# Improving Subscriber Anonymity in the 5G Authentication and Key Agreement (AKA) Protocol — Three Approaches and Their Tradeoffs

No Author Given

No Institute Given

**Abstract.** We consider 5G Authentication and Key Agreement (AKA), a procedure for a cellular network to authenticate and establish a secret key with a subscriber that seeks access to it. We observe that the 5G cellular standards require the Serving Network (SN) to know the Subscription Permanent Identifier (SUPI) of the subscriber before the SN provides communication services to the subscriber's User Equipment (UE). We observe that removal of this requirement substantially improves subscriber anonymity. We identify that the main reason behind the disclosure of the SUPI to the SN in the 3GPP standards is Lawful Interception. We propose a version of 5G AKA in which the SUPI is not disclosed to the SN, prove that it maintains all prior security properties and propose three approaches via which a Law Enforcement Agency can be provided the SUPI if it demands it of the SN. We then identify tradeoffs along several axes each of the three approaches incurs. Taken together, our work proposes a version of 5G AKA that is very close to the original, provides tangible improvement to subscriber anonymity and proposes and analyses three approaches to address the lawful interception requirements in the standards that are potentially impacted.

**Keywords:** 5G, protocol security, formal verification, lawful interception

## 1 Introduction

*Subscribers* who communicate on *cellular networks* should have some measure of privacy and security from mobile network operators. For example, a subscriber may not wish a foreign operator to be able to track them without their explicit permission. The state of the art in cellular networks, however, is complete trust in operators. This expression of trust has pervaded standards documents, and persists in specifications for the next-generation cellular standard, 5G [7].

In this work, we focus on an identity of a subscriber, the Subscription Permanent Identifier (SUPI). 3GPP defines the SUPI as "a globally unique 5G Subscription Permanent Identifier allocated to each subscriber in the 5G System" [4]. Thus, a SUPI uniquely identifies a subscriber. Typically, a SUPI is

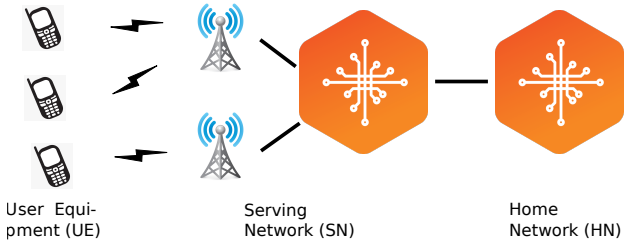


**Fig. 1.** The setting that 5G Authentication and Key Agreement (AKA) targets.

issued to a subscriber by their Home Network (HN), and bound strongly to the subscriber's UE, e.g., by having to physically insert (or program) a Subscriber Identity Module (SIM) card (embedded) into a mobile phone. The privacy of such identifiers is seen by security researchers as an important property. For example, International Mobile Subscriber Identity (IMSI) catching[1], in which an IMSI that is transmitted in the clear over the air is disclosed to an unauthorized entity, is considered an attack [15, 21, 22, 35]. Since a SUPI uniquely identifies a subscriber, at the minimum, an attacker to whom a SUPI is disclosed more than once can track a subscriber as they roam. The 3GPP specification for 5G's security architecture [7] now calls for the SUPI to be "privacy protected over-the-air" — specifically, a Subscription Concealed Identifier (SUCI) replaces the SUPI over the air [40]. Thus, the sensitivity of the SUPI, and the need to protect it, are acknowledged by the 5G specifications themselves.

While 5G specifications and academics acknowledge the sensitivity of the SUPI, certain parties in the 5G ecosystem have reason to oppose this interest. For example, depending on the jurisdiction they operate in, an operator may have to comply with lawful interception requirements, which require an operator to be able to isolate target subscriber communications upon request by a Law Enforcement Agency (LEA). An operator may also benefit from knowing a permanent identifier for a subscriber, e.g., to support business analysis. A vendor likely has to support lawful interception through their equipment and technology solutions, and may benefit from being able to correlate subscriber communications, e.g., in trying to optimize their equipment.

When a subscriber seeks to communicate over the cellular network using their UE, they must first gain access to an operator's network. We refer to such a network to which the subscriber seeks access as the SN (see Figure 1). The SN may be distinct from the HN, i.e., the subscriber may roam. To gain access to an SN, the subscriber must be authorized; this authorization is based on authentication of the subscriber by both the HN and SN, and effected by a protocol that involves all three parties – the UE, SN and HN.

[1] An IMSI is one type of SUPI.

A standard protocol for this is AKA [7]. AKA has been around since at least the 3G cellular standard [2]. Over time, its security has improved. For example, between the versions in 4G and 5G, cryptographic protection of the SUPI when it is transmitted over-the-air has been added [17]: the SUCI now replaces the SUPI over the air. We refer to the version currently in specifications for 5G as 5G AKA [7]. Notwithstanding the improvements from prior versions, 5G AKA still contains vulnerabilities as it relates to its explicated security goals. Recent work, for example, points out that the design, and not any particular implementation, of the protocol leaves the confidentiality of the sequence number of a protocol instance susceptible to compromise, which in turn undermines the privacy of the subscriber [13]. In particular, knowledge of a portion of the sequence number of a 5G AKA instance, that work shows, enables an attacker to learn the subscriber's service consumption pattern (number of calls and SMSs sent per time unit, etc.) We discuss this, and other related work, in Section 4.

To us, even more fundamental than such flaws are directives incorporated into the specifications that violate a user's privacy and security. In this regard we return to our discussion of the SUPI, and cite two properties 5G AKA seeks to meet that are relevant to our work, subscription authentication and UE authorization [7].

> Subscription authentication: The serving network shall authenticate the Subscription Permanent Identifier (SUPI) in the process of authentication and key agreement between UE and network.
> 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 issue we raise** We do not challenge the properties above as needs to be met by 5G AKA. We take exception, however, to the disclosure of the SUPI to the SN by the HN as part of 5G AKA (see Figure 2). Indeed, the specification is emphatic about this [7]:

*No communication services will be provided to the UE until the SUPI is known to the serving network.*

We are able to glean three reasons in the specifications for requiring the disclosure of the SUPI to the SN. The first is implicit: for the HN to confirm to the SN that the subscriber has been authenticated. We observe, however, that this is not necessary. In Section 3.1, we establish that even if the SUPI is not transmitted to the SN in 5G AKA, no security property other than agreement (between the subscriber and the SN) is impacted. We do this by amending the model in a verification system from prior work [11], and running it against all the properties encoded by that work.

The second is explicit in the specifications [7]:

*By including the SUPI as input parameter to... key derivation... additional assurance on the correctness of SUPI is achieved by the serving network from both, home network and UE side.*

The above passage refers to a key derivation that is carried out by each of the UE and the SN after 5G AKA completes. To us, such "additional assurance

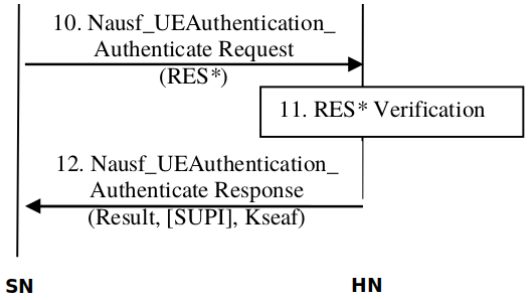


**Fig. 2.** A portion of 5G AKA from the specification [7] that shows the SUPI being disclosed to the SN by the HN in Step 12. The reason the SUPI is shown as optional, i.e., within '[]' is, "In case HN received [an encrypted SUPI] from the [UE via the SN] in the authentication request... then [HN] shall also include the SUPI in [Step 12]".

on the correctness of SUPI," serves no purpose: in 5G AKA, the SN delegates authentication of the UE to the HN [11]. This trust in the HN seems appropriate from the standpoint of 5G AKA: as we discuss in Section 3.1, the HN can easily violate the specifications with regards to the SUPI. The HN can, for example, authenticate a UE based on a SUPI that changes with every instance of 5G AKA in which the UE engages, which would violate the specifications with regards to what a SUPI is [4]. 5G AKA cannot realistically be expected to mitigate this. Thus, to us, this reason for disclosing the SUPI to the SN is not valid.

A final reason we can see in the specifications for disclosure of the SUPI to the SN is *lawful interception*. According to the specifications for 5G [3], lawful interception comprises "actions taken by the Communication Service Provider (CSP) that include: provisioning the target identity in the network to enable isolation of target communications (separating it from other users' communications), duplicating the communications for the purpose of sending the copy to the Law Enforcement Agency (LEA), and handing over the Interception Product to the LEA that served the Communication Service Provider (CSP) with the warrant." The specification for 5G's security architecture [7] states:

"*For lawful interception, the [HN] sending SUPI to the [SN] is necessary...*" This is the only valid reason we see for disclosure of the SUPI to the SN, and is a focus of our work.

**Our contributions** We observe that the 5G specifications (a) call for disclosure of the SUPI to the SN, and, (b) assert a false equivalence between allowance for lawful interception, and disclosure of the SUPI to the SN. We argue that (a) is unnecessary and compromises a subscriber's anonymity, and in (b), the latter is not necessary for the former. We modify 5G AKA so the SUPI is not disclosed to the SN. We confirm that a prior approach based on a theorem prover [11] validates the new version of the protocol in the same manner as the prior one.

Towards lawful interception, we provide two new protocols. Each of the new protocols is associated with what we call a query from an LEA to an SN to which it issues a warrant: (i) given an identifier for an instance of 5G AKA,

**Table 1.** The symbols used in Figure 3 and their meaning.

| Symbol | Meaning |
|---|---|
| K | Symmetric key shared by the UE and HN. |
| $\text{PUB}_{\text{HN}}$, $\text{PRI}_{\text{HN}}$ | Public-private key pair of the HN. |
| $\text{SQN}_{\text{UE}}$, $\text{SQN}_{\text{HN}}$ | Sequence numbers at the UE and HN. |
| SUCI | SUbscription Concealed Identifier. The UE's SUPI, but cryptographically protected. |
| SNname | Identity of the SN. |
| RAND | Random string for this instance of the protocol. |
| AUTN | Authentication token. |
| ABBA | Anti-Bidding down Between Architectures; a constant, from our standpoint. |
| $\text{RES}^*$ | Keyed Message Authentication Code (MAC) of a response string. |
| $\text{HXRES}^*$ | Hash of a keyed MAC of an expected response. |
| Result | Authentication result $\in \{\mathsf{success}, \mathsf{failure}\}$ |
| $\text{K}_{\text{SEAF}}$ | Key for this instance of 5G AKA from which the SN derives a key to communicate with the UE. |

what is the SUPI with which that instance is associated?, and, (ii) given a SUPI, what are the identifiers for all instances of 5G AKA that involve it? We require, in answering a query, that the SN continues to not learn the SUPI, and yet require that the SN and the HN consent for the LEA to learn the answer to the query it poses to the SN. Both protocols include a mechanism for the LEA to confirm that what it learns under either query above is indeed the correct answer. These features go beyond the capabilities of the current 5G AKA. Both protocols guarantee that the HN does not learn the identity of the interception target, although both protocols involve the HN.

## 2 5G AKA

In this section, we describe 5G AKA that is current as of the writing of this paper. The level of abstraction we adopt is the same as the standards documents that pertain to our work (see, for example, Sections 6.1.2 and 6.1.3 in [7]) and prior work on 5G AKA [11]. Steps (1) and (2) in Figure 3 are the initiation of authentication; those steps are not part of 5G AKA, but of the Extensible Authentication Protocol (EAP) [8], which 5G adopts for this purpose. We adopt the following conventions and notation in our exposition. We use "||" for string concatenation, "$\oplus$" for bitwise XOR and $|\cdot|$ to denote string-length. A field in a message that is received is denoted with a prime; for example, $\text{Eph}'_{\text{pub}}$. We denote checks by a recipient with a question mark on an operator; for example, $a' \stackrel{?}{=} b$. If a check fails, unless we specify otherwise, this instance of the protocol ceases. 5G AKA uses a few different Key Derivation Functions and Message Authentication Code (MAC)s. We adopt the notation $kdf_i$ and $mac_j$, respectively, i.e., with subscripts. We adopt the convention that $kdf_i = kdf_j$ if and only if $i = j$, and similarly for $mac_i$ and $mac_j$.

1. In this step, the UE sends the SUCI to the SN to initiate authentication. As Table 1 states, the SUCI is a cryptographically protected SUPI. More specifically:

$$\text{SUCI} = \text{Eph}_{\text{pub}} \,||\, \text{Ciphertext} \,||\, \text{MAC-tag}$$

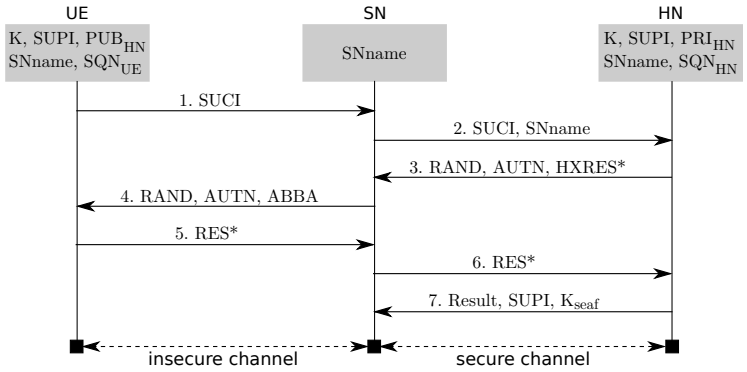


**Fig. 3.** The portion of 5G EAP and AKA on which fwe focus. The numbers indicate chronology. Steps (1) and (2) are initiation of authentication, which is not part of 5G AKA, but EAP. Table 1 explains the symbols. We explain each step and the actions a sender and recipient perform in the prose.

- $\text{Eph}_{\text{pub}}$is the public part of an ephemeral public-private key pair the UE generates for this instance. The private part of this key is then combined with the HN's public key, $\text{PUB}_{\text{HN}}$ to yield an ephemeral shared key, which in turn is used to derive ephemeral encryption and MAC keys, which we denote "$\text{Eph}_{\text{enc}}$" and "$\text{Eph}_{\text{mac}}$". That is, given a key-generation function $\mathsf{Gen}$, distinct key-derivation functions $kdf_1$ and $kdf_2$, and $n$ a desired key-length:

$$\begin{aligned}\langle \text{Eph}_{\text{pub}}, \text{Eph}_{\text{priv}} \rangle &= \mathsf{Gen}(1^n)\\ \text{Eph}_{\text{shared}} &= kdf_1(\text{Eph}_{\text{priv}}, \text{PUB}_{\text{HN}})\\ \langle \text{Eph}_{\text{enc}}, \text{Eph}_{\text{mac}} \rangle &= kdf_2(\text{Eph}_{\text{shared}})\end{aligned}$$

- Ciphertext = $encrypt(\text{Eph}_{\text{enc}}, \text{SUPI})$, is a semantically secure, symmetric-key encryption of the SUPI with the key $\text{Eph}_{\text{enc}}$.
- MAC-tag = $mac_1(\text{Eph}_{\text{mac}}, \text{Ciphertext})$, is a MAC of Ciphertext with the key $\text{Eph}_{\text{mac}}$.

2. In this step, the SN, acting as a "pass-through authenticator" in EAP parlance [8], forwards the SUCI to the HN along with its own identity, SNname. Upon receipt of this message, the HN first extracts the three different fields from the SUCI, denote these $\text{Eph}'_{\text{pub}}$, Ciphertext$'$, MAC-tag$'$. It then computes the following.

$$\begin{aligned}\text{Eph}'_{\text{shared}} &= kdf_1(\text{Eph}'_{\text{pub}}, \text{PRI}_{\text{HN}})\\ \langle \text{Eph}'_{\text{enc}}, \text{Eph}'_{\text{mac}} \rangle &= kdf_2(\text{Eph}'_{\text{shared}})\\ \text{SUPI}' &= decrypt(\text{Eph}'_{\text{enc}}, \text{Ciphertext}'),\end{aligned}$$

Also, it checks the MAC-tag:

$$\text{MAC-tag}' \stackrel{?}{=} mac_1(\text{Eph}'_{\text{mac}}, \text{Ciphertext}')$$

We have abused notation and used $kdf_1$ with different kinds of arguments at the UE and HN. Our intent is to convey that $kdf_1$ may be used with either set of arguments, and the resultant ephemeral key should agree if the epheremal public-private key, and public-private keys of the HN are provided in a manner consistent to one another.
The HN, using what appears to be the SUPI of the UE, i.e., SUPI$'$, extracts from its database the symmetric key, $K$, that it shares with the UE.

3. This step is the initiation of a challenge from the HN to the UE. Not only does the HN verify that the UE meets the challenge in the subsequent Step (6), but also, the SN is able to verify that the UE meets the challenge in Step (5). In this step, the HN responds to what it receives in Step (2) with (i) RAND, a random string it chooses for this instance of authentication, (ii) AUTN, an authentication token, and, (iii) HXRES$^*$, the hash of the MAC of a response it expects from the UE via the SN.

$$\begin{aligned}
&\text{AK} = kdf_3(K, \text{RAND})\\
&\text{AUTN} = \text{SQN}_{\text{HN}} \oplus \text{AK} \,||\, mac_2(K, \text{SQN}_{\text{HN}} \,||\, \text{RAND})\\
&\text{XRES} = mac_3(K, \text{RAND})\\
&S = \text{SNname} \,||\, |\text{SNname}| \,||\, \text{RAND} \,||\, |\text{RAND}| \,||\\
&\qquad \text{XRES} \,||\, |\text{XRES}|\\
&\text{XRES}^* = kdf_4(K, S)\\
&\text{HXRES}^* = hash(\text{RAND} \,||\, \text{XRES}^*)
\end{aligned}$$

4. In this step, the SN forwards RAND and AUTN that it received in the previous step to the UE, along with a value, ABBA, as protection from being bid down to a lower, possibly less secure version of the protocol. From our standpoint, ABBA is a constant; indeed, the current specification sets it to all 0's. We point out that the AUTN is opaque to the SN as it possesses neither $K$ nor $\text{SQN}_{\text{HN}}$.
Upon receipt, the UE checks for authenticity and integrity. Specifically, it does the following. The UE extracts $\text{RAND}'$, $\text{SQN}'_{\text{HN}}$, and $mac_2(K, \text{SQN}_{\text{HN}}, \text{RAND})'$. To extract $\text{SQN}'_{\text{HN}}$, it first computes $\text{AK}' = kdf_3(K, \text{RAND}')$, and then XOR's with the leading bits of AUTN$'$.
It checks the MAC against what it receives, and the sequence number against its own:

$$\begin{aligned}
&mac_2(K', \text{SQN}'_{\text{HN}}, \text{RAND}') \stackrel{?}{=} mac_2(K, \text{SQN}_{\text{HN}}, \text{RAND})'\\
&\text{SQN}_{\text{UE}} \stackrel{?}{<} \text{SQN}'_{\text{HN}}
\end{aligned}$$

This is to check whether the random string, authentication token, sequence number and AK$'$, it received are what were sent by the HN. The last of these it uses to infer whether its shared symmetric key $K$, is the same as that at the HN. If the first check fails, the UE sends a "`MAC_Failure`" message (not depicted in Figure 3) to the SN [11]. If the first check passes but the second fails, the UE initiates a "resynchronization" procedure [11], the details of which we omit here for brevity.

5. The UE computes and sends a response, $\text{RES}^*$, to the challenge. The intent is for $\text{RES}^* = \text{XRES}^*$, where the expected response, $\text{XRES}^*$, was computed by the HN prior to Step (3). $\text{RES}^*$ is computed by the UE exactly as $\text{XRES}^*$ is computed by the HN (see Step (3) above), except using the values it possesses for the corresponding fields.
   Suppose we denote what the SN receives as $\text{RES}^{*\prime}$. Also denote as $\text{RAND}'$ and $\text{HXRES}^{*\prime}$ the values the SN receives beforehand in Step (3) from the HN. The SN now checks:

$$\mathit{hash}(\text{RAND}' \,||\, \text{RES}^{*\prime}) \stackrel{?}{=} \text{HXRES}^{*\prime}$$

6. If the check for $\text{RES}^*$ passes at the SN, it forwards it to the HN in this step. The HN performs its own check of $\text{RES}^*$; in particular, it checks:

$$\text{RES}^{*\prime} \stackrel{?}{=} \text{XRES}^*,$$

   and if it passes, deems the UE to be authentic.
7. The HN sends (i) an authentication result, $\text{Result} \in \{\mathsf{success}, \mathsf{failure}\}$, and in the case of success, (ii) the SUPI of the UE, and, (iii) a key, $\text{K}_{\text{SEAF}}$, to the SN. As before, we use "$|\cdot|$" for string-length.

$$\begin{aligned}
S_{\text{ausf}} &= \text{SNname} \,||\, |\text{SNname}| \,||\, \text{SQN}_{\text{HN}} \oplus \text{AK} \,||\, |\text{SQN}_{\text{HN}} \oplus \text{AK}| \\
\text{K}_{\text{ausf}} &= \mathit{kdf}_4(\text{K}, S_{\text{ausf}}) \\
S_{\text{seaf}} &= \text{SNname} \,||\, |\text{SNname}| \\
\text{K}_{\text{SEAF}} &= \mathit{kdf}_4(\text{K}_{\text{ausf}}, S_{\text{seaf}})
\end{aligned}$$

   As the key derivation function used above is the same as the one used in Step (3) to generate $\text{XRES}^*$, we use the same mnemonic, $\mathit{kdf}_4$, for it.

**Key derivation** Immediately following 5G AKA, the SN and UE derive a key $\text{K}_{\text{AMF}}$ as follows.

$$\text{K}_{\text{AMF}} = \mathit{hmac}(\text{K}_{\text{SEAF}}, \text{SUPI} \,||\, |\text{SUPI}| \,||\, \text{ABBA} \,||\, |\text{ABBA}|),$$

where $\mathit{hmac}(K, S)$ denotes the HMAC [25] of string $S$ using key $K$.

## 3 New Protocols

This section comprises the core technical contributions of this work. We first present and then address the consequences of adopting our new version of 5G AKA. The contributions are a verification that the new version is no worse than the current one (except for its inability, by definition, to meet certain authentication properties), and that a particular key derivation which in the current specification takes the SUPI as an input retains its security even without the SUPI as input (Section 3.1). We finally discuss the tradeoffs our version introduces with respect to the ability of the SN to meet lawful interception requirements (Section 3.2), and discuss mechanisms for resolving these tradeoffs.

### 3.1 New 5G AKA

Our new version is a seemingly simple modification to the existing version of 5G AKA. In Step (7) of the protocol in Figure 3, the HN does not send the SUPI. That is, in Step (7), the HN sends Result and $K_{SEAF}$ only. We refer to this version as Anonymous AKA.

**Anonymous AKA is no worse than 5G AKA** We claim that Anonymous AKA maintains all the security properties of the original 5G AKA, minus Lowe's (injective and non-injective) agreement [33] between the UE and the SN. Perhaps this is obvious, as we remove only a portion of a message in the last step of the protocol. However, we know from history that even seemingly simple and innocuous changes to a protocol can cause vulnerabilities. Consequently, to validate that Anonymous AKA is indeed no worse that 5G AKA, we first clearly assert our claim and then discuss the manner in which we establish it.

*Claim.* With the exception of agreement between the SN and the UE, Anonymous AKA provides the same security guarantees as the original under the same security assumptions.

The manner in which we have established the above claim is the following. We updated a formal model of 5G AKA provided by prior work [11] so it concurs with the specification, then modified the model so the HN does not send the SUPI in Step (7), but rather a random string of numbers. We then used Tamarin [34] to check whether the new model has all the properties of the original that prior work [11] verifies (with the exception of agreement between the SN and the UE). We describe the model and the changes we made to it in Appendix B.A repository of the code is here [1]. We verified, using Tamarin, that Anonymous AKA has all properties of the original AKA (assuming the same threat model is applied to each protocol), with the exception of agreement properties between the SN and the UE (Tamarin failled to either conclude or produce a counterexample that Anonymous AKA has these properties). We also verified, using the same model, that an SN does not learn the SUPI in an instance of Anonymous AKA (Appendix B). We point out however that underlying our work is the assumption that a standalone 5G network is in operation: until the old network is phased out, a subscriber can always fall back to 4G AKA and risk their SUPI being revealed to the SN at authentication time.

**An important detail** We point out, in the context of our above modification to 5G AKA, that in the original 5G AKA, the SN entirely trusts the HN to communicate the correct SUPI to it. That is, in absence of such an unqualified trust in the HN, the 5G AKA protocol itself does not guarantee that HN will send the correct SUPI to the SN. This is because the SN accepts immediately after receiving message (7) from the HN, i.e., it does not validate the SUPI it receives from the HN[2].

[2] After key derivation (Section 2), assuming the HN cannot modify the SUPI of the UE remotely, and that the subscriber has no control over the SUPI, the SN has assurance

**Key Derivation** Having proposed that the HN withhold the SUPI from the SN, we now consider whether we incur damage if we require that the SN and the UE use $K_{SEAF}$ (and the ABBA value) only as input to HMAC when deriving $K_{AMF}$ (Section 2).

**Theorem 1.** *Let $D$ be the distribution of the key $K_{SEAF}$ that the UE and the HN compute, and let $U$ be the uniform distribution over $\{0,1\}^{256}$. Then assuming $D$ is $(t,\epsilon)$-indistinguishable from $U$, and assuming HMAC is a $(t,\epsilon,1)$-PRF, the distributions of the following two random variables:*

$$hmac(K_{SEAF}, SUPI \,||\, |SUPI| \,||\, ABBA \,||\, |ABBA|)$$

*and*

$$hmac(K_{SEAF}, ABBA \,||\, |ABBA|)$$

*are $(t-T, 4\epsilon)$-indistinguishable, where $T$ is the time it takes to compute HMAC of $SUPI \,||\, |SUPI| \,||\, ABBA \,||\, |ABBA|$ given a 256-bit long key. In other words, our adversary's chance of guessing the result of the coin toss in time $t-T$ is at most $4\epsilon$.*

The proof is in Appendix C. The theorem says that assuming HMAC is a Pseduorandom Function (PRF) — a customary assumption about HMAC [12]— and assuming that the value of $K_{SEAF}$ a UE, an SN, and an HN compute in a 5G AKA instance is indistinguishable by an adversary from a random 256-bit bitstring ($K_{SEAF}$ is 256 bits long), there should be no difference in quality of key $K_{AMF}$ if we omit the SUPI entirely. Note that being a $(t,\epsilon,1)$-PRF is a very weak requirement on a PRF, since it requires that the adversary tells whether its oracle is a PRF or a whether it is a random oracle after making only one query, i.e., the adversary is really weak.

### 3.2 Lawful Interception

As we discuss in Section 1, a potential consequence of the SUPI not being revealed to the SN is that requirements from lawful interception are not met. We now consider three options for resolving this tradeoff. The first is for the the HN to encrypt the SUPI before sharing it with the SN in Step (7), using a *semantically-secure* encryption scheme, so only the LEA can decrypt it. The second is for the HN to encrypt the SUPI using a *deterministic* encryption scheme. The third is for the the LEA to involve the HN in lawful interception, without informing it of the interception target's identity. Each of the three solutions has tradeoffs. We discuss these tradeoffs and analyze them from the perspective of each of the parties involved: the subscriber, the HN, the SN, and the LEA. We frame the discussion in terms of two queries that an LEA can issue to an SN:

that the SUPI received from the HN matches the SUCI supplied by the UE. We consider a threat model where the HN and subscriber collude so the subscriber can control the SUPI in 5G AKA on an instance-by-instance basis.

i Given an identifier for an instance of Anonymous AKA, what is the SUPI associated with the instance?
ii What are all the identifiers for instances of Anonymous AKA that involve a given SUPI?

We summarize the tradoffs of each solution in Table 2 in terms of (i) **Performance**: the relative speed with which the LEA can obtain a response to a query of type (i), (ii) **Allows for tracking**: whether the SN can link different sessions of the same subscriber, (iii) **Other subscriber privacy**: whether the solution entails that the LEA learns the SUPIs associated with AKA instances other than the one identified in a query of type (i), (iv) **Changes to architecture**: whether the solution requires changes to ETSI's lawful interception architecture [5], (v) **Requires HN availability**: whether the solution requires the HN to be available for the SN to be able to respond correctly to a query by the LEA, (vi) **HN trusted to give actual SUPI**: whether the solution assumes the HN would give out the actual SUPI (in Step (7) of AKA for the deterministic and semantically-secure encryption solutions, or in a response to an LEA query of type (i) for the solution that involves the HN), and (vii) **HN becomes aware**: whether the solution results in the HN knowing that one of its subscribers is a target for interception.

**Requirements** We make four requirements of each of our solutions.

1. We require that the SN does not learn the SUPI in an answer to a query.
2. We require that the SN and the HN consent for the LEA to learn the answer to a query it poses to the SN.
3. We require that the LEA can check whether the answer it learns is correct.
4. We require that the HN remains oblivious to the identity of the interception target.

In all solutions, the LEA receives a value that is supposed to match the ephemeral shared key $\mathrm{Eph}'_{\mathrm{shared}}$ the HN obtained in Step (2) of the Anonymous AKA instance (see Section 2). We require the LEA to obtain the SUCI corresponding to the Anonymous AKA instance from the SN and to extract the Ciphertext field from the SUCI in the same manner an HN does in Step (2) of 5G AKA. The LEA then computes:

$$\langle \mathrm{Eph}'_{\mathrm{enc}}, \mathrm{Eph}'_{\mathrm{mac}} \rangle = kdf_2(\mathrm{Eph}'_{\mathrm{shared}})$$
$$\mathrm{SUPI}' = decrypt(\mathrm{Eph}'_{\mathrm{enc}}, \mathrm{Ciphertext}')$$

Then, it checks SUPI$'$ against the SUPI in question. If the two match, the LEA concludes that the SUPI is correct; otherwise the LEA concludes that the HN is dishonest, and proceeds accordingly.

**Semantically-Secure Encryption** The first option is to require the HN to encrypt the SUPI of the UE, along with the ephemeral shared key $\mathrm{Eph}'_{\mathrm{shared}}$, using a semantically secure encryption scheme, prior to including the the SUPI in message (7) of 5G AKA. Specifically, we require the HN to encrypt the SUPI

| Property/feature | Deterministic Encryption | Semantically-Secure Encryption | Involving the HN |
|---|---|---|---|
| Performance | fast | depends on #HN subscribers at SN | depends on blockchain & #subscribers |
| Allows for tracking | yes | no | no |
| Other subscriber privacy | protected | unprotected (as part of solution) | protected |
| Changes to architecture | none | yes | yes |
| Requires HN availability | no | no | availability enforced by miners |
| HN trusted to give actual SUPI | no | no | if HN lies, it will be penalized |
| HN/subscriber becomes aware | no | no | yes |

**Table 2.** Three solutions for interception and their security/performance properties. Semantically-secure encryption requires changes to 3GPP's lawful interception architecture (search would have to be performed by LEA, while in case of deterministic encryption search can be performed by the SN, once it is given the encrypted SUPI). Involving the HN also requires changes to the architecture as HN would have to be involved in the warrant execution process. By 'actual SUPI' we do not mean the **SUPI!** that the HN issues to the subscriber; we mean the plaintext corresponding to the ciphertext that the UE sends to the SN in Step (1).

using a public key that belongs to an LEA that the HN recognizes, prior to sending the SUPI to the SN along with Result and $K_{SEAF}$[3]. By the semantic security of the encryption scheme, no SN can then feasibly extract any information about the SUPI. The LEA would have the advantage of being the only party (other than the HN) that can determine the SUPI, and the ephemeral shared key $Eph'_{shared}$, given a transcript of the 5G AKA instance. Figure 4 depicts the solution.

Such a version of 5G AKA would have the disadvantage of requiring the LEA to decrypt the SUPI associated with every AKA instance at the SN, for it to get an answer to a query of type (ii). There is also the overhead of encrypting the SUPI, but note that Step (1) of AKA involves encrypting the SUPI as well.

*Remark 1.* The UE generates a fresh public-private key pair $\langle Eph_{pub}, Eph_{pri} \rangle$ for each instance of authentication [16], and the Elliptic Curve Integrated Encryption Scheme (ECIES) — the scheme the specification prescribes for a UE to generate the SUCI [7] — ensures that the UE never never uses the same ephemeral encryption key twice. Disclosing the ephemeral shared key corresponding to an Anonymous AKA instance thus should not affect security of other Anonymous AKA instances.

[3] Note that this process is different from key escrow, as the third party in question, the LEA, here already has the decryption key.

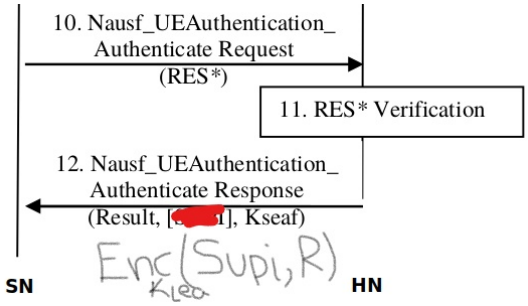


**Fig. 4.** Semantically-secure encryption: do not disclose SUPI to the SN; rather, encrypt it using a public key that belongs to an LEA in the SN's jurisdiction.

**Deterministic Encryption** The second option is for the HN to use a deterministic encryption scheme to encrypt the SUPI, separately from the shared ephemeral key $\text{Eph}'_{\text{shared}}$, rather than a semantically secure one. The LEA can then easily get an answer to a query of type (i): it asks the SN for the (now encrypted) SUPI, and decrypts it using the corresponding private key. To get an answer to a query of type (ii), the LEA encrypts the SUPI using the same public key, and identifies the subscriber to the SN using the resulting ciphertext. One might wonder what the privacy benefit is to a subscriber from replacing a permanent subscriber identity with a "semi-permanent" identity. The privacy benefit is that the HN does not disclose personal data of the subscriber (data that identifies the subscriber in the Home Subscriber Server (HSS)) to a third party unnecessarily. A legitimate downside of this approach is that the SN can link AKA instances that involve the same subscriber. One might propose here that the LEA changes public key periodically. Ideally, an LEA would change its public key every time the HN of a foreign subscriber authenticates the subscriber to an SN that falls under the LEA's jurisdiction. Changing the public key every so often however may incur too much overhead on the LEA.

**Involving the HN** The third option is for the LEA to involve the HN of the subscriber in lawful interception, without informing the HN of the subscriber's identity. This option gives the HN the upper hand: it requires the HN to cooperate with the LEA and the SN as needed, rather than to surrender the SUPI beforehand. The SN however needs assurance that the HN will respond when needed: otherwise the HN might just drop off the face of the earth once it has authenticated the subscriber. Cryptography alone cannot force the HN to cooperate. One might propose that the LEA invokes whatever authority it has to force the HN to cooperate, but (1) the LEA may not have any authority over the HN (e.g., if the HN operates in a foreign jurisdiction), and (2) such a solution might be an anathema to a subscriber who wishes to have transparency on the conditions under which the LEA can appeal to authority in matters relating to their privacy. An option here is to have the HN and the SN rely on a blockchain

to resolve potential disputes. The use of blockchain to resolve disputes between operators is already been considered by companies like IBM [23], KPMG [24], and Infosys [27], in the context of invoicing, with performance as the key driver. The expected proliferation of virtual network operators in 5G further suggests to us that a blockchain-like approach has promise. For example, before the SN provides services to the subscriber, the SN can require the HN to commit to sending the SUPI when required by recording a promise on a blockchain. If the HN later rescinds, the blockchain can penalize the HN by, e.g., decreasing the HN's reputation score. Note that an SN does not need to wait for the blockchain to approve the promise and can proceed with service immediately if it is satisfied with the HN's reputation.

We now discuss two protocols which allow the LEA to issue queries (i) and (ii) to the SN and have them answered by the HN. We call the protocol associated with query (i) 'SUPI Disclosure', and the one associated with query (ii) 'AKA Instance ID Disclosure'. We assume in designing the protocols that the LEA and the HN cannot communicate with each other directly, but that each (naturally) can communicate with the SN.

Both SUPI Disclosure and AKA Instance ID Disclosure protocols involve the LEA and the HN in an instance of oblivious transfer where the HN answers the LEA's query without knowing which SUPI is involved, and with the SN acting as a pass-through.

**SUPI Disclosure** We first describe the SUPI Disclosure protocol. The LEA here has an identifier $i$ for an Anonymous AKA instance, and the HN has $n$ SUPIs $\{\mathrm{SUPI}_1, \mathrm{SUPI}_2, \ldots, \mathrm{SUPI}_n\}$, where $\mathrm{SUPI}_j$ is the SUPI associated with Anonymous AKA instance $j$. The LEA and the HN engage in a 1-out-of-$n$ oblivious transfer protocol where the HN transfers $\mathrm{SUPI}_i$ to the LEA, along with the ephemeral shared key $\mathrm{Eph}'_{\mathrm{shared}}$ it obtained in Step (2) of Anonymous AKA instance $i$. The LEA and the HN use the SN as a pass-through.

*Remark 2.* Oblivious transfer ensures the HN remains oblivious to which SUPI and ephemeral shared key it transferred to the LEA. The HN's prior on $i$ therefore changes only negligibly throughout the protocol. The LEA learns $\mathrm{SUPI}_i$ and the ephemeral shared key the HN intends to transfer, but none of the other SUPIs or their associated ephemeral shared keys.

*Remark 3.* The messages exchanged between the LEA and the HN through the SN reveal neither $\mathrm{SUPI}_i$ nor the ephemeral shared key.

**AKA Instance ID Disclosure** In the AKA Instance ID Disclosure protocol, the LEA has a SUPI, $\mathrm{SUPI}^{\mathsf{target}}$, and the HN has a set $A = \{(s, i) : \text{Anonymous AKA instance } i \text{ involves SUPI } s\}$. The LEA and the HN engage in a generalized oblivious transfer (GOT) protocol [28] where the HN transfers to the LEA the set

$$\{(\mathsf{i}, \mathrm{Eph}^{\mathsf{i}}_{\mathrm{shared}}) : (s, i) \in A,\ s = \mathrm{SUPI}^{\mathsf{target}}\}$$

where $\mathrm{Eph}^{i}_{\mathrm{shared}}$ is the value of the ephemeral shared key the HN computed for the UE in Anonymous AKA instance $i$. The LEA and the HN use the SN as a pass-through.

*Remark 4.* Oblivious transfer ensures the HN remains oblivious to the subset of $A$ it transferred to the LEA.

**Tradeoffs** Even though both protocols keep the target's identity secret from the HN, both protocols still involve the HN. The HN therefore knows that interception is taking place at the SN's side by virtue of being a protocol participant, even as it does not know who the target of interception is. If the HN operates in a different country, an SN that adopts our solution violates the following requirement from the specification [3]:

> Undetectability Across Countries: The CSP shall ensure the performance of interception in one country cannot be detected in other countries.

One can conceive of a couple of other problematic scenarios. First, since the HN receives a subscriber's SUPI at authentication time and can keep track of which subscriber is located in which country, the HN can use any secret message passed as part of AKA to inform a subscriber that interception is taking place in their country of location. For instance, the subscriber and HN can agree beforehand that a certain value for RAND is a danger signal relating to interception. Legal regulations in the SN's jurisdiction may also require the SN to be able to carry out interception without involving the HN of a foreign subscriber. We finally note that our choice of an oblivious transfer protocol may impact performance of interception even though the state of the art in the domain [31] appears promising. (Note here that a legal interception is an infrequent event, so performance may not matter as much as if the HN and the SN would expect to handle such requests on a daily basis.)

## 4 Related Work

Our work pertains to AKA and lawful interception in 5G. As such, specifications from the 3GPP are relevant to our work. The most relevant specifications are the following: (i) TS 33.102 [2] on 5G's security architecture which discusses 5G AKA, (ii) TS 23.003 [4], which specifies what a SUPI is, and (iii) TS 33.126 [3] and TS 33.127 [6] which discuss requirements and an architecture for lawful interception, respectively.

From the standpoint of research, there are several pieces of work that are relevant to ours. One rather important line of work is a precise encoding of 5G AKA and related protocols, and the automated verification of them using tools, specifically model checkers and theorem provers. The work of Basin *et al.* [11], Cremers and Dehnel-Wild [20], Hussain *et al.* [26], Rellstab [37] and Zhang *et al.* [42] fall within this line of work. We have chosen the encoding of Basin *et al.* [11] to check whether changes to 5G AKA that we propose in this work result in a protocol that is no worse from the standpoint of security (see Section 3.1).

There has been work also that assesses security properties of 5G AKA rigorously but not using automated tools. The work of Koutsos [32] falls in this category. That work points out that apart from IMSI catching, all other attacks on privacy to which prior cellular specifications of AKA are susceptible

are still possible in 5G AKA. It proposes fixes, with proofs for the new version under a new notion called $\sigma$-unlinkability. We leave an assessment as to whether $\sigma$-unlikability is a notion that is appropriate in our context for future work.

There have been security assessments of protocols and architectures in 5G that are informal, but more holistic, specifically, the work of Jover and Marojevic [36] and Khan *et al.* [30]. Similar is the work of Choudhary *et al.* [19], which surveys security issues and mechanisms in 5G, particularly for the backhaul network.

Another body of work identifies new attacks in 5G's security mechanisms. For example, the work of Borgaonkar *et al.* [13] points out a possible compromise of confidentiality of a particular sequence field, which then can be used to compromise the privacy of a subscriber. Khan and Martin [29] question some of the claims of Borgaonkar *et al.* [13], particularly with regards to the claim that custom fixes are needed for each attack discovered in that work. We do not investigate nor attempt to fix such issues with 5G AKA, but rather, focus on removal of the requirement to provide the SUPI to the SN while still supporting lawful interception, but without introducing new vulnerabilities.

There is work also that proposes fixes, or alternatives, to 5G AKA with claims of better security properties. The work of Braeken *et al.* [14] claims to fix a number of security issues by proposing modifications to 5G AKA. Further still, it claims to provide new security properties with the modified version. No rigorous security analysis is carried out, however. Cao *et al.* [18] propose a new authentication protocol, LSAA, as an alternative to 5G AKA. They carry out a formal analysis of the new protocol in two different verification tools, and claim better security and a new feature related to what they call "massive device concurrent connections".

Lawful interception in cellular networks is the subject of several patents. For example, Sirotkin *et al.* [39]. The work of Sharevski [38] approaches lawful interception in 5G from the standpoint of forensics. In particular, it reviews the digital forensics mechanisms for lawful interception and user localization properties in the context of protocols that are part of 5G.

As to the use of smart contracts and blockchain in the context of 5G, the only prior work of which we are aware is on leveraging blockchain for network slicing [9,41]. Both pieces of work seek to support an architecture for a customer to acquire a network slice. Thus, there are complementary settings in 5G to ours in which the use of smart contracts appears to be appropriate.

## 5 Conclusion

In this paper, we have investigated whether disclosing the SUPI to the serving network (SN) in a 5G AKA instance is necessary for the protocol to provide a meaningful security guarantee, and whether it is indeed necessary for lawful interception as the specification asserts. The answer was no: disclosing the SUPI is neither necessary for security nor for lawful interception. To establish the first, we modified a formal model of 5G AKA provided by prior work so the HN

does not disclose the SUPI, then verified, using a theorem prover, that the new model has all the security properties of the original that prior work confirms. To demonstrate that disclosing the SUPI to the SN is not necessary for lawful interception, we described two protocols by which a LEA can obtain an answer to two types of queries to an SN. Neither protocol requires the SN to know the SUPI at the time of authentication. Our protocols provide five security guarantees. (1) The SN does not learn the SUPI. (2) The SN and the HN must consent for the LEA to learn the answer to the query it poses to the SN. (3) Each protocol exits successfully only if the LEA learns a correct answer. (4) The HN remains oblivious to the identity of the interception target in both protocols.

Both protocols rely on a blockchain to force the HN to cooperate in the protocol. To prevent the HN from learning interception target's identity, we use oblivious transfer. Future work might implement a state-of-the-art oblivious transfer protocol to estimate the overhead an HN and an LEA incur.

## References


1. Analysis of 5g-aka. `https://github.com/tamarin-prover/tamarin-prover/tree/develop/examples/ccs18-5G` (2018), (Accessed: 2020-05-07)
2. 3GPP: 3gpp ts 33.102 v15.1.0; 3rd generation partnership project; technical specification group services and system aspects; 3g security; security architecture; (release 15). `https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2262` (Dec 2018)
3. 3GPP: 3gpp ts 33.126 v16.1.0; 3rd generation partnership project; technical specification group services and system aspects; security; lawful interception requirements (release 16). `https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3181` (Sep 2019)
4. 3GPP: 3gpp ts 23.003 v16.2.0; 3rd generation partnership project; technical specification group core network and terminals; numbering, addressing and identification; (release 16). `https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=729` (Mar 2020)
5. 3GPP: 3gpp ts 33.107 v16.0.0; 3rd generation partnership project; technical specification group services and system aspects; 3g security; lawful interception architecture and functions (release 16). `https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2266` (Jul 2020)
6. 3GPP: 3gpp ts 33.127 v16.3.0; 3rd generation partnership project; technical specification group services and system aspects; security; lawful interception (li) architecture and functions (release 16). `https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3182` (Mar 2020)
7. 3GPP: 3gpp ts 33.501 v16.2.0; 3rd generation partnership project; technical specification group services and system aspects; security architecture and procedures for 5g system (release 16). `https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3169` (Mar 2020)

8. B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson and H. Levkowetz, Ed.: Extensible Authentication Protocol (EAP). Internet Request for Comments (RFC) 3748 (Jun 2004), https://tools.ietf.org/html/rfc3748
9. Backman, J., Yrjola, S., Valtanen, K., Mammela, O.: Blockchain network slice broker in 5g: Slice leasing in factory of the future use case. In: 2017 Internet of Things Business Models, Users, and Networks. pp. 1 – 8 (2017)
10. Basin, D., Cremers, C., Dreier, J., Radomirovic, S., Sasse, R., Schmid, L., Schmidt, B.: The tamarin manual (2019)
11. Basin, D., Dreier, J., Hirschi, L., Radomirovic, S., Sasse, R., Stettler, V.: A formal analysis of 5g authentication. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. pp. 1383–1396. CCS '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3243734.3243846, https://doi.org/10.1145/3243734.3243846
12. Bellare, M.: New proofs for nmac and hmac: Security without collision-resistance. In: Dwork, C. (ed.) Advances in Cryptology - CRYPTO 2006. pp. 602–619. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
13. Borgaonkar, R., Hirschi, L., Park, S., Shaik, A.: New privacy threat on 3g, 4g, and upcoming 5g aka protocols. Proceedings on Privacy Enhancing Technologies **2019**(3), 108 – 127 (2019), https://content.sciendo.com/view/journals/popets/2019/3/article-p108.xml
14. Braeken, A., Liyanage, M., Kumar, P.V., Murphy, J.: Novel 5g authentication protocol to improve the resistance against active attacks and malicious serving networks. IEEE Access **7**, 64040–64052 (2019)
15. van den Broek, F., Verdult, R., de Ruiter, J.: Defeating imsi catchers. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. pp. 340–351. CCS '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2810103.2813615, https://doi.org/10.1145/2810103.2813615
16. Brown, D.: Standards for efficient cryptography, sec 1: elliptic curve cryptography. Released Standard Version **1** (2009)
17. CableLabs: A comparative introduction to 4g and 5g authentication. https://www.cablelabs.com/insights/a-comparative-introduction-to-4g-and-5g-authentication (2019)
18. Cao, J., Yan, Z., Ma, R., Zhang, Y., Fu, Y., Li, H.: Lsaa: A lightweight and secure access authentication scheme for both ues and mmtc devices in 5g networks. IEEE Internet of Things Journal (2020)
19. Choudhary, G., Kim, J., Sharma, V.: Security of 5g-mobile backhaul networks: A survey. CoRR **abs/1906.11427** (2019), http://arxiv.org/abs/1906.11427
20. Cremers, C., Dehnel-Wild, M.: Component-based formal analysis of 5g-aka: Channel assumptions and session confusion. In: Proceedings of the Network and Distributed Systems Security (NDSS) Symposium (2019), https://dx.doi.org/10.14722/ndss.2019.23394
21. Dabrowski, A., Petzl, G., Weippl, E.R.: The messenger shoots back: Network operator based imsi catcher detection. In: International Symposium on Research in Attacks, Intrusions, and Defenses. pp. 279–302. Springer (2016)
22. Dabrowski, A., Pianta, N., Klepp, T., Mulazzani, M., Weippl, E.: Imsi-catch me if you can: Imsi-catcher-catchers. In: Proceedings of the 30th annual computer security applications Conference. pp. 246–255 (2014)

23. Dickinson, A.: Blockchain for invoice reconciliation and dispute resolution. https://www.ibm.com/blogs/blockchain/2020/11/blockchain-for-invoice-reconciliation-and-dispute-resolution/ (11 2020), (Accessed: 2021-05-31)
24. Gosh, A.: How blockchain is helping telecom companies prepare for 5g. https://info.kpmg.us/news-perspectives/technology-innovation/blockchain-streamlining-settlements-between-telecom-companies.html (11 2021), november 2019. (Accessed: 2021-05-31)
25. H. Krawczyk and M. Bellare and R. Canetti: HMAC: Keyed-hashing for message authentication. Internet Request for Comments (RFC) 2104 (Feb 1997), https://tools.ietf.org/html/rfc2104
26. Hussain, S.R., Echeverria, M., Karim, I., Chowdhury, O., Bertino, E.: 5greasoner: A property-directed security and privacy analysis framework for 5g cellular network protocol. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 669 – 684. CCS '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3319535.3354263, https://doi.org/10.1145/3319535.3354263
27. Infosys: 5 major blockchain use cases for the telecom industry — infosys. https://www.infosys.com/insights/other-insights/documents/blockchain-5g.pdf (11 2019), (Accessed: 2021-05-31)
28. Ishai, Y., Kushilevitz, E.: Private simultaneous messages protocols with applications. In: Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems. pp. 174–183. IEEE (1997)
29. Khan, H., Martin, K.M.: On the efficacy of new privacy attacks against 5g aka. In: Proceedings of the 16th International Joint Conference on e-Business and Telecommunications — Volume 2: SECRYPT,. pp. 431 – 438. INSTICC, SciTePress (2019). https://doi.org/10.5220/0007919704310438
30. Khan, R., Kumar, P., Jayakody, D.N.K., Liyanage, M.: A survey on security and privacy of 5g technologies: Potential solutions, recent advancements, and future directions. IEEE Communications Surveys Tutorials **22**(1), 196 – 248 (2020)
31. Kolesnikov, V., Kumaresan, R., Rosulek, M., Trieu, N.: Efficient batched oblivious prf with applications to private set intersection. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 818–829 (2016)
32. Koutsos, A.: The 5g-aka authentication protocol privacy. In: 2019 IEEE European Symposium on Security and Privacy (Euro S & P). pp. 464 – 479 (2019)
33. Lowe, G.: A hierarchy of authentication specifications. In: Proceedings 10th Computer Security Foundations Workshop. pp. 31–43. IEEE (1997)
34. Meier, S.: Advancing automated security protocol verification. Ph.D. thesis, ETH, Zurich, Switzerland (2013), https://doi.org/10.3929/ethz-a-009790675
35. Ney, P., Smith, I., Cadamuro, G., Kohno, T.: Seaglass: enabling city-wide imsi-catcher detection. Proceedings on Privacy Enhancing Technologies **2017**(3), 39–56 (2017)
36. Piqueras Jover, R., Marojevic, V.: Security and protocol exploit analysis of the 5g specifications. IEEE Access **7**, 24956 – 24963 (2019)
37. Rellstab, A.: Formalizing and Verifying Generations of AKA Protocols (Master's thesis). Master's thesis, ETH Zurich, Zurich, Switzerland (2019)
38. Sharevski, F.: Towards 5g cellular network forensics. EURASIP Journal on Information Security (Jul 2018), https://doi.org/10.1186/s13635-018-0078-7

39. Sirotkin, A., Wu, G., Luft, A., Stojanovski, A.S.: Apparatus, system and method of lawful interception (LI) in a cellular network (Jun 2018), patent No. US 10,009,813 B2, Issued Jun. 26, 2018
40. Technplayon: 5g identifiers supi and suci - techplayon nas signalling. `https://www.techplayon.com/5g-identifiers-supi-and-suci/` (11 2019), november 2019. (Accessed: 2021-05-31)
41. Valtanen, K., Backman, J., Yrjola, S.: Creating value through blockchain powered resource configurations: Analysis of 5g network slice brokering case. In: 2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW). pp. 185 – 190 (2018)
42. Zhang, J., Yang, L., Cao, W., Wang, Q.: Formal analysis of 5g eap-tls authentication protocol using proverif. IEEE Access **8**, 23674 – 23688 (2020)

## A Table of Acronyms

**SIM** Subscriber Identity Module
**SUPI** Subscription Permanent Identifier
**SUCI** Subscription Concealed Identifier
**AKA** Authentication and Key Agreement
**HN** Home Network
**SN** Serving Network
**UE** User Equipment
**IMSI** International Mobile Subscriber Identity
**CSP** Communication Service Provider
**LEA** Law Enforcement Agency
**SEAF** Security Anchor Function
**EAP** Extensible Authentication Protocol
**MAC** Message Authentication Code
**ABBA** Anti-Bidding down Between Architectures
**PRF** Pseduorandom Function
**RAP** Random Access Procedure
**MME** Mobility Management Entity
**HSS** Home Subscriber Server
**NAS** Non-access Stratum
**RRC** Radio Resource Control

## B Tamarin Models of 5G AKA

The model we adopt is a transition system that describes the behaviour of a UE, an SN, an HN, and an adversary in a 5G AKA instance. The state of the system consists of a (multi)set of "facts" that currently hold. An example of a fact is `St_1_SEAF(c, x, <'5G:', x>, suci, y)`, which holds when SN `x` has sent a SUCI to HN `y` on channel `c`, but is yet to receive a response [1]. The transition relation governs the behaviour of each party. The construct shown in the Figure 5 for instance, called a "rewrite rule", specifies that an SN is to forward RAND and AUTN to the UE as received from the HN. The construct

has two parts: a set of "source" facts and a set "sink" facts [10]. Tamarin, in exploring the state space of the system, attempts to find a rule with a set of source facts that all hold currently, then replaces these facts with the "sink" facts of the rewrite rule.

```
rule seaf_receive_aia_send_authReq:
    let
        5G_AV = < RAND, HXRES_star, AUTN >
        msgIn = 5G_AV
        msgOut = < RAND, AUTN, snID >
    in
[St_1_SEAF(~tid, ~idSN, snID, conc_supi, idHN),
 RcvS(idHN, ~idSN, <'aia', msgIn>)]
--[...]->
[St_2_SEAF(~tid, ~idSN, snID, conc_supi, idHN,
RAND, HXRES_star, K_seaf), Out(msgOut)]
```

**Fig. 5.** Part of a rewrite rule from the formal model of 5G AKA we study in this work [1]: the rule specifies that a serving network that has completed Step (2) of 5G AKA and has received a message containing three values from the home network on the channel dedicated for this 5G AKA instance, can proceed by forwarding the message to the UE if the last of the three values matches the SN Id. The `St_1_SEAF` predicate on the lefthand side of the rule indicates that the SN has completed Step (2): in particular, that the SN has sent a concealed SUPI, `conc_supi`, to HN `idHN` on channel ("thread") `tid`. The `Rcvs` predicate indicates that the SN has received a message from the HN that begins with the string '`aia`' ("authentication initiation answer" [1]). The `Out` predicate on the righthand side of the rule indicates that the message the SN constructs, `msgOut`, will become known to the adversary if the SN proceeds; and the `St_2_SEAF` predicate indicates that the SN will have completed Step (4). The message the SN constructs becomes available to the adversary since the air interface, over which the SN sends the message, is untrustworthy. A transition from a state to another has a set of labels; each of which is a fact a party believes to hold at the moment of transition. For example, the transition by a UE from a state in which it has completed Step (1) to a state in which it has completed Step (5) has, as a label, the fact "`Secret(<'UE', supi>, 'supi', supi)`", where `supi` is the SUPI bound to the UE.

Basin *et al.* [11] provide seven different models of 5G AKA: two of these assume the channel between an SN and an HN is "binding", which Basin *et al.* [11] determine is necessary for 5G AKA to guarantee non-injective agreement between the UE and the SN on $K_{SEAF}$, which means it is necessary if no adversary is to deceive an SN into associating $K_{SEAF}$ in one instance of 5G AKA with an incorrect SUPI. Agreement between the SN and the UE is however inapplicable for us (see below). Two of the other models model untraceability: one against a passive attacker and the other against an active attacker [1], but each consider only the air-interface messages of 5G AKA. Since we have not made any changes to 5G AKA that apply to messages exchanged between the UE and the SN, the

results implied by these two models remain unaffected, i.e., Anonymous AKA guarantees untraceability against a passive attacker only. Of the three remaining models, one discovers an active "linkability" [11] attack against 5G AKA [1]. We conjecture that the same attack would succeed on our new protocol (Section 3.1).

The model named '`5G_AKA_fix.spthy`' includes a fix to Step (3) of 5G AKA, which Basin *et al.* [11] propose to eliminate the need for a "key confirmation" step where the UE and the SN each sign a public string using $K_{SEAF}$ and sends the signature to the other. As this fix has not been incorporated into the specifications [7], we adopt the plain '`5G-AKA-nonbindingChannel\5G_AKA.spthy`' model, which incorporates an explicit key confirmation step, for analysis.

Basin *et al.* [11] verified that 5G AKA has 35 authentication properties, 8 secrecy properties, and one privacy property which is equivalent to a secrecy property [11]. Of the 35 authentication properties, 16 are inapplicable for us. These are agreement properties to the SN with the UE, and agreement properties to the UE with the SN, which require that the SN establishes the identity of the UE at some point during the 5G AKA instance [11]. This, again, goes counter to our premise.

**Our modifications** Our first set of modifications to this model was to make it concurrent with the specifications [7]. We noticed two discrepancies. First, Basin *et al.* [11] include $K_{SEAF}$ in message (3) of 5G AKA (and exclude it from message (7)). Second, Basin *et al.* [11] include SUCI in message (6). We updated the model file so the message does not include $K_{SEAF}$. We also updated the file so the message includes $K_{SEAF}$. We finally updated the file so message (6), so the message does not include SUCI. These three changes necessitated that we make other modifications to the model so it is well-formed. For example, we modified line 382 of the file to have the HN maintain $K_{SEAF}$ as a state variable so the HN is able to share it with the SN in Step (7).

Our second set of modifications was to introduce an `Fr` fact into the "sink" facts of rewrite rule "`hss_receive_ac_send_aca`" [1], which models the final step of 5G AKA. According to the Tamarin Manual [10], the `Fr` fact denotes a "freshly generated name", i.e., a data item that is distinct from all other data items in the model. We then replaced the second component of the message that the HN sends so it is the freshly generated name rather than the actual SUPI. The new rule is shown in Figure 6.

**Anonymity in Anonymous AKA** We also verified, using the same model, that an SN does not learn the SUPI in an instance of Anonymous AKA: We did this by modifying a "SUPI secrecy" property from the model which we reproduce in Figure 7 and interpret here as [1]:

> "No adversary can compute the SUPI in any instance of 5G AKA, unless the adversary compromises at least one channel that connects an HN to an SN, or the adversary forces the HN to disclose the private portion of its public-private key pair, or the adversary forces the UE to disclose the SUPI."

```
rule hss_receive_ac_send_aca:
let
    SNID = <'5G', idSN>
    CK = f3(~k, ~RAND)
    IK = f4(~k, ~RAND)
    AK = f5(~k, ~RAND)
    K_seaf = KDF(KDF(<CK, IK>, <SNID, Sqn XOR
    AK>), SNID)
    msgIn = <XRES_star, suci, SNID>
    msgOut = <'confirm', ~rsupi, K_seaf>
in
    [St_1_HSS(~tid, ~idHN, ~supi, suci, idSN,
    SNID, ~k, Sqn, XRES_star, ~RAND, ~sqn_root,
    ~sk_HN, K_seaf),
    RcvS(idSN, ~idHN, <'ac',msgIn>), Fr(~rsupi)]
    --[...]->
    [SndS(~idHN, idSN, <'aca', msgOut>)]
```

**Fig. 6.** Rewrite rule that models the final step of Anonymous AKA (updated from [1]): the identifier `rsupi` here is a fresh name, and `msgOut` is the message that the HN sends. So instead of sending the SUPI in message (7), the HN now sends a random string of numbers.

'Compromise a channel' here means the adversary impersonates an SN to an HN, and can therefore read every message that the HN sends the SN, including message (7) of 5G AKA which includes the SUPI. Making this qualification is clearly necessary for 5G AKA, but it should not be for Anonymous AKA. We modified this property by removing the "unless the adversary compromises..." qualification. We then used Tamarin to check whether Anonymous AKA has the modified property; Tamarin produced a a proof that the protocol indeed has the property, implying that the protocol possesses the property regardless of whether the SN is honest, malicious, or honest-but-curious.

## C Proof of Theorem 1

We begin by recalling the concept of a pseudorandom function and the notion of 'indistinguishability from a random bitstring by an adversary'.

**Definition 1 (Pseudorandom Function).** *A function* $f : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^m$ *is a* $(t, \epsilon, q)$*-PRF if*

- *Given a key* $K \in \{0,1\}^k$ *and an* $s$*-bit input* $S$ *there is an efficient (read: "polynomial in both* $k$ *and* $s$*"-time) algorithm to compute* $f_K(S) = f(K, S)$.
- *For any* $t$*-time oracle algorithm* $A$*, we have*

$$\left| \Pr\left[A^{f_K}, K \leftarrow_{\$} \{0,1\}^k\right] - \Pr\left[A^{\mathcal{F}}\right] \right| < \epsilon$$

```
All supi t #i.
    Secret(<'UE', supi>, 'supi', t)@i
==> not (Ex #j. K(t)@j)
| (Ex X #r. Rev(X, 'secureChannel')@r)
| (Ex X k #r. Rev(X, <'skHN',k>)@r
        & Honest(X)@i)
| (Ex X s #r. Rev(X, <'supi',s>)@r
        & Honest(X)@i)
```

**Fig. 7.** A security property from a formal model of 5G AKA [1]: the `All` keyword stands for universal quantification; the `Ex` keyword for existential quantification. A `Secret(<'UE', supi>, 'supi', supi)` fact labels a transition by a UE whose SUPI is `supi` from a state in which it has performed Step (1) of 5G AKA to a state in which it has completed Step (5), and indicates that the UE believes the SUPI is secret [10]. The same transition has, as labels, the following three facts: "`Honest(supi)`", "`Honest(idHN)`", and "`Honest(idSN)`", where `idSN` is the identity of the SN from which the UE has received message (4), and `idHN` is the home network identifier. The `K(t)` fact holds when the adversary knows the value of `t`. A `Rev(X, 'secureChannel')` fact holds if the adversary has "compromised" the party whose identifier is `X`, which is either an HN or an SN. A `Rev(idHN, <'skHN',k>)` fact holds when a disclosure of the private portion of the public-private key pair of HN `idHN` has occurred; a `Rev(supi, <'supi',supi>)` fact holds when the UE whose SUPI is `supi` has disclosed the SUPI, `supi`. The variables `i`, `j`, and `r` are "timepoint" [10] variables. The syntax `P(x)@i` where `P` is a fact, `x` is a variable, and `i` is a timepoint variable reads: "Fact `P` holds on variable `x` at time `i`."

*where $\leftarrow_{\$}$ denotes sampling from a distribution,* $\Pr[A]$ *denotes the probability distribution function of A's output, the absolute difference between two probability distributions is their statistical difference, $\mathcal{F}$ is a random oracle, and A makes at most q queries to the oracle.*

**Definition 2 (Indistinguishability).** *Let D and E be two probability distributions; we say D and E are $(t, \epsilon)$-indistinguishable if for any t-time algorithm A, we have*

$$\left| \Pr[A(x), x \leftarrow_{\$} D] - \Pr[A(x), x \leftarrow_{\$} E] \right| < \epsilon.$$

*Proof.* Suppose $D$ is $(t, \epsilon)$-indistinguishable from random. Then the distribution of:

$$hmac(\mathrm{K_{SEAF}}, \mathrm{SUPI} \,||\, |\mathrm{SUPI}| \,||\, \mathrm{ABBA} \,||\, |\mathrm{ABBA}|)$$

should be $(t - T, \epsilon)$-indistinguishable from that of:

$$hmac(\mathcal{K}_0, \mathrm{SUPI} \,||\, |\mathrm{SUPI}| \,||\, \mathrm{ABBA} \,||\, |\mathrm{ABBA}|),$$

where $\mathcal{K}_0$ is a bitstring uniformly distributed in $\{0,1\}^{256}$. Now assume HMAC is a $(t, \epsilon, 1)$-PRF. It follows that the distribution of each of:

$$hmac(\mathcal{K}_0, \mathrm{SUPI} \,||\, |\mathrm{SUPI}| \,||\, \mathrm{ABBA} \,||\, |\mathrm{ABBA}|)$$

and

$$hmac(\mathcal{K}_0, \text{ABBA} \,||\, |\text{ABBA}|)$$

is $(t, \epsilon)$-indistinguishable from the uniform distribution over $\{0,1\}^{256}$. It then follows that the distributions of:

$$hmac(\mathcal{K}_0, \text{SUPI} \,||\, |\text{SUPI}| \,||\, \text{ABBA} \,||\, |\text{ABBA}|)$$

and

$$hmac(\mathcal{K}_0, \text{ABBA} \,||\, |\text{ABBA}|)$$

are $(t, 2\epsilon)$-indistinguishable from each other, and, since each is $(t-T, \epsilon)$-indistinguishable from their counterpart computed using key $\text{K}_{\text{SEAF}}$, it follows that the distributions of:

$$hmac(\text{K}_{\text{SEAF}}, \text{SUPI} \,||\, |\text{SUPI}| \,||\, \text{ABBA} \,||\, |\text{ABBA}|)$$

and

$$hmac(\text{K}_{\text{SEAF}}, \text{ABBA} \,||\, |\text{ABBA}|)$$

are $(t - T, 4\epsilon)$-indistinguishable.

*Remark 5.* Note that the proof assumes a single instance of AKA is running and that we implicitly assumed that the sequence number check at the UE passes (otherwise the UE will not compute $\text{K}_{\text{SEAF}}$). That our proof assumes a single session of AKA leaves open the question of whether the proof composes. We defer an answer this question for future work.