# Security vulnerability in 5G-AKA draft

(3GPP TS 33.501 draft v0.7.0)

Martin Dehnel-Wild and Cas Cremers

Department of Computer Science, University of Oxford.
{martin.dehnel-wild,cas.cremers}@cs.ox.ac.uk

v1.0 – 8th February 2018

## Summary

The 5th Generation (5G) mobile networks and telecommunications standards are currently under development, and are nearly finalised. We analyse the security properties of the main 5G-AKA protocol within the February 2018 version of the draft standard. Our analysis reveals a security vulnerability in the proposed 5G-AKA protocol as specified within 3GPP TS 33.501 v0.7.0. The discovered protocol vulnerability would allow a malicious actor (with no privileged network access) to impersonate another user to a Serving Network, for example in a roaming scenario.

We found the vulnerability by performing formal symbolic analysis of the protocol standard using the Tamarin Prover. After describing the protocol and the vulnerability, we provide possible fixes.

## 1 Introduction

We present a newly-found security vulnerability in the proposed 5G-AKA protocol, as specified within 3GPP TS 33.501 v0.7.0 [1]. The protocol vulnerability would allow an attacker (with no privileged network access) in a roaming scenario to impersonate another user to a Serving Network. This could potentially allow malicious users to bill expensive phone calls or access charges to other legitimate users, after eavesdropping on their initial connection. This attack was found through use of the Tamarin Prover, a tool for the symbolic verification of cryptographic protocols [3]. As it is exceptionally challenging to perform manual security analyses of complex, multiparty protocols such as 5G-AKA, we believe this further demonstrates the necessity of automated verification for security critical functionality.

This summary document aims to give sufficient information to allow the 3GPP SA3 committee and others to consider the attack, its implications, and its potential mitigations. A full technical report will follow.

In Section 2 we recall the main protocol features and describe the found vulnerability in Section 3. We then discuss implications in Section 4 and provide two possible fixes in Section 5.

# 2 The 5G-AKA protocol

The 5G-AKA protocol is the flagship "Authentication and Key Agreement" protocol within the newly proposed 5G standard. The protocol is specified within §6.1.3.2 of 3GPP Technical Specification 33.501 (we model v0.7.0) [1], and is proposed as the main method of authentication and key agreement between a mobile device and its Home Network. The design of the 5G-AKA protocol is directly based on the EPS-AKA* protocol as used by 4G/LTE [2].[1]

5G-AKA is a four-party protocol (in both the roaming and non-roaming context). These parties are:

- **UE**: the 'User Equipment'.
  This can be for example mobile phones or USB 5G dongles. Each UE is uniquely identified by its SUbscription Permanent Identifier (**SUPI**). In 5G, the SUPI performs the same role as the 'IMSI' in pre-5G standards.
- **SEAF**: the 'Security Anchor Function'.
  In the roaming context, this is within the Serving Network.
- **AUSF**: the 'Authentication Server Function'.
  This role falls within the Home Network.
- **ARPF**: the 'Authentication credential Repository and Processing Function'.
  This also falls within the Home Network, and may typically reside within a secure location, such as a Hardware Security Module.

The UE and ARPF alone share the user's long-term secret symmetric key, **K**. In 5G-AKA, the SUPI should never be exposed publicly; a 'SUbscribption Concealed Identifier' or **SUCI** is used to achieve this. The **SIDF**, or the Subscriber Identity De-concealing Function is used to decrypt a SUCI value into a SUPI: this functionality resides within (or is co-located with) the ARPF.

At the end of a successful run of the protocol, all parties should all share (or at least be able to derive) *and agree upon* an 'anchor key' $\mathbf{K_{SEAF}}$, from which session keys for communication between the mobile device and base station(s) within the local network are derived. The secrecy of the $\mathrm{K_{SEAF}}$ key is therefore crucial to ensure the security of subsequent operations and communications.

## 2.1 Channel properties

The 5G-AKA protocol leverages the communications channels between the four involved parties. The channels specified by TS 33.501 for the 5G-AKA protocol are:

1. UE ↔ SEAF
2. SEAF ↔ AUSF
3. AUSF ↔ ARPF

The standard specifies which of these connections should be secured in which way. Concretely, the communications between SEAF, AUSF, and ARPF are within the "5G Core Network". We cite the precise properties required of "e2e core network interconnection" channels as described in TS 33.501, verbatim:

[1] We have no direct reason to believe the EPS-AKA* protocol is similarly vulnerable to the presented attack, due to major architectural differences.

> **5.7.4: Requirements for e2e core network interconnection security** (from [1] p. 21)
>
> A solution for e2e core network interconnection security shall satisfy the following requirements.
>
> - The solution shall provide confidentiality and/or integrity end-to-end between source and destination network for specific message elements identified in this specification. [...]
> - The destination network shall be able to determine the authenticity of the source network that sent the specific message elements protected [...]
> - The solution should be using standard security protocols.
> - The solution shall cover prevention of replay attacks.

In the process of this research, we precisely modeled Channels 2 and 3 to meet these properties. Note however that these channel properties do not explicitly require (or guarantee) delivery of messages, nor of ordering of the receipt of messages. We believe that these properties are analogous (or at least very close) to setting up and maintaining long-term IPSec, (D)TLS, or DIAMETER sessions over these channels, between the named parties. We discuss the further implications of stronger channel properties than these (in light of the discovered attack) in Appendix C.

The standard does not specify any assumed security for the channel between UE and SEAF: in some sense, this is part of what 5G-AKA aims to provide. We therefore assume that channel between the UE and SEAF is considered insecure; we model it such that it is attacker-controlled, eavesdroppable, replayable, and the channel itself is without any cryptographic protections (individual messages may of course use their own cryptographic protections directly). We therefore arrive at the following overview:

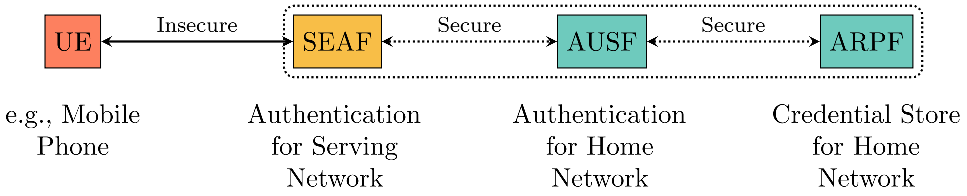
UE
Insecure
SEAF
Secure
AUSF
Secure
ARPF
e.g., Mobile Phone
Authentication for Serving Network
Authentication for Home Network
Credential Store for Home Network

Figure 1: Parties and channels involved in the 5G-AKA protocol. Parties inside the dashed box are within the 5G Core Network; dashed channels are considered 'secure'. The left two parties (UE and SEAF) can be remote, or roaming, and the right two parties are within the Home Network.

## 2.2 Normal execution of the 5G-AKA protocol

We now give a simplified overview of the 5G-AKA protocol execution, for illustrative purposes. See Figure 2 for a message sequence chart of the normal flow of the protocol, which we describe below. We omit the details of messages that are not needed to understand the attack later.

1. The UE sends its ephemerally encrypted 'concealed ID' (SUCI) and the name of its Home Network to the nearest SEAF.
2. The SEAF sends a `5G-AIR` message containing this and the name of the Serving Network to an AUSF in the relevant Home Network.
3. The AUSF then transmits this information in an '`Auth-Info Request`' message to the Home Network's ARPF.

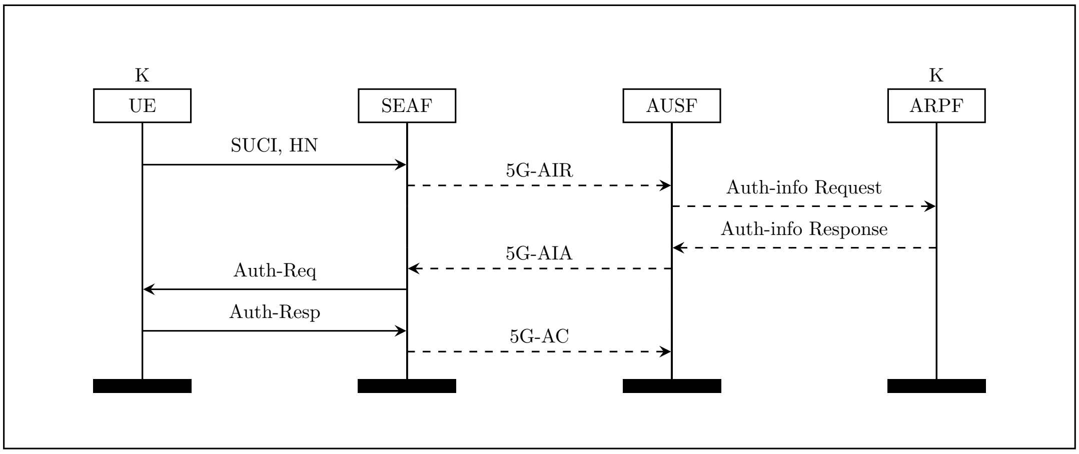

Figure 2: The normal flow of the 5G-AKA Protocol. Only the UE and ARPF know the user's long-term key, K. Dashed lines indicate messages sent over secure channels.

4. The ARPF
   (a) De-conceals the SUCI into its respective SUPI. This technically occurs in the SIDF, but these are normally co-located. The ARPF uses the SIDF to find the relevant K for this user, and then
   (b) Generates a random number 'RAND', a number of Authentication Vectors (derived from RAND and the user's long term key K), an 'Expected Response' value (XRES*) so that other parties can verify that the user responded correctly without knowing the long term K, and a session key for the AUSF, $K_{\mathrm{AUSF}}$. These are transmitted to the AUSF in an '`Auth-Info Response`' message.
5. The AUSF sends a `5G-AIA` message containing the Authentication Vectors, a hash of the 'Expected Response' (i.e. HXRES*), the new 'anchor key' $K_{\mathrm{SEAF}}$ (derived from $K_{\mathrm{AUSF}}$), and the SUPI of the intended recipient.
6. The SEAF sends RAND and the Authentication Vectors to the UE in an Auth-Req message.
7. The UE proves its identity (and implicitly, ownership of K) by responding to the SEAF with RES* (i.e. the 'Response') within an Auth-Resp message; the UE can now calculate the keys $K_{\mathrm{AUSF}}$ and $K_{\mathrm{SEAF}}$.
8. The SEAF calculates the hash of RES* (i.e. HRES*) received from the UE, and checks that it matches with the hash of the '*Expected* Response', HXRES* value from the AUSF. If it matches, the SEAF considers the authentication to have been successful, and sends an Authentication Confirmation (`5G-AC`) message containing the user's SUPI, the Serving Network's ID, and optionally the response, to the AUSF.

## 3 Protocol vulnerability

Using formal analysis, we found an attack on this protocol. We first give an informal overview before giving an in-depth description.

A malicious actor 'B' starts two 5G-AKA sessions with a local Serving Network at roughly the same time. One session is initiated by replaying an overheard SUCI (of the target, user 'A'), and the other session is with the malicious actor's own USIM and SUCI (for user 'B'). The

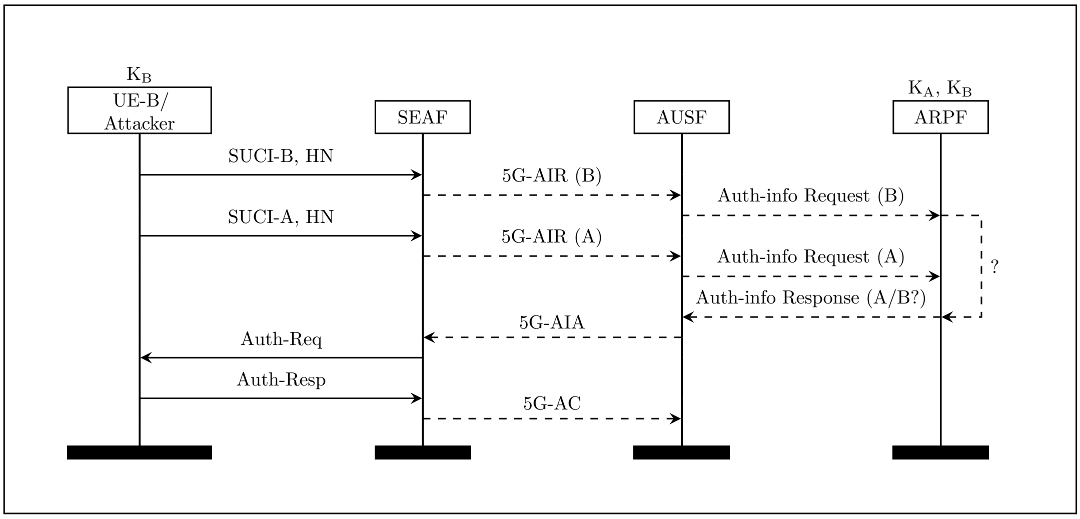

Figure 3: The **attack** flows of the 5G-AKA Protocol.

sessions run in parallel, and result in a race-condition; if this occurs, the AUSF will be unable to distinguish between the two responses containing the Authentication Vectors from the credential store (ARPF), and is liable to associate the wrong response (and resultant keys) with the wrong user. In the case that this occurs, the AUSF and SEAF will incorrectly believe that a set of Authentication Vectors and 'anchor key' were intended for user A (and derived from user A's long term key $K_A$), when they were in fact derived from user B's long term key $K_B$. As a result, the malicious user B will now be able to derive the anchor key, and use it to impersonate user A to the network. See Figure 3 for the message sequence chart of the attack.

## 3.1 What does the attack break?

We now give a more in-depth description of the attack and violated properties.

The specific property which is violated is the secrecy of the 'anchor key' $K_{SEAF}$ (and its cryptographic parent, $K_{AUSF}$), *from the points of view of the SEAF and AUSF.* That is, at the end of the 5G-AKA protocol run:

- the SEAF, AUSF, and a user will have agreed and be in possession of a cryptographic anchor key ($K_{SEAF}$),
- the SEAF and AUSF believe this key is for an honest and un-compromised user (in our example, user 'A' with 'SUPI-A'), and,
- both the SEAF and AUSF believe this key is secret from the attacker.

Thus, the protocol draft lacks a crucial containment property: an attacker that can compromise the long-term key of a user (e.g., 'B') will be able to impersonate *any* user (e.g., 'A') to the SEAF and the AUSF, because it knows the $K_{SEAF}$ of sessions that the SEAF and AUSF assume to be from 'A'.

## 3.2 Detailed attack scenario

The attack takes place in two (possibly temporally and even geographically) separate phases. In the first phase, the attacker eavesdrops and records a legitimate 'encrypted SUPI', also known as a SUCI. In the second phase, the main body of the attack takes place. Full message definitions can be found in TS 33.501 [1].

### 3.2.1 Setup to the attack

1. A legitimate user 'A' with ID 'SUPI-A' is registered with its Home Network (HN). We are not interested in its long term key $\mathrm{K_A}$, as the attack does not require access to it. This honest user initialises the 5G-AKA protocol, sending the SUCI-A (user A's ephemerally encrypted SUPI) and 'HN' to a SEAF. The user might then complete the protocol as normal.

2. The attacker eavesdrops on the public radio transmissions from the previous step, and records the message containing SUCI-A and HN.[2]

3. The attacker purchases a legitimate USIM from the same Home Network as his intended victim; this has ID 'SUPI-B'. The attacker physically attacks and compromises the USIM, and extracts the long term key $\mathrm{K_B}$ of this USIM in its possession.

### 3.2.2 Main phase of the attack

The message sequence chart of the main phase of the attack can be found in Figure 3.

1. After the setup phase, the attacker (perhaps within the physical realm of a visiting network) initiates the 5G-AKA protocol by replaying to a SEAF the pre-recorded SUCI-A. This is the concealed ID of SUPI-A, but the attacker does not need to know whose ID it is, or its decrypted form. The attacker sends a message containing 'SUCI-A' and the name of the user's Home Network ('HN') to a SEAF in a Serving Network (of name SNID).

2. The protocol proceeds as normal: the SEAF communicates with an AUSF in the specified Home Network by sending the "`5G-AIR`" message. This contains 'SUCI-A', and SNID (i.e. the ID of the Serving Network being used).

3. In parallel with the session for SUCI-A, the attacker starts a 5G-AKA session for the USIM it owns (SUPI-B) with the same Home Network, via the same Serving Network (and SEAF). The attacker is already in possession of SUPI-B's long term key ($\mathrm{K_B}$), as it has compromised the USIM in the setup phase. As before, it starts the 5G-AKA session by sending its own concealed ID ('SUCI-B') and the name of the Home Network (HN), to the same SEAF as in the other, parallel session. The SEAF clearly and correctly treats this as a separate session.

4. As before, the SEAF communicates with the AUSF in the Home Network by sending the "`5G-AIR`" message, containing 'SUCI-B' and SNID. The AUSF then sends the '`Auth-Info Request`' message to the Home Network's ARPF, as per the protocol.

5. The SIDF (within the ARPF) de-conceals SUCI-B into SUPI-B, and the ARPF then responds by sending the '`Auth-Info Response`' message to the AUSF. This message contains terms derived from (the compromised) $\mathrm{K_B}$, and the terms RAND, SQN, and SNID, but notably contains no reference to either the SUPI or the SUCI.

[2] See Appendix B for a minor comment on the use of replayed SUCIs.

6. The ‘`Auth-Info Response`’ message is received by the AUSF, but as this message does not have a SUPI or SUCI attached to it, the AUSF *does not know whether this message was for the session with ‘SUCI-A/SUPI-A’, or whether it was for the session with ‘SUCI-B/SUPI-B’*. The AUSF can legitimately continue its session intended for ‘SUCI-A/SUPI-A’ with the ‘`Auth-Info Response`’ message that was actually intended for the session with ‘SUPI-B’.

7. The AUSF then proceeds with the protocol, sending the `5G-AIA` message for ‘SUPI-A’ to the SEAF; this contains the anchor key $\mathrm{K_{SEAF}}$ that the ARPF generated for ‘SUPI-B’, but now the AUSF associates it with ‘SUPI-A’ (and as a result, so does the SEAF). As the attacker has compromised SUPI-B’s long term key $\mathrm{K_B}$ (and RAND and SQN are publicly transmitted during the protocol), the attacker can now construct the anchor key $\mathrm{K_{SEAF}}$ that the AUSF and SEAF now believe is the anchor key for ‘SUPI-A’. That is, the attacker can derive the $\mathrm{K_{SEAF}}$ that the AUSF and SEAF believe to be for the (honest) ‘SUPI-A’ (and *not* ‘SUPI-B’ which the attacker has compromised). Hence, we have an attack.

It is worth noting two things:

This attack cannot work accidentally. The race condition occuring benignly, and the wrong Authentication Vectors (AVs) being accidentally delivered to the wrong (honest) user will not cause a violation of the security property. An honest USIM receiving the wrong AVs would calculate a different MAC to the value contained within the received AVs; at this point, the UE would reject the authentication attempt (as this failure might indicate to the user that an attacker had attempted to modify the messages en route) and try the protocol again.

Secondly, **counters** or SQN values do not have any bearing on this attack, as only the ARPF and UE store what the ‘correct’ value of SQN is. The AUSF and SEAF do not use SQN directly in any calculations or derivations, and hence do not check whether it matches (or is greater than) stored values for a particular user. The attacker can of course accept Authentication Vectors (leading to an anchor key) generated with any SQN value, and is able to deduce directly which SQN value was used. In other words, while counters are used in the protocol to prevent some forms of replay, they are used in exactly the other direction as the one the attack proceeds in.

# 4 Statement of implications

This protocol attack allows a malicious actor, or an attacker that compromised the long-term key of an actor, to impersonate *another* user to a Serving Network. Strictly from the point of view of the 5G-AKA protocol, this attack allows the attacker to agree upon an anchor key (and thereby gain access to a Serving Network) dishonestly, under the newly generated false credentials of a legitimate user. This is a substantial containment problem.

We note that this attack holds whether the protocol uses encrypted/concealed SUPIs or not, i.e. the attack holds both when the protocol uses the null-scheme for ‘SUPI (non-)concealment’, *and* when SUPIs are concealed. We make this distinction because the SEAF and AUSF know strictly *more* about the claimed identities of session owners in the non-concealed situation than they do when identities are concealed. This, in turn, implies that there is a deeper identity mis-binding issue that is not caused directly by SUPI/SUCI encryption. (We first discovered this attack in the non-concealed setting, and then saw that it also applied to the concealed protocol.)

This attack relies on a race-condition between two sessions of the protocol. This means the attack is probabilistic, and an attacker would not be able to guarantee success on every run; however, in any secure protocol, there ought not exist *any* run of the protocol under normal circumstances which violates the required security properties.

N.B. This does **not** allow an attacker to decrypt any radio traffic (past or present) originally generated by the impersonated, legitimate user.

## 4.1 Potential practical implications

In the real world, we conjecture that this attack might allow an attacker to access a Serving Network (and its services) in the name of a legitimate user other than itself. This attacker could then bill services, air-time, or access charges to another user account, rather than its own; this is clearly not the intended behaviour or level of security required within 5G networks.

We are not confident of the range of further authentication and authorisation procedures which may or may not be in place distinct from the 5G-AKA protocol, or any billing–authentication procedures: e.g., whether specific billing actions sent back to a user's Home Network are tied to and verified against a named anchor key or not (we note that an ARPF *would* be able to establish that the anchor key was not for the correct user; but that an AUSF or any other party within the 5G Core Network would not). We do, however, believe it is plausible that once access is granted in the form of an anchor key,[3] this key is sufficient to allow a user to perform the normal range of actions within a network.

It is also plausible that the same attack works when the SEAF is within the Home Network (rather than the attack we describe, with the SEAF in the Serving Network) – and thus the attacker would not have to be in the physical realm of a Serving Network. We have not yet modeled any distinctions between the two situations, so only make the weaker claim.

We acknowledge that there may be other technical measures within 5G that make full implementation of this attack impossible in the real world. Regardless, an authentication and key agreement protocol must meet its own required security properties.

If there are other methods or mechanisms for further authentication, then the strongest statement we can make is that the 5G-AKA protocol on its own does not meet its security requirements. However, as the primary method for authentication and key agreement within 5G, we believe that 5G networks should not rely solely upon secondary mechanisms for security; we believe this is sufficient reason on its own to fix the protocol to prevent this and similar attacks.

# 5 Proposed fixes

The essence of the attack is identity mis-binding that can occur when the sessions in the channel between AUSF and ARPF are not bound tightly enough. This leads to two possible ways to prevent the attack: one can either bind the identities of the intended parties to each message all the way through the 5G-AKA protocol, or one can ensure a one-to-one mapping between the high-level 5G-AKA sessions and its internal AUSF ↔ ARPF sessions.

Using the Tamarin Prover, we have formally verified that both solutions prevent the attack.

### Fix 1: Explicit identity binding

To improve identity binding, we propose the following two minor changes to the protocol:

- In the `Auth-Info Response`[4] message sent from the ARPF to the AUSF, the SUPI *and SUCI* of the intended UE should be added to the message.
- In the `5G-AIA`[5] message sent from the AUSF to the SEAF, the SUCI should be added to the message. The SUPI is already included in the concealed setting; this must additionally be included in the non-concealed setting.

[3] And the resultant ability to derive the other keys (such as $K_{gNB}$, CK, and IK) associated with it; see the 5G Key Hierarchy in [1, Figure 6.2.1-1]

[4] As specified in 6.1.3.2-2 of TS 33.501 v0.7.0.

[5] As specified in 6.1.3.2-5 of TS 33.501 v0.7.0.

These minor changes (adding the relevant identities to each of the `Auth-Info Response` and `5G-AIA` messages) now successfully bind the correct parties to the messages throughout the full flow of the protocol, preventing this (and other) identity mis-binding attacks from working.

The proposed modifications have negligible impact on the computational efficiency of the protocol. If the addition of the full SUCI value to these two messages is deemed to require too much bandwidth, using a cryptographically secure hash of the SUCI can also suffice.

### Fix 2: Tighter session binding

The attack is dependent on the ability of messages between the AUSF and ARPF from one 5G-AKA session to end up in that channel for another 5G-AKA session. Currently, there is nothing in the protocol specification that prevents this. This binding can be fixed in several ways, for example:

1. Including a fresh (unique, random) value in Auth-info Request. The ARPF should include this in Auth-info Response, and the AUSF can check that they match.

2. Emulating individual sessions within a long-lived TLS or DIAMETER session between AUSF and ARPF by an intermediate layer. In practice, this boils down to turning the previous solution into a separate emulation layer that creates new session-identifiers for each Auth-info Request and expects the Auth-info Response to be bound to this session-identifier.

3. Initiating an entirely new TLS or DIAMETER session for each Auth-info Request, and expecting the Auth-info Response within that session.

### Alternative fixes

We have considered several alternative fixes, but they all seem either more complex or insufficient. For example, one might consider putting unique nonces in other ways in the channels to solve this attack, especially since this is likely to be implemented at a lower level due to engineering concerns. However, in Appendix C we give four separate techniques for including such nonces and evaluate their effectiveness.

## 6 Conclusion

In this document we have demonstrated an attack against the draft 5G-AKA protocol, which would allow a malicious actor to impersonate an honest user to a network. We propose two possible fixes, and we have verified the correctness of the proposed solutions using the Tamarin Prover. We strongly encourage 3GPP SA3 to adopt one of our proposed fixes immediately.

We recognise that standards often make implicit assumptions about the reality of engineering solutions, and that there may be other mechanisms in place to mitigate the real-world impact of this protocol attack. In particular, we conjecture that Fix 2, method 2 might be proposed as an implementation choice in practice, accidentally mitigating the attack. Even if that were the case, our analysis has newly revealed that such a mechanism would in fact be security-critical.

In general, we emphasise strongly that security critical properties of any security protocol *must not depend on implicit engineering solutions.* In other words, the specification of the standard should be such that any implementation provides the security properties. Currently, this seems not yet to be the case for 5G-AKA 0.7.0.

We recognise that identity binding is hard to get right (especially within complex, multi-party protocols with subtle channel assumptions); to this end, we believe that the discovery of this attack further demonstrates the necessity and importance of automated verification for security-critical functionality and protocols.

## 7 Acknowledgments

We would like to acknowledge and thank Ralf Sasse, Lucca Hirschi, and Vincent Stettler (ETH Zürich), Steve Babbage and Tim Evans (Vodafone Group), Nicholas Moore and Ravishankar Borgaonkar (University of Oxford) for their contributions, comments, and explanations throughout the process.

## References

[1] 3rd Generation Partnership Project (3GPP); Technical Specification Group Services and System Aspects (SA3): TS 33.501: Security Architecture and Procedures for 5G System (December 2017), version 0.7.0.

[2] 3rd Generation Partnership Project (3GPP); Technical Specification Group Services and System Aspects (SA3): TS 33.401: 3GPP System Architecture Evolution (SAE); Security architecture (January 2018), version 15.2.0.

[3] Meier, S., Schmidt, B., Cremers, C., Basin, D.A.: The TAMARIN Prover for the Symbolic Analysis of Security Protocols. In: Computer Aided Verification - 25th International Conference, CAV. pp. 696–701 (2013)

# A Responsible disclosure

We have opted not to follow any responsible disclosure path, as 5G (and 5G-AKA) is not yet an implemented, used, or complete standard; this document is intended to highlight the stated attack *during* the standardisation phase, rather than after it has been finalised.

# B Comment on replaying ephemerally encrypted SUCIs

The purpose of sending an ephemerally encrypted SUCI is to conceal the SUPI, maintaining the privacy and unlinkability of the UE's global ID. This attack does not violate that property. The UE uses an ephemeral public key (rather than static) to maintain its own unlinkability; hence the onus is on the UE to use a new ephemeral key with each run of the protocol to maintain this property. While the ARPF or SIDF maintaining a complete list of previously used ephemeral public keys is touched upon as a possible suggestion in TR 33.899, we find no evidence within TS 33.501 that this is required or even formally proposed, and therefore no evidence that an ARPF or SIDF will not accept a re-used ephemeral key or SUCI.

In terms of fixing the protocol (and preventing the discovered attack), including a counter within the ECIES-encrypted SUCI happens to be sufficient for the concealed setting, but isn't sufficient overall: in the case that the null-scheme is used for SUPI/SUCI 'encryption', the attack then holds again. In the non-concealed case adding a counter isn't sufficient: there are no cryptographic protections at this stage, so the attacker can simply increment the last counter it observed. TS 33.501 (and notably §6.12.2) does not mention a counter or any similar values within the SUCI.

If a counter is added to the SUCI, while this prevents the direct attack described in this document from succeeding (because replaying an overheard SUCI will no longer be sufficient), it means that an attacker merely has to learn a target user's un-concealed SUPI by some means to be able to impersonate them. It is our belief that the purpose of concealed IDs (and introduction of SUCIs rather than just SUPIs) is solely to maintain the privacy of the user's ID, not as a means to ensure the overall authentication and key-agreement properties of the protocol.

# C Varying channel properties

As described in Section 2.1, in the process of finding this attack, we have deliberately and explicitly modeled the network channels between the SEAF, AUSF, and ARPF precisely as TS 33.501 v0.7.0 describes, and with no greater or lesser protections or security capabilities.

We recognise that standards often make implicit assumptions about the reality of engineering solutions, so in this section we explore some possible additional properties that the secure network channels may observe in any real-world implementation: the properties which we now explore are **not** specified or required by TS 33.501.

We emphasise that the security properties of any authentication and key-agreement protocol *must not depend on implicit engineering solutions* – instead, such implicit requirements should be made explicit.

We note that the reason this specific attack works is identity mis-binding: there is nothing to ensure that the '`Auth-Info Response`' message sent by the ARPF isn't accidentally fed into a session initiated by an attacker.

In the following, we explore the range of outcomes when we bind these sessions together more tightly with specific Channel Session IDs, i.e. unique, random nonces inserted into each message to ensure that 'response' messages are paired up correctly with their original 'initiator' message's session. We first list four options for the channels and then discuss their resultant properties and implications.

## C.1 Four secure network models

There are several ways in which underlying channels might maintain session state IDs (or not) which have different consequences in the context of our attack. We define four different models of channel properties, with new session state IDs variously included in all messages transmitted over certain channels; this is to ensure pairing of specific session states before and after sending and receiving a message to another party.

**All four models** listed below maintain confidentiality, integrity, replay protection over the channels, and authenticity of the involved parties (i.e., the authenticity of the parties at either end of the secure channels, namely the SEAF, AUSF, and ARPF) as a minimum: this means that all of the below models meet or exceed the properties required by TS 33.501. The listed properties are in addition to the requirements of the standard.

1. In the first model, we do not include any channel-session IDs in transmitted messages. We believe this is identical to the required channel properties specified by TS 33.501. This model corresponds, for example, to using long-lived TLS sessions between two parties.

2. In the second model, we augment one channel's security properties by adding channel-session IDs just to messages sent over the AUSF ↔ ARPF channel. This means that all pairs of `Auth-Info Request` and `Auth-Info Response` messages now contain unique IDs to ensure that the `Auth-Info Response` message's contents are definitely transmitted directly to the same session state that sent the original `Auth-Info Request` message (and that it therefore shouldn't end up in the state for the wrong user). This model effectively approximates a new TLS session per protocol run between these two actors.

3. In the third model, we do broadly the same thing as in the second, but just between the SEAF and AUSF. Note that the AUSF does not forward on any information about the SEAF↔AUSF channel-session ID to the ARPF. This means that a single, unique ID is added (per session) to each of the `5G-AIR`, `5G-AIA`, and `5G-AC` messages sent between the SEAF and AUSF. This models channel-session IDs across the boundary of the Serving and Home Networks.

4. In the fourth model, we include *both* channel-session IDs in their respective messages, i.e. the combination of models 2 and 3.

We believe that all of these models are possible candidates for actual network implementation.

## C.2 Results of varying channel properties on the attack

We detail the formal analysis results of different channel properties and channel-session IDs on the attack in Figure 4. We consider the secrecy of the anchor key from the points of view of both the AUSF and the SEAF.

The first result (Model 1) is as stated in the main attack: it shows that when no channel-session IDs are included, there exist attacks such that both the AUSF and SEAF are fooled into thinking the anchor key (for an honest user) is secret, when in reality it is not.

Adding a channel-session ID *per protocol run* between the AUSF and ARPF (i.e. Model 2) correctly prevents the violation of the secrecy of the anchor key. This is a rough approximation of Fix 2 from Section 5, as the SUPI is sufficiently unique to mitigate the attack. It is worth

| Model vs. Property | Secrecy of Anchor Key |
|---|---|
| Model 1: No channel-session IDs | x |
| Model 2: IDs for AUSF ↔ ARPF | ✓ |
| Model 3: IDs for SEAF ↔ AUSF | x |
| Model 4: Channel-session IDs for **both** channels | ✓ |

Figure 4: How do the underlying channel properties affect the attack? (i.e. with or without channel-session IDs)

noting that the connection between AUSF and ARPF is completely internal to a specific "Home Network": connections between these two actors are internal and do not necessarily cross any network boundaries. As such we believe there is an argument to be made that requiring a brand new channel-session ID per protocol run might not be seen as an engineering / implementation necessity or priority above e.g. the same requirement across the Serving Network ↔ Home Network divide.

Adding a channel-session ID per protocol run between the AUSF and SEAF (i.e. Model 3) does **not** prevent the attack. This is the boundary between the Serving and Home Networks, so would seem to be a likely and realistic priority candidate for a requiring a new channel-session ID per protocol run.

Adding channel-session IDs to both network channels (i.e. Model 4) prevents the attack; this is a result of having correctly fixed the identity binding issue across the AUSF ↔ ARPF channel.

## C.3 Implications of alternative channel properties

Our analysis indicates that:

1. Confidentiality, Integrity, and Replay Protection on channels, and Authenticity of involved parties are necessary but not sufficient protections on secure network channels,

2. Channel-session ID binding to messages are not always sufficient, and that stronger, protocol-level solutions are required to guarantee a protocol's desired security properties, and

3. Protocols cannot and must not rely on implicit assumptions about the properties of underlying network implementations; these assumptions cannot be left as implementation details, especially when these details have tangible security consequences.

It might be acceptable for identity binding or channel-session IDs to be left as implementation details when they only affect the reliability of a protocol, but when these details have security impact, they must not be left to the implementers of a specific system. This is even more important than usual when concealed and ephemeral identities are involved in a protocol.

While some implementations might effectively use e.g. a new TLS or DIAMETER session per protocol run to ensure the correct pairing of messages to original sessions (for the sake of efficiency and not un-necessarily confusing packets), this is not specified in the standard. Our analysis shows it such a mechanism would not just prevent message confusion for reliability, but would in fact be security-critical.

In any case, to ensure that *all* implementations of the standard provide strong security guarantees, it is clear that the standard is currently at least underspecified, and needs to include one of our suggested fixes.

# D Table of acronyms

| Acronym | Definition |
|---|---|
| `5G-AC` | 5G Authentication Confirmation message |
| `5G-AIA` | Authentication Initiation Answer message |
| `5G-AIR` | Authentication Initiation Request message |
| 5G-AKA | Fifth Generation Authentication and Key Agreement Protocol |
| ARPF | Authentication credential Repository and Processing Function<br>(This resides within the Home Network, and is often within an HSM) |
| AUSF | Authentication Server Function (within Home Network) |
| EPS-AKA* | Evolved Packet System Authentication and Key Agreement Protocol<br>(The predecessor authentication protocol to 5G-AKA used by 4G/LTE) |
| HN | Home Network |
| HSM | Hardware Security Module |
| $\mathrm{K}_{\mathrm{AUSF}}$ | AUSF's anchor key |
| K | Long-term secret master key shared between mobile device and HN |
| $\mathrm{K}_{\mathrm{SEAF}}$ | SEAF's anchor key (derived directly from $\mathrm{K}_{\mathrm{AUSF}}$) |
| MAC | Message Authentication Code |
| RAND | Random number generated by the ARPF |
| SEAF | Security Anchor Function (within Serving Network) |
| SIDF | Subscriber Identity De-concealing Function |
| SNID | Serving Network Identifier |
| SQN | Sequence Number |
| SUCI | Subscription Concealed Identifier |
| SUPI | Subscription Permanent Identifier: previously "IMSI" |
| TLS | Transport Layer Security |
| TS | Technical Specification |
| TR | Technical Report |
| UE | User Equipment (e.g., mobile phone) |
| USIM | Universal Subscriber Identity Module (e.g., SIM Card) |