, to guide the proof search in TAMARIN.

The key helping lemmas we prove state that the *SQN* associated to a subscriber stored on his side (respectively on its *HN*'s side) is strictly increasing (respectively monotonically increasing). Thanks to our chosen modeling of the states' *SQNs* as multisets and the comparisons of *SQNs* based on pattern-matching, we were able to prove the aforementioned lemmas by induction with a simple, general proof strategy. The security properties, however, require dedicated and involved proof strategies (~1000 LoC of Python).

4.4.3 Our models. Based on our modeling choices, we built a complete model of 5G AKA (preceded by the initialization protocol) that is amenable to automation. We model fully parametric compromise scenarios that enables one to easily choose what kind of reveals or compromises are considered when proving properties. We also implement the key confirmation roundtrip in a modular way: one can consider authentication properties after or without this roundtrip. The protocol model itself consists of ~500 LoC.

Our model includes all the necessary lemmas: helping lemmas, sanity check lemmas, and the lemmas we eventually use to check the relevant security properties against the 5G AKA protocol. Since

we aim at identifying the minimal assumptions required for the stated properties to hold, we prove several lemmas for each security property. First, we state a lemma showing that the property holds under a certain set of assumptions. Second, we show the minimality of this set of assumptions. We do this by disproving all versions of the previous lemma where the set of assumptions has been reduced by just one assumption. This represents ~1000 LoC.

Our model of 5G AKA is general in that it can be used to model different protocols from the AKA family with localized modifications. Part of the model (creation or role instantiations, reveal and compromise modelings, *etc.*) would not change, but the roughly 300 LoC defining the main flow of the protocol would have to be adapted. The size of this change depends on how different the chosen protocol is to 5G AKA. We expect our oracle to be still valid, at least after minor modifications to the model. Furthermore, given that our analysis is fully automatic (thanks to our proof strategies), our model can be easily kept up-to-date with future evolutions of the standard and any change in terms of provided security guarantees can then be automatically identified by the tool.

5 SECURITY ANALYSIS

5.1 Results

We present the results of our comprehensive analysis of the 5G AKA protocol. We emphasize that we *automatically* analyze the *formal security guarantees* that the protocol provides for an *unbounded number* of sessions executed by subscribers, *SNs*, and *HNs* when used *in combination* with the initiation protocol. Thus, our analyses account for all potential unintended interactions an attacker could exploit between these sub-protocols run by all those instantiations of the three roles we consider.

We depict the outcome of our analysis of authentication properties in Table 1. For each pair of parties, we present the minimal assumptions required to achieve authentication properties: *i.e.,* weak agreement, non-injective agreement, and injective agreement. We only consider agreement on relevant data (*i.e.,* K_{SEAF}, the *SUPI*, and *SNname*). The assumptions are minimal in that strengthening the attacker's capabilities in any direction violates the property. The symbol ✗ denotes that the property is violated for the weakest threat model we consider (*i.e.,* no compromised agents and an active attacker with no agent's secrets in its initial knowledge). Similarly, we present our results concerning secrecy properties in Table 2.

5.2 Discussion

Table 1 clearly shows the extent that the 5G standard underspecifies authentication requirements (recall that [·] denotes explicit goals); see Section 3.3. We also indicate a number of properties that are violated even in the weakest threat model where all participants are honest (✗). We discuss why in Section 5.2.1. Afterwards, we explain and critique the use of key-confirmation in Section 5.2.2. We discuss privacy properties in Section 5.2.3. Finally, our results concerning secrecy properties are as expected and are not discussed further. Also, perfect forward secrecy of K_{SEAF} is violated as expected.

5.2.1 Missing Security Assumption. The 5G AKA protocol fails to meet several security goals that are explicitly required as well as other critical security properties. This is still true under the

assumptions specified in the standard, even after a successful key-confirmation phase (see Table 1). More specifically, the agreement properties on K_{SEAF} between the subscribers and *SNs* are violated. So is weak agreement from the subscribers towards the *SNs*. This is caused by the lack of a binding assumption on the channel between *SNs* and *HNs* and because the *SUPI* is sent to the *SN* in a different message than the message containing K_{SEAF}, which is sent earlier. Therefore, as soon as a pair of a *SN* and a *HN* runs two sessions concurrently, there is no assurance that the *SUPI* the *SN* receives at the end of the protocol actually corresponds to the K_{SEAF} it has received earlier (it could correspond to another concurrent session). As a consequence, a *SN* may associate the session key K_{SEAF} to the correct subscriber (a necessary condition for the key confirmation to be successful), but to the wrong *SUPI*, violating the aforementioned properties. Thus, the binding property for the channel between the *SNs* and *HNs* appears to be a critical security assumption, and should be explicitly mentioned in the standard. This weakness has been introduced in the last version of the standard. In the previous version (`v0.7.1`), the *SUPI* was sent by the *HN* to the *SN* together with the challenge[3], thus preventing the aforementioned attack and therefore making the binding assumption unnecessary. However, the new version of the standard requires this additional assumption. A similar looking issue, but between two parts of the *HN* has been previously observed by [19], but it is an entirely independent concern of the one we describe.

Table 3 depicts additional security properties the 5G AKA protocol provides when the channel between the *SNs* and *HNs* is assumed to be binding. Under this assumption, the previously violated properties are now satisfied under reasonable threat models. We only show results for *UEs* and *SNs* to show how their guarantees change.

5.2.2 Implicit Authentication. A successful key-confirmation roundtrip is required to obtain crucial security guarantees. More precisely, this roundtrip is required for all agreement properties from the subscribers' point of view except weak agreement towards the *HNs*. Indeed, an attacker can impersonate a *SN* towards a subscriber but is unable to learn the K_{SEAF} key the subscriber has computed.

> **[TS 33.501, Sec. 5.1.2]** The meaning of 'implicit key authentication' here is that authentication is provided through the successful use of keys resulting from authentication and key agreement in subsequent procedures.

The 5G standard only requires *implicit* authentication properties for the subscribers. However, the standard neither specifies that subscribers must wait for this key confirmation to be successful before continuing nor does it specify this additional roundtrip *as part* of the authentication method. As a consequence, the standard makes a choice that we consider risky: it postpones the handling and the verification of the additional key confirmation roundtrip to *all possible subsequent procedures* (*e.g.,* NAS security mode command procedure [TS 33.501, Sec. 6.7.2]). The standard fails to specify a standalone authentication protocol that provides a reasonable set

---

[3]To the best of our knowledge, the rationale behind the new version is to let *HN* wait for the proof of the subscriber's recent aliveness before disclosing the corresponding *SUPI* to *SNs* that may be malicious or dishonest.

Point of view	UE				SN				HN			
Partner	SN		HN		UE		HN		UE		SN	
Agreement	NI	I	NI	I	NI	I	NI	I	NI	I	NI	I
on K_{SEAF}	✗	✗	$\neg K \wedge$k-c	$\neg K \wedge$k-c	✗	✗	$\neg$ch	$\neg K \wedge \neg$ch	$\neg K$	$\neg K$	$\neg$ch	$\neg$ch
on $SUPI$	wa	×	wa	×	wa	×	[$\neg$ch]	×	wa	×	×	×
on $SNname$	wa	×	[$\neg K \wedge$k-c]	×	wa	×	wa	×	[$\neg K$]	×	wa	×
Weak agreement	[✗]		$\neg K$		[$\neg K \wedge \neg$ch]		$\neg$ch		$\neg K$		$\neg$ch	

Table 1: Minimal assumptions required for 5G AKA to achieve authentication properties. We denote subscribers by *UE,* non-injective by NI and injective by I. Assumptions are expressed in terms of forbidden reveals (*e.g.,* $\neg K$, meaning the property only holds when K is not revealed). We also indicate whether a key confirmation phase is needed with k-c while $\neg$ch denotes an uncompromised channel between *SN* and *HN*. ×: the property is violated *by definition* (*e.g.,* because *SUPI* is constant). wa: the property coincides with weak agreement and requires same assumptions. The explicit goals given in the specification are denoted by [·] around them.

Point of view	UE	SN	HN
K_{SEAF}	$\neg K \wedge \neg$ch	$\neg K \wedge \neg$ch	$\neg K \wedge \neg$ch
PFS(K_{SEAF})	✗	✗	✗
$SUPI$	$\neg sk_{\text{HN}} \wedge \neg$ch*	−	$\neg sk_{\text{HN}} \wedge \neg$ch*
K	∅	∅	∅

Table 2: Minimal assumptions for 5G AKA to achieve secrecy properties. We omit the assumption that data that is supposed to be secret is not revealed. See Table 1 for the legend. The symbol * indicates that there is no dishonest *SNs* at all and the underlying property is always violated otherwise. PFS(·): perfect-forward secrecy. −: property not relevant.

P.o.V.	UE		SN	
Partner	SN		UE	
Agre.	NI	I	NI	I
on K_{SEAF}	$\neg K \wedge$k-c$\wedge \neg$ch	$\neg K \wedge$k-c$\wedge \neg$ch	$\neg K \wedge \neg$ch	$\neg K \wedge \neg$ch
Weak agre.	[$\neg K \wedge$k-c$\wedge \neg$ch]		[$\neg K \wedge \neg$ch]	

Table 3: Minimal assumptions required for 5G AKA to achieve authentication properties between *UEs* and *SNs,* assuming that the channel between *HNs* and *SNs* is binding. Agreements on *SUPI* and *SNname* are not impacted.

of security guarantees since some critical properties are provided only when the protocol is used in specific, appropriate contexts.

More importantly, since the standard makes the overall security of the authentication rest on subsequent procedures, it is very challenging, and out of the scope of the present paper, to assess if all currently specified subsequent procedures (as well as future ones that may be added) either correctly mandate the use of this key confirmation roundtrip or do not require authentication properties from the subscribers' point of view towards the *SNs*. We believe that there are at least two potential use cases where the above weakness represents a vulnerability. First, the standard specifies that *SNs* can initiate key change on-the-fly [TS 33.501, Sec. 6.9.4.1] as well as switch security contexts [TS 33.501, Sec. 6.8], including keys, parameters, *etc.*. This raises the question whether a malicious *SN* or a

fake base station could not *fully* impersonate a genuine *SN* towards the subscribers by changing the session key immediately after 5G AKA. Second, in a scenario where subscribers use the presence of *SNs* for geo-localization or for making sensitive decisions (related to *e.g.,* emergency calls), an active attacker could impersonate an *SN* since the (mismatched) K_{SEAF} key may not be needed or used.

Finally, the key confirmation roundtrip is not the only option to achieve the aforementioned missing security guarantees. We provide and discuss in Section 5.3.3 two alternative solutions that fix this issue while reducing neccessary communications.

5.2.3 On Privacy. As mentioned in Section 3.2.3, the 5G standard aims to protect privacy against passive attackers only. 5G AKA provides an identifier hiding mechanism and sends the *SUPI* only in a randomized public key encryption (the Subscription Concealed Identifier, *SUCI*). We show with TAMARIN that the *SUPI* indeed remains confidential, even against active attackers (see Table 2), hence also against passive attackers. 5G AKA defeats thus previous IMSI-catcher attacks[4] [32]. We also have modelled a weak, passive attacker and have automatically proven that he cannot trace subscribers.

We believe that active attackers are realistic threats for most use cases. Moreover, since privacy is a real concern to 3GPP, 5G AKA should protect subscribers' privacy also against active attackers. Unfortunately, we have found that this is not the case as the 5G AKA protocol suffers from a traceability attack.

Using TAMARIN (see our model [5]), we automatically find the following attack in 5G AKA. In this attack, the attacker observes one 5G AKA authentication session and later replays the *SN*'s message to some subscriber. From the subscriber's answer (MAC failure or Synchronization failure), the attacker can distinguish between the subscriber observed earlier (in case of Synchronization failure) and a different subscriber (in case of MAC failure). This attack can be exploited to track subscribers over time. A variant of this attack was first described in [6] for the AKA protocol as used in 3G.

5.3 Recommendations

Throughout the paper, we have highlighted weaknesses in the standard and suggested improvements and refinements. We now

---

[4]Those attacks relied on the subscribers sending IMSI (matching *SUPI* in 5G) in the clear.

summarize some of them and propose more precise, provably secure fixes as a replacement for the key-confirmation and the binding channel assumptions. Again, we emphasize the critical role played here by our formal interpretation of the standard and our formal analysis of the described 5G AKA protocol.

5.3.1 Explicit Requirements. As shown in Table 1 and discussed in Section 3.3, the standard underspecifies security requirements for the 5G AKA protocol. We suggest that the standard explicitly requires the missing intended security properties. In particular, it should be clear that 5G AKA aims at achieving *injective agreement on K_{SEAF} between the subscribers and the SNs* which is central to the protocol's purpose. The subscribers should obtain *injective agreement on K_{SEAF} with the HNs*; they are thereby assured the *HNs recently* authorized this session, since K_{SEAF} derives from the random *R*. Finally, the *HNs* should have *injective agreement on K_{SEAF} with the subscribers*, obtaining recent aliveness as a consequence.

5.3.2 Binding Channel. As discussed in Section 5.2.1, the last version of the standard introduced attacks under the given security assumptions. There are two solutions to fix this: either the standard explicitly states an additional security assumption (*i.e.,* the channel between the *SNs* and *HNs* must be binding), or alternatively the 5G AKA protocol is fixed (without the need for a new assumption) using the following minor modification: ⟨*SUPI, SUCI*⟩ is sent instead of *SUPI* in the final message from *HN* to *SN*. However, it is our understanding that the binding assumption is a property that is required for other reasons anyway, such as reliability.

5.3.3 On the Key-Confirmation. We already have discussed the danger of missing key-confirmation in 5G AKA in Section 5.2.2. We now propose two simple modifications to the protocol that would make key-confirmation redundant and unnecessary, therefore reducing the number of roundtrips that are needed to achieve intended security guarantees. Before explaining our fix, note that the key-confirmation was necessary in the first place because the *HNs* never commit to a specific *SNname* when computing the challenge *R*, *AUTN*. Only the key K_{SEAF} is bound to *SNname*, but the challenge itself is not.

Our first fix consists of binding *AUTN* to *SNname* so that subscribers directly have the proof the *HN* has committed to a specific *SNname*, without even using K_{SEAF}. Formally, *AUTN* currently refers to ⟨$SQN_{HN} \oplus AK$, *MAC*⟩ where *MAC* = f1(K, ⟨SQN_{HN}, R⟩). In our fix, *MAC* is replaced by f1(K, ⟨SQN_{HN}, R, *SNname*⟩). Therefore, the subscribers can verify the authenticity of the challenge that commits to a specific *SNname*. We have formally verified [5] that a key-confirmation roundtrip is no longer necessary with this fix.

Our second, alternative, fix consists in replacing the full key-confirmation roundtrip by an unidirectional key-confirmation from the *SN* only. More precisely, we could add (any) message MACed with a key derived from K_{SEAF}, sent by the *SNs* to the subscribers, at the very end of the protocol. We have proven with TAMARIN that no further guarantees are provided by a full key-confirmation, compared to our (less costly) unidirectional key-confirmation.

5.3.4 On Privacy. We recall that the functions f1 and f1* are not explicitly required to protect the confidentiality of their inputs (see Section 3.1.2). This is however necessary for privacy as these MAC functions take *SQN* as input, among others. If these functions were not confidentiality-preserving, a passive attacker could learn the subscribers' *SQNs* and perform location attacks [32] by tracking nearby *SQNs* over time or perform activity monitoring attacks [12].

We also recommend for the standard to explicitly aim at protecting privacy against active attackers and take steps in this direction. Unfortunately, this would involve significant modifications to the protocol since at least the failure reasons (MAC/Synchronization failure) must be hidden from the attacker [6, 22] and the *SQN* concealment mechanism should be strengthened against active attackers [12], possibly by using proper encryption or using an anonymity key *AK* based on subscriber-generated randomness. We leave a complete evaluation of possible solutions for future work and we expect our model to be valuable in this process.

5.3.5 Redundancies. A close look at the cryptographic messages (see detailed list in Appendix A) and their purposes shows many redundancies. For instance, in *RES**, the proof of possession of *K* is in *CK*, *IK*, and *RES*. *R* appears to be redundant as well. Similarly, *SNname* is redundant in the key derivation of K_{SEAF}. Legacy reasons may explain these redundancies, but these design choices could be questioned and the protocol simplified.

5.3.6 On the role of SQN. The purpose of the *SQN* counters is to provide replay protection for the subscribers. This mechanism was introduced in 3G, when the USIM was incapable of generating good randomness. This is no longer the case in 5G, where USIMs can perform randomized asymmetric encryption (required to compute *SUCI* from *SUPI*). Therefore, authentication protocols should be rethought and more standard challenge-response mechanisms could be used to replace the *SQN* counters. This would benefit the current authentication methods, which can suffer from de-synchronization and must keep the privacy sensitive *SQNs* up-to-date and sometimes fail to protect them against attackers (see Section 5.3.4).

5.3.7 On the benefits of Formal Methods. As argued throughout this section, the standard could be simplified and improved in various directions. We recall that formal models, such as our model of the 5G AKA protocol, have proven to be extremely valuable to quickly assess the security of such modifications and simplifications. Our model can serve as a basis to accompany the future evolution of the standard and can be used as a tool to quickly evaluate the security of modification proposals.

6 CONCLUSION

We have formally analyzed one of the two authentication methods in 5G, the one which enhances the previous variant currently used in 4G. This included a detailed analysis of the standard to identify all assumptions and security goals, a formal model of the protocol and security goals as specified in the standard, the automated security analysis using the TAMARIN prover, and a detailed discussion of our findings. Our models are substantially more detailed than those of previous work and account for detail of the state machine, counters, the re-synchronization procedures, and the XOR operations.

While analyzing the standard we discovered that security goals and assumptions are underspecified or missing in the standard, including central goals like agreement on the session key. Moreover, our analysis in TAMARIN shows that some properties are violated

without further assumptions, which is notably the case for agreement properties on the session key. We also critique the standard's choice of implicit authentication and the lack of key confirmation as this introduces weaknesses if the protocol is used incorrectly. Finally, our privacy analysis shows that the 5G version of AKA still fails to ensure unlinkability against an active attacker; this scenario is, in our opinion, completely realistic.

As future work, we intend to analyze other variants of the AKA protocol, notably those used in 3G and 4G networks, to see which security guarantees they provide compared to 5G AKA. We also plan to follow the future development of the 5G standard as our analysis can serve as the basis for improving the protocol's design, in particular to evaluate ideas and avoid regressions. For example, we identified one weakness that was introduced in revision v1.0.0, which was not present in v0.7.1. This is a major benefit of tool-based analysis of protocol design: once the model is constructed, one can quickly test changes and evaluate different design options.

REFERENCES

[1] 2018. The Tamarin Manual. https://tamarin-prover.github.io/manual/. (2018). Accessed: 2018-05-05.
[2] 2018. Tamarin prover. https://github.com/tamarin-prover/tamarin-prover/releases/tag/1.4.0. (2018). Accessed: 2018-05-08.
[3] 3GPP. 2018. 3G Security: Formal Analysis of the 3G Authentication Protocol. (2018). TS 33.902.
[4] 3GPP. 2018. Security architecture and procedures for 5G system. (2018). TS 33.501.
[5] Anonymous. 2018. Tamarin models, proofs and instructions for reproducibility. https://drive.google.com/open?id=1XizCcUWCGMDuxu5Z4TOEO-ZQXFjK07FN. (2018). Accessed: 2018-05-09.
[6] Myrto Arapinis, Loretta Mancini, Eike Ritter, Mark Ryan, Nico Golde, Kevin Redon, and Ravishankar Borgaonkar. 2012. New privacy issues in mobile telephony: fix and verification. In *Proceedings of the 2012 ACM conference on Computer and communications security.* ACM, 205–216.
[7] David Basin, Cas Cremers, and Simon Meier. 2013. Provably repairing the ISO/IEC 9798 standard for entity authentication. *Journal of Computer Security* 21, 6 (2013), 817–846.
[8] David Basin, Cas Cremers, Kunihiko Miyazaki, Sasa Radomirovic, and Dai Watanabe. 2015. Improving the security of cryptographic protocol standards. *IEEE Security & Privacy* 13, 3 (2015), 24–31.
[9] David Basin, Jannik Dreier, and Ralf Sasse. 2015. Automated symbolic proofs of observational equivalence. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security.* ACM.
[10] K. Bhargavan, B. Blanchet, and N. Kobeissi. 2017. Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate. In *2017 IEEE Symposium on Security and Privacy (SP).* 483–502. https://doi.org/10.1109/SP.2017.26
[11] Bruno Blanchet. 2016. Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif. *Foundations and Trends in Privacy and Security* 1, 1–2 (Oct. 2016), 1–135.
[12] R. Borgaonkar, L. Hirshi, S. Park, A. Shaik, A. Martin, and J.-P. Seifert. 2017. New Adventures in Spying 3G & 4G Users: Locate, Track, Monitor. (2017). https://www.blackhat.com/us-17/briefings.html#new-adventures-in-spying-3g-and-4g-users-locate-track-and-monitor Briefing at BlackHat USA 2017.
[13] Colin Boyd and Wenbo Mao. 1993. On a limitation of BAN logic. In *Workshop on the Theory and Application of of Cryptographic Techniques.* Springer, 240–247.
[14] Vincent Cheval, Steve Kremer, and Itsaka Rakotonirina. 2018. DEEPSEC: Deciding Equivalence Properties in Security Protocols - Theory and Practice. In *Proceedings of the 39th IEEE Symposium on Security and Privacy (S&P'18).* IEEE Computer Society Press, San Francisco, CA, USA. Accepted for publication.
[15] Katriel Cohn-Gordon, Cas Cremers, and Luke Garratt. 2016. On post-compromise security. In *Computer Security Foundations Symposium (CSF), 2016 IEEE 29th.* IEEE, 164–178.
[16] C. Cremers, M. Horvat, J. Hoyland, S. Scott, and T. van der Merwe. 2017. A Comprehensive Symbolic Analysis of TLS 1.3. In *ACM CCS 2017: Proceedings of the 24th ACM Conference on Computer and Communications Security, Dallas, USA, 2017.* To appear.
[17] Cas Cremers, Marko Horvat, Jonathan Hoyland, Sam Scott, and Thyla van der Merwe. 2017. A Comprehensive Symbolic Analysis of TLS 1.3. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu (Eds.). ACM, 1773–1788. https://doi.org/10.1145/3133956.3134063
[18] Cas Cremers, Marko Horvat, Sam Scott, and Thyla van der Merwe. 2016. Automated Analysis and Verification of TLS 1.3: 0-RTT, Resumption and Delayed Authentication. In *IEEE Symposium on Security and Privacy.*
[19] Martin Dehnel-Wild and Cas Cremers. 2018. Authentication vulnerability in the most recent 5G AKA drafts (February 2018). http://www.cs.ox.ac.uk/people/cas.cremers/tamarin/5G/. (February 2018).
[20] D. Dolev and Andrew C. Yao. 1981. On the security of public key protocols. *Information Theory, IEEE Transactions on* 29, 2 (March 1981), 198–208. https://doi.org/10.1109/TIT.1983.1056650
[21] Jannik Dreier, Lucca Hirschi, Sasa Radomirovic, and Ralf Sasse. 2018. Automated Unbounded Verification of Stateful Cryptographic Protocols with Exclusive OR. In *Accepted at Computer Security Foundations (CSF).*
[22] Pierre-Alain Fouque, Cristina Onete, and Benjamin Richard. 2016. Achieving better privacy for the 3GPP AKA protocol. *Proceedings on Privacy Enhancing Technologies* 2016, 4 (2016), 255–275.
[23] Nico Golde, Kévin Redon, and Jean-Pierre Seifert. 2013. Let Me Answer That for You: Exploiting Broadcast Information in Cellular Networks. In *22Nd USENIX Conference on Security.* USENIX Association.
[24] GSMA. 2017. Global Mobile Trends 2017. https://www.gsma.com/globalmobiletrends/. (2017). Accessed: 2018-05-06.
[25] Syed Rafiul Hussain, Omar Chowdhury, Shagufta Mehnaz, and Elisa Bertino. 2018. LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE. In *Network and Distributed Systems Security (NDSS) Symposium 2018.*
[26] Nadim Kobeissi, Karthikeyan Bhargavan, and Bruno Blanchet. 2017. Automated verification for secure messaging protocols and their implementations: A symbolic and computational approach. In *IEEE European Symposium on Security and Privacy (EuroS&P).*
[27] Robert Künnemann and Graham Steel. 2013. YubiSecure? Formal Security Analysis Results for the Yubikey and YubiHSM. In *Security and Trust Management*, Audun Jøsang, Pierangela Samarati, and Marinella Petrocchi (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 257–272.
[28] Gavin Lowe. 1997. A Hierarchy of Authentication Specifications. In *10th Computer Security Foundations Workshop.* IEEE Computer Society Press.
[29] Simon Meier. 2013. *Advancing automated security protocol verification.* Ph.D. Dissertation. ETH Zurich.
[30] Simon Meier, Benedikt Schmidt, Cas J. F. Cremers, and David Basin. 2013. The TAMARIN Prover for the Symbolic Analysis of Security Protocols. In *CAV (LNCS),* Vol. 8044. Springer, 696–701.
[31] Piers O'Hanlon, Ravishankar Borgaonkar, and Lucca Hirschi. 2017. Mobile subscriber WiFi privacy. In *Proceedings of Mobile Security Technologies (MoST'17), held as part of the IEEE Computer Society Security and Privacy Workshops (SPW'17).* To appear.
[32] David Rupprecht, Adrian Dabrowski, Thorsten Holz, Edgar Weippl, and Christina Pöpper. 2017. On Security Research towards Future Mobile Network Generations. *arXiv preprint arXiv:1710.08932* (2017).
[33] Benedikt Schmidt, Simon Meier, Cas Cremers, and David Basin. 2012. Automated analysis of Diffie-Hellman protocols and advanced security properties. In *Computer Security Foundations Symposium (CSF).* IEEE, 78–94.
[34] Benedikt Schmidt, Ralf Sasse, Cas Cremers, and David Basin. 2014. Automated Verification of Group Key Agreement Protocols. In *Proceedings of the 2014 IEEE Symposium on Security and Privacy (SP '14).* IEEE Computer Society, Washington, DC, USA, 179–194. https://doi.org/10.1109/SP.2014.19
[35] Altaf Shaik, Jean-Pierre Seifert, Ravishankar Borgaonkar, N. Asokan, and Valtteri Niemi. 2016. Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems. In *23nd Annual Network and Distributed System Security Symposium, NDSS.*
[36] Fabian van den Broek, Roel Verdult, and Joeri de Ruiter. 2015. Defeating IMSI Catchers. *Proceedings of the 2015 ACM Conference on Computer and Communications Security - CCS '15* (2015).

A   NOTATIONS AND ACRONYMS

We list all the acronyms we introduced throughout the paper in Table 4 and give a correspondence table between our simplified terminology and the 3GPP terminology in Table 5.

We describe the cryptographic messages format in Table 6 (abstracting away AMF, other constants and sub-message lengths).

B   THE EAP-AKA' PROTOCOL

We depict the core flow of the EAP-AKA' protocol in Figure 4. We omit the MAC failure and Re-synchronization failure phases that are the same as for 5G-AKA (see Section 2.2.2 and Figure 3). We also omit the message headers specific to the EAP framework such as 'EAP Request' and 'EAP Success'. The key derivation is a bit different compared to 5G AKA. K_{SEAF} is derived from K_{AUSF} exactly as in 5G AKA:

$$K_{\text{SEAF}} = \text{KDF}(K_{\text{AUSF}}, SNname)$$

but K_{AUSF} is derived differently: $K_{\text{AUSF}} = \text{MK}[1152...1407]$ (we write $[x..y]$ for the substring from bit x to y) where the master key MK is:

$$\text{KDF}(\langle \text{KDF}(IK, SNname), \text{KDF}(CK, SNname)\rangle, \langle\text{'EAP-AKA''}, SUPI\rangle).$$

Therefore,

$$
\begin{aligned}
K_{\text{SEAF}} \quad = \quad & \text{KeySeed}(K, R, SQN_{\text{HN}} SNname, SUPI)\\
= \quad & \text{KDF}(\text{KDF}(\langle \text{KDF}(IK, SNname), \text{KDF}(CK, SNname)\rangle,\\
& \quad \langle\text{'EAP-AKA''}, SUPI\rangle)[512..767],\\
& \quad SNname).
\end{aligned}
$$

The messages AT_MAC are MAC messages over the other sub-messages as part of the same message. The key in use is $K_{\text{aut}} = \text{MK}[1152...1407]$.

Conceptually, the main difference of EAP-AKA' compared to 5G AKA are as follows:

- the challenge $xRES$ does not directly bind the SN's identity $SNname$. However, since the challenge is MACed (with the session key K_{aut}) together with $SNname$, both are *de facto* bound together.
- SN serves as a pass-through until the authentication is considered successful by the HN. Only at this time SN obtains $SUPI$ and K_{SEAF} from the HN, while it obtains K_{SEAF} already in the first message in 5G AKA.

C   LOWE'S TAXONOMY AND TAMARIN MODELING

Lowe's Taxonomy [28] notably defines *aliveness, recent aliveness, weak agreement, non-injective agreement*, and *injective agreement*. After an introductory example showing how properties are typically modelled in TAMARIN, we show how aliveness and non-injective agreement properties are modelled. The process for the other properties is similar.

C.1   Introductory Example: Secrecy

As an introductory example, let us see how secrecy properties are modeled in TAMARIN. For instance, we model the property that the $SUPI$ of subscribers is never revealed to the attacker. Formally, such a property is formalized in TAMARIN using the formula defined below, where facts $\text{Claim\_Secret}(a, k)$ are produced for each rule of agent a (some subscriber or some HN) who accesses or stores the identifier $SUPI$. Note that $\text{K}(t)$ denotes the fact that t is in the attacker's Knowledge.

Definition C.1. Secrecy is modeled via the following formula:

$$\forall a\ t\ \#i.\ \text{Claim\_secret}(a, t)@i \Rightarrow \neg(\exists \#j.\ \text{K}(t)@j).$$

C.2   Aliveness

Lowe defines *aliveness* as follows (excerpt from [28]):

> We say that a protocol guarantees to an initiator A aliveness of another agent B if, whenever A (acting as initiator) completes a run of the protocol, apparently with responder B, then B has previously been running the protocol.

Let us see how this property is mathematically modelled in TAMARIN. We assume that the TAMARIN model is equipped with facts $\text{Claim\_commit}(a, b, \langle A, B\rangle)$ (i.e., an agent a of role A claims it has established aliveness of b whose role is B) and $\text{Claim\_running}(b, B)$ (i.e., an agent b of role B claims it has run the protocol).

Definition C.2. Aliveness of a role A towards a role B is modeled via the following formula:

$$
\begin{aligned}
\forall a\ b\ \#i.\ \text{Claim\_commit}(a, b, \langle A, B\rangle)@i \Rightarrow\\
(\exists \#j.\ \text{Claim\_running}(b, B)@j).
\end{aligned}
$$

Note that we do not restrict the timestamp $\#j$ to be before the timestamp $\#i$ (e.g., with a constraint $\#j \lessdot \#i$) since the trace semantics of TAMARIN already accounts for this constraint. More precisely, if, for some execution, there was a fact $\text{Claim\_running}(b, B)@j$ but only after the fact $\text{Claim\_commit}(a, b, \langle A, B\rangle)@i$ (i.e., $\#i \lessdot \#j$), then it would suffice, for falsifying the property, to consider a prefix of the considered execution that contains $\text{Claim\_commit}(a, b, \langle A, B\rangle)@i$ but not $\text{Claim\_running}(b, B)@j$.

C.3   Non-injective Agreement

Another example is the *non-injective agreement* property, defined as follows (excerpt from [28]):

> We say that a protocol guarantees to an initiator A non-injective agreement with a responder B on a set of data items t_s (where t_s is a set of free variables appearing in the protocol description) if, whenever A (acting as initiator) completes a run of the protocol, apparently with responder B, then B has previously been running the protocol, apparently with A, and was acting as responder in his run, and the two agents agreed on the data values corresponding to all the variables in t_s.

We assume that the TAMARIN model is equipped with facts $\text{Claim\_commit\_agr}(a, b, \langle A, B, t_s\rangle)$ (i.e., an agent a of role A claims it has established agreement on data t with b whose role is B) and $\text{Claim\_running\_agr}(b, a, \langle A, B, t_s\rangle)$ (i.e., an agent b of role B claims it tries to establish agreement on data t with a whose role is A). The above property is modelled as follows.

Acronym	Full name	Reference	Description (if needed)
AK	Anonymity Key	Section 2.2	
AMF	Authentication Management Field	Appendix A	
AMF$^{(*)}$	Access and Mobility Management Function	Appendix A	
ARPF	Authentication credential Repository and Processing Function	Appendix A	
AUSF	Authentication Server Function	Appendix A	
EAP	Extensible Authentication Protocol	Section 2.2	
IMSI	International Mobile Subscriber Identity	Section 2.1	Uniquely identity subscribers
gNB	NR Node B	Appendix A	new generation base station
GUTI	Globally Unique Temporary UE Identity	Section 4.3	
ME	Mobile Equipment	Section 2.1	
MNC	Mobile Country Code	Appendix A	Uniquely identify *HNs*' countries
MNC	Mobile Network Code	Appendix A	Uniquely identity *HNs* in a country
RAND (*R*)	Random Challenge	Section 2.2	
RES	RESponse	Appendix A	
SEAF	SEcurity Anchor Function	Appendix A	
SUCI	Subscription Concealed Identifier	Section 2.2	
SUPI	Subscription Permanent Identifier	Section 2.1	Uniquely identity a subscriber and its *HN*
SQN	SeQuence Number	Section 2.1	
UDM	Unified Data Management	Appendix A	
USIM	Universal Subscriber Identity Module	Appendix A	
XRES	eXpected RESponse	Appendix A	
SIDF	Subscription Identifier De-concealing Function	Appendix A	
SUPI	Subscription Concealed Identifier	Section 2.1	
SUCI	Subscription Permanent Identifier	Section 2.2	Randomized encryption of *SUPI*
SNid	SN identity (?)	Appendix A	Uniquely identity *SNs*

Table 4: Acronyms and abbreviations (mostly from [TS 33.501, Sec. 3.2])

Our notion	Correspondent notion in TS33.501
Serving Network	Combination of SEAF, AMF$^{(*)}$ and gNB
Home Network	Combination of AUSF, ARPF, UDM and SIDF

Table 5: Correspondence between our terminology with the one of TS33.501

Definition C.3. Non-injective agreement on data *t* of a role *A* towards a role *B* is modeled via the following formula:

$$\forall a\, b\, t\, \#i.\ \mathsf{Claim\_commit\_agr}(a, b, \langle A, B, t_s\rangle)@i \Rightarrow \\ (\exists \#j.\ \mathsf{Claim\_running\_agr}(b, a, \langle A, B, t_s\rangle)@j).$$

D SECURITY ASSUMPTIONS AND GOALS

This section extends Section 3.

This section is dedicated to our interpretation of security assumptions and goals that are relevant to authentication methods in 5G as precise formal statements. We shall support our interpretation by relevant excerpts from Technical Specification (TS) documents and Technical Requirement (TR) documents issued by 3GPP. Note that we may cite documents specifying aspects of earlier generations (3G and 4G) when relevant.

D.1 Security Assumptions and Threat Model

D.1.1 Assumptions on Channels.

Channel SN-HN. As part of the E2E core network, the channel between the *SN* and *HN* is supposed to provide confidentiality, integrity, authenticity, and, replay protection. Those assumptions are explicitly specified:

[TS 33.501, Sec. 5.9.3] Requirements for E2E core network interconnection security:
- The solution shall provide confidentiality and/or integrity end-to-end between source and destination network for specific message elements identified in the present document. For this requirement to be fulfilled, the SEPP - cf [2], clause 6.2.17 shall be present at the edge of the source and destination networks dedicated to handling e2e Core Network Interconnection Security. The confidentiality and/or integrity for the message elements is provided between two SEPPs of the source and destination PLMN-.

Message Name	Content	Internal Ref.	Specification
SUPI	$\langle IMSI, MMC, MNC \rangle$	Section 2.1	[TS23.501, Sec. 5.9.2]
SUCI	$\langle \text{aenc}(\langle SUPI, R_s \rangle, pk_{\text{HN}}), MMC, MNC \rangle$	Section 2.2	[TS 33.501, Sec. C.3]
SNname	$\langle '5G', ':', SNid \rangle$	Section 2.1	[TS 33.501, Sec. 6.1.1.4]
MAC	$\text{f1}(K, \langle SQN_{\text{HN}}, R \rangle)$	Section 2.2	[TS 133.102, Sec. 6.3.2]
AK	$\text{f5}(K, R)$	Section 2.2	[TS 133.102, Sec. 6.3.2]
AUTN	$\langle SQN_{\text{HN}} \oplus AK, MAC \rangle$	Section 2.2	[TS 33.501, Sec. 6.1.3]
RES	$\text{f2}(K, R)$	Section 2.2	[TS 133.102, Sec. 6.3.2]
CK	$\text{f3}(K, R)$	Appendix A	[TS 133.102, Sec. 6.3.2]
IK	$\text{f4}(K, R)$	Appendix A	[TS 133.102, Sec. 6.3.2]
*RES**	$\text{KDF}(\langle CK, IK \rangle, \langle SNname, R, RES \rangle)$	Section 2.2	[TS 33.501, Sec. A.4]
Challenge$(K, R, SNname)$	*RES**	Section 2.2	None
*HXRES**	$\text{SHA256}(\langle R, RES^* \rangle)$	Section 2.2	[TS 33.501, Sec. A.5]
MACS	$\text{f1}^*(K, \langle SQN_{\text{UE}}, R \rangle)$	Section 2.2	[TS 133.102, Sec. 6.3.3]
AKS	$\text{f5}^*(K, R)$	Section 2.2	[TS 133.102, Sec. 6.3.3]
AUTS	$\langle SQN_{\text{UE}} \oplus AKS, MACS \rangle$	Section 2.2	[TS 133.102, Sec. 6.3.3]
K_{AUSF}	$\text{KDF}(\langle CK, IK \rangle, \langle SNname, SQN \oplus AK \rangle)$	Section 2.2	[TS 33.501, Sec. A.2]
K_{SEAF}	$\text{KDF}(K_{\text{AUSF}}, SNname)$	Section 2.2	[TS 33.501, Sec. A.6]
KeySeed$(K, R, SQN_{\text{HN}}, SNname)$	K_{SEAF}	Section 2.2	None

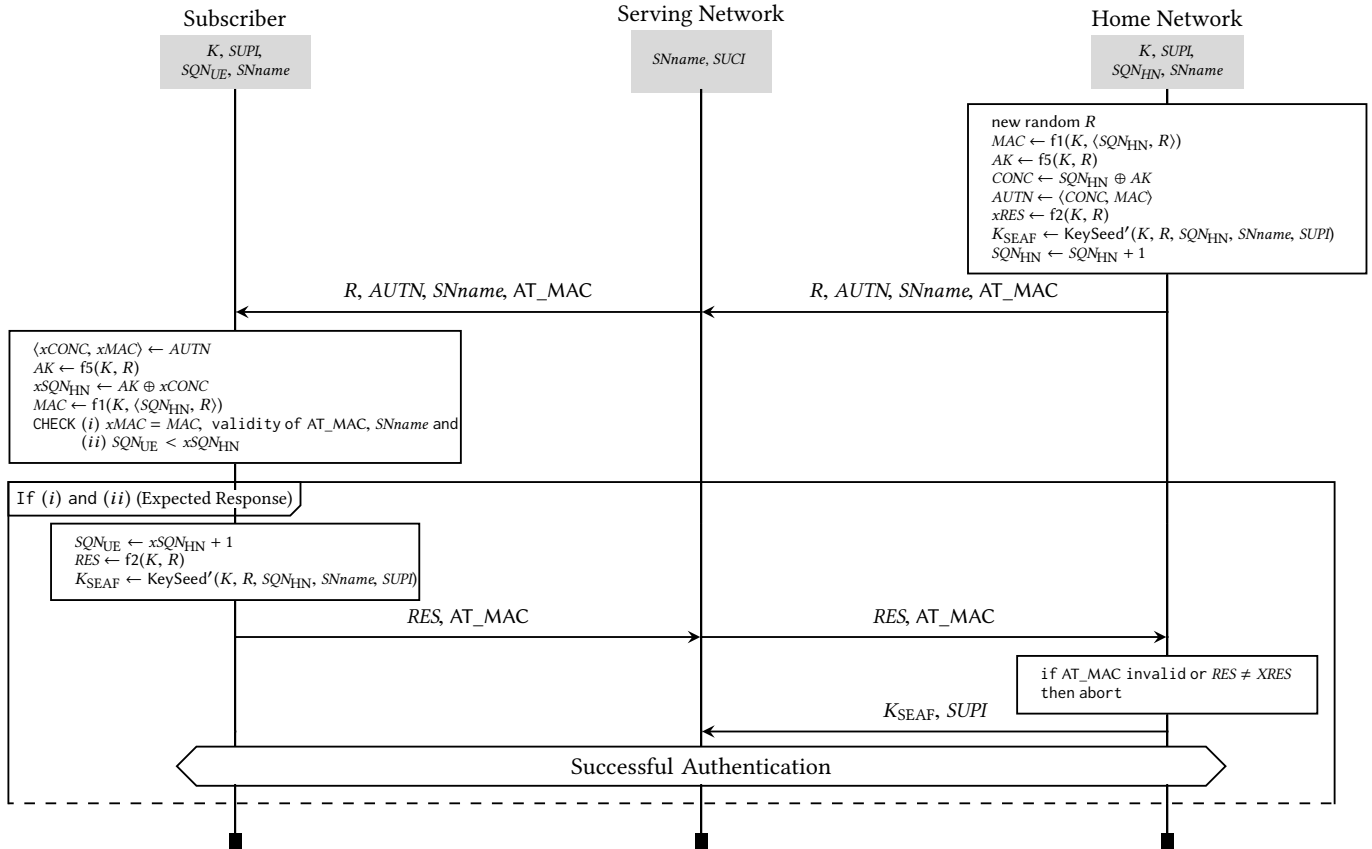**Table 6: Notations and Messages**

Figure 4: The EAP-AKA' protocol (continuing Figure 2). AT_MAC denotes a MAC over the other messages with key K**.**

- The destination network shall be able to determine the authenticity of the source network that sent the specific message elements protected according to the preceding bullet. For this requirement to be fulfilled, it shall suffice that a SEPP in the destination network that is dedicated to handling e2e Core Network Interconnection Security can determine the authenticity of the source network.
- The solution shall cover prevention of replay attacks.

Channel Subscribers-SNs. The channel between the subscribers and *SNs*, on the radio physical layer, is subject to eavesdropping (by passive attackers) or manipulations, interception, and injection of messages (by an active attacker). A *passive attacker* listens to signaling messages (*i.e.,* messages sent on the radio physical layer) on specific bandwidths and can therefore easily eavesdrop on all messages exchanged in its vicinity. An *active attacker* sets up a fake base station to receive and send signaling messages; *e.g.,* to impersonate *SNs*. While no 5G-specific hardware is publicly available yet, we recall how easily an attacker can set-up fake base stations in 4G using open-source and freely available software and hardware [23, 35]. From now on, we shall consider active attackers, except when explicitly stated otherwise.

D.1.2 Assumptions on Cryptographic Primitives. According to [TS 33.102, Sec. 3.2,6.3.2], the functions f1, f1*, f2 are message authentication functions while f3, f4, f5, f5* are key derivation functions. To the best of our knowledge, there is no standardized, explicit security requirements for these functions. One could infer from the informal presentation [TS 33.102, Sec. 3.2] that the former are integrity protected and the latter are integrity and confidentiality protected. However, since f1 and f1* are used to MAC sensitive pieces of data such as *SQN* (see the Section dedicated to privacy in Appendix D.2.3), it is our understanding that they should additionally preserve the confidentiality of their inputs.

Therefore, we assume f1, f1*, f3, f4, f5, f5* are integrity and confidentiality protected while f2 is integrity protected. We also stress that f1 and f1* are underspecified.

D.1.3 Assumptions on Parties. We consider compromised scenarios in order to provide stronger and more fine-grained guarantees. Our analyses will be parametrized by those compromised scenarios; in the worst case, a property will hold only when the attacker cannot compromise any agent. First, we consider an attacker who can compromise certain *SNs*. This means that the attacker gets access to an authenticated channel between the compromised *SN* and *HNs*, which he can use to eavesdrop on and inject messages. This is a reasonable assumption in 5G, where authentication methods should provide security guarantees even in presence of genuine but malicious *SNs*. In such situations, the *HNs* may cooperate with such *SNs* to authenticate some subscriber. In practice, this may happen in roaming situations. The following excerpt shows that in 5G, this is a threat model that should be considered (home refers to *HN* and visited network refers to *SN*):

[TS 33.501, Sec. 6.1.4.1] Increased home control: The authentication and key agreement protocols mandated to provide increased home control [compared to previous generations]. The feature of increased home control is useful in preventing certain types of fraud, e.g. fraudulent Nudm_UECM_Registration Request for registering the subscriber's serving AMF in UDM that are not actually present in the visited network.

Furthermore, we consider that the attacker may have genuine USIMs and compromised USIMs under its control. For those compromised subscribers, the attacker can access all secret values stored in the USIMs; *i.e., SUPI, K,* and *SQN*. Finally, the attacker can access all long-term secrets K, sk_{HN}, and *SUPI* from compromised *HNs*.

D.1.4 Assumptions on Data Protection.

Subscriber credentials. The subscriber credentials, notably the key K and the identifier *SUPI*, shared between subscribers and *HNs*, should be initially secret (provided they belong to uncompromised agents):

[TS 33.501, Sec. 3.1] Subscription credential(s): set of values in the USIM and the ARPF, consisting of at least the long-term key(s) and the subscription identifier SUPI, used to uniquely identify a subscription and to mutually authenticate the UE and 5G core network.

[TS 33.501, Sec. 5.2.4] The following requirements apply for the **storage and processing of the subscription credentials** used to access the 5G network:
- The subscription credential(s) shall be integrity protected within the UE using a tamper resistant secure hardware component.
- The long-term key(s) of the subscription credential(s) (i.e. K) shall be confidentiality protected within the UE using a tamper resistant secure hardware component.
- The long-term key(s) of the subscription credential(s) shall never be available in the clear outside of the tamper resistant secure hardware component.
- The authentication algorithm(s) that make use of the subscription credentials shall always be executed within the tamper resistant secure hardware component.
- It shall be possible to perform a security evaluation / assessment according to the respective security requirements of the tamper resistant secure hardware component.

NOTE: The security assessement scheme used for the security evaluation of the tamper resistant secure hardware component is outside the scope of 3GPP specifications.

Sequence Number. The sequence number *SQN* is a 48-bit counter (a 43-bits counter in some situations, see [TS 33.102, Sec. C.3.2]), therefore guessable with a very low probability. We consider a reasonable threat model where the value of *SQN* is unknown to the attacker when the attack starts, but the attacker knows how it is incremented during the attack. This corresponds to an attacker who

(i) can monitor the activity of targeted subscribers in its vicinity during the attack but, (ii) cannot guess the initial value of *SQN*, (iii) nor he can monitor targeted subscribers all the time (*i.e.,* from the very first use of the USIM up to the attack time).

> **[TS 33.102, Sec. 6.3.7]** (3G) Sequence numbers (SQN) shall have a length of 48 bits.

Other data. While not explicitly stated in the specification, we shall assume that the private asymmetric key sk_{HN} is initially private.

D.2 Security Requirements

We now extract and interpret from the 5G documents the security goals the authentication method 5G-AKA should achieve according to the 5G standard.

D.2.1 Authentication Properties. 5G specifications make semi-formal claims about authentication properties at different places in the documents. We have identified relevant claims and translate them into formal security goals, indicated in *purple and cursive text*. When doing this, we rely on Lowe's taxonomy of authentication properties [28]. The first benefit is that the Lowe's taxonomy provides precise properties that are now well established and understood, which can very often clarify an ambiguity [8]. The second benefit is that there exists a formal relation between the Lowe's taxononmy and mathematical definitions of security properties that can be directly modeled in TAMARIN [1]. We give an overview of this taxonomy and its relation with mathematical formulations in Appendix C. Intuitively, it specifies, from an agent A's point of view, four levels of authentication between two agents A and B: (i) aliveness, which only ensures that B has been running the protocol previously, but not necessarily with A; (ii) weak agreement, which ensures that B has previously been running the protocol with A, but not necessarily with the same data; (iii) non-injective agreement, which ensures that B has been running the protocol with A and both agree on the data; and (iv) injective agreement, which additionally ensures that for each run of the protocol of an agent there is a unique matching run of the other agent, which prevents replay attacks.

We start by recalling the (informal) goals of authentication methods in 5G:

> **[TS 33.501, Sec. 6.1.1.1]** The purpose of the primary authentication and key agreement procedures is to enable mutual authentication between the UE and the network and provide keying material that can be used between the UE and network in subsequent security procedures. The keying material generated by the primary authentication and key agreement procedure results in an anchor key called the KSEAF provided by the AUSF of the home network to the SEAF of the serving network.

As we shall see, 5G aims at providing stronger guarantees than in older generations, *e.g.,* than in 3G:

> **[TS 133.102, Sec. 5.1.2]** (3G) The following security features related to entity authentication are provided:
> - user authentication: the property that the serving network corroborates the user identity of the user;
> - network authentication: the property that the user corroborates that he is connected to a serving network that is authorised by the user's HE to provide him services; this includes the guarantee that this authorisation is recent.

We now list the security goals in terms of authentication by pairs of entities. Note that the specification considers some authentication properties to be *implicit*. This means that the guarantee is provided only after an additional *key confirmation roundtrip* (w.r.t. K_{SEAF}) between the subscribers and the *SN*. We discuss and criticize this design choice in Section 5.

Authentication between subscribers and HNs. First, the subscribers must have the assurance that authentication can only be successful with *SNs* authorized by their *HNs*.

> **[TS 33.501, Sec. 5.1.2]** **Serving network authorization by the home network:** Assurance shall be provided to the UE that it is connected to a serving network that is authorized by the home network to provide services to the UE. This authorization is 'implicit' in the sense that it is implied by a successful authentication and key agreement run.

> **[TS 33.501, Sec. 6.1.1.3]** The binding to the serving network prevents one serving network from claiming to be a different serving network, and thus provides implicit serving network authentication to the UE.

Formally, a *subscriber must obtain non-injective agreement on SNname with its HN after key confirmation.*

In 5G, the trust assumptions are balanced differently than in previous standards (*e.g.,* 3G or 4G). Most notably, the level of trust the system needs to put into *SNs* has been reduced. One important property provided by 5G is that a *SN* can no longer fake authentication requests with the *HNs* for subscribers that are not attached to one of its base station:

> **[TS 33.501, Sec. 6.1.4.1]** **Increased home control:** The authentication and key agreement protocols mandated to provide increased home control [compared to previous generations]. The feature of increased home control is useful in preventing certain types of fraud, e.g. fraudulent Nudm_UECM_Registration Request for registering the subscriber's serving AMF in UDM that are not actually present in the visited network.

Formally, *the HNs obtain aliveness of its subscribers at that SN, which is non-injective agreement on SNname from the HNs' point of view with the subscribers.*

Authentication between subscribers and SNs. As expected, the *SNs* shall be able to authenticate the subscribers:

[TS 33.501, Sec. 5.1.2] **Subscription authentication**: The serving network shall authenticate the Subscription Permanent Identifier (SUPI) in the process of authentication and key agreement between UE and network.

Formally, *the SNs must obtain non-injective agreement on SUPI with the subscribers, which is weak agreement from the SNs towards subscribers* (since the *SUPI* is the subscriber's identifier).

Conversely, the subscribers shall be able to authenticate the *SNs*:

[TS 33.501, Sec. 5.1.2] **Serving network authentication**: The UE shall authenticate the serving network identifier through implicit key authentication.
    NOTE 1: The meaning of 'implicit key authentication' here is that authentication is provided through the successful use of keys resulting from authentication and key agreement in subsequent procedures.
    NOTE 2: The preceding requirement does not imply that the UE authenticates a particular entity, e.g. an AMF, within a serving network.

Formally, and because *SNname* is the *SN*'s identifier, *the subscribers must obtain weak agreement with the SNs after key confirmation.*

Authentication between SNs and HNs. The *SNs* shall be able to authenticate the subscribers that are authorized by their corresponding *HN*:

[TS 33.501, Sec. 5.1.2] **UE authorization**: The serving network shall authorize the UE through the subscription profile obtained from the home network. UE authorization is based on the authenticated SUPI.

Formally, *the SNs must obtain non-injective agreement on SUPI with the HNs*.

D.2.2   Confidentiality Properties. While it is not clearly specified, it is obviously the case that 5G authentication methods should achieve *secrecy of K_{SEAF}, K, and sk_{HN}*. We recall similar goals for 3G:

[TS 133.102, Sec. 5.1.3] (3G) The following security features are provided with respect to **confidentiality of data** on the network access link:
  - cipher algorithm agreement: the property that the MS and the SN can securely negotiate the algorithm that they shall use subsequently;
  - cipher key agreement: the property that the MS and the SN agree on a cipher key that they may use subsequently;
  - confidentiality of user data: the property that user data cannot be overheard on the radio access interface;
  - confidentiality of signalling data: the property that signalling data cannot be overheard on the radio access interface.

5G should ensure that knowing the K_{SEAF} established in a certain session is insufficient to deduce a K_{SEAF} key that has been

established in a previous session or that will be established in a later session:

[TS 33.501, Sec. 3] **backward security**: The property that for an entity with knowledge of K_n, it is computationally infeasible to compute any previous K_{n-m} $(m > 0)$ from which K_n is derived.
    NOTE 5: In the context of K_{gNB} key derivation, backward security refers to the property that, for a gNB with knowledge of a K_{gNB}, shared with a UE, it is computationally infeasible to compute any previous K_{gNB} that has been used between the same UE and a previous gNB.

[TS 33.501, Sec. 3] **forward security**: The property that for an entity with knowledge of Km that is used between that entity and a second entity, it is computationally infeasible to predict any future K_{m+n} $(n > 0)$ used between a third entity and the second entity.
    NOTE 6: In the context of K_{gNB} key derivation, forward security refers to the property that, for a gNB with knowledge of a K_{gNB}, shared with a UE, it is computationally infeasible to predict any future K_{gNB} that will be used between the same UE and another gNB. More specifically, n hop forward security refers to the property that a gNB is unable to compute keys that will be used between a UE and another gNB to which the UE is connected after n or more handovers $(n = 1$ or more$)$.

Since we do not consider the full key hierarchy and how K_{gNB} can be derived from K_{SEAF}, we shall consider those properties for K_{SEAF} directly. Formally, it should be the case that K_{SEAF} *established in a given session remains confidential even when the attacker learns the K_{SEAF} keys established in all other sessions*. Note that this is different from *forward secrecy* and *post-compromise secrecy [15]* which fail to hold as we shall see in Section 5.1.

Note that some other confidentiality properties are considered to be privacy properties (see Appendix D.2.3).

D.2.3   Privacy Properties. We first emphasize the importance given to privacy in 5G:

[TR 33.899, Sec. 4.1,4.2] Subscription privacy deals with various aspects related to the protection of subscribers' personal information, e.g. identifiers, location, data, etc. [...] The security mechanisms defined in NextGen shall be able to be configured to protect subscriber's privacy.

[TR 33.899, Sec. 5.7.1] The subscription privacy is very important area for Next Generation system as can be seen by the growing attention towards it, both inside and outside the 3GPP world.
    Outside the 3GPP, an alliance of mobile network operators, vendors, and universities called NGMN [9] has identified security and privacy as an enabler and essential value proposition of NextGen system and has presented that built-in privacy should be included as a design principle [10]. Similarly, a 5G PPP project called 5G-Ensure [11] has also identified privacy

as one of the topmost priorities for the NextGen system stating that the privacy has an important social impact [12]. [...]

> **[TS 33.501, Sec. F.2]** EAP-AKA' includes optional support for identity privacy mechanism that protects the privacy against passive eavesdropping.

This important role given to privacy can be explained by numerous and critical attacks that have breached privacy (*e.g.,* with IMSI-catcher [35, 36]) in previous generations; see the survey [32].

We also recall that privacy was already a concern in 3G:

> **[TS 133.102, Sec. 5.1.1]** (3G) The following security features related to user identity confidentiality are provided:
> - **user identity confidentiality**: the property that the permanent user identity (IMSI) of a user to whom a services is delivered cannot be eavesdropped on the radio access link;
> - **user location confidentiality**: the property that the presence or the arrival of a user in a certain area cannot be determined by eavesdropping on the radio access link;
> - **user untraceability**: the property that an intruder cannot deduce whether different services are delivered to the same user by eavesdropping on the radio access link.

Thus, already for 3G, user identity confidentiality, anonymity, and untraceability were security requirements. However, those properties were required against a passive attacker only (we discuss and criticize this restriction to a passive attacker in Section 5). Note that anonymity and untraceability (often called unlinkability) are not clearly defined. We propose formalization in Section 4.3.

We now list more precise statements specifying how privacy should be protected in 5G.

Confidentiality of SUPI. In 5G, the *SUPI* is considered sensitive and must remain secret since it uniquely identifies users. Indeed, an attacker who would be able to obtain this value from a subscriber would be able to identify him, leading to classical user location attacks (*i.e.,* see [TS 133.102, Sec. 5.1.1] above), much like IMSI-catcher attacks.

> **[TS 33.501, Sec. 5.2.5]** The SUPI should not be transferred in clear text over 5G RAN except routing information, e.g. Mobile Country Code (MCC) and Mobile Network Code (MNC).

> **[TS 33.501, Sec. 6.12]** **Subscription identifier privacy:** In the 5G system, the globally unique 5G subscription permanent identifier is called SUPI as defined in 3GPP TS 23.501 [2]. The SUCI is a privacy preserving identifier containing the concealed SUPI. [...]

> **[TS 133.102, Sec. 5.1.1]** (3G) **User identity confidentiality** (see above).

Formally, *the SUPI shall remain secret in the presence of a passive attacker*.

Confidentiality of SQN. For similar reasons, *SQN* must remain secret. An additional reason that is not explicitly stated is that *SQN* leaks the number of successful authentications the corresponding USIM has performed since it was manufactured, which is strongly correlated to its age and activity. This is even more critical when the attacker learns *SQN* at different times.

> **[TS 33.102, Sec. 6.2.3]** (3G) Here, AK is an anonymity key used to conceal the sequence number as the latter may expose the identity and location of the user. The concealment of the sequence number is to protect against passive attacks only. If no concealment is needed then f5 ≡ 0 (AK = 0).

> **[TS 133.102, Sec. C.3.2]** (3G) **User anonymity**: the value of SQN may allow to trace the user over longer periods. If this is a concern then SQN has to be concealed by an anonymity key as specified in section 6.3.

Formally, *the SQN shall remain secret in the presence of a passive attacker*.

Anonymity and Untraceability. Preventing the attacker from learning pieces of data that are identifying (*e.g., SUPI, SQN*) is insufficient to protect against traceability attacks, user location attacks, or even anonymity attacks (we explain why and discuss definitions in Section 4.3). While no formal statement is made on the necessity of ensuring untraceability or anonymity for 5G, the following excerpts and the fact that it was required for 3G ([TS 133.102, Sec. 5.1.1], see above), seem to imply that those properties are relevant for 5G as well.

On untraceability (also called unlinkability): [TS 133.102, Sec. 5.1.1], item 2 and:

> **[TS 33.501, Sec. C.2]** The reason for mentioning the nonfreshness is that, normally, in order to attain **unlinkability** (i.e., to make it infeasible for over-the-air attacker to link SUCIs together), it is necessary for newly generated SUCIs to be fresh. But, in case of the null-scheme, the SUCI does not conceal the SUPI. So unlinkability is irrelevant.

> **[TR 33.899, Sec. 5.2.3.8.2]** **Security threats:** Over-use of a single UE key-pair may harm user privacy (allowing a user's actions to be linked and tracked across multiple domains and services).

On anonymity: [TS 133.102, Sec. 5.1.1], item 3 and:

> **[33.849, Sec. 6.4.2]** (TR on Privacy in 3GPP, 2016) The UMTS authentication procedure (TS 33.102 [10]) design is an example of how to fulfil anonymity:
> (1) Analysis of the authentication process: identity and location of the user may be exposed.
> (2) Identify an identifying attribute: sequence number may bring a risk of personal identification.

(3) Risk: The **sequence number may expose the identity and thus the location of the user**.
(4) Anonymizing technique used: use Anonymity Key in the Authentication Token to conceal (blind) the sequence number.

> **[TR 33.899, Sec. 5.1.4.14.3]** If there was no single NAS security termination then the unencrypted part of a signalling message would have to contain parameters that would allow routing to the correct NAS entity, e.g. SM entity in a network slice. This information about the slice may give away information on the services used. However, user identity privacy should prevent that an eavesdropper can associate a particular signalling message with a particular subscriber. Editor's Note: The above paragraph has been included for completeness. It is ffs whether leaving parameters unencrypted that are required for NAS-internal routing would endanger privacy.

Formally, it seems that 5G authentication methods are required to provide *anonymity and untraceability of the subscribers in the presence of a passive attacker*.

D.2.4 Other Properties. As specified below, the established keys should never be the same twice:

> **[TS 133.102, Sec. 6.2.3]** (3G) **Key reuse:** A wrap around of the counter SQN could lead to a repeated use of a key pair (CK, IK). This repeated key use could potentially be exploited by an attacker to compromise encryption or forge message authentication codes applied to data sent over the 3GPP-defined air interfaces.

This will be analyzed as part of Injective *agreement properties on the established key* K_{SEAF} for different pairs of parties.

Finally, 5G specify some security goals in the context of backward compatibility with older generations. We do not analyze those properties as they would require us to analyze the combination of the 5G authentication protocols with the older generations authentication protocols. This is left as future work.

> **[TS 33.501, Sec. 6.1.1.3]** **Key separation:** Furthermore, the anchor key provided to the serving network shall also be specific to the authentication having taken place between the UE and a 5G core network, i.e. the KSEAF shall be **cryptographically separate** from the key KASME delivered from the home network to the serving network in earlier mobile network generations.

> **[TS 33.501, Sec. 5.11]** An attacker could attempt a **bidding down attack** by making the UE and the network entities respectively believe that the other side does not support a security feature, even when both sides in fact support that security feature. It shall be ensured that a bidding down attack, in the above sense, can be prevented.