A Formal Analysis of 5G Authentication — Source link ↗

David Basin, Jannik Dreier, Lucca Hirschi, Saša Radomirović ...+2 more authors

Institutions: ETH Zurich, University of Lorraine, University of Dundee

Related papers:

- A Formal Analysis of 5G Authentication

- Formal verification and analysis of primary authentication based on 5G-AKA protocol

- A comprehensive formal analysis of 5G handover

- Formal Analysis of a Privacy-Preserving Billing Protocol

- Formal analysis of modern security protocols

A Formal Analysis of 5G Authentication

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

A Formal Analysis of 5G Authentication

David Basin
Department of Computer Science
ETH Zurich
Switzerland
basin@inf.ethz.ch

Jannik Dreier
Universite de Lorraine
CNRS, Inria, LORIA
Nancy, France
jannik.dreier@loria.fr

Lucca Hirschi
Department of Computer Science
ETH Zurich
Switzerland
lucca.hirschi@inf.ethz.ch

Saša Radomirović
School of Science and Engineering
University of Dundee
UK
s.radomirovic@dundee.ac.uk

Ralf Sasse
Department of Computer Science
ETH Zurich
Switzerland
ralf.sasse@inf.ethz.ch

Vincent Stettler
Department of Computer Science
ETH Zurich
Switzerland
svincent@student.ethz.ch

ABSTRACT

Mobile communication networks connect much of the world's population. The security of users' 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 automated analysis 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 protocol, symbolic verification, formal analysis

1 INTRODUCTION

Two thirds of the world's population, roughly 5 billion people, are mobile subscribers [25]. 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 the 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 new use cases. The 5G standard will be deployed in two phases. The first phase (Release 15, June 2018) addresses the most critical requirements needed for commercial deployment and forms the basis for the first deployment. The second phase (Release 16, to be completed by the end of 2019) will address all remaining requirements.

In June 2018, the 3GPP published the *final* version v15.1.0 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 actually 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 precisely which 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 [11, 17, 19], messaging protocols [27], and entity authentication protocols [6]. 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 [31], ProVerif [12], and DeepSec [15].

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 the discussion on related work below. Given these features, we are left with just the verifier Tamarin [31] as a suitable tool, and Tamarin has only recently been extended to handle XOR [22].

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 flaws in the stated goals. We target a wide range of properties — confidentiality, authentication, and privacy — and their fine-grained variants. 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 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 authentication properties are violated prior to key confirmation, which is not clearly mandated by the standard. Some other properties are not met, except under assumptions on the 5G ecosystem that are missing from the standard. Additionally, we show that a privacy attack (enabling traceability) is possible for an active attacker. 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 proposal updates and the standard's evolution (*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 the 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, which was given 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 [14]. Moreover, the proof considered a simplified protocol without *SQN* concealment or re-synchronization as *SQNs* were always assumed to be synchronized. This misses, for example, the privacy attack based on the desynchronization error message that we observed.

ProVerif has also been used to formally check untraceability and basic authentication properties of simplified AKA protocols [5, 32]. These prior works acknowledge the challenges of formally verifying AKA protocols but only offered limited solutions. For instance, the *SQN* counters were abstracted away by nonces that are 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, [26] 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. In Section 4 we explain the basics of the Tamarin verifier 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. Note that additional details are given in the companion report [8].

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 [8]. 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 which 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.,*
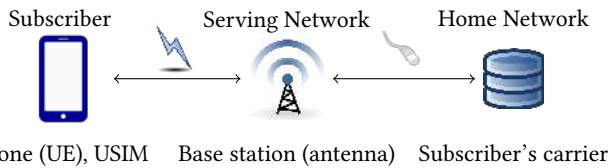

Figure 1: Overall architecture: The subscriber uses his phone (UE), equipped with a USIM, to communicate with a base station run by the SN over an insecure wireless channel. The SN communicates with the subscriber's carrier (HN) on the right over an authenticated (wired) channel.

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 over 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 the *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 have thus chosen to model these three larger logical entities (see [8] for more details on the sub-entities).

2.2 Authentication Protocols

To enable *SNs* and subscribers to establish secure channels and authenticate each other, the 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 [8].)

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): $SUCI = \langle \text{aenc}(\langle SUPI, R_s \rangle, pk_{HN}), idHN \rangle$, where $\text{aenc}(\cdot)$ 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* can retrieve the *SUPI*, the subscribers' identity, and choose an authentication method. Note that *SUPI* also contains *idHN* and therefore identifies both a subscriber and its *HN*.

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* provides replay protection[1] for the subscriber. While *SQN* should 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

---

[1]This design choice is for historical reasons: old USIMs (*e.g.,* in 3G and 4G) did not have the capability to generate random nonces.

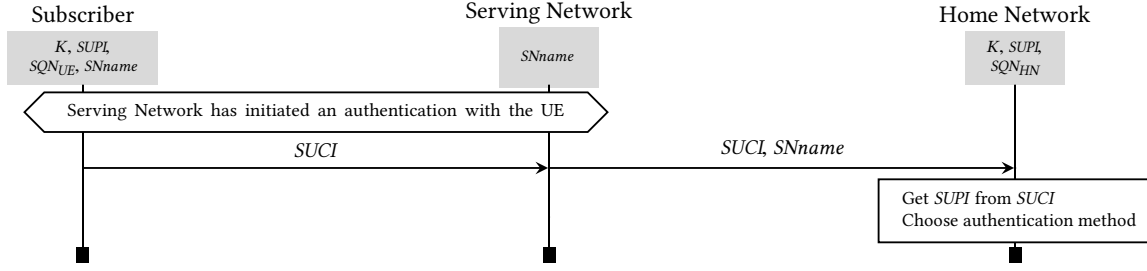D. Basin, J. Dreier, L. Hirschi, S. Radomirović, R. Sasse and V. Stettler

Figure 2: Initiation of Authentication

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 with each other, and ⊕ denotes Exclusive-OR. Challenge(·) and KeySeed(·) are complex Key Derivation Functions (KDFs); see [8] 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 thereerof; 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 challenge's expected response 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,

(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 the *SN* and the subscriber can be secured using the key seed K_{SEAF}.

---

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

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 enables 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*; namely, it is XORed with a value that should remain private: $AK^* = f5^*(K, R)$. Formally, the concealed value is $CONC^* = SQN_{UE} \oplus AK^*$, which 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 similar cryptographic messages. The main difference is the flow and some key derivation functions are slightly 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 [8].

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 interpretation 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 an explanation of our interpretation is given in [8].

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
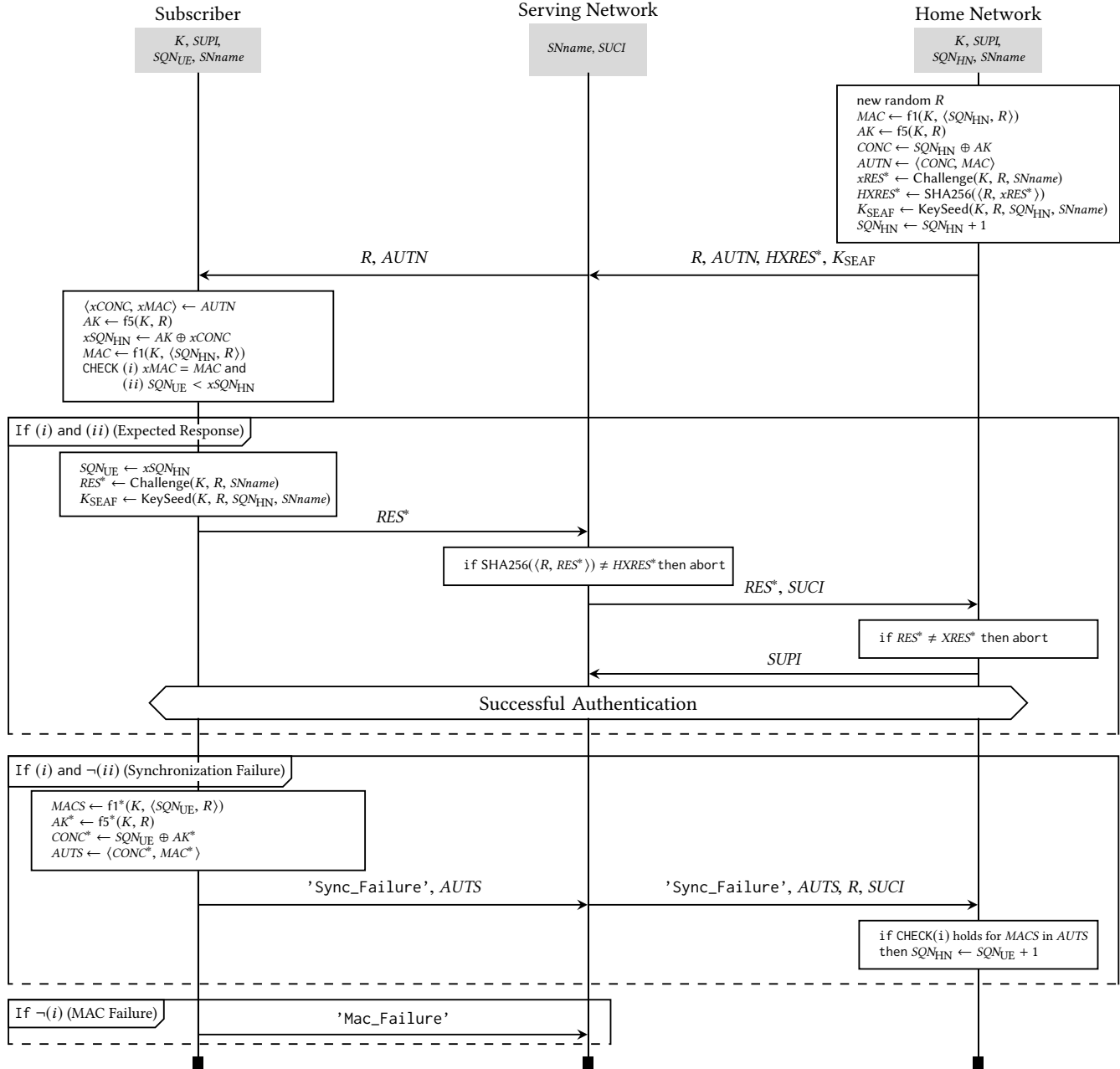
Subscriber	Serving Network	Home Network
K, $SUPI$, SQN_{UE}, $SNname$	$SNname$, $SUCI$	K, $SUPI$, SQN_{HN}, $SNname$

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

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

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

If (i) and (ii) (Expected Response)
- $SQN_{UE} \leftarrow xSQN_{HN}$
- $RES^* \leftarrow \mathrm{Challenge}(K, R, SNname)$
- $K_{SEAF} \leftarrow \mathrm{KeySeed}(K, R, SQN_{HN}, SNname)$

Subscriber → Serving Network: RES^*

Serving Network: if $\mathrm{SHA256}(\langle R, RES^* \rangle) \neq HXRES^*$ 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)
- $MACS \leftarrow \mathrm{f1}^*(K, \langle SQN_{UE}, R \rangle)$
- $AK^* \leftarrow \mathrm{f5}^*(K, R)$
- $CONC^* \leftarrow SQN_{UE} \oplus AK^*$
- $AUTS \leftarrow \langle CONC^*, MAC^* \rangle$

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)

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 send and receive 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 [24, 36]. From now on, we shall consider active attackers, except when explicitly stated otherwise.

3.1.2 Assumptions on Cryptographic Primitives. The functions f1, f1*, and 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 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 their 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. The 5G specifications make claims about authentication properties at different places in the documents. We have identified relevant claims and translated them into formal security goals, indicated in *purple, cursive text*. We use Lowe's taxonomy of authentication properties [29] to make the goals precise, prior to formalization. These properties are well established and understood, avoiding ambiguity [7]. Moreover, there is a formal relationship 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 [8]. 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 the 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.

Formally, *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*. Moreover, since *SUPI* also contains id_{HN}, an agreement on *SUPI* entails an agreement on id_{HN}.

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 knowledge of 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 [16]*, which fail to hold as we shall see in Section 5.1. Forward and post-compromise secrecy require session key secrecy even when 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.

> **[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. [...]

This important role given to privacy can be explained by numerous, critical attacks that have breached privacy (*e.g.,* with IMSI-catchers [36, 37]) in previous generations; see the survey [33]. 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, 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, i.e., one who only eavesdrops on the radio link. We criticize this restriction in Section 5.2.3. We now list more precise requirements on privacy 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 passive IMSI-catcher attacks. Formally, *the SUPI shall remain secret in the presence of a passive attacker*.

Similarly, the *SQN* must remain secret [TS 133.102, 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 or explicit statement is made on the necessity of ensuring untraceability for 5G, several claims in TR and TS documents (see [8]) 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. Therefore, formally, 5G authentication methods should 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 or not the proposed protocols actually fulfill these properties (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, K_{SEAF} should be different for each session. This is a critical requirement, especially for typical use cases for 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, in particular 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 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*, from 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 using TAMARIN (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 analyze 5G AKA, we used the TAMARIN prover [34]. 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 [10], 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 [18]. Moreover, it was recently extended with support for XOR [22], 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*.

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

These rules yield a labeled transition system describing the possible protocol executions (see [1, 34] for details on syntax and semantics). TAMARIN combines the protocol semantics with a Dolev-Yao [21] style attacker. This attacker controls the entire network and can thereby intercept, delete, modify, delay, inject, and build new messages. However, the attacker is limited by the cryptography: he cannot forge signatures or decrypt messages without knowing the key (black box cryptography assumption). He can nevertheless apply any function (e.g., hashing, XOR, encryption, pairing, …) to 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*:

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

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 [8].

For each specified property, TAMARIN checks that the property holds for all possible protocol executions, and all possible attacker 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 [10]), i.e., it compares the two resulting systems and checks that the attacker cannot distinguish them for any protocol execution and any adversarial behaviors.

In fully automatic mode, TAMARIN either returns a proof that the property holds, or a counterexample/attack if the property is violated, or it 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 protocol's complexity, 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, 35]). 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 [22].

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 hold without key confirmation, and what is gained by including 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, ABBA, 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 online [9], 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 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. 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 an automated formal verifier in the unbounded setting. Previous examples are limited to the case of just an internal counter for a TPM [30] or small examples, like simplified Yubikey [28].

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 [22]. Intuitively, this is because of the intricate algebraic properties of XOR (*i.e.,* associativity, commutativity, cancellation, and neutral element). This 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, in particular with the re-synchronization procedure, induces complex state-changes 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). The effort of writing such generic proof strategies represents several person-months.

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 this roundtrip or without. The protocol model itself consists of roughly 500 LoC.

Our model includes all the necessary lemmas: helping lemmas, sanity-check lemmas, and the lemmas that 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 is reduced by just one assumption. This requires 124 different lemmas and ca. 1000 LoC. Tamarin needs ca. 5 hours to automatically establish all the proofs and find all the attacks.

Our model of 5G AKA is general in that it can be used to model all other protocols from the AKA family requiring only localized modifications in the model. 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 as the standard further evolves 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 honest and compromised subscribers, *SNs*, and *HNs* when used *in combination* with the initiation protocol. Thus, our analysis accounts for all potential unintended interactions an attacker could exploit between these sub-protocols run by all possible 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}, *SNname*, and the *SUPI* (recall that the *SUPI* already contains id_{HN}). 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 where all participants are honest, none of the compromise scenarios is considered, and key confirmation is systematically enforced. Similarly, we present our results concerning secrecy properties in Table 2.

We only check for 2-party authentication properties, which expresses well the security goals of 5G AKA. Note however that we obtain a form of 3-party agreement property (where all 3 parties' views coincide) as a corollary of three 2-party agreement properties. This is because we check for strong 2-party agreement properties on several data points and identifiers simultaneously.

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 best-case scenario (✗). 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 ✗ in 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 an *SN* and an *HN* runs two sessions concurrently, there is no assurance that the *SUPI* the *SN* receives at the end of the protocol actually corresponds to

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*. When not otherwise specified, the worst-case scenario is considered; that is $K, SQN, SUPI, sk_{\text{HN}}$, and the channel between *SN* and *HN* are compromised and the key confirmation phase is skipped. ×: 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	$\emptyset$	$\emptyset$	$\emptyset$

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.

the K_{SEAF} it has received earlier (it could correspond to another concurrent session). As a consequence, an *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. In practice, this could allow an attacker to make the *HN* bill someone else (*i.e.,* with a different *SUPI*) for services he consumes from an *SN* (*i.e.,* encrypted with K_{SEAF}). 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 version v0.8.0 of the standard (published in March 2018). 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 making the binding assumption unnecessary. However, the final version of the standard requires this additional assumption. A similar looking issue, but between two parts of the *HN* has been previously observed by [20], but it is an entirely different concern than 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 an *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.,* the 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 of security guarantees since some critical properties are

---

[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.

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 only against passive attackers. 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) and hence also against passive attackers. 5G AKA thus defeats previous active IMSI-catcher attacks [33], which relied on the subscribers sending the IMSI (matching *SUPI* in 5G) in the clear. 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 the 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 [9]), 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 [5] 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 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} is derived 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, a recent update in 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 ⟨$SQN_{\text{HN}} \oplus AK, MAC$⟩ where $MAC = \text{f1}(K, \langle SQN_{\text{HN}}, R\rangle)$. In our fix, *MAC* is replaced by $\text{f1}(K, \langle SQN_{\text{HN}}, R, SNname\rangle)$. Therefore, the subscribers can verify the authenticity of the challenge that commits to a specific *SNname*. We have formally verified [9] 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 [33] by tracking nearby *SQNs* over time or perform activity monitoring attacks [13].

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 [5, 23] and the *SQN* concealment mechanism should be strengthened against active attackers [13], 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 for this process.

5.3.5 Redundancies. A close look at the cryptographic messages (see the detailed list in [8]) 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 standard's future evolution and provides a tool for quickly evaluating 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 details 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, including central goals like agreement on the session key. Moreover, our analysis in TAMARIN shows that some properties are violated without further assumptions. A striking example of this is 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 in ways other than intended. 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 plan 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 will also 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 a recent update (from v0.7.1 to v0.8.0). 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/. Accessed: 2018-05-05.
[2] 2018. Tamarin prover. https://github.com/tamarin-prover/tamarin-prover/releases/tag/1.4.0. Accessed: 2018-05-08.
[3] 3GPP. 2001. 3G Security: Formal Analysis of the 3G Authentication Protocol. TS 33.902, v4.0.0.
[4] 3GPP. 2018. Security architecture and procedures for 5G system. TS 33.501, v15.1.0.
[5] 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.
[6] 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.
[7] David Basin, Cas Cremers, Kunihiko Miyazaki, Saša Radomirović, and Dai Watanabe. 2015. Improving the security of cryptographic protocol standards. *IEEE Security & Privacy* 13, 3 (2015), 24–31.
[8] David Basin, Jannik Dreier, Lucca Hirschi, Saša Radomirović, Ralf Sasse, and Vincent Stettler. 2018. A Formal Analysis of 5G Authentication. *arXiv preprint arXiv:1806.10360* (2018).
[9] David Basin, Jannik Dreier, Lucca Hirschi, Saša Radomirović, Ralf Sasse, and Vincent Stettler. 2018. Tamarin models, proofs and instructions for reproducibility. https://github.com/tamarin-prover/tamarin-prover/tree/develop/examples/ccs18-5G. Accessed: 2018-08-10.
[10] 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.
[11] Karthikeyan Bhargavan, Bruno Blanchet, and Nadim 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
[12] 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.
[13] Ravishankar Borgaonkar, Lucca Hirshi, Shinjo Park, Altaf Shaik, Andrew Martin, and Jean-Pierre Seifert. 2017. New Adventures in Spying 3G & 4G Users: Locate, Track, Monitor. 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.
[14] 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.
[15] 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, 529–546.

[16] 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.

[17] Cas Cremers, Marko Horvat, Jonathan Hoyland, Sam Scott, and Thyla 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.* 1773–1788.

[18] 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

[19] 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.*

[20] 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/.

[21] Danny 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

[22] Jannik Dreier, Lucca Hirschi, Saša Radomirović, and Ralf Sasse. 2018. Automated Unbounded Verification of Stateful Cryptographic Protocols with Exclusive OR. In *31st IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9-12, 2018.* IEEE Computer Society, 359–373. https://doi.org/10.1109/CSF.2018.00033

[23] 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.

[24] 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.

[25] GSMA. 2017. Global Mobile Trends 2017. https://www.gsma.com/globalmobiletrends/. Accessed: 2018-05-06.

[26] 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.*

[27] 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).*

[28] 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.

[29] Gavin Lowe. 1997. A Hierarchy of Authentication Specifications. In *10th Computer Security Foundations Workshop.* IEEE Computer Society Press.

[30] Simon Meier. 2013. *Advancing automated security protocol verification.* Ph.D. Dissertation. ETH Zurich.

[31] 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.

[32] Piers O'Hanlon, Ravishankar Borgaonkar, and Lucca Hirschi. 2017. Mobile subscriber WiFi Privacy. In *2017 IEEE Security and Privacy Workshops, SP Workshops 2017, San Jose, CA, USA, May 25, 2017.* 169–178.

[33] David Rupprecht, Adrian Dabrowski, Thorsten Holz, Edgar Weippl, and Christina Pöpper. 2018. On Security Research towards Future Mobile Network Generations. *IEEE Communications Surveys & Tutorials* (2018).

[34] 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.

[35] 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

[36] 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.*

[37] 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).