A Systematic Study of Practical & Formal Privacy in the 5G AKMA Procedure

Abstract

We systematically scrutinise all the facets of privacy in a recent 5G delegated authentication procedure called AKMA (Authentication and Key Management for Applications based on 3GPP credentials in the 5G Systems). We define, in general terms, what we argue to be the appropriate threat model and requirements of privacy for this protocol. Then, we formalise and analyse these using the various, state-of-art notions of privacy in Dolev-Yao protocol analysis. Using the Tamarin prover, we find that AKMA has several privacy breaches, for which we then propose fixes. Our patch, called AKMA^P, imposes minimal changes on AKMA, yet we formally show that it attains all our privacy requirements.

1 Introduction

Brought to the forefront of our awareness by the EU's General Data Protection Regulation (GDPR) [2] and then by the more recent COVID tracing apps [38], the importance of our privacy, as users of online systems cannot be overstated. Meanwhile, there are many web-based protocols (such as OAuth 2.0 [1], AAuth [37], Droplet [36]), which allow one application (say LibreOffice, Zoho Office, draw.io, Slack) to borrow a user's data and authorisation in another system to gain access to the user's resources (say Word documents stored on Dropbox, Google Drive or OneDrive). There is a history of abuses [18, 24, 30] by such inter-operable servers, which may make users more careful over delegated authorisations.

More worryingly still, newer platforms, drawing inspiration from Federated Identity Management (FIM) paradigms such as OAuth and Single Sign On [21], may expand users' authentication and authorisation even without their explicit knowledge. One such system is the new *Authentication and Key Management for Applications based on 3GPP credentials in the 5G Systems*, *AKMA* for short [7]. Indeed, in the 5G (5th Generation) mobile networks, this AKMA procedure was added, to allow for delegated authentication from the network to third-party providers called *application functions* (AFs). For instance, using AKMA, a driver inside a connected car *securely* employs a proprietary system to pay for road services and tolls automatically, without needing to authenticate in any other way to any of these third-party systems/AFs, but by sheer virtue of the car including a SIM-card (Subscriber Identification Module) registered onto a mobile network. For this, the third-parties/AFs will have provisioned AKMA with the mobile operator, and it is the AKMA protocol that the SIM will use when accessing different third-party services. Moreover, as we said, the owners of AKMA-capable devices, be it phones or cars, may not even know that their device is connecting to such a third-party service/AF using their SIM/mobile-network credentials. Importantly, the user's privacy can be severely impacted by these connections: e.g., their connection to a given third-party service or the time of it, or their location, their mobile-network provider, all may leak. Finally, the privacy aspects in AKMA also go the other way: a third-party, AKMA-functionality provider/AF may not wish to be linked to a given set of users or connected to other providers/AFs who also serve AKMA traffic to those users.

We recognise that AKMA is a procedure still undergoing standardisation by 3GPP, with the technical specifications (TS) [7] having changed recently, i.e., April 2023. Furthermore, privacy is not yet included as a requirement in these TS. That said, studies [10, 40] of the privacy in AKMA have very recently emerged. This work adds to that and is collaborating with 3GPP in terms of new privacy measures for AKMA.

Contributions. We scrutinise, in a systematic way, all the facets of privacy in the recent 5G delegated authentication procedure AKMA. We start by defining, in general terms, what we argue to be the appropriate threat model and privacy requirements for this application. We then formalise these using various state-of-art privacy notions in symbolic/Dolev-Yao analysis. Using the Tamarin prover, we find several privacy breaches in AKMA, which we patch, with utmost consideration for the current technical specifications [7]. We also formally verify our patch, called *AKMA^P*, and find that despite the minimal changes it imposes on AKMA, it does bring all privacy guarantees we stipulate.

2 Background

2.1 A Glance on (5G) Mobile Networks

We first give a simplified overview of the relevant 5G network entities for AKMA:

1. the *User Equipment (UE)* – a device (e.g., phone, car with a SIM onboard) subscribing to a mobile service;
2. the *Radio Access Network (RAN)* – the 5G radio "towers" providing network connectivity to the UEs;
3. the *5G core* [5, 6] – servers implementing the operator's logic, split into services: e.g., the *Authentication Server Function (AUSF)* and the *Access and Mobility Management Function (AMF)* authenticate the UE via the protocol called the *5G Registration / AKA (Authentication and Key Agreement)*; the *Unified Data Management (UDM)* service that manages access authorization; *Network Exposure Function (NEF)* is an API-based proxy for 5G to allow third-party applications' queries; e.g., some of these calls go to the core via the *Applications (AKMA) Anchor Function* (*AANF*);
4. *Application Functions (AFs)* – 3rd-party application-servers leveraging, e.g., the network's authentication.

2.2 AKMA & Its Privacy-Relevant Aspects

2.2.1 AKMA – An Overview

AKMA [7] is a delegated authentication service in 5G mobile networks. It aims to extend a subscriber's authentication onto the 5G network further into applications outside the network. In this way, an *application-function (AF) server* identifies a subscriber's UE indirectly, mediated by the core – which, in fact, does the UE's authentication. After this proxied authentication, the UE and the application function (AF) server will (re)establish a channel, secured with a key called K_{AF} *(application function key)*.

2.2.2 The AKMA Protocol

The AKMA protocol is succinctly represented here, in Figure 1 and Figure 2, and explained further now. The AKMA protocol can be seen as executing in two phases, which we denote as follows: the "initial phase" of AKMA shown in Figure 1, and the "K_{AF}-key generation" phase of AKMA given in Figure 2. The two phases are generally not executed in immediate sequence and, in fact, there is not a one-to-one mapping in these executions: i.e., for the same UE, there can be several runs of the "initial phase" of AKMA and just one run of the "K_{AF}-key generation" phase, and vice-versa.

Even if Figure 1 and Figure 2 show more 5G entities being involved in this protocol, for the purposes of this work, we can consider that the AKMA protocol is executed between a UE, an AF, and a part of the core[1], e.g., the *Application Anchor Function (AAnF)*.

We proceed in describing these phases.

The "Initial Phase" of AKMA As per points 3.a and 3.b on Figure 1, the end of 5G authentication (i.e., Registration/AKA [4]), an *AKMA-ready UE*[2] and the AAnF will both hold a new K_{AKMA} key for all AFs that the UE may connect to, and index this key under a so-called *AKMA Key Identifier (AKID)* (or *A-KID*, as both terms are used interchangeably in the specifications [7]).

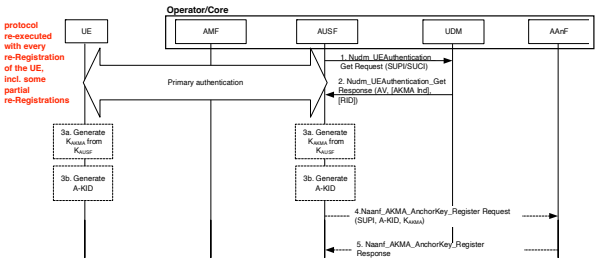


Figure 1: Deriving the K_{AKMA} key (after Fig. 6.1-1 [7]).

Note 1: In the current specifications [7], there is one AKID and one K_{AKMA} derived for all AFs registered with that AKMA-capable UE. Hence, for a given AKMA-capable UE set up to connect to an AF owned by a particular car manufacturer as well as one AF owned by a specific social media platform, the two *different* links to the two *different* AFs will use the *same* AKID and the *same* K_{AKMA} for this UE.

Note 2: As per Section 4.2.2. of the specifications [7], given an AKID, any AF can identify the network/AAnF which has generated that AKID, and –as such– can (later) contact that AAnF.

The Derivations for AKID, K_{AKMA} and K_{AF} The AKID is an identifier formed of a part containing routing information about the user-equipment (denoted below as "ue_routing_info") and a cryptographically derived identifier called "A-TID (AKMA Temporary UE Identifier)" in AKMA's 3GPP specifications [7]. Such an AKID is a pointer to an associated K_{AKMA} key in the core's database. These derivations are described in the Equations 1 below.

Based on the K_{AKMA} key, the UE and *each* AF associated with the newly derived AKID can derive a new K_{AF} key. In more detail, the key derivations relevant to the AKMA

[1] It is run between the AF and the AAnF – if the AF is internal to the operator, and between the AF, the AAnF (Application Anchor Function) and the NEF (network exposure function) – if the AF is external to the operator.

[2] This readiness is something one buys as part of their mobile-phone contract, or when they purchase a modern car with a SIM onboard, etc.

protocol are as follows:

$$A\text{-}TID = KDF(const, K_{AUSF}, \text{“}ATID\text{”}, \text{SUPI}); \quad (1)$$
$$AKID = ue\_routing\_info \,||\, A\text{-}TID \quad (2)$$
$$K_{AKMA} = KDF(const, K_{AUSF}, \text{“}AKMA\text{”}, \text{SUPI}) \quad (3)$$
$$K_{AF} = KDF(const, K_{AKMA}, AF\_ID), \quad (4)$$

whereby: KDF is a hash, *const* symbolises some hex constant values, *SUPI (Subscription Permanent Identifier (SUPI)* is a long-term identifier of the UE described in [3], and AF_ID is constructed as $AF\_ID = AF\_qualified\_name||Ua^*$ with Ua^* being the identifier[3] of the protocol eventually used at the application level between the UE and an application server associated with an AKMA Application Function (AF).

Note 3: Two different AFs will have different AF_ID (see the derivations above), so they will establish two different K_{AF}s with the same UE, despite the fact that these K_{AF}s will be “sharing” the same seed in the shape of the same K_{AKMA}.

The “K_{AF}-key Generation” of AKMA We now describe the second phase of AKMA, given on Figure 2:

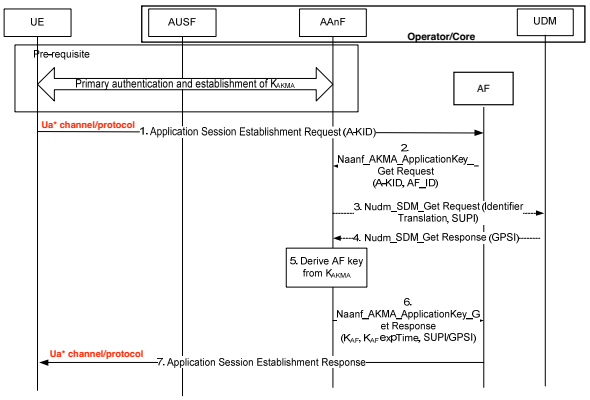


Figure 2: Deriving the K_{AF} key (based on Fig.6.2-1 [7]).

1. In step 1, the UE sends its current, global AKID to one AF with whom it intends to communicate; this is done over the Ua^* protocol that the UE and this AF are set up to use. We refer to this interchangeably as the Ua^* *protocol*, the Ua^* *channel*, or the Ua^* *connection*.

Note 4: According to Annex H of 3GPP 33.220 TS [8] where the Ua^* protocol is specified, there are multiple choices for this protocol and multiple options for each choice: e.g., HTTP, HTTP with digest authentication, TLS 1.2. with various cipher suites, etc.

2. In step 2, since the AKID identifies the network associated with it (see Note 2), then the AF at hand sends the received AKID to the right AAnF, alongside with its own identifier AF_ID.

   In line with the specifications [7], this is done over a secure or an insecure channel between the AF and the AAnF.

3. In steps 3 and 4, the AAnF now checks that the contacting AF can provide the service to the UE linked to the AKID, based on subscribers' information.

4. If the above steps are successful, then a new K_{AF} key is generated in step 5.

5. Then, in step 6, this new K_{AF} key, its time-to-live (*TTL*) and, optionally, a long-term identifier of the UE, such as the SUPI or equivalent, is sent by the AAnF to the AF.

   The UE does the same K_{AF} key-computation.

6. In step 7, the AF replies to the UE, to say if the AKMA-session establishment requested by the UE in step 1 was successful or not, including the potential reason of failure.

7. After step 7, provided the UE and the AF deem the process as successful at both ends, all communication between the UE and this AF is encrypted from then on with this K_{AF} inside the Ua^* protocol.

8. If, for whichever reason, the AF requires a new K_{AF} key (i.e., because its TTL has expired, or the Ua^* protocol demands it via its tickets, etc.), then the AF will contact the AAnF, and this phase is re-run from step 2 onwards.

Note 5: The freshness of the K_{AF} key depends on that of the K_{AKMA} key. That is, if the AF asks for a new K_{AF} key, but no new Registration has taken place since the last K_{AF}-generation (i.e., the “initial phase” of AKMA has not re-occurred), then the new K_{AF} key will be the same as the last K_{AF} key. In fact, the 3GPP specifications states: 1. *“K_{AKMA} and AKID can only be refreshed by a new successful primary authentication*[4]*”* (Section 6.1.1 [7]); 2. *“... when a new K_{AKMA} is derived, the K_{AF} will not be re-keyed automatically.”* (Section 6.4.2 [7]); 3. *“When the K_{AF} lifetime expires and the K_{AKMA} has not changed in AAnF, according to the Annex A.4, the AKMA Application Key which is established based on the current AKMA Anchor Key K_{AKMA} is not a new one”* (Section 5.2 [7]).

The expiration-time of the K_{AF}s can be made longer/shorter via policies in the Ua^* protocol, as per the 3GPP specifications: 1. *“The Ua^* protocol shall be able to handle the expiration of KAF”*. (Sec. 4.4.1 [7]); 2. *“If the Ua^* protocol supports refresh of K_{AF}, the AF may refresh the K_{AF} at any time using the Ua^* protocol.”* (Sec. 6.4.3 [7])

[3] This is specified in Annex H of 3GPP 33.220 TS [8].

[4] This means Registration/AKA.

2.3 Recent Privacy Analyses of AKMA

In recent work [10], Akman *et al.* have looked at the privacy of AKMA. The analysis in [10] is done from the perspective of hiding the AKID and other identifiers of the UE such as the SUPI from various parties. Their standpoints and modifications to AKMA by [10] are:

1. the Ua^* protocol run between the UE and the AF is considered always to be run over insecure channels, and all analyses are done under these assumptions;

2. step 1 of the Ua^*/AKMA protocol (see Figure 2) is modified: the AKID is sent to the AF, in a form only decryptable by the core. It is forwarded to the core in the step after step 1 of the Ua^* protocol (see Figure 2);

3. the AF no longer receives the SUPI or equivalent identifiers of the UE, but some ephemeral ones are sent/received instead, in the stage before step 7 in the Ua^* protocol (see Figure 2).

We now discuss [10]'s standpoints above. W.r.t. 1), please refer to our Note 4 in Section 2.2.2: it is not necessary that the Ua^* protocol run between the UE and the AF is insecure. In fact– in current times, this protocol will likely be secure: the UE would be on-boarded with the AF's provider (e.g., a car would come shipped with X.509 certificates for the car-manufacturer's AF, and a secure Ua^* protocol will likely be pre-selected to be used in the car-to-AF communications). Note that if the Ua^* protocol is secure, then all developments in [10] are nullified (their attacks are not possible anymore).

W.r.t. 2), the AF knowing the AKID and using it as an identifier of the UE are main features of protocol-design in AKMA: "The AKID functions as a temporary user identifier" and "AKMA AF shall be able to identify the AAnF serving the UE from the A-KID"; see, respectively, Section 6.2.2 and Section 4.4.2 (Requirements on AKMA Key Identifier) in the AKMA specification [7]. So, changing/removing this AKID-based identification of the AF equates to modifying central requirements of the standard documents for AKMA.

With this AKID concealment, [10] also changes the way the core indexed the K_{AF} and even the derivation of K_{AF}, with an additional key called K_{sh} being added to the mobile-key hierarchy. Their change to the indexing of the K_{AF} key violates another core principle of the AKMA design, that is: "AKID shall be usable as a key identifier in protocols used in the reference point Ua^*" (see [7], p.11). Then, their change to the derivation of the K_{AF} key is also hard to achieve in practice, as the 3GPP-specified key-hierarchy is almost immutable; further still, (this part of the) 3GPP specifications (see, e.g., [3]) intrinsically resides on the fact that only symmetric-key cryptography is used between the UE and the core, and, in their new proposal [10] involved key-encapsulation mechanisms whereby the UEs employ public keys of the core. [10] also change the logic of all protocol parties, the structure of the protocol messages, etc., producing in fact an entirely new design, arguably very far from the original AKMA procedure. Therefore, Akman *et al.* [10]'s work is going against core 3GPP principles.

So, whilst looking at privacy in a protocol that can identify the UEs is laudable, we find that the assumptions and approaches in [10] are not realistic. Thus, we look into plausible privacy notions for AKMA, in a systematic manner, also aligned with the 3GPP specifications. In Appendix A, we detail the view we take of this quest.

3 Generic Execution & Threat Models

Our generic model aims to be accessible, and follows a commonplace description of protocol executions, whereby one omits if the protocol measures are computationally bounded, or they follow Dolev-Yao abstraction.

Generic Execution Model & Environment $\mathcal{E}$. In our generic execution model, we assume the following settings. The AKMA protocol has several concurrent executions, successful as well as unsuccessful over various AF, UE, and *core* entities (e.g., $AF_1, AF_2, \ldots, UE_1, UE_2, \ldots, core_1, core_2, \ldots,$). The latter entities are denoted *parties* and have been *enrolled* in the system such as to have all the cryptographic material and knowledge to execute the AKMA protocol.

One part-taking by a party in an AKMA execution describes an *instance* of that party. In enrolment or provisioning[5], also *channels* between parties are created, emulating that the instances of different party[6] *types* (AF, UE, and *core*) can communicate with one other, with incoming and outgoing messages, as per the AKMA protocol; we speak of the channels on which the Ua^* protocol is run as the Ua^* *channels*.

Each party can have several concurrent instances running at any given point. The interleaving of at least three instances of an AF, a UE and a *core* party respectively, creating an execution of the AKMA protocol is denoted as an AKMA-protocol *session*. A session can be *partial* – if not all the instances involved in it have reached the final step in AKMA, or –otherwise, it is said to be *complete*.

We consider an *AKMA execution environment* $\mathcal{E}$ in which several parties of each type (AF, UE, and *core*) are involved, several instances of each party are present, and several partial and complete sessions are under way.

Our threat model and security assumptions, also complemented by details in Appendix A, are stated below.

[5] This is when a third-party such as a car manufacturer partners with an operator to provide AKMA to some of its subscribers.

[6] Any AKMA-capable device on the UE-side, be it phone, car is a party of type UE. Operators such as Orange, Vodafone, Telefonica, are parties of type core. Third-party AKMA providers, say BMW, YouTube, etc., are parties of type AF.

Our Threat Model $\mathcal{T}$ This is defined as follows:

(T1). an AKMA execution environment $\mathcal{E}$, as described above;

(T2). the presence of an active adversary $\mathcal{A}$ who can corrupt AF and UE parties from the enrolment phase or during the protocol execution;

(T3). a party compromised at enrolment phase will be totally controlled by the adversary, including the attacker controlling all its long-term cryptographic material, which need not be the case for a party compromised during the protocol execution;

(T4). trusted *core*s, meaning that the attacker cannot corrupt any party of type *core*;

(T5). corrupted parties will not follow the AKMA protocol;

(T6). honest parties (i.e., not corrupted) follow the AKMA protocol;

(T7). upon corruption, the attacker is subsumed by itself together with all the parties it has corrupted, and excluding any honest parties;

(T8). as per usual, if AKMA is studied against a privacy notion underpinning one party P_1, then this party P_1 cannot be corrupt;

(T9). as per usual, if AKMA is studied against a privacy notion underpinning collectively a certain UE_i and a certain AF_j, then not both parties UE_i and AF_j can be corrupt;

(T10). the channels between the core and the AFs remain secure, i.e., ensuring authenticity, confidentiality and integrity. I.e., our corruption of an AF is not at the level where it can defeat the authentication on the channels with the core[7];

(T11). the active adversary can listen on all channels, at the security-level of the channel (as per specified in AKMA) compound with his corruptions. I.e., if the UE-to-AF channel is secure and the end-points of the channel are not corrupt, the attacker cannot defeat the security of the channel; the UE-to-core communication can be compromised, if the UE is compromised; the AF-core channels remain secure.

[7] Arguably, this is realistic. If corrupted AFs can create fake certificates to connect to the core impersonating other, honest AFs and thus breaking the security of these channels, then this is a security flaw w.r.t. the (public-key) infrastructure in general, defeating entire sets of security procedures relying on it, not just AKMA. If this fails, then all security assumptions and procedures by 3GPP [3], are futile ab initio.

Security Settings $\mathcal{S}$ We consider the following security settings denoted, as a whole, by $\mathcal{S}$:

(S1). All the privacy properties we study are from the perspective of our attacker, i.e., from the perspective of corrupted parties and not honest parties.

(S2). Any privacy property that we studied with respect to a party UE_i and a party AF_j, then the Ua^*-channel between UE_i and a party AF_j is secure (i.e., ensuring authenticity, integrity, and confidentiality).

The above means the following: (a) we study privacy in the strongest possible setting for any given parties UE_i and AF_j, that is when there is a secure channel between them; (b) whilst ask that the Ua^*-channel between such parties of privacy-interest UE_i and AF_j be secure, we do allow that other Ua^*-channels, e.g., between UE_i and AF_k, or between UE_p and AF_m are insecure. This is exactly in line with the possibilities implied by the 3GPP specifications, and a stronger setting than in any of the previous works (see Section 2.3). Notably, (i) there may be privacy attacks which occur because the insecurity of one Ua^*-channel leads to a privacy-attack on another Ua^*-channel even if the latter was secure; and, (ii) there may be privacy attacks which occur even if all Ua^*-channel are secure.

4 Privacy Notions for AKMA

We mean to analyse AKMA from the viewpoint of all privacy notions possible, i.e., with regard to all identifiers used in the protocol and pertaining to this protocol, in our threat model. In the latter, the core is trusted, so there are two identifiable parties that remain for us to look at: the users and the application servers/functions.

4.1 Privacy of Users in AKMA

4.1.1 Which identifiers to consider and why

The privacy notions laid out in this section are referring to fields/properties that identify the UEs specifically during and/or via AKMA. Let us start by finding which parts of the protocol identify the UEs in AKMA per se. To this end, we cite the 3GPP specifications for AKMA [7], Section 6.2.2:

> "The AKID functions as a temporary user identifier." [7], p. 16

In Section 2.2.2, we saw that the AKID contains some "$ue\_routing\_info$" and the so-called ATID. The "$ue\_routing\_info$" contains static information, which can be inverted, i.e., linked backed to a UE by any party that can corrupt the core; however, in our threat model, the core is trusted (see (T4) in Section 3). Also, this $ue\_routing\_info$ is a 4-digit number, which is not necessarily UE-specific. Finally,

one given $ue\_routing\_info$ will be the same outside of the AKMA protocol, so this is not an AKMA-centric identifier; as such, looking at privacy via the lens of the $ue\_routing\_info$ is not of interest. The latter part of the AKID (i.e., the ATID), however, is cryptographically re-generated per UE, with every Registration and it is used to identify the UE by parts of the network (i.e., the AF) which we do consider corruptible (see (T2) in Section 3). Thus, all things considered, w.r.t. any privacy linked to AKID, we will in fact be interested in privacy analysis w.r.t. the A-TID, and henceforth, we use "AKID" to mean "A-TID" for purposes of privacy analysis.

Why not SUPI-based privacy? Receiving, in AKMA's step 7 in the Ua^* protocol (see Figure 2), a SUPI *illicitly* would require the compromise of the channel between the receiving AFs and the core, which is outside of our threat model as per its entry (T10). Also, as we discussed in Section 2.3, honest AFs would know SUPIs anyway, and –in any case– as per our security assumption (S1), we are not pursuing privacy-analyses from the viewpoint of honest parties. So, threats of leaking SUPIs in the AKMA protocol are not of interest herein, and possibly not of realistic interest, in general.

4.1.2 What privacy notions

All AKMA-centric identifiers are ephemeral: i.e., as Section 2.2.2 explained, each of these are re-generated with every Registration (see Figure 1) or with every re-run of AKMA (see Figure 1). Or, we cite the 3GPP specifications for AKMA [7], Section 6.1:

> "AKID can only be refreshed by a new ... authentication", [7], p. 14

So, we are interested in strong forms of privacy, which will refer to tracking UEs based on ephemeral identifiers, rather than long-term ones (e.g., their mobile phone):

Definition 1 ***Strong Secrecy of the UE in AKMA.*** *In the threat model $\mathcal{T}$, under the security assumptions $\mathcal{S}$,* strong secrecy of the UE *(*SS_UE*) holds if:*

- *for any attacker $\mathcal{A}$ in the settings $\mathcal{T}$,$\mathcal{S}$, spanning any AKMA execution environment $\mathcal{E}$ such that $\mathcal{A}$ does not know[8] (now) the active AKID of an honest UE, the attacker $\mathcal{A}$ cannot follow (future) presences of this AKID within secure Ua^* connections within $\mathcal{E}$.*

Definition 1 has a flavour of confidentiality, hence it is being called "strong secrecy of the UE". Concretely, in the spirit of our earlier setting (S2) in Section 3, Definition 1's focus on secure Ua^* connections (as opposed to any Ua^* connections) can be seen as asking: if AKIDs remain secret, do secure channels imply that one cannot track, observe, link-together the executions of an AKID (maybe based on a some protocol data linking to specific AKIDs)?

Now, we move to a weaker notion of privacy, which we call "post-compromise privacy"; see Definition 2.

Definition 2 ***Post-Compromise Privacy of the UE in AKMA.*** *In the threat model $\mathcal{T}$, under the security assumptions $\mathcal{S}$,* post-compromise privacy of the UE *(*PP_UE*) holds if:*

- *for any attacker $\mathcal{A}$ in the settings of $\mathcal{T}$ and $\mathcal{S}$, spanning any AKMA execution environment $\mathcal{E}$ such that $\mathcal{A}$ knows that at an AKID of an honest UE was present in the execution environment $\mathcal{E}$, the attacker $\mathcal{A}$ cannot follow future presences of this AKID within secure Ua^* connections within $\mathcal{E}$.*

In the spirit of our earlier setting[9] (S2) in Section 3, *PP_UE* asks this: is the existence of insecure Ua^* channels fatal to privacy? Once an AKID leaks, is it all lost w.r.t. tracking that AKID, or do the secure Ua^* channels and the AKMA protocol per se add to the (repair or the future of) privacy-preservation w.r.t. AKID?

The AKMA protocol provides PP_UE if no attacker would be able to tell that a leaked AKID will be present on secure Ua^* channels/protocols, after the leak. This leakage could occur, e.g., as follows: (a) due to the fact that the Ua^* protocol can be insecure between one honest UE and one honest AF_1; (b) due to the attacker having corrupted an AF and thus getting honest AKIDs anyway, etc.; moreover, this AKID will be the same between this UE and all AFs, including those with secure Ua^*s, so this further legitimises the question.

4.2 Privacy of AKMAApplication-Functions

We now move to privacy properties pertaining to the application functions (AFs). The only identifier to pursue w.r.t. privacy now is AF_ID. We also propose just one privacy notion, in Definition 3 below.

Definition 3 ***Weak Privacy of the AF in AKMA.*** *In the threat model $\mathcal{T}$, under the security assumptions $\mathcal{S}$,* weak privacy of the AF *(*WP_AF*) holds if:*

- *any attacker $\mathcal{A}$ in the settings of $\mathcal{T}$,$\mathcal{S}$, spanning any AKMA execution environment $\mathcal{E}$, cannot detect the presence of an AF_ID pertaining to an honest AF, within the execution environment $\mathcal{E}$.*

Definition 3 says that weak privacy of the AF (*WP_AF*) holds if no attacker, in our model, is able to learn, by executing AKMA, if one AF or another is present onto a network.

[8] Here, knowing an AKID is equivalent to knowing that this AKID was/is present in the execution environment $\mathcal{E}$.

[9] That is, the privacy of AKID should be studied on secure Ua^* channels, bearing in mind insecure Ua^* channels exist and not just on insecure Ua^* channels a la [10].

Table 1: Our Privacy Notions for AKMA

Property	About	Meaning	Setting
strong secrecy of the UE (*SS_UE* Def. 1)	AKID	Can $\mathcal{A}$ track AKIDs, if AKIDs' confidentiality holds?	secure Ua^* channels
post-compromise privacy of the UE (*PP_UE*, Def. 2)	AKID	Can $\mathcal{A}$ track an AKID even when only sent encrypted, once this AKID's confidentiality was breached?	secure & insecure Ua^* channels
weak privacy of the AF (*WP_AF*, Def. 3)	AF_ID	Can $\mathcal{A}$ track AF_IDs even if all channels are secure?	secure & insecure Ua^* channels, secure core-AF channels
unlinkability of UEs and AFs (*UE-AF_L*, Def. 4)	AKID and AF_ID	A mix of PP_UE and SP_AF: i.e., can the AKID and the AF_ID be put together by $\mathcal{A}$?	secure & insecure Ua^* channels, secure core-AF channels

WP_AF is dubbed as "weak' since it asks whether, not honest, but corrupt parties are able to track honest AFs in the AKMA protocol; i.e., any attacker in our threat model (see entry (T7)) is formed by corrupt parties, that is, parties which deviate from the protocol description.

4.3 Unlinkability of UEs and AFs in AKMA

Now, we move to describing the notions of whether an attacker can find a link between a given UE and a given AF, or vice-versa, in the case where the two are honest and are communicating securely. Asking around such linking makes sense primary due to asynchronous nature of executing the AKMA protocol: i.e., (1) first, the UE executes with the core independently of the AF; (2) after this, the UE executes with the AF; (3) as a result of the latter, the AF executes with the core, but — at this stage — the core and the UE are no longer in contact via AKMA; the channels on which these happen are also different, with different security provisions. Also, corruptions of the UEs vs. the AFs lead to different observability levels, since the AKID is global to all AFs, and one AF may serve one AKID but not another.

Definition 4 ***Unlinkability of UEs and AFs in AKMA.*** *In the threat model $\mathcal{T}$, under the security assumptions $\mathcal{S}$,* unlinkability of UEs and AFs *(*UE-AF_L*) holds if*

- *any attacker $\mathcal{A}$ in the settings of $\mathcal{T}$,$\mathcal{S}$, spanning any AKMA execution environment $\mathcal{E}$, cannot detect if a honest UE is communicating with a honest AF, within secure Ua^* connections within $\mathcal{E}$.*

Specifically:

(a) In the threat model $\mathcal{T}$, under the security assumptions $\mathcal{S}$, unlinkability of UEs to AFs *($\exists UE \rightarrow \forall AF$) holds if: any attacker $\mathcal{A}$ in the settings of $\mathcal{T}$,$\mathcal{S}$, spanning any AKMA execution environment $\mathcal{E}$, cannot detect if a given honest UE is communicating with any given honest AF, within secure Ua^* connections within $\mathcal{E}$.*

(b) In the threat model $\mathcal{T}$, under the security assumptions $\mathcal{S}$, unlinkability of AFs to UEs *($\exists AF \rightarrow \forall UE$) holds if: any attacker $\mathcal{A}$ in the settings of $\mathcal{T}$,$\mathcal{S}$, spanning any AKMA execution environment $\mathcal{E}$, cannot detect if a given honest AF is communicating with any given honest UE, within secure Ua^* connections within $\mathcal{E}$.*

In AKMA, our unlinkability notion in Definition 4 does not necessarily follow from the previous two notions on privacy. This is because of the following three aspects.

Firstly, an attack w.r.t. Definition 3 does not imply an attack w.r.t. Definition 4. That is, an attacker may be able to break post-compromise privacy of the UE on AKIDs, by observing just something on the first message on the Ua^* channel between the UE and the AF, but meanwhile being unable to say/see anything of the AF_ID sent on the channel between the AF and the core.

Second, an attack w.r.t. Definition 2 does not imply an attack w.r.t. Definition 4. That is, one can break weak privacy of the AF by corrupting a UE in the enrolment phase and getting an AF_ID from the K_{AF} derivation function, but not knowing which other AKIDs this AF/AF_ID serves.

To summarise, all notions we introduced thus far are recounted in Table 1. Note that investigating all our notions recalled in Table 1 is valid pursuit at this stage, as we do not know whether they imply one another in AKMA, or not[10].

5 Symbolic Analysis of Privacy of AKMA

5.1 Dolev-Yao Verification of Privacy

Dolev-Yao (DY)/symbolic analysis [13, 16, 32] is algorithmically tailored to verifying *path/trace properties*, which can be expressed on one or on all the protocol's execution paths/*traces* (i.e., there is one possible protocol-execution path where A disagrees with B on some value; or all paths exhibit such agreement). Meanwhile, most privacy properties are not expressible as trace properties (see, e.g., [28]). Fortunately, some privacy and anonymity notions can be modelled as *trace equivalences*, i.e., checking equivalence between

[10]Perhaps, there is enough information in AKMA executions that once, e.g., an AKID is trackable, it also follows which is the AF_IDs that this AKID connects to, or vice-versa, or not all, or it may depend on the security of the Ua^* channels, or on the corruptions' settings.

two traces, and this is mechanised, e.g., in DEEPSEC [22], SAT-Equiv [23], AKISS [20]. Yet, other properties, such as flavours of unlinkability [12], would require checking equivalences not between two traces but rather between (any) two sets of traces, or checking of stronger equivalences, not between traces but between processes, known as *observational equivalences* [9, 28]. Looking to verify the former but with trace-based protocol provers, two mainstream[11] approximations for their observational-equivalence verification exist.

(1) **Diff-equivalences**. Observational equivalences are over-approximated via coarser processes equivalences known as ***diff-equivalences*** [35]. A diff-equivalence is a strong reachability condition, which often does not hold even if observational equivalence does. To give an example, when proving diff-equivalence between $P|Q$ and $P'|Q'$, diff-equivalence requires that P is equivalent (observationally by an attacker but sometimes even w.r.t. internal reductions) to P' and Q is equivalent (in the same sense) to Q'.

Despite being too-fine a relation for most privacy notions and thus prone to leading to false attacks for privacy [17], diff-equivalence can be safely trusted in the case of positive results: if diff-equivalence holds, then observational equivalence (and therefore some associated privacy notion) holds. To this end, it is advantageous that diff-equivalence can be proved automatically in ProVerif, Tamarin, and Maude-NPA.

(2) ***Indistinguishability***$^{\textit{Arapinis,Delaune}}$. Beyond diff-equivalence being too strong sometimes, there are also some privacy properties which are encoded via more complex notions of observational equivalences (over configurations of executing processes rather than processes themselves). This is the case of strong unlinkability, introduced by Arapinis et al. in [12], which speaks about linking protocol executions/sessions together. One flavour of Arapinis' et al. (session-)unlinkability is that a protocol-execution in which $alice$ has just one presence/sessions should be indistinguishable to an attacker from the protocol-execution in which $alice$ has two or more presences/sessions.

The authors in [14, 26] soundly reduced[12] the verification of the strong session-indistinguishability by Arapinis et al. to the verification of two reachability properties called ***well authentication (WA)*** and ***no desynchronisation (ND)***, and one property (which is not a trace-property) called ***frame opacity (FO)***. I.e., if the strong session-indistinguishability by Arapinis fails, then at least one of these three properties, WA or FO or ND, by [14, 26] fails. Simply put, ND denotes that an honest interaction between A and B cannot/should not fail. And, WA encodes that whenever a conditional is positively evaluated, the agents involved are having so far an honest interaction. These clearly stem from the fact that to induce a privacy fault, often an attacker distinguishes evaluation on branches (WA), or observes the state of process via some leak in the systems book-keeping (ND). In tandem, frame-opacity (FO) is a property generally required to check privacy in protocols when ND and WA hold. Intuitively, FO ensures that an attacker cannot learn identification aspects from the protocol-messages themselves (e.g., headers, counters, etc.); so, FO is a strong, idealistic requirement: that is, that all messages visible to the attacker in the protocol are random-looking.

Due to the aforesaid reduction by Delaune et al. of the verification of the privacy in Arapinis' observability notion, we henceforth refer to the latter notion as "***indistinguishability***$^{\textit{Arapinis,Delaune}}$", or "***indist.***$^{\textit{Ar.,Del.}}$" for short. Note that, since $indist.^{Ar.,Del.}$ in fact reduces strong observational equivalence to properties verifiable by DY, trace-oriented tools, it has actually been put to use in verifying protocols, in both in `ProVerif` (in [26], in a weaker form without ND), and in Tamarin [14].

5.2 Symbolic Models for AKMA's Privacy

We took our generic threat model and properties and under-approximated them into Dolev-Yao models and a set of symbolic expressions of privacy as per Section 5.1. Our models are inspired by those in [40], upgraded to `Tamarin` v 1.6.1, and made for privacy, unlike the original that focused only on correctness and agreement. So, we now discuss our various models of AKMA, resulted from varying all security assumptions underpinning our privacy notions in Section 4.

- All our Tamarin files are at `https://bit.ly/3QjnWPM`.

DY Models Varying the Security of the Ua^* Channels. Our privacy properties –in line with our threat model in Section 3– vary the security assumptions: e.g., strong secrecy of the UE in Definition 1 requires secure Ua^* channels, whereas post-compromise privacy of the UE in Definition 2 allows for the possibility for the Ua^* channels to be both secure and insecure (i.e., one way for AKIDs to leak is insecure Ua^* channels). Thus, we start by yielding two classes of models:

- $\mathcal{M}^{\text{Sec\_Insec-}Ua^*}$ – models where Ua^* channels can be both secure or insecure.

  The above models are suited to modelling *PP_UE* (Def. 2), *WP_AF* (Def. 3), unlinkability of UEs to AFs as well as unlinkability of AFs to UEs (Def. 4).

- $\mathcal{M}^{\text{Sec-}Ua^*}$ – models where Ua^* channels can only be secure.

  The above models are suited to modelling *SS_UE* (Def. 1).

Main Characteristics of Models $\mathcal{M}^{\text{Sec\_Insec-}Ua^*}$ and $\mathcal{M}^{\text{Sec-}Ua^*}$. In these models, we have followed the following settings:

- There are no restrictions on the number of AFs, *core*s or UEs.

[11] This is alongside, newer ones: [34].

[12] This is for two-party, stateful (authentication) protocols.

- Each UE is associated with one and only one *core*, which is standard for mobile-networks' subscribers.
- Each UE is assigned 2 AFs, at random, from all the AFs "on-boarded" in the setup of the model (i.e., point 1 above). These AFs remain the same throughout the protocol's multiple execution: i.e., we do not model full subscription or re-subscription by UEs to AFs, as this is not part of the AKMA specifications.
- In the $\mathcal{M}^{\text{Sec-}Ua^*}$ models, the Ua^* channels between the UEs and its AFs are always secure.
- In the $\mathcal{M}^{\text{Sec\_Insec-}Ua^*}$ models, the Ua^* channels between the UEs and its AFs are chosen to be secure or insecure, non-deterministically, in the setup of the UEs and AFs in the model. We ensure that in each model there is at least an UE with an insecure Ua^* channel and one with a secure channel.
- The channel between AF and the *core* is always secure.
- UEs can re-Register with the *core* (i.e., as such, change of their AKID, and renew their K_{AKMA}) unboundedly many times, and all parties can re-run the AKMA protocol, in line with specifications [7] and Fig. 1 and 2.

DY Models Varying the Privacy Encoding. As we saw in Section 5.1, there are two main ways of encoding and verifying privacy in DY tools: via diff-equivalences and/or via $\mathit{indist.}^{Ar.,Del.}$. One could arguably encode a given privacy property (e.g., unlinkability of UEs to AFs) in the same model, via both diff-equivalences or via $\mathit{indist.}^{Ar.,Del.}$. To be clearer, we create another two classes of models:

- $\mathcal{M}^{\text{diff}}$ – models where we use diff-equivalence to prove privacy properties in Section 4;

  The models in the above are used to prove, e.g., unlinkability of UEs to AFs and unlinkability of AFs to UEs in Def. 4.
- $\mathcal{M}^{\text{indist.}^{\text{Ar.,Del.}}}$ – models where we use $\text{indist.}^{\text{Ar.,Del.}}$ to prove privacy properties in Section 4;

  The models in the above are used to prove, e.g., *PP_UE* (Def. 2).

Unlike with the case of $\mathcal{M}^{\text{Sec\_Insec-}Ua^*}$ and $\mathcal{M}^{\text{Sec-}Ua^*}$ models, the choice on which models from $\mathcal{M}^{\text{diff}}$ vs. $\mathcal{M}^{\text{indist.}^{\text{Ar.,Del.}}}$ to use for verifying privacy properties is less stringent, e.g., $\exists UE \rightarrow \forall AF$ can be modelled both as $\mathit{indist.}^{Ar.,Del.}$ as well as *diff.-equiv* on AKIDs.

Main Characteristics of Models $\mathcal{M}^{\text{diff}}$ and $\mathcal{M}^{\text{indist.}^{\text{Ar.,Del.}}}$. In these models, we have followed the following settings:

- Both these models can be of the $\mathcal{M}^{\text{Sec\_Insec-}Ua^*}$ type or of the $\mathcal{M}^{\text{Sec-}Ua^*}$ type.
- The $\mathcal{M}^{\text{diff}}$ models are simplified models of AKMA in that they contain at most three UEs and two AFs; these simplifications come from the fact that we aim to prove specific *diff*-equivalences driven by our notions unlinkability of UEs to AFs and unlinkability of AFs to UEs, and this is a safe abstraction to undertake: if the attacker can distinguish protocol aspects in this simplified setting, then the attacker can also distinguish these in richer settings. To this end, the models focus on proving *diff*-equivalence primarily based on AKIDs (driven by the unlinkability of UEs to AFs notion) or proving *diff*-equivalence primarily based on AFIDs (driven by the unlinkability of AFs to UEs notion); we call the former models $\mathcal{M}^{diff}_{AKID}$ and the latter models $\mathcal{M}^{diff}_{AFID}$.

To see at a glance, the relationship between our classes of models mentioned thus far, please refer to Figure 4, in Appendix B. As we said above, the grey area (denoting *diff*-equivalence properties as well as $\mathit{indist.}^{Ar.,Del.}$ in the same model) makes sense from a modelling perspective, but we do not use this. It is also clear that the models of most interest are those which can best express our properties and our threat models, i.e., those that fit the property-summarising Table 1. To this end, the models of most interest are those in the sub-families marked via the checkered areas in Figure 4 (i.e., most of our properties require this setting, as per Table 1), followed by sub-families within the dotted areas (i.e., these are required by *SS_UE* in Def. 1, as per Table 1). Finally, we note that for some of these sub-families, in fact, we produce variations on one model, such as to show slight nuances or angles of a privacy notion (e.g., *PP_UE* in Def. 2) failing.

Note: From here on, we express the intersection of the classes of models (i.e., the checkered and the dotted areas in Figure 4) in a natural way: e.g., $\mathcal{M}^{Sec\_Insec\text{-}Ua^*,diff}_{AKID}$ is a model in which we prove *diff*-equivalence based on AKIDs, and the Ua^* channels can be both secure and insecure.

5.3 Verifying of AKMA's Privacy

We now explain how we modelled the various privacy properties from Section 4, in an order that eases understanding.

5.3.1 Analysing Post-compromise Privacy of the UE

In line with the secure-channels' setting required by *PP_UE*, we encoded *PP_UE* in a model in the subclass $\mathcal{M}^{\text{indist.}^{\text{Ar.,Del.}},\text{Sec\_InSec-}Ua^*}$ and in a model in the $\mathcal{M}^{\text{indist.}^{\text{Ar.,Del.}},\text{Sec-}Ua^*}$. Given the denotation of these models, the property *PP_UE* was analysed by checking all or some of the WA, ND, and FO notions by Delaune et al.

In fact, the ND lemma we encoded for this checks if all honest UE and honest AF are synchronised: i.e., if an arbitrary UEs with a given AKID at some point and an arbitrary AF that this UE has contacted based on this AKID stay in a state

of synchronous/aligned view of the executions of AKMA over repeated executions. To be able to show this, we modelled a crude version of sessions' management across executions of the AKMA protocol (i.e., an UE's view to have contacted a given AF once and still have this one live-session with them; an AF's view that a given UE has contacted them twice, etc.); this was done via an emulation of counters[13] in Tamarin (via multisets), both inside the UE and inside the AF. At the high-level, the UE and the AF will respect the ND lemma if these counters are in sync at the two ends. In Appendix E, in Figure 8, we show this no-desynchronisation lemma.

This ND lemma fails, which means, without WA and FO needing to be verified in this model, that *indist.*$^{Ar.,Del.}$ fails for AKMA. The exact attack trace, in a $\mathcal{M}^{\text{indist.}^{\text{Ar.,Del.}},\text{Sec\_InSec-}Ua^*}$ model of ours as per the above, in Tamarin, is as follows: (1) An honest UE1 established an AKMA session with an honest AF1, successfully; (2) UE1 then contacts AF2 with the same AKID, for a session; (3) AF2 is corrupted and leaks the AKID; (3) The DY attacker uses the AKID through a corrupt UE2 to contact AF1 again, thus de-syncing the counters in the honest AF1 from those in the honest UE1. An immediate consequence of this no-desynchronisation attack is that the attacker learns of the state of UE1 on the network. That is, in the Ua^*-connection opened via the corrupt UE2, the attacker also receives information from AF1 on UE1. The information leaked is whether the request for a new the Ua^*-connection has gone through, and if not – why not: e.g., because another UE1 connection (to AF1) is live. This refutes our *PP_UE*.

In Figure 3, we explain the attack-trace above via an image, and –primarily via lower, left-hand-side rectangle– we also summarise again why this entails that our *PP_UE* fails.

As Figure 3 also depicts, note that this *PP_UE* fails both on $\mathcal{M}^{\text{Sec\_InSec-}Ua^*}$ models and on $\mathcal{M}^{\text{Sec-}Ua^*}$ models (i.e., irrespective if the Ua^* channel is secure or not), as long as the attacker – as per our threat model – can corrupt UEs and AFs.

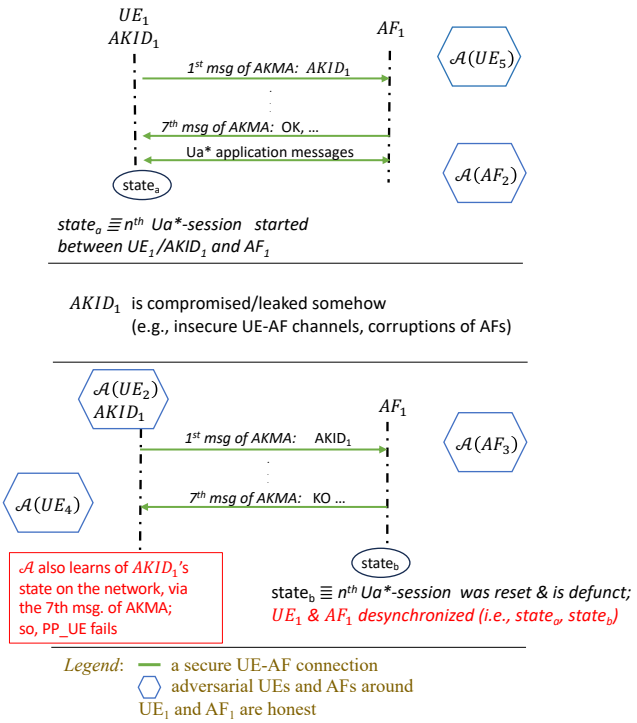


Figure 3: *PP_UE* Failing as a Consequence of Failure of indist.$^{\text{Ar.,Del.}}$ (Failure of ND between UEs and AFs)

5.3.2 Analysing Weak Privacy of the AF

In line with the secure-channels' setting required by *WP_AF*, we encoded *WP_AF* in a model in the subclass $\mathcal{M}^{\text{indist.}^{\text{Ar.,Del.}},\text{Sec\_InSec-}Ua^*}$ and in a model in the $\mathcal{M}^{\text{indist.}^{\text{Ar.,Del.}},\text{Sec-}Ua^*}$. And, to check property *WP_AF*, we actually looked at analysing the WA, ND, and FO notions.

The WA lemma we encoded for these checks says: if every time the *core* evaluates positively the start of an AKMA session for a UE, this UE is honest and has requested the session per se. The modelling to make this possible does not require the counters as per the above. In Appendix E, in Figure 9, we show this UE-to-*core* well-authentication lemma.

This WA lemma fails for AKMA, which means, without ND and FO needing to be verified in this model, that *indist.*$^{Ar.,Del.}$ fails. The exact attack trace, on an $\mathcal{M}^{\text{indist.}^{\text{Ar.,Del.}},\text{Sec\_InSec-}Ua^*}$ model of ours as per the above, in Tamarin, is as follows: (1) An honest UE1 tries to establish an AKMA session with an honest AF1, but the attacker blocks this getting the UE1's AKID; (2) The attacker uses the AKID through a corrupt UE2 to contact AF1 again, thus making the core allow an AKMA-session for UE1 but via the corrupt UE2. This also means that the attacker gets information about AF1 illicitly (e.g., which UEs they serve, when); this directly refutes our weak privacy of the AF.

In Figure 5, we explain the attack-trace above via an image, and –primarily via lower, left-hand-side rectangle– we also summarise again why this entails that our *WP_AF* fails.

As Figure 5, in Appendix C also depicts, note that *WP_AF* fails both on $\mathcal{M}^{\text{Sec\_InSec-}Ua^*}$ models and on $\mathcal{M}^{\text{Sec-}Ua^*}$ models (i.e., irrespective if the Ua^* channel is secure or not), if the attacker – as per our threat model – can corrupt UEs and AFs.

We note that property *WP_AF* can also be shown to fail as via a ND lemma failing between the *core* and UE1 above; we have done this too, modelling session-booking via counters as in the case of the model dis-proving *PP_UE*, only that this time the counters used and quantified over are, of course, inside the *core*[14] and UEs. Using Figure 5, we can see that $state_c$ of honest UE1 and $state_d$ of the (honest) *core* are de-synchronised. This corresponding ND lemma, which fails

[13] To make things more tractable, in the models submitted for review, the counters in the AF go up to a maximum of 3 requests for each UE.

[14] To make things more tractable, in the models submitted for review, the counters in the *core* go up to a maximum of 3 requests for each UE.

showing the same attack as per the WA lemma failing (on an $\mathcal{M}^{\text{indist.}^{\text{Ar.,Del.}},\text{Sec\_InSec-}Ua^*}$ model of ours as per the above, in Tamarin), is given in Appendix E, in Figure 10.

5.3.3 Analysing Strong Secrecy of the UE

In line with the secure-channels' setting required by *SS_UE*, we encoded *SS_UE* in a model in the sub-class $\mathcal{M}^{\text{diff,Sec-}Ua^*}$. And, to check property *SS_UE*, we actually looked at analysing a diff-equivalence quantifying over two AKIDs (the same or different, belonging the same UE or not).

In fact, in the case of strong secrecy of the UE, as per Def. 1, we need that the attacker necessarily does not learn the AKIDs. Since we operate in a DY setting, we therefore need the Ua^* to be secure, hence building a model in $\mathcal{M}^{\text{Sec-}Ua^*}$. In this case, for simplicity and clarity of the diff proof, we restrict[15] the model to 2 UE and 2 AFs. W.r.t. the AKIDs to distinguish, we take the following approach. We use three types of AKIDs: (a) for UE1, we single out AKID1 is for AF1 (and AF2), which becomes AKID2 after the re-Registration; (b) for UE2, we single out AKID3 for AF1 (and AF2). Then, we prove `diff`-equivalence of AKID1 to all the other AKIDs, in fact w.r.t. also to other AKID-indexed information, i.e., long-terms keys, K_{AF}s, K_{AKMA}s, etc.: e.g.,

```
diff(<A_KID1, K_AF1, K_AKMA1, ~K_sh1, ~K_sh3>,
 <A_KID2,K_AF2, K_AKMA2,~K_sh2, ~K_sh4>).
```

This `diff` is clearly a faithful representation of Def. 1: that is, " for the active AKID of an honest UE" (i.e., AKID1) , "the attacker $\mathcal{A}$ cannot follow (future) presences of this AKID within secure Ua^*-connections" (i.e., AKID2), compared to other presences in the AKMA executions (i.e., AKID3).

The strong secrecy of the UE property holds in AKMA.

5.3.4 Analysing Unlinkability Users and AFs

As per Def. 4, there are two types of unlinkability: one from the direction of linking "known" UEs to AFs (i.e., unlinkability of UEs to AFs), and one from the direction of linking "known" AFs to UEs (i.e., unlinkability of AFs to UEs). We discern between the analysis of the two in the below.

Analysing unlinkability of UEs to AFs. Looking at the way post-compromise privacy of the UE fails (see Section 5.3.1 and/or Figure 3), it should become clear that, in the case of the AKMA protocol, the failure of post-compromise privacy of the UE also leads to a failure of unlinkability of UEs to AFs a.k.a. $\exists UE \rightarrow \forall AF$ (Def. 4). That is, the attacker in the attack trace in Figure 3 not only tracks an AKID, but via the details in msg. 1 and msg. 7 of the AKMA protocol, it links this AKID (i.e., AKID1) to an AF (i.e., AF1). In Tamarin, we can easily quantify over an extra variable (i.e., the AF) in the lemma (i.e., ND) that shows post-compromise privacy of the UE failing, and thus show $\exists UE \rightarrow \forall AF$ failing.

However, for completeness, we do the above not in $\mathcal{M}^{\text{indist.}^{\text{Ar.,Del.}}}$ models, but in $\mathcal{M}^{\text{diff}}$ models.

In fact, the *diff*-equivalence to consider here is similar to that of strong secrecy of the UE, with the significant difference in settings whereby the attacker knows one AKID.

For the former setting, we have to do nothing other than consider an $\mathcal{M}^{\text{Sec\_InSec-}Ua^*}$ model (unlike in the case of strong secrecy of the UE where we worked in $\mathcal{M}^{\text{Sec-}Ua^*}$). Similarly to Section 5.3.3, w.r.t. the AKIDs to distinguish, we take the following approach. We use 3 different types of AKIDs: (a) for UE1, we single out AKID1 is for AF1 (and AF2), which becomes AKID2 after the re-Registration, and make sure that UE1/AKID1 uses an insecure Ua^*-channel with AF2 such that AKID1 leaks to the attacker (as needed by unlinkability of UEs to AFs), but a secure Ua^*-channel with AF1; (b) for UE2, we single out AKID3 for AF1 (and AF2). Then, we check diff-equivalence of AKID1 to all the other AKIDs, with the `diff` as per strong secrecy of the UE. This `diff` is clearly attempting to find a faithful representation of the refutation of Def. 4: that is, "can" the attacker "detect if a honest UE" (here, UE1) "is communicating with some honest AF, within secure Ua^*-connections within $\mathcal{E}$" (here, AF1).

Analysing unlinkability of AFs to UEs. The privacy attacks shown thus far (e.g., weak privacy of the AF) do not imply an attack on unlinkability of AFs to UEs; this is because the attacks before stem from knowing/learning an AKID, whereas unlinkability of AFs to UEs is about starting from knowing an AFID and linking it to (potentially as-of-then unknown) AKIDs.

We do this in a $\mathcal{M}^{\text{diff,Sec\_InSec-}Ua^*}$ model, where we leak one AFID, say AF1 to the attacker, and constrain, in the model setup, that it connects with some UE1 over secure channels; this UE1 however is allowed, in our model setup, to connect to other AFs (e.g., AF2) over insecure channels. Then, we check a `diff` as per the above. This `diff` is clearly attempting to find a faithful representation of the refutation of Def. 4: that is, "can" the attacker "detect if a honest AF" (here, AF1) "is communicating with some honest UE" (here, UE1), "within secure Ua^*-connections".

Interestingly, because the AKIDs are global, it turns out that the `diff` above holds if there is only one UE (e.g., UE1 as named above) present, but it fails if there are at least two UEs present (e.g., UE1 as named above, and a different UE2).

[15] Note that if the attacker distinguishes AKIDs in this case, it will also distinguish them in the ampler models, without restrictions on the numbers of parties.

5.3.5 Other Tamarin Verification of AKMA

All our models are also checked w.r.t. standard security requirements, such as Lowe's hierarchy [31], and the results are as expected: e.g., weak agreement holds between parties two-by-two, except for between the UE and the AF when the two communicate on an insecure channel; most secrecy and key-agreement properties hold (e.g., on AKID, and K_{AF}, respectively), except for when, again, the UE and the AF communicate on an insecure channel. Such findings are recounted inside our models as comments.

5.3.6 Summary of Models and Analyses

The summary of the most relevant privacy analyses and finding discussed above is recounted in Table 2.

6 AKMA^p: Practical & Private AKMA

We now give a practical solution to the privacy attacks we exhibited in the previous section.

Vehicles to Our Privacy Attacks. Our privacy attacks are possible due to three main reasons:

(a) in the attack against *WP_AF* (Figure 5) has its onset via an adversarial injection of the global AKID of an UE to an AF1, when this AKID was aimed for an AF2; this global nature of the AKID is also at fault for unlinkability of UEs to AFs failing;

(b) the final parts of the attack against *WP_AF* (Figure 5) as well as against *PP_UE* (Figure 3) are stemming from the fact that the success or the reason of failure in the 7th message of AKMA is sent, on the Ua^* channel, to any UE (adversarial or honest) without any checks w.r.t. who this success/failure details are intended for; i.e., the attacker manipulating an UE2 learns of the AKIDs or the AFs via information meant for UE1 but sent to this UE2, on the Ua^* channel;

(c) the main vehicle by which attacker can learn AKIDs and thus have one way to start mounting most attacks, e.g., *PP_UE* (Figure 3), unlinkability of UEs to AFs, etc., is in the case where the Ua^* channels/protocols are insecure.

AKMA^p: Patching Our Privacy Attacks. A solution against all four problems above is compound:

(i) to counteract problem (a) above, we ask that for each UE, during each its Registrations, an AKID unique per every AFID is (re)generated;

For this, equation (1) in AKMA (see Section 2.2.2) is modified to the equation below, as follows, where the addition is marked in bold-face:

$$A\text{-}TID = KDF(const, K_{AUSF}, \text{“}ATID\text{”}, \textbf{AF\_ID}, \text{SUPI})$$

All the other equations (i.e., derivations of identifiers and keys in AKMA) stay the same.

(ii) to counteract problem (b) above, the success or failures in step 6 and 7 of AKMA should be sent encrypted with a key pertaining to the UE that these messages are for.

To this end, the *core* has to be used to issue UE-specific messages for step 7 of the AKMA protocol. This is because it is only the *core* (and not the AF), who –at the stage before step 7 of the AKMA protocol– has UE-specific keys. Moreover, in our trust model (namely, (T2) and (T4)), it is only the core who is trusted and therefore can construct such a message in ways unattainable to the attacker. Concretely, if the K_{AF} generation succeeds or fails, this is encrypted by the *core* with the K_{AMF} key[16] it shares[17] with the UE for which this *core* tried to calculate the K_{AF} key; this encryption is forwarded by the AF to the UE who opened the Ua^*-channel with it.

The reader can also see Figure 7 for a diagrammatic version of this modification.

(iii) to counteract problem (c) above, all Ua^* channels/protocols should be secure;

• We refer to AKMA patched as *AKMA^p* and its diagrammatic description is given in Appendix D.

About AKMA^p. We recount that our privacy-attacks' counteractions. • Secure Ua^* channels prevent AKID leaks, stopping the onset of attacks. • Encrypted handling of successes/errors w.r.t. K_{AF} derivations stop malicious UEs from learning other UEs' data. • One AKID per AFID stops the tracking of AFIDs and/or the linking of AFIDs to AKIDs.

As the Figure in Appendix D shows, AKMA^p is extremely close to AKMA, with minimal changes: e.g., a KDF change for unique AKIDs, and management of error-handling. Judicious error-handling is common practice when it comes to privacy preservation. One AKID per AF is also meaningful: if Twitter/X and Facebook/Meta both offered AKMA to UEs, it would not be desirable that these UEs would share the same AKID to connect over-the-air to both services.

We do not make the series of recommendations leading to AKMA^p just based on the attacks we find. We do formally show that the modifications (i)-(iii) together, bring full privacy guarantees w.r.t. to our notions. We discuss this next.

[16] This is the lowest-level key in the mobile-network key hierarchy which is derived out of K_{AUSF}, and re-generated at each UE re-Registration.

[17] We note that an encryption (by the *core* or the AF) of success/failure with K_{AF} itself does not work, as it may be the case (e.g., in the case of failure) that if the right K_{AF} has not been computed, and then there will no way for the UE to decrypt the reason of failure.

Table 2: Our Systematic Privacy Verification of AKMA in Tamarin

Property	**About**	**Model Class**	**Verification Method**	Status	Filename for Model(s)	Time
strong secrecy of the UE (Def. 1)	AKID	$\mathcal{M}^{\text{diff,Sec-Ua*}}$	diff-equiv on two unknown AKIDs	holds	SS_AKMA	cca. 1min
post-compromise privacy of the UE (Def. 2)	AKID	$\mathcal{M}^{\text{indist.}^{\text{Ar.,Del.}},\text{Sec(\_Insec)-Ua*}}$	$\text{indist.}^{\text{Ar.,Del.}}$	ND_UE_AF fails	PP_AKMA	manual (with oracle cca. 10 mins)
weak privacy of the AF (Def. 3)	AF_ID	$\mathcal{M}^{\text{indist.}^{\text{Ar.,Del.}},\text{Sec(\_Insec)-Ua*}}$	$\text{indist.}^{\text{Ar.,Del.}}$	ND_UE_CORE fails WA fails	WP_AKMA	manual (with oracle cca. 30 mins)
unlinkability of UEs to AFs (Def. 4)	AKID AF_ID	$\mathcal{M}^{\text{diff,Sec\_Insec-Ua*}}$	no known AFID, one known AKID1 diff-equiv over AKID1-AFID when '-' secure	fails	Unlink1_AKMA	cca. 3 mins
unlinkability of AFs to UEs (Def. 4)	AKID AF_ID	$\mathcal{M}^{\text{diff,Sec\_Insec-Ua*}}$	no known AKID, one known AFID1 diff-equiv over AKID-AFID1 when '-' secure	holds/fails	Unlink2(oneUE)_AKMA Unlink2(twoUEs)_AKMA	cca. 3 mins

Verifying AKMA^p's Privacy in Tamarin

Since the AKMA^p has only secure Ua^* channels, then in our Tamarin modelling we will only have $\mathcal{M}^{\text{Sec-}Ua^*}$ models and no $\mathcal{M}^{\text{Sec\_Insec-}Ua^*}$. That said, that are other clear modelling changes from the AKMA-based models, as now there is one AKID per AFID. For instance– in the $\mathcal{M}^{\text{diff}}$ models– where we restricted for AKMA to 3 AKIDs (for tractability reasons), now we will need to consider at least 4 AKIDs to be able to formulate the right `diff`s (e.g., for checking unlinkability of UEs to AFs) . We applied such changes to our Tamarin models for AKMA, and easily produced Tamarin models for AKMA^p.

Note that, especially in AKMA^p's settings of secure-communications only, it is the case that some privacy holding implies trivially other privacy-notions holding. Concretely, if strong secrecy of the UE holds ("diff-equiv" on all UE-related data holds for any two unknown AKIDs) and unlinkability of AFs to UEs holds ("diff-equiv" on all UE-AF-related data holds even when one knows one AFID) and unlinkability of UEs to AFs holds ("diff-equiv" on all UE-AF-related data holds even when one knows one AKID), then also post-compromise privacy of the UE must hold ("diff-equiv" on all UE-related data holds even when one knows one AKID) and weak privacy of the AF("diff-equiv" on all UE-related data holds even when one knows one AFID) must hold. This is the case in AKMA^p.

This is summarised in Table 3, in Appendix F.

In fact, in AKMA^p, we verified all the properties we had verified for AKMA (e.g., all the $\text{indist.}^{\text{Ar.,Del.}}$ via WA, ND, ...), and they all hold form AKMA^p. We do not report further but one can see this in our files.

Responsible Disclosure. We are in discussions with the 3GPP security working group about these attacks, and the recommendations of securing the Ua^* channels have seen practical hurdles, yet the communications are on-going.

7 Related Work

In Section 2.3, we looked, in detail, at the privacy analysis of AKMA in [10] (done in ProVerif). In Section 5.1, we covered the state-of-the-art on privacy verification in symbolic/Dolev-Yao tools. Aside of this, in this section, we are discussing other related aspects.

In terms of formal-verification of privacy, this is non-trivial in terms of tool-building: see e.g., [15, 17, 27, 29]. But, with this work, we are rather belonging to another sub-community of privacy analysis, interested in: (a) definitions of privacy, and (b) in their formal treatments with relevance to realistic practical settings. In the context of line (a), Pfitzmann and Hansen offer a consolidated report of privacy-related terminology in [33]. Their report combines a number of concepts of anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. Tsukada et al. investigate and organise some of these concepts in [39], particularly relating the concept of unlinkability to that of minimal anonymity. Goriac [25] expands on the work by Pfitzmann and Hansen by adding definitions for *involvement* and *unobservability*, and investigates privacy in terms of *behavioural equivalence*. Concerning line (b), a recent work is [19], which looks at the notion of "trackability" from various perspectives, all rooted in some practical modification. Their notion of existential trackability is closest to what is commonly known as unlinkability, and –in fact– our own notions of unlinkability are inspired by their work.

W.r.t. the symbolic verification of AKMA which is not privacy-related (but for Dolev-Yao agreement), this has been looked at very recently, in: ProVerif – by [11], and Tamarin – by [40].

8 Conclusions

We systematically and generically defined numerous facets of privacy in the recent 5G procedure of delegated authentication called AKMA. We formalised and verified these using the various, state-of-art notions of privacy in Dolev-Yao analysis, using the Tamarin prover. We found privacy breaches in AKMA. Having proposed a series of patches which are minimally changing the 3GPP specifications for AKMA, we then proved that our new version of AKMA attains our privacy guarantees. We are in on-going talks with 3GPP about this.

References

[1] The OAuth 2.0 Authorization Framework. RFC 6749, IETF, October 2012. https://datatracker.ietf.org/doc/html/rfc6749.

[2] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union*, L119:1–88, 2016.

[3] 3GPP. System architecture for the 5G System (5GS). Technical Specification (TS) 23.501, 3rd Generation Partnership Project (3GPP), 10 2020. Version 16.0.0.

[4] 3GPP. Procedures for the 5G System. Technical Specification (TS) 23.502, 3rd Generation Partnership Project (3GPP), 10 2021. Version 16.7.0.

[5] 3GPP. 5G Security Assurance Specification (SCAS); Access and Mobility management Function (AMF). Technical Specification (TS) 33.512, 3GPP, 07 2022. Version 16.3.0.

[6] 3GPP. 5G Security Assurance Specification (SCAS) for the Session Management Function (SMF) network product class. Technical Specification (TS) 33.515, 3GPP, 07 2022. Version 16.2.0.

[7] 3GPP. Authentication and Key Management for Applications (AKMA) based on 3GPP credentials in the 5G System. Technical Specification (TS) 33.535, 3GPP, 09 2022. Version 17.7.

[8] 3GPP. Digital cellular telecommunications system (Phase 2+), ... Technical Specification (TS) 33.220, 3GPP, 01 2023. Version 17.4.0.

[9] Ki Yung Ahn, Ross Horne, and Alwen Tiu. A characterisation of open bisimilarity using an intuitionistic modal logic. *Logical Methods in Computer Science*, 17, 2021.

[10] Gizem Akman, Philip Ginzboorg, Mohamed Taoufiq Damir, and Valtteri Niemi. Privacy-enhanced AKMA for multi-access edge computing mobility. *Computers*, 12(1), January 2023.

[11] Gizem Akman, Philip Ginzboorg, and Valtteri Niemi. AKMA for secure multi-access edge computing mobility in 5g. In Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Ana Maria A. C. Rocha, and Chiara Garau, editors, *Computational Science and Its Applications - ICCSA 2022 Workshops - Malaga, Spain, July 4-7, 2022, Proceedings, Part IV*, volume 13380 of *Lecture Notes in Computer Science*, pages 432–449. Springer, 2022.

[12] Myrto Arapinis, Tom Chothia, Eike Ritter, and Mark Ryan. Analysing unlinkability and anonymity using the applied pi calculus. In *2010 23rd IEEE Computer Security Foundations Symposium*, pages 107–121, Edinburgh, UK, 2010. IEEE, IEEE Computer Society.

[13] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, P. C. Heám, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, and L. Vigneron. The AVISPA tool for the automated validation of internet security protocols and applications. In Kousha Etessami and Sriram K. Rajamani, editors, *Computer Aided Verification*, pages 281–285, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[14] David Baelde, Stéphanie Delaune, and Solene Moreau. A method for proving unlinkability of stateful protocols. In *2020 IEEE 33rd Computer Security Foundations Symposium (CSF)*, pages 169–183, Los Alamitos, CA, USA, June 2020. IEEE Computer Society.

[15] David Baelde, Stéphanie Delaune, and Solène Moreau. A method for proving unlinkability of stateful protocols. In *2020 IEEE 33rd Computer Security Foundations Symp. (CSF)*, pages 169–183, 2020.

[16] B. Blanchet. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In *IEEE CSFW*, 2001.

[17] Bruno Blanchet and Ben Smyth. Automated reasoning for equivalences in the applied pi calculus with barriers. In *2016 IEEE 29th Computer Security Foundations Symp. (CSF)*, pages 310–324, 2016.

[18] Rich Brown. SkyDrive content restrictions among the toughest in the cloud. *CNET*, August 2012. https://www.cnet.com/tech/services-and-software/skydrive-content-restrictions-among-the-toughest-in-the-cloud/.

[19] K. Budykho, I. Boureanu, S. Wesemeyer, F. Rajaona, D. Romero, Lewis, Y. Rahulan, and S. Schneider. Fine-Grained Trackability in Protocol Executions. In *Network and Distributed System Security Symposium (NDSS) 2023*, 2023.

[20] Rohit Chadha, Ştefan Ciobâcă, and Steve Kremer. Automated verification of equivalence properties of cryptographic protocols. In *Programming Languages and Systems*, pages 108–127, 2012.

[21] David W. Chadwick. *Federated Identity Management*, pages 96–120. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[22] Vincent Cheval, Steve Kremer, and Itsaka Rakotonirina. The DEEPSEC prover. In Hana Chockler and Georg Weissenbacher, editors, *International Conference on Computer Aided Verification*, pages 28–36. Springer International Publishing, 2018.

[23] Véronique Cortier, Antoine Dallon, and Stéphanie Delaune. SAT-Equiv: An Efficient Tool for Equivalence Properties. In *2017 IEEE 30th Computer Security Foundations Symp. (CSF)*, pages 481–494, 2017.

[24] Emily Dreyfuss. Was It Ethical for Dropbox to Share Customer Data with Scientists? *Wired*, July 2018. https://www.wired.com/story/dropbox-sharing-data-study-ethics/.

[25] Iulian Goriac. An epistemic logic based framework for reasoning about information hiding. In *2011 Sixth International Conference on Availability, Reliability and Security*, pages 286–293, Vienna, Austria, 2011. IEEE.

[26] Lucca Hirschi, David Baelde, and Stéphanie Delaune. A method for unbounded verification of privacy-type properties. *Journal of Computer Security*, 27(3):277–342, 2019.

[27] Lucca Hirschi, David Baelde, and Stéphanie Delaune. A method for unbounded verification of privacy-type properties. *J. of Computer Security*, 27(3):277–342, 2019.

[28] Ross Horne. A bisimilarity congruence for the applied pi-calculus sufficiently coarse to verify privacy properties, 2018.

[29] Ross Horne, Sjouke Mauw, and Semen Yurkov. Unlinkability of an improved key agreement protocol for EMV 2nd gen payments. In *2022 IEEE 35th Computer Security Foundations Symposium (CSF)*, pages 364–379. IEEE, 2022.

[30] Joel Khalili. Google Drive is locking some people's files for no reason. *TechRadar*, January 2022. https://www.techradar.com/news/google-drive-is-locking-some-peoples-files-for-no-reason.

[31] G. Lowe. A Hierarchy of Authentication Specifications. In S. Foley and J. Millen, editors, *In Proceedings of the 10th IEEE workshop on Computer Security Foundations (CSFW'97)*, pages 31–43, Massachusetts, US, 1997. IEEE Computer Society.

[32] S. Meier, B. Schmidt, C. Cremers, and D. Basin. The TAMARIN Prover for the Symbolic Analysis of Security Protocols. In *CAV*, pages 696–701, 2013.

[33] Andreas Pfitzmann and Marit Hansen. Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management - A consolidated proposal for terminology. *Version v0*, 31:15, 2008.

[34] L. Vigano S. Gondron, S. Moedersheim. Privacy as reachability. http://www.imm.dtu.dk/~samo/abg.pdf, 2021.

[35] Sonia Santiago, Santiago Escobar, Catherine Meadows, and José Meseguer. A formal definition of protocol indistinguishability and its verification using Maude-NPA. In *Int. Workshop on Security and Trust Management*, pages 162–177. Springer, 2014.

[36] Hossein Shafagh, Lukas Burkhalter, Sylvia Ratnasamy, and Anwar Hithnawi. Droplet: Decentralized Authorization and Access Control for Encrypted Data Streams. In *Proceedings of the 29th USENIX Security Symposium (USENIX Security '20)*, pages 2469–2486. USENIX, August 2020. https://www.usenix.org/conference/usenixsecurity20/presentation/shafagh.

[37] Anuchart Tassanaviboon and Guang Gong. OAuth and ABE based authorization in semi-trusted cloud computing: aauth. In *Proceedings of the Second International Workshop on Data Intensive Computing in the Clouds - DataCloud-SC '11*, page 41, Seattle, Washington, USA, 2011. ACM Press. http://dl.acm.org/citation.cfm?doid=2087522.2087531.

[38] Carmela Troncoso, Dan Bogdanov, Edouard Bugnion, Sylvain Chatel, Cas Cremers, Seda F. Gürses, Jean-Pierre Hubaux, Dennis Jackson, James R. Larus, Wouter Lueks, Rui Oliveira, Mathias Payer, Bart Preneel, Apostolos Pyrgelis, Marcel Salathé, Theresa Stadler, and Michael Veale. Deploying decentralized, privacy-preserving proximity tracing. *Commun. ACM*, 65(9):48–57, 2022.

[39] Yasuyuki Tsukada, Ken Mano, Hideki Sakurada, and Yoshinobu Kawabe. Anonymity, privacy, onymity, and identity: A modal logic approach. In *2009 International Conference on Computational Science and Engineering*, volume 3, pages 42–51, Bellaterra, Catalonia, Spain, January 2009.

[40] Tengshun Yang, Shuling Wang, Bohua Zhan, Naijun Zhan, Jinghui Li, Shuangqing Xiang, Zhan Xiang, and Bifei Mao. Formal analysis of 5g authentication and key management for applications (AKMA). *Journal of Systems Architecture*, 126:102478, 2022.

A Specification-sensitive & Realistic Goals for the Privacy Analysis of AKMA

Following from the discussions in Section 2.3, we now state our series of *goals (G1, . . . , G4)* with respect to analysing the AKMA protocol from a holistic privacy viewpoint.

(G1. We aim to pursue a *systematic* and *non-judgemental* privacy analysis of AKMA.

By "systematic analysis" here we mean that we are not targeting the study of privacy from the viewpoint of one single identifier (e.g., the AKID) in AKMA, chosen in some bespoke manner, but instead our goal is to:
– ascertain all identifiers which are at risk of privacy attacks, and look at privacy-analysis for each and all in turn;
– find and describe the interplay between such identifiers, if more than one identifier exists;
– treat, where possible, the privacy threats w.r.t. each and all these identifiers through the same lens or, via the same technique(s).

By "non-judgemental" here we mean that, in our systematic approach above, we also divorce ourselves of promoting one privacy concern over another: i.e., some may be of more interest, or more serious, and we do discuss such aspects, but –ultimately– we are not ruling over the matter[18].

(G2. We assume the *strongest possible security settings* for AKMA, since any (cryptographic) privacy that holds in such a setting will hold in any weaker-security setting. We undertake (G1) above under this assumption.

Notably, we mean that if there is any alternative given by the specifications, then –in our study– we will consider the one implying more security. Concretely, here, we mean if the channel underpinning the Ua^* protocol can be both secure and insecure, as the specifications [7] say, then –in our analyses– we allow indeed these both cases to coexist.

Further, we are interested in seeing if privacy holds or can fail with respect to parties using the Ua^* protocol strictly over a secure channel, which was not studied before and which –of course– is of realistic interest.

It is rather clear that privacy would be broken for parties using insecure Ua^* protocols, which [10] studied. In turn, we are interested to see if secure Ua^* protocols bring privacy guarantees. Also, we are further seeking to see if secure Ua^* channels can cause the recovery of privacy, after an initial privacy-breaches due to other insecure Ua^* channels.

(G3. We aim to propose *pragmatic/realistic solutions*, to any privacy issues we may find. This is to say that any possible privacy-enhancing improvements of AKMA will be as close as possible to the current, general 3GPP specifications of mobile networks (e.g., no use of public key cryptography within the network as proposed in [10]), and as close as possible to the current, specific 3GPP specification of AKMA, with the view of adoption.

(G4. We aim to use *formal verification* throughout our investigation.

[18] One reason to do this is the fact that we imparted this work to the 3GPP security group, and the wide-scope discussions are on-going.

B A Representation of Our Models for AKMA

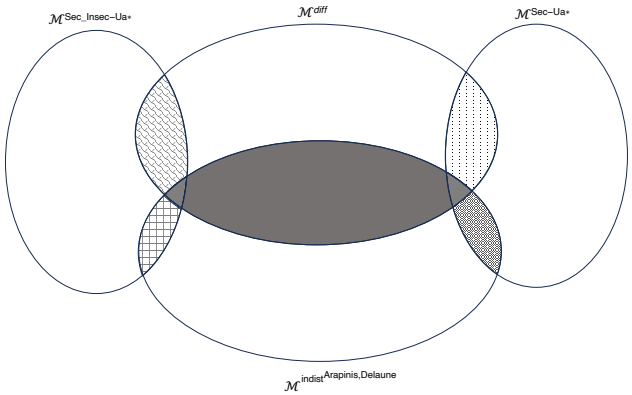


Figure 4: Our Classes of Privacy Models for AKMA (of most interest are checkered areas: $\mathcal{M}^{\text{indist.}^{\text{Ar.,Del.}},\text{Sec\_Insec-Ua*}}$, $\mathcal{M}^{\text{diff},\text{Sec\_Insec-Ua*}}$; of interest for one property are dotted areas: $\mathcal{M}^{\text{indist.}^{\text{Ar.,Del.}},\text{Sec-Ua*}}$, $\mathcal{M}^{\text{diff},\text{Sec-Ua*}}$; dark area – not used)

C Well-Authentication/No-desynchronisation Failing in AKMA

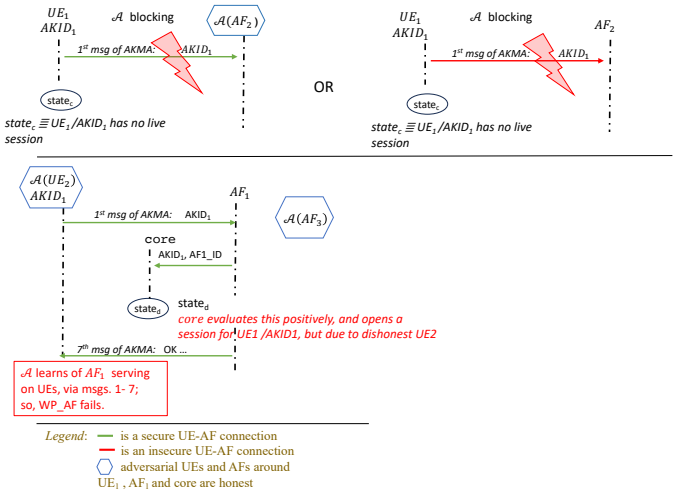
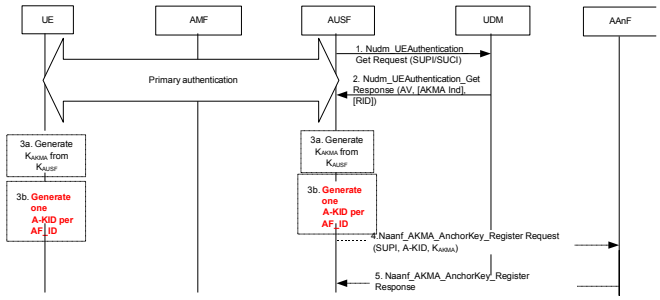


Figure 5: *WP_AF* Failing as a Consequence of Failure of indist.$^{\text{Ar.,Del.}}$ (Failure of WA between UEs and *core*)

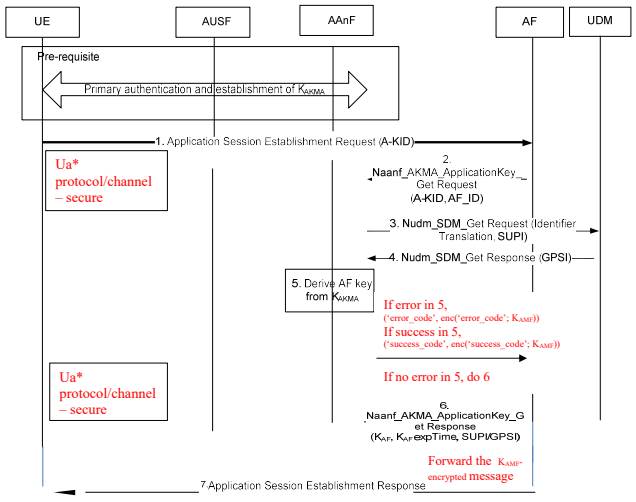
D The Diagrammatic Description of AKMA^P

In this section, we give the diagrammatic description of our AKMA^P protocol, which –as one can easily see– is very intentionally close to AKMA.



Deriving K_{AKMA} after primary authentication and A-KID in AKMA^P

Figure 6: Modifying Fig. 6.1-1 in [7]



K_{AF} generation from in K_{AKMA} in AKMA^P

Figure 7: Modifying Fig. 6.2-1 in [7]

E Tamarin Code

```
lemma No_Desynchronisation_UE_AF [use_induction]:
"All AKID idAF idHN KAF SUPI count1 #t01 #t02 #t03 #t04
        .
        (
                AF_WA_ND(AKID, idAF, 'ok', '1'+'1') @ #t04 //the AF was contacted
                   twice by the AKID
                &
                HN_Response(idHN, idAF, AKID, KAF,'ok') @ #t03
                &
                AF_send_KeyRequest(idAF,idHN, AKID) @ #t02
                &
                UE_WA_ND(SUPI, AKID, idAF, 'secure', count1) @ #t01  //the channel
                   between UE and AF is secure
                & #t01 < #t02
                & #t02 < #t03
                & #t03 < #t04
                & // no key reveal
                not (
                Ex X m #r.
                Reveal(X, m) @ #r
                &
                Honest(X) @ #t04 // the AF has all the parties that should be honest
                    and not leak.
                )
        )
        ==>
        (
                count1='1'+'1' // the UE must have sent its AKID at least twice to
                   the AF as well - which is clearly impossible - so this should
                   fail
        )
"
```

Figure 8: Tamarin lemma — No Desynchronisation between the UEs and the AFs in AKMA

```
lemma Well_Authentication:
"All AKID idAF #t04
        .
        (
                //AF_WA_ND(AKID, idAF, 'ok', '1') @ #t04
                AF_WA_ND(AKID, idAF, 'ok') @ #t04
        )
        ==>
        (
                Ex idHN KAF SUPI UaType #t01 #t02 #t03
                .
                        HN_Response(idHN, idAF, AKID, KAF,'ok') @ #t03
                        &
                        AF_send_KeyRequest(idAF,idHN, AKID) @ #t02
                        &
                        UE_WA_ND(SUPI, AKID, idAF, UaType, '1') @ #t01
                        & #t01 < #t02
                        & #t02 < #t03
                        & #t03 < #t04
        )
        | //key reveal
        (
                Ex X m #r.
                Reveal(X, m) @ #r
                &
                Honest(X) @ #t04 // the AF has all the parties that should be honest
                    and not leak.
        )
"
```

Figure 9: Tamarin lemma — Well-Authentication between the *core* and the UEs in AKMA

```
lemma No_Desynchronisation_UE_Core:
"All AKID idAF idHN SUPI count1 #t01 #t02 #t03
        .
        (
                Core_WA_ND(AKID, idAF, 'ok', '0'+'1') @ #t03 //the core was
                    contacted for this AKID by the AF twice
                &
                AF_send_KeyRequest(idAF, idHN, AKID) @ #t02
                &
                UE_WA_ND(SUPI, AKID, idAF, 'insecure', count1) @ #t01 //the channel
                    between UE and AF is insecure and the UE only sent count1
                    requests
                & #t01 < #t02
                & #t02 < #t03
                & // no key reveal
                not (
                Ex X m #r.
                Reveal(X, m) @ #r
                &
                Honest(X) @ #t03 // the core has all the parties that should be
                    honest here
                )
        )
        ==>
        (
                count1='0'+'1' //This is clearly impossible as the UE only ever
                    sends 1 request
        )
"
```

Figure 10: Tamarin lemma — No Desynchronisation between the UEs and the *core* in AKMA

F Summative Results on the Verification of AKMA^p

Table 3: Our Systematic Privacy Verification of AKMA^p in Tamarin

Property	**About**	**Model Class**	**Verification Method**	Status	Ref. Model(s)	Time
strong secrecy of the UE (Def. 1)	AKID	$\mathcal{M}^{\text{diff,Sec-Ua*}}$	diff-equiv on two unknown AKIDs	holds	SS_AKMAP	cca. 1 min
unlinkability of UEs to AFs (Def. 4)	AKID and AF_ID	$\mathcal{M}^{\text{diff,Sec-Ua*}}$	no known AFID, one known AKID1, diff-equiv over AKID1-AFID	holds	Unlink1_AKMAP	cca. 1 min
unlinkability of AFs to UEs (Def. 4)	AKID and AF_ID	$\mathcal{M}^{\text{diff,Sec-Ua*}}$	no known AKID, one known AFID1, diff-equiv over AKID-AFID1	holds	Unlink2_AKMAP	1 min
post-compromise privacy of the UE (Def. 2)	AKID	$\mathcal{M}^{\text{diff,Sec-Ua*}}$	diff-equiv "between a known AKID and an unknown one"	holds	from the first 3 rows	1 min
weak privacy of the AF (Def. 3)	AF_ID	$\mathcal{M}^{\text{diff,Sec-Ua*}}$	diff-equiv "between a known AFID and an unknown one"	holds	from the first 3 rows	1 min