Trashing IMSI Catchers in Mobile Networks

Mohammed Shafiul Alam Khan*
ISG, Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
Mohammed.Khan.2013@live.rhul.ac.uk

Chris J Mitchell
ISG, Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
me@chrismitchell.net

ABSTRACT

We address the decades-old privacy problem of disclosure of the permanent subscriber identity (IMSI), a problem that arises in all generations of mobile networks and that makes IMSI catchers a real threat. A number of possible modifications to existing protocols have been proposed to address the problem; however, most require significant changes to existing deployed infrastructures. We propose a novel authentication approach for 3G and 4G systems that does not affect intermediate entities, notably the serving network and mobile equipment. It prevents disclosure of the subscriber's IMSI by using a dynamic pseudo-IMSI that is only identifiable by the home network for the USIM. A major challenge in using dynamic pseudo-IMSIs is possible loss of identity synchronisation between USIM and home network, an issue that has not been adequately addressed in previous work. We present an approach for identity recovery to be used in the event of pseudo-IMSI desynchronisation. The scheme requires changes to the home network and the USIM, but not to the serving network, mobile phone or other internal network protocols, enabling simple, transparent and evolutionary migration. We provide a detailed analysis of the scheme, and verify its correctness and security properties using ProVerif.

ACM Reference format:
Mohammed Shafiul Alam Khan and Chris J Mitchell. 2017. Trashing IMSI Catchers in Mobile Networks. In *Proceedings of WiSec '17 , Boston, MA, USA, July 18-20, 2017,* 12 pages.
DOI: 10.1145/3098243.3098248

1 INTRODUCTION

The possibility of user tracking by intercepting air interface traffic was considered in the design phase of 2G GSM, and was addressed by use of a changing pseudonym, the *temporary mobile subscriber identity* (TMSI), instead of the permanent *international mobile subscriber identity* (IMSI). Newly allocated TMSIs are transmitted in encrypted form to prevent linking of TMSIs or of a TMSI to the IMSI. The same mechanism was adopted in 3G [50] and 4G [25] systems. Although the temporary identity is used instead of the IMSI in most cases, in some situations the IMSI is sent in cleartext by the mobile. One such case is when a mobile is switched on and wishes to connect to a new network, and so will not have an assigned TMSI. Another is where the network is unable to identify the IMSI from the presented TMSI [11]; an active adversary can exploit this feature and masquerade as a legitimate network to request the IMSI. Such an adversary is known as an *IMSI catcher* [44], and such attacks have been observed in several countries [28, 43, 48]. In GSM networks, man-in-the-middle attacks are also possible, [37], although we do not consider such attacks further here. The discussion below applies to both 3G and 4G systems, although we use 4G terminology throughout.

A wide range of solutions to the IMSI catcher problem have been proposed [16, 17, 21, 26, 33, 40, 41]; unfortunately, they all require significant changes to deployed network infrastructures, making them impractical. However, van den Broek et al. [51] and Khan and Mitchell [31] recently described possible system enhancements which reduce the privacy impact of IMSI catchers without requiring any changes to serving networks or mobile phones. We refer below to the respective schemes as BVR and KM. In both cases, changeable (temporary) IMSIs are used to help protect privacy; however, as described below, this also makes possible a potentially disastrous loss of synchronisation between USIM and home network, an issue addressed by the scheme described in this paper.

Both BVR and KM involve updating the IMSI held by the home network as soon as use of the new IMSI by the mobile is observed, so a failed serving network could cause the home network to update its database prematurely by falsely claiming to have seen use of a new IMSI. This could cause loss of identity synchronisation between USIM and home network. In both schemes it is implicitly assumed that serving networks always behave honestly and that communications between networks is secure. However, several recent papers [18, 22, 24, 27, 36, 38, 39, 49] have reported security flaws in the core network protocols and their implementation, which makes this assumption questionable; thus loss of identity synchronisation is possible for both schemes. This observation motivates the work described here. In summary, the contributions of the paper are as follows.

- We propose a new authentication approach using dynamic pseudonyms that does not affect the serving networks and the mobile equipment.
- We present an approach for identity recovery for use in the event of identity desynchronisation.

*The author is a commonwealth scholar, funded by the UK government.

WiSec '17 , Boston, MA, USA

DOI: 10.1145/3098243.3098248

- We provide a detailed analysis of the scheme, and verify its correctness and security properties using ProVerif.

The remainder of the paper is structured as follows. In Section 2 key terminology for, and features of, mobile telephony systems are introduced. The BVR and KM schemes are introduced in Section 3. The proposed new authentication scheme is described in detail in Section 4, and Section 5 gives a novel approach for identity recovery. Section 6 provides analyses, and Section 7 reports on a formal verification of the scheme. Section 8 provides a summary of other related work, and conclusions are drawn in Section 9.

2 MOBILE TELEPHONY SYSTEMS

2.1 Terminology

A complete mobile phone is referred to as a *user equipment* (UE), where the term covers both the *mobile equipment* (ME), i.e. the phone, and the *universal subscriber identity module* (USIM) within it, where the USIM takes the form of a cut-down smart card. The USIM embodies the relationship between the human user and the issuing home network, where this relationship includes the IMSI, the telephone number of the UE, and other user (subscriber) data, together with a secret key K which forms the basis for all the air interface security features. Within the USIM, information is stored in hierarchically structured files, known as *elementary files* (EFs), *dedicated files* and *application dedicated files*. The USIM data storage capabilities are specified in Section 10.1 of 3GPP TS 21 111 [5].

To attach to a mobile network, a UE connects via its radio interface to a radio tower, linked to the core network *mobility management entity* (MME). The MME is responsible for authenticating and tracking the location of UEs, paging, and interfacing with the *home subscriber server* (HSS), e.g. to obtain authentication data and update location data. The HSS resembles the *home location register* (HLR) of 3G systems, and stores mobility and service data for subscribers. It contains the *authentication centre* (AuC), which stores the unique secret key K for each USIM. The MME is part of the *serving network* (SN), whereas the HSS is the *home network* (HN) component.

2.2 Identities

IMSI: An IMSI is a string of 15 decimal digits, of which the first three are the *mobile country code* (MCC), and the next two or three form the *mobile network code* (MNC) that identifies the network operator. The MNC length, i.e. whether it is two or three digits, is a national matter. The remaining nine or ten digits, known as the *mobile subscriber identification number* (MSIN), are administered by the operator in line with national policy [6, 47]. IMSIs thus have geographical significance, and are typically managed by the network operator in blocks. The MCC/MNC combination uniquely identifies the HN, and is known as the *public land mobile network identity* (PLMN-ID). The MSIN is used by the network operator to identify the subscriber for billing and other operational purposes. Each IMSI uniquely identifies the mobile and the HN. The IMSI is stored in the USIM and is normally fixed. An IMSI is represented in binary coded decimal, and is stored in the elementary file EF_{IMSI} [9].

International Mobile Equipment Identity (IMEI): An IMEI is a string of 15 decimal digits used to uniquely identify an ME, allowing stolen equipment to be blacklisted. Since it is permanent, if sent in cleartext it could compromise user privacy. However, the 4G standards prohibit a UE from transmitting an IMEI until after a security context has been activated [25]; as a result, if equipment behaves in accordance with the standards, the IMEI should not pose a privacy threat, and so we do not discuss it further here.

Temporary Identities: To enhance user privacy, both 3G and 4G networks employ temporary subscriber identities, namely the TMSI (3G) and the GUTI (4G), avoiding the need to send the IMSI across the air interface. When a subscriber moves from one location area to another, its temporary identity is updated by the SN. A location area is a geographical sub-division of a network coverage area; depending on the service, similar sub-divisions are known as routing areas and tracking areas. Although a subscriber is temporarily traceable via its temporary identity, the length of time a single temporary identity remains valid is configurable by the SN.

2.3 Authentication

Authentication in 3G and 4G systems is performed using the *authentication and key agreement* (AKA) protocol. In AKA, the HN's AuC generates *authentication vectors* (AVs) containing the required data to perform authentication, and sends them to the SN. More specifically, an AV contains a random 'challenge' $RAND$, an expected response $XRES$, an authentication token $AUTN$, and a master session key K_{ASME}. AKA starts with the SN sending a *user authentication request* to the UE. The UE checks the validity of this request, and then sends a *user authentication response*. The SN checks this response to authenticate the UE. This enables the UE and network to authenticate each other and simultaneously establish a shared secret session key.

To participate in AKA, the USIM has a long term secret key K, known only to the USIM and the USIM's HN, and a sequence number SQN, maintained by both USIM and HN. The key K never leaves the USIM, and the values of K and SQN are protected by the USIM's physical security features. The 48-bit SQN enables the UE to verify the 'freshness' of the user authentication request. More specifically, the request message contains two 128-bit values: $RAND$ and $AUTN$, where $RAND$ is a random number generated by the HN, and $AUTN$ consists of the concatenation of: $SQN \oplus AK$ (48 bits), AMF (16 bits), and MAC (64 bits), where $\oplus$ denotes bitwise exclusive-or. The MAC is a *message authentication code* computed as a function of $RAND$, SQN, AMF, and the long term secret key K, using a MAC algorithm known as $f1$. The value AK is computed as a function of K and $RAND$, using a cipher mask generating function known as $f5$. The AK

functions as a means of encrypting SQN; this is necessary since, if sent in cleartext, the SQN value would potentially compromise user identity confidentiality, given that the value of SQN is USIM-specific.

On receipt of these two values, the USIM uses the received $RAND$, along with its stored value of K, to regenerate the value of AK, which it can then use to recover SQN. It next uses its stored key K, together with the received values of $RAND$, SQN, and AMF, in function $f1$ to regenerate $XMAC$; if the newly computed value agrees with the MAC value received in $AUTN$ then the first stage of authentication has succeeded. The USIM next checks that SQN is a 'new' value; if so it updates its stored SQN value and the network has been authenticated. If authentication succeeds, the USIM computes another message authentication code, called RES, and sends it to the network as part of the user authentication response. If this RES agrees with the value expected by the SN ($XRES$ in the AV) then the UE is deemed authenticated by the SN.

AKA relies on a set of cryptographic functions known as $f1$–$f5$, $f1^*$, and $f5^*$. Although implementation of these functions is operator-specific and not fully standardized, the 3GPP documents [13, 14] give an example set of algorithms for these functions. In this paper we assume that the network operators use functions with similar characteristics to those proposed by 3GPP [13, 14].

Error Reporting: The AKA protocol could fail for a variety of reasons, as discussed in 3GPP TS 24.008 [8]. In this paper we are interested in the failures involving the authentication challenge: a *MAC-failure* arises if the MAC sent by the network does not match the value the USIM computes, and a *sync-failure* occurs if the SQN value sent by the network is not greater than the USIM's stored value. A *MAC-failure* is reported via a signalling message, whereas a *sync-failure* is communicated via a token known as $AUTS$. An $AUTS$ is constructed by concatenating a masked version of the USIM's SQN and a MAC computed by the USIM, known as $MAC\text{-}S$. The USIM computes $MAC\text{-}S$ using the function $f1^*$ with inputs: the received $RAND$, its stored key K, its own SQN, and a dummy value of AMF [11]. To conceal SQN, the USIM masks it with AK, computed using a variant of $f5$ known as $f5^*$.

2.4 Location Update

A UE can initiate a location update procedure for a range of reasons, including: routing area update, tracking area update, and routing/tracking area update with IMSI attach [7]. On receiving a *location update* request, the SN runs AKA, enables the security context, and sends a new temporary identity to the UE; it then initiates a location update with the subscriber's HN (see 3GPP TS 23.012 [2] and 3GPP TS 23.401 [7]).

3 THE BVR AND KM SCHEMES

We next introduce the operation of the BVR and KM schemes (see Section 1), which relate to the scheme we propose in this paper. We also discuss the desynchronisation problem present in both schemes, which motivates the design of the scheme described in the remainder of this paper.

3.1 The BVR Scheme

In BVR [51], the subscriber's IMSI is replaced with a changing pseudonym called the *pseudo mobile subscriber identifier* (PMSI), which is only resolvable by the USIM's HN. The structure of a PMSI is the same as that of an IMSI, and it is treated like an IMSI by the SN which is unaware of the fact it is not an IMSI. The scheme requires the AuC to store three additional values for each USIM: a new shared secret key k_e, the current PMSI in use by the subscriber, and a future PMSI. When requested by an SN for an AV for a particular PMSI, the AuC executes the following steps.

(1) It compares the PMSI with all current and future PMSIs, and continues if it finds a match.
(2) If the match is with a future PMSI, the AuC updates the subscriber's current PMSI to equal the received PMSI, and assigns a randomly chosen PMSI from the pool of free PMSIs to be the subscriber's future PMSI. If the match is with a current PMSI then the AuC database is unchanged.
(3) The AuC computes the $RAND$ by AES-encrypting the concatenation of the 34-bit binary-coded MSIN part of the future PMSI and the subscriber's current SQN, using the key k_e.
(4) The AuC computes the other authentication parameters in the AV as a function of the computed $RAND$ (in the standard way), and transmits the AV to the SN.

The USIM must possess k_e, along with the current and future PMSIs. On receiving $RAND$ and $AUTN$, the USIM executes the usual AKA steps; if AKA is successful, the USIM decrypts the received $RAND$ and compares the recovered SQN value with the received SQN. If they agree, the USIM extracts the binary-encoded MSIN, converts it to binary coded decimal, and uses it to compute the new future PMSI. The USIM continues for the moment to use its current PMSI. When the USIM next receives an identity request, it sets its stored current PMSI to the stored future PMSI, and responds with this new value. That is, an updated PMSI is only used on the next occasion that the USIM receives an identity request, preventing linking of new and old PMSI values by observers. However, how the ME will be made aware of the new PMSI is not discussed in [51].

3.2 The KM Scheme

Analogously to BVR, KM [31] makes use of multiple IMSIs for an individual USIM. The scheme requires the AuC to store an *IMSI-change flag* for each subscriber indicating whether an IMSI change is required, and up to two IMSIs: $IMSI_{allocated}$ and $IMSI_{intransit}$. The event that triggers setting of the *IMSI-change flag* is a policy matter for the network. When requested by an SN for an AV for a particular IMSI, the AuC executes the following steps.

(1) It compares the IMSI with all $IMSI_{allocated}$ and $IMSI_{in\ transit}$ values, and continues if it finds a match.
(2) If the match is with an $IMSI_{intransit}$, the AuC adds the subscriber's current $IMSI_{allocated}$ to the pool of unused IMSIs, sets $IMSI_{allocated}$ to the received IMSI, and clears the subscriber's $IMSI_{intransit}$ value.
(3) It checks the IMSI-change flag, and computes an AV in the standard way if the flag is cleared; otherwise it computes an AV as follows.
    (a) The AuC computes a 64-bit MAC as a function of the subscriber's SQN and the shared secret key K, using a variant of the existing $f1$ function, the presence of which is used to instruct the USIM to make an IMSI change (see below).
    (b) The AuC generates a 48-bit mask key EK as a function of the subscriber's SQN and K, using a variant of the existing $f5$ function.
    (c) If the subscriber's $IMSI_{intransit}$ is null, the AuC assigns a free IMSI to it.
    (d) The AuC XORs the MSIN part of the subscriber's $IMSI_{intransit}$ (binary coded decimal) with EK, and computes $RAND$ by concatenating the MAC, the masked MSIN, and a 16-bit random number.
    (e) The AuC computes the other authentication parameters in the AV in the standard way.

The USIM must temporarily store the value of the last received IMSI. On receipt of an authentication request, the USIM uses the standard AKA procedure. After successfully completing AKA, the USIM computes MAC as a function of the received SQN, and compares it with the appropriate part of $RAND$. If they match, the USIM regenerates EK and XORs it with the masked MSIN extracted from $RAND$. The USIM then constructs the new IMSI by prefixing the retrieved cleartext MSIN with the PLMN-ID. If the constructed IMSI is different to its current stored value, the USIM updates both the stored value and the elementary file EF_{IMSI}, which contains the IMSI. The USIM then uses the proactive UICC feature [10] to instruct the ME to read this file, causing it to start using the new IMSI.

3.3 Practical Issues with BVR and KM

Unfortunately, both schemes are susceptible to possible identity desynchronisation, i.e. where the USIM and its HN hold different IMSI values. An active adversary capable of presenting an IMSI of its choice to the HN could cause the HN to incorrectly update its database if the presented IMSI is equal to a stored future PMSI (for BVR) or a stored $IMSI_{intransit}$ (for KM). If the corresponding USIM has not received the new IMSI (or PMSI), this could cause a permanent denial of service, since there will be no way for synchronisation to be regained.

Desynchronisation could arise in a variety of ways. One possible scenario involves a compromised SN sending randomly chosen IMSIs to an HN, e.g. embedded in an AV request. If a received IMSI is equal to a currently free IMSI, the HN will respond with an error in the standard way. If the IMSI is equal to a current IMSI (a current PMSI for BVR, or an $IMSI_{allocated}$ for KM), the HN will provide an AV. Finally, and most importantly here, if the IMSI is equal to a future IMSI (a future PMSI for BVR, or an $IMSI_{intransit}$ for KM), then there will be a loss of IMSI synchronisation if the mobile to which the future IMSI belongs has not received it, and this could cause a permanent denial of service.

A related but distinct scenario involves a malicious or malfunctioning ME submitting random IMSIs to an SN. The SN will use a received IMSI to request an AV from the HN indicated by the PLMN-ID part of the IMSI. The remainder of the attack will work exactly as in the scenario described in the previous paragraph; that is, if the random IMSI happens to equal a stored future IMSI for a USIM which has not yet received this IMSI, then the corresponding USIM will suffer from a catastrophic identity desynchronisation. Hence any such changeable IMSI scheme needs to incorporate an identity recovery mechanism.

Also, both BVR and KM involve changing the IMSI at the HN. This introduces major management concerns because the IMSI is the only permanent unique identifier in the HN, and other services depend on fixed IMSIs. Thus, a scheme supporting pseudonyms without changing the IMSIs in the HN would be highly advantageous.

4 PRIVACY ENHANCED AKA

We now describe a new set of modifications to the operation of 3G and 4G systems designed to address the same privacy concerns as the BVR and KM schemes, and using very similar ideas. However, it incorporates significant additional features that protect against permanent loss of identity synchronisation between a USIM and its HN. The scheme also avoids changing the IMSI in the HN, i.e. it addresses both the major concerns raised in Section 3.3.

4.1 Overview

Just like BVR and KM, the scheme uses changing pseudonyms. The scheme avoids sending the subscriber MSIN across any communication channels, including the radio path and the core network, and instead uses a *transient identity* (TID), i.e. a temporary identity managed by the HN with the same length as an MSIN. The TID works rather like the TMSI except that TIDs are managed by the HN instead of the SN. The TID provides subscriber pseudonymity on the air interface, avoiding the need for cleartext transmission of the IMSI. TIDs are mapped to fixed MSINs in the HN database, and the IMSI is only accessible by the HN.

The initial *pseudo-IMSI*, equal to the concatenation of the PLMN-ID and the initial TID, must be stored in a USIM during personalisation. The pseudo-IMSI is indistinguishable from an IMSI to any party other than the USIM and HN. As in BVR and KM, the pseudo-IMSI is treated as an IMSI by the SN, and is periodically refreshed by the HN. An unauthenticated UE (without a valid TMSI/GUTI) identifies itself to the SN using the pseudo-IMSI, exactly as an IMSI

is used currently. The SN learns the identity of the HN from the pseudo-IMSI, and forwards it to the HN as part of an AV request. The HN uses the TID from the pseudo-IMSI to learn the IMSI. It then chooses a new TID to refresh the pseudo-IMSI, embeds this new TID into the *RAND* analogously to BVR and KM, and computes the AV, which is sent to the SN for use in AKA.

After a successful AKA, the USIM has the new TID, and uses it to construct the next pseudo-IMSI. At the same time the SN allocates the UE a local TMSI or GUTI. As in BVR and KM, the old pseudo-IMSI remains in use by the current SN, e.g. in mobile terminated services. The new pseudo-IMSI is used in subsequent AKA executions. We also introduce the notion of a *recovery identity* (RID) for each USIM, to enable pseudo-IMSI recovery in a privacy-preserving way.

4.2 Modifications to AKA

The pseudo-IMSI is indistinguishable from an IMSI to the SN, and is used in exactly the same way, i.e. when polled for its IMSI the phone will respond with its pseudo-IMSI. The operation of AKA is also unchanged; the only difference is in the composition of *RAND*, as discussed in Sections 4.3 and 4.4, and this difference is transparent to the SN.

The only other change of relevance here is to error handling. As noted in Section 2.3, if a USIM fails to authenticate a network, it responds with either a *MAC-failure* message, or a *sync-failure* message. As part of the novel scheme (discussed in detail in Section 4.4.2) if the authentication fails due to a MAC mismatch, the USIM incorporates a novel failure type indication in an error token that is sent to the SN (and hence to the HN) just like a *sync-failure* error. Since the SN does not process the error token, the incorporation of a new type of error code is transparent to it.

That is, the only change from the SN perspective is that it will never see a *MAC-failure* message; a malicious third party could use this to distinguish between an IMSI and a pseudo-IMSI by sending a modified (and hence incorrect) MAC value to a UE and seeing what error message is returned.

4.3 Modifications to Home Network

4.3.1 Fundamentals. We first introduce certain data structures which form part of the modified scheme.

- *TID*: A TID substitutes for the MSIN, and so its maximum length is 40 bits (i.e. ten binary coded decimal digits). If the HN implementation encodes the HSS/HLR identity in the MSIN, as is done in some networks, the bit-length of a TID could equal the number of bits used to uniquely identify a subscriber of that specific HSS/HLR.
- *RID*: Like the TID, a RID is a 48-bit USIM identifier, used only in the pseudo-IMSI recovery procedure described in Section 5.
- *Linked-TID*: This is a specific TID, sent with a RID to a subscriber. This TID (if not null) is equal to the value of the HN stored TID_{future} (see below).
- *RID Flag*: This bit, maintained for every subscriber in the HN database, indicates whether or not a RID value should be embedded into the next *RAND* for a subscriber. Initially the flag is cleared. The use of the flag is similar to that of the *IMSI-change flag* in KM.
- *Keystream 1 (EK_1)*: Like *EK* in the KM scheme, the 48-bit EK_1 is derived from subscriber-specific secrets using a *key derivation function* (KDF). It is used to mask the TID when sent from the HN to a USIM.
- *Keystream 2 (EK_2)*: This is another 48-bit string generated using a different KDF from subscriber-specific secrets. It is used to mask the RID when it is transferred from the HN to a USIM.
- *Padding data*: Pad_1 and Pad_2 (each of 80 bits) are used to extend the subscriber's *SQN* value to a 128-bit string, as required for input to the $f5$ variant functions.
- *Instruction byte*: This is used to instruct a USIM to perform certain operations. We use specific values of the byte ($INS_{regular}$, INS_{RID} and INS_{reset}) to instruct the USIM to perform certain tasks. All other values are reserved for future use — on receipt of a reserved value the USIM should not take any action.

For each subscriber account in the database, the HN must maintain a *RID flag* indicating whether a RID change is under way. The database must also hold up to three TIDs (TID_{past}, $TID_{current}$ and TID_{future}), three RIDs (RID_{past}, $RID_{current}$ and RID_{future}) and a *linked-TID* for each subscriber. It will always hold a $TID_{current}$ and a $RID_{current}$. When a new TID is transferred to the USIM, the database will hold a TID_{future}; on a TID update, described in Section 4.3.3, it will hold a TID_{past}. If the *RID flag* is set, the database will also hold a *linked-TID* and a RID_{future}. Analogously to the TID, on a RID update, the database will hold a RID_{past}. The HN must also maintain a pool of unused TIDs and RIDs, enabling the AuC to dynamically assign a new TID and RID to an existing subscriber.

The following functions are used in the scheme described below.

- $f5^{**}$: This KDF is a variant of the existing $f5$. It takes as input a 128-bit string and the shared secret key K, and generates EK_1. The function must be chosen so that it is computationally infeasible to derive the key K from knowledge of the string and EK_1, the requirements of $f5$ as described in [12].
- $f5^{***}$: This function, used to generate EK_2, must have similar properties to $f5^{**}$, but for cryptographic cleanliness we use a distinct function.
- *Randgen*: This function takes as input an integer, and returns a pseudo-random number of the length specified.

4.3.2 AV Generation. On receiving a request for an AV for a pseudo-IMSI, the AuC first retrieves the TID from the pseudo-IMSI and searches its subscriber database for

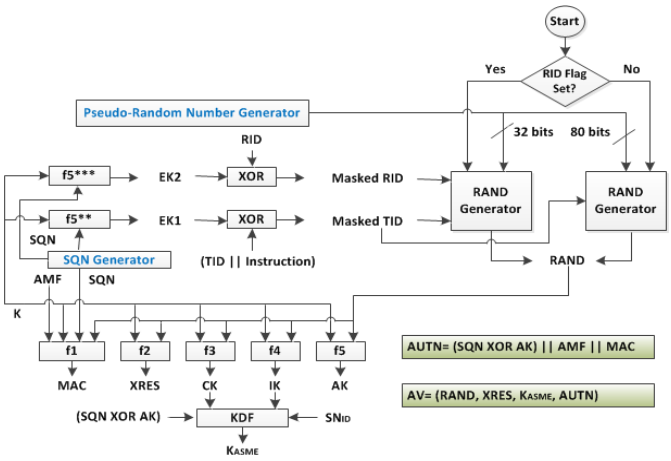

Figure 1: Computation of an AV

the corresponding IMSI. If the IMSI is not found, the AuC generates an arbitrary AV in which all the values are chosen at random (to initiate pseudo-IMSI recovery by the receiving USIM, as described in Section 5), and sends it to the SN. Otherwise, the AuC retrieves the IMSI and proceeds as follows (see Figure 1).

(1) It retrieves the shared key K, SQN, RID flag, and stored value of TID_{future} for this IMSI.
(2) It sets EK_1 to equal $f5^{**}(SQN||Pad_1, K)$.
(3) If the value of TID_{future} is null, the AuC sets it to equal a fresh TID selected from the pool of unused TIDs. The AuC also removes the allocated TID from the unused TID pool.
(4) If the RID flag is not set, the AuC sets *masked-TID* to $(TID_{future}||INS_{regular}) \oplus EK_1$, and sets $RAND$ to equal *masked-TID*$||$*randgen(80)*. Otherwise, the AuC computes $RAND$ as follows.
    (a) It sets EK_2 to equal $f5^{***}(SQN||Pad_2, K)$, and *linked-TID* to equal TID_{future}.
    (b) If the value of RID_{future} is null, the AuC sets it to equal a fresh RID selected from the pool of unused RIDs. The AuC also removes the RID from the pool of unused RIDs.
    (c) It sets *masked-TID* to $(TID_{future}||INS_{RID}) \oplus EK_1$, and *masked-RID* to $RID_{future} \oplus EK_2$.
    (d) It sets $RAND$ to equal *masked-TID*$||$*masked-RID*$||$*randgen*(32).
(5) The AuC generates the AV using the computed $RAND$ in the standard way.

4.3.3 Identity Update. Although an HN keeps the subscriber's security credentials to itself, it delegates authentication to an SN by passing it an AV; thus the HN is not aware when a specific AV is used in AKA. Thus the HN does not have a direct means of knowing when a USIM receives a new TID. As discussed in Section 2.4, a *location update* request from an SN is preceded by a successful AKA; we therefore use the receipt by an HN of a *location update* request as implicit indication that a USIM has received the TID in the provided pseudo-IMSI; we use this to trigger a TID update. This approach differs from BVR and KM, and to some extent restricts[1] unauthorised updates to the HN database.

When an HSS receives a location update request, it sends the embedded pseudo-IMSI to the AuC, which may trigger an identity update. (In 3G, the location update request is sent to the HLR and not the AuC although, since they are controlled by the same network, adding the necessary intercommunication should not be difficult.)

On receiving the pseudo-IMSI, the AuC first retrieves the embedded TID, and searches its subscriber database for the corresponding IMSI. If the retrieved TID is not found (as might occur with a maliciously generated location update request), the AuC takes no further action; otherwise, the AuC retrieves the IMSI and compares the embedded TID with the value of TID_{future} for this IMSI. If they do not agree (including if the TID matches either TID_{past} or $TID_{current}$ for this IMSI), the AuC takes no further action; otherwise, the AuC performs the following steps.

(1) It deletes the value of TID_{past} for the retrieved IMSI, and adds the value of TID_{past} to the pool of unused TIDs.
(2) It sets TID_{past} to equal $TID_{current}$, $TID_{current}$ to equal TID_{future}, and sets TID_{future} to null.
(3) It checks the RID flag. If the flag is clear, the AuC takes no further action. Otherwise, it retrieves the value of *Linked-TID* for the IMSI, and compares this value with the TID retrieved from the pseudo-IMSI. If they do not agree, the AuC takes no further action; otherwise, it updates the RID information in its subscriber database as follows.
    (a) It deletes the value of RID_{past} for the retrieved IMSI, and adds this to the unused RID pool.
    (b) It sets RID_{past} to equal $RID_{current}$, and $RID_{current}$ to equal RID_{future}.
    (c) It sets RID_{future}, RID flag, and *Linked-TID* to null.

4.3.4 RID Flag Setting. The RID flag is used to indicate to the AuC that the RID should be updated at the next opportunity. The flag is cleared initially, and it is also cleared during identity update (when the RID is changed); however, we do not specify when it is set, since this is a matter for network policy. A possible trigger for setting the flag would be if the network believes the RID may have been disclosed, e.g. if a *MAC-failure* token is received.

4.4 Modifications to USIM

To support the scheme, a USIM will need to store certain additional information. We propose that the pseudo-IMSI is stored in the existing file EF_{IMSI}, and that the RID is stored in a new EF. We further suppose that the initial RID value is set during USIM personalisation.

[1] A maliciously modified ME cannot falsely claim to possess a TID which it has not been allocated, as discussed in Section 3.3.

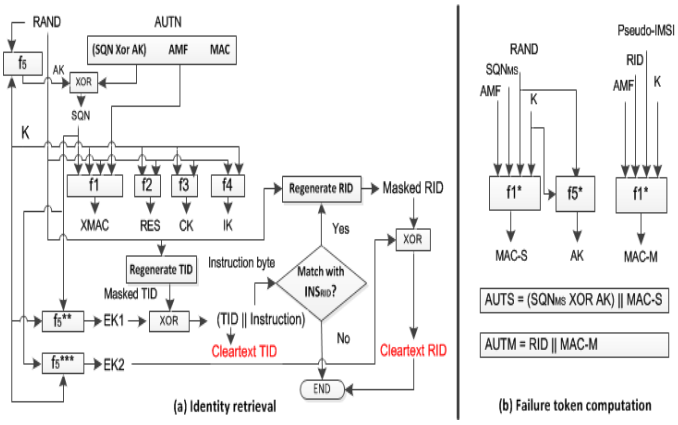

Figure 2: Modified processes in USIM

4.4.1 New Identity Retrieval. On receipt of an authentication challenge, the USIM proceeds using the standard AKA procedure. After successful completion of AKA, the USIM processes the $RAND$ to retrieve the new identities as follows (see Figure 2(a)).

(1) It sets EK_1 to equal $f5^{**}(SQN||Pad_1, K)$.
(2) It retrieves *masked-TID* from $RAND$, and parses *masked-TID*$\oplus EK_1$ to obtain the TID and instruction byte.
(3) It compares the retrieved instruction byte with the three predefined values $INS_{regular}$, INS_{RID} and INS_{reset}. If it equals any of these values then the USIM concatenates the received TID with the PLMN-ID to obtain the new pseudo-IMSI, and compares the concatenated value with the stored pseudo-IMSI (an essential step since the same TID may be received multiple times). If they are the same, the USIM takes no further action; otherwise, it keeps a record of the new pseudo-IMSI and later updates its pseudo-IMSI using the procedure described in Section 4.4.3.
   (a) If the received instruction byte equals INS_{RID}, then the USIM also performs the following steps.
      (i) It sets EK_2 to equal $f5^{***}(SQN||Pad_2, K)$. It retrieves *masked-RID* from $RAND$, and parses *masked-RID*$\oplus EK_2$ to obtain a RID.
      (ii) It compares the retrieved RID with the stored RID (an essential step since the same RID may be received multiple times). If they are the same the USIM takes no further action; otherwise it updates its stored RID.
   (b) If the received instruction byte equals INS_{reset}, then the USIM immediately updates its pseudo-IMSI with the received pseudo-IMSI, using the procedure described in Section 4.4.3. This case will arise when an HN wishes to synchronise the pseudo-IMSI between the USIM and the HN, as described in Section 5.

4.4.2 Failure Token Generation. As mentioned in Section 4.2, we introduce a novel error token, the $AUTM$, to enable a USIM to report a *MAC-failure* arising during authentication. To construct $AUTM$, the USIM computes a 64-bit MAC, the *MAC-M*, as a function of its stored RID, the current pseudo-IMSI, the key K, and a dummy AMF, using the existing $f1^*$ function (see Figure 2(b)), and sets $AUTM$ to $(RID||MAC-M)$. The structure of $AUTM$ is similar to that of the *sync-failure* token $AUTS$, discussed in Section 2.3. When a USIM detects a *MAC-failure* in authentication, it computes $AUTM$, and reports it to the SN in the same way as an $AUTS$.

4.4.3 New Pseudo-IMSI Notification. The scheme requires a way to transfer a new pseudo-IMSI from the USIM to the ME for subsequent use. As in KM [31], we propose using the USIM application toolkit feature [10] to instruct the ME to read the new pseudo-IMSI and thereby start using the new value.

5 PSEUDO-IMSI RECOVERY

5.1 A Desynchronisation Scenario

Unfortunately, the authentication scheme described here is not completely free from possible pseudo-IMSI desynchronisation. However, it allows pseudo-IMSI synchronisation to be regained as soon as its loss is detected. Analogous to the discussion in Section 3.3, one possible scenario for such an event is as follows.

Suppose a malicious entity, e.g. a compromised SN, able to initiate a *location update* request, sends such a request containing a randomly chosen pseudo-IMSI. If, by chance, the TID in this pseudo-IMSI happens to match a stored TID_{future}, then the HN will incorrectly update its database. If this 'new' TID has not been received by the corresponding USIM, then the current TID in the USIM will equal the value of TID_{past} in the HN database (see steps 1 and 2 in Section 4.4.3). Unlike the analogous scenario discussed in Section 3.3, this scenario does not cause pseudo-IMSI desynchronisation in the scheme described here, since the AuC is still able to map the IMSI to the TID currently stored in the USIM.

However, even in the scheme described here, the AuC will lose its mapping from the IMSI to the received pseudo-IMSI if the malicious entity could successfully cause another unauthorised TID update in the HN database. If pseudo-IMSI synchronisation is lost (and cannot be recovered), AKA will always fail, and the UE will not be able to receive any network service until the USIM is replaced. The likelihood of such a pseudo-IMSI desynchronisation is reduced in the new scheme, in that it requires two false updates to the HN database; however, such an event is disastrous, and so a way to recover from this failure state is needed. We describe the recovery process in Section 5.2 below.

5.2 Synchronisation Recovery Process

In current systems, there are scenarios in which desynchronisation can occur. For example, TMSI-IMSI synchronisation

can be lost by an SN if it receives repeated *MAC-failure* messages as a result of failed authentications. In such a situation, the SN recovers by requesting the cleartext IMSI, obtaining a new set of AVs from the HN, running AKA, and allocating a new TMSI.

Also, as discussed in Section 2.3, if a USIM identifies loss of *SQN* synchronisation between it and the HN, it resynchronises *SQN* by sending a *sync-failure* token to the HN. On receiving such a token, the HN adjusts its stored value of *SQN*, computes an AV with the new value, and sends the AV to the SN for use in AKA.

Analogously, in the new scheme described here, if a USIM detects a possible pseudo-IMSI desynchronisation when authentication fails because of a MAC mismatch, it sends an $AUTM$ error token to the HN (via the SN). The SN includes other relevant information, notably the pseudo-IMSI and the failure cause. As $AUTS$ and $AUTM$ are indistinguishable to the SN, the SN reports both types of token as a *sync-failure*.

On receiving the token, the AuC first runs the standard validation steps for an $AUTS$. If validation succeeds, the AuC performs the standard process for *SQN-recovery*; otherwise, it further verifies the token to confirm its validity as an $AUTM$. As part of $AUTM$ validation, the AuC proceeds as follows.

(1) It parses the token to retrieve a RID and *MAC-M*.
(2) It searches its subscriber database for the IMSI for this RID. If the RID is not found, validation fails; otherwise, the AuC retrieves the IMSI and the corresponding K.
(3) It computes *MAC* as $f1^*$*(RID, pseudo-IMSI, AMF, K)*, and compares it with the retrieved *MAC-M*. If they agree, the token is validated; otherwise, validation fails.

If $AUTM$ validation fails, the AuC reports the issue to the SN; otherwise, it rectifies the TID entries in its subscriber database using the procedure described in the following paragraphs, computes an AV, and sends the AV to the SN for authentication.

As noted in the previous paragraph, if $AUTM$ validation succeeds, the SN must adjust its TID values appropriately; this is achieved as follows. The AuC first retrieves the TID from the pseudo-IMSI sent with the $AUTM$ token by the SN, and compares it with the three stored TID values (TID_{past}, $TID_{current}$ and TID_{future}) for this IMSI. If it equals any of the these values then the *MAC-failure* does not indicate that a pseudo-IMSI desynchronisation has occurred. Hence, in this case the AuC does not modify its subscriber database, computes an AV using the procedure described in Section 4.3.2, and sends the AV to the SN.

Otherwise, the AuC proceeds as follows.

(1) It deletes the values of TID_{past} and $TID_{current}$ for the IMSI concerned, and adds them to the pool of unused TIDs.
(2) It checks whether the retrieved TID is in the pool of unused TIDs. If the TID is available, the AuC sets $TID_{current}$ to equal this TID, computes an AV using the procedure described in Section 4.3.2, and sends the AV to the SN. Otherwise, the pseudo-IMSI reported by the USIM must have been allocated to another subscriber, and the AuC performs the following steps.
   (a) It computes EK_1 and selects a TID using steps 2 and 3 of Section 4.3.2.
   (b) The AuC sets $RAND$ to equal $((TID_{future}||INS_{reset}) \oplus EK_1)||randgen(80)$.
   (c) It generates an AV using the computed $RAND$ in the standard way, and sends the AV to the SN.

Note that a *MAC-failure* token reported by a legitimate subscriber due to loss of synchronisation between its TMSI and the corresponding pseudo-IMSI will be deemed invalid by the HN. This is because the MAC component of the token is computed over the USIM's actual pseudo-IMSI, whereas the SN reports a different pseudo-IMSI. To deal with this case, the SN could either use the HN's response in processing the failure token, or simply request the cleartext pseudo-IMSI before forwarding a failure token.

6 ANALYSIS

6.1 Authentication

The modified AKA protocol described here is as secure as the existing AKA, since we have not modified it except to replace the random $RAND$ with a cryptographically constructed $RAND$. The constructed $RAND$ is indistinguishable from a random value if the input key is not known, on the assumption that data strings masked using the output of the functions $f5^{**}$ and $f5^{***}$ are indistinguishable from random data, cf. [3, 4]. The security properties of AKA have been widely analysed, [1, 34, 46].

Since a USIM accepts the identities sent embedded in $RAND$ only after AKA has completed successfully, i.e. after the SN has been authenticated, the modified AKA guarantees the origin, integrity and timeliness of the new pseudo-IMSI. Hence, an active adversary cannot force a USIM to change its pseudo-IMSI to something other than a value selected by the HN. Moreover, because the SQN value is checked by the USIM during AKA, an active adversary cannot force a USIM to accept an 'old' pseudo-IMSI, since this checking requires AVs to be used in strict order of generation.

The scheme does not change the way the data confidentiality and integrity keys are generated.

We formally verified our security claims using ProVerif (see Section 7 below).

6.2 User Identity Confidentiality

An adversary cannot infer any confidential information from $RAND$, since the private values TID, RID, and the instruction byte embedded in $RAND$ are all masked. An adversary without access to the key K is unable to learn a pseudo-IMSI before it is used, ensuring unlinkability between consecutive pseudo-IMSIs.

The scheme diminishes the impact of IMSI catchers and improves user identity confidentiality by preventing the IMSI

ever being sent across the air interface. However, air interface interactions are not completely anonymous, since the pseudo-IMSI functions as a pseudonym, potentially enabling the interactions of a single phone to be tracked for a period; of course, this is always true if a subscriber resides in a single location area, even where only a temporary identity, i.e. a TMSI or GUTI, is used.

Frequent AKA execution could lessen the impact of such tracking, which would also alleviate the problem of long TMSI validity periods over multiple geographic areas, as reported by Arapinis et al. [15]. Although the requirement to execute AKA frequently is reduced in 4G, the importance of frequent AKA execution in preventing security attacks is discussed in recent research [29].

6.3 Pseudo-IMSI Recovery

A request for pseudo-IMSI recovery cannot be forged, since the request contains a MAC computed over the subscriber's RID, pseudo-IMSI, and a value of AMF, using the key K. The MAC guarantees detection of malicious changes to the RID or reported pseudo-IMSI, preventing an adversary falsely initiating a pseudo-IMSI recovery. We formally verify this claim using ProVerif (see Section 7).

The pseudo-IMSI synchronisation recovery process is similar to the existing *SQN* synchronisation recovery process. Unlike for *SQN*, the RID in the pseudo-IMSI recovery request is transferred in cleartext, since the HN is unable to identify the subscriber from the reported pseudo-IMSI while pseudo-IMSIs are desynchronised. This allows possible user tracking using the RID. As the RID changes over time, the traceability of the RID is the same as that of the existing temporary identity.

6.4 Identity Synchronisation

As discussed in Section 6.1, if the cryptographic assumptions for AKA hold, an adversary cannot force a subscriber to change its pseudo-IMSI illegitimately. Nevertheless, an adversary can stop or delay the arrival of a *RAND* containing a new pseudo-IMSI at a USIM. However, such an event does not affect pseudo-IMSI desynchronisation, since, as in BVR and KM, the HN retransmits a pseudo-IMSI until it has reliable evidence that it has been received by the USIM.

Unlike BVR and KM, an HN updates its subscriber database only when it receives a specific *location update* request from an SN, and it keeps the immediate past pseudo-IMSI for each subscriber. These changes help to minimise the likelihood of identity desynchronisation, as discussed in Section 5.1. Moreover, inclusion of a pseudo-IMSI recovery process guarantees synchronisation of pseudo-IMSI between USIM and HN.

Although RIDs and TIDs are managed in the same way, a RID is only used to recover pseudo-IMSI synchronisation. The frequency of RID updates is a policy matter for the network. It might be possible to deploy other methods to guarantee the synchronisation between RID and IMSI, a possible avenue for future research.

6.5 Deployment and Interoperability

The scheme modifies only the USIM and the HN, owned by a single entity, and is transparent to the SN and mobile phone. This allows phased deployment, e.g. by including the additional functionality in newly issued USIMs while existing USIMs continue to function as at present. In addition, the scheme does not affect existing services dependent on the IMSI, e.g. lawful interception and billing, making deployment simpler than for BVR and KM.

There are certain practical issues to be considered. For example, the set of 'normal' IMSIs used by existing USIMs needs to be kept distinct from the range of pseudo-IMSIs used by the new USIMs. The HN should use location information from both the user's current and past pseudo-IMSI in supporting mobile terminated services; that is, the HN might use the location information of the pseudo-IMSI containing TID_{past}, if delivery of a mobile terminated service using the pseudo-IMSI containing $TID_{current}$ fails.

The scheme will not work for GSM, as it depends on the mutual authentication feature of 3G and 4G AKA. If a UE using a new-style USIM needs to connect to a GSM network, it should continue to use the fixed pseudo-IMSI as long as it is connected to that network. The pseudo-IMSI can be updated when the UE next roams to a 3G or 4G network.

6.6 Performance and Overhead

The scheme introduces minimal additional overhead to a USIM. We add two KDFs (to retrieve identities) and one MAC function (to support identity recovery), all of which are similar to the existing USIM functions. Transferring a new pseudo-IMSI to the ME is a new task for a USIM, which could be performed when the ME is idle. We believe that this overhead should be manageable, even for a USIM with limited computational power.

The scheme requires the HN to manage new identities. It increases database transactions, adds two KDFs for computing an AV, and introduces new functions, notably the need to refresh an identity on receiving an appropriate *location update* request and identity recovery in the event of pseudo-IMSI desynchronisation. Since none of these are particularly complex, it seems likely that this could be achieved with some combination of allocating more resources, clustering subscribers in multiple HSSs, and efficient database design.

The scheme does not affect any functionality in the SN or introduce any additional communications. The only impact is an increase in the apparent number of subscribers at the SN, since subscribers switching to a new pseudo-IMSI appear like new subscribers.

Pseudo-IMSIs and IMSIs for a single HN must all be distinct. Since multiple pseudo-IMSIs are allocated for each subscriber, pressure could be created on the number of IMSIs available to an operator. To address this issue, multiple MNC codes could be allocated to an operator and three-digit MNCs could be used to avoid wastage of IMSIs allocated to small operators.

6.7 Impact on Other Attacks

Arapinis et al. [16] describe an attack allowing an adversary to distinguish between UEs based on the error messages arising from a failed AKA execution. The change to use of $AUTM$ for reporting *MAC-failure* messages, which are thereby indistinguishable from *sync-failures*, invalidates this attack.

An attack [42] on the USIM provisioning process compromises the key K, compromising all the security features. However, the scheme described here could reduce the effect of key compromise, since the initial mapping from pseudo-IMSI to the key K is lost as soon as the subscriber changes its pseudo-IMSI, and even an adversary knowing K would not be able to readily track a device.

7 FORMAL VERIFICATION

We used *ProVerif*, an automatic cryptographic protocol verifier [19], to analyse the proposed scheme. ProVerif is used to check the secrecy and authentication properties of cryptographic protocols. It can also be used to verify privacy properties [20]. The ProVerif checker is not complete, which means that it may not be capable of proving a property that holds (it might output invalid attacks). However, it is sound; that is, if ProVerif verifies that a property is satisfied then the model ensures the property.

We modelled the modified AKA protocol using the ProVerif formalism, and verified the security and privacy properties discussed in Section 6 above[2]. We checked whether the model ensures mutual authentication between the USIM and the SN, and this holds. We also verified the secrecy property of the transferred identities, i.e. a TID and a RID, and the model ensures their secrecy when transferred from the HN to the USIM. We further verified correctness of the pseudo-IMSI recovery process; to do so we proved that if the HN initiates a recovery for a USIM, it is indeed requested by a legitimate subscriber possessing the USIM. Finally, we followed a similar approach to van den Broek et al. [51] to successfully verify that the unlinkability property is achieved by the scheme.

These proofs increase confidence that the proposed scheme could defend against IMSI catchers while maintaining the original functionality of the protocol.

8 OTHER RELATED WORK

Apart from the recent work of van den Broek et al. [51] and Khan and Mitchell [31] (see Section 3), stopping IMSI catchers has been the subject of research for over 20 years; we review some of the most relevant.

Samfat and Molva [40] first addressed the underlying privacy issue, and suggested the use of public key encryption to hide the permanent identity. Herzberg et al. [30] and Ateniese et al. [17] discussed anonymity in mobile networks, and suggested use of a fixed shared secret key to encrypt permanent identities managed by an HN when sent to the SN. Mitchell et al. [35] proposed identity and location privacy mechanisms for 3G systems, using a new temporary user identity $TMUI_s$ for each subscriber. However, analysis of the usability of the scheme was left as future work.

Choudhury et al. [21] proposed a scheme to improve user identity confidentiality in the LTE network using a *dynamic mobile subscriber identity* (DMSI) instead of the IMSI, analogous to the scheme described here. However, their scheme changes the protocol messages and the entities in the mobile network. Køien [33] proposed a privacy enhanced mutual authentication scheme for LTE using identity-based encryption, again imposing significant modifications on all the major system elements.

Sung et al. [45] proposed a scheme to provide identity and location privacy making use of multiple IMSIs for a single SIM. However, their scheme involves an additional party in its operation, needs support by the ME, and requires wireless data connectivity for sending and receiving calls. The threat model is also very different, in that the HN is considered as a potential adversary.

Fouque et al. [26] have recently proposed a scheme addressing the IMSI privacy problem. Their scheme introduces a public key infrastructure for the HN, changes the message structure and adds new parameters to the AKA.

We use the *RAND*, sent from HN to UE during AKA, as a way of conveying information. This idea was apparently first described in a patent due to Dupré [23]. Later, Vodafone [52], and Khan and Mitchell [32] used a similar concept of a special RAND. However, these prior uses of the special RAND were completely different to that proposed here.

9 CONCLUSIONS

We described a novel authentication approach for 3G and 4G mobile systems that does not affect existing SNs and mobile phones. The IMSI is never sent across a communication channel; instead a changing pseudo-IMSI is used. The pseudo-IMSIs appear as new subscribers to the SN, and are unlinkable. We address the possible loss of pseudo-IMSI synchronisation between USIM and HN, and present an approach for pseudo-IMSI recovery to be used if pseudo-IMSI synchronisation is lost.

The scheme introduces changes to the operation of the HN and USIM, but not to the SN, the mobile device, or other internal network protocols, which enables transparent migration. We discussed the strengths and limitations of the scheme, and reported the results of a formal analysis. The scheme alleviates the decades-old privacy problem of IMSI disclosure on the air interface, and hence 'trashes the IMSI catchers in mobile network'.

[2]The model is available at https://pure.royalholloway.ac.uk/portal/files/28145762/ProVerif_Model.zip

REFERENCES

[1] 3rd Generation Partnership Project (3GPP) 2001. *3GPP TR 33.902 V4.0.0 : Technical Report; Technical Specification Group Services and System Aspects; 3G Security; Formal Analysis of the 3G Authentication Protocol (Release 4)*. 3rd Generation Partnership Project (3GPP).

[2] 3rd Generation Partnership Project (3GPP) 2015. *3GPP TS 23.012 V13.0.0 : Technical Specification; Technical Specification*

Group Core Network and Terminals; Location management procedures (Release 13). 3rd Generation Partnership Project (3GPP).

[3] 3rd Generation Partnership Project (3GPP) 2016. *3GPP TR 35.909 V13.0.0 : Technical Report; Technical Specification Group Services and System Aspects; 3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 5: Summary and results of design and evaluation (Release 13)*. 3rd Generation Partnership Project (3GPP).

[4] 3rd Generation Partnership Project (3GPP) 2016. *3GPP TR 35.934 V13.0.0 : Technical Specification; Technical Specification Group Services and System Aspects; Specification of the TUAK algorithm set: A second example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 4: Report on the design and evaluation (Release 13)*. 3rd Generation Partnership Project (3GPP).

[5] 3rd Generation Partnership Project (3GPP) 2016. *3GPP TS 21.111 V13.0.0 : Technical Specification; Technical Specification Group Core Network and Terminals; USIM and IC card requirements (Release 13)*. 3rd Generation Partnership Project (3GPP).

[6] 3rd Generation Partnership Project (3GPP) 2016. *3GPP TS 23.003 V14.1.0 : Technical Specification; Technical Specification Group Core Network and Terminals; Numbering, addressing and identification (Release 14)*. 3rd Generation Partnership Project (3GPP).

[7] 3rd Generation Partnership Project (3GPP) 2016. *3GPP TS 23.401 V14.1.0 : Technical Specification; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access (Release 14)*. 3rd Generation Partnership Project (3GPP).

[8] 3rd Generation Partnership Project (3GPP) 2016. *3GPP TS 24.008 V14.1.0 : Technical Specification; Technical Specification Group Core Network and Terminals; Mobile radio interface Layer 3 specification; Core network protocols; Stage 3 (Release 14)*. 3rd Generation Partnership Project (3GPP).

[9] 3rd Generation Partnership Project (3GPP) 2016. *3GPP TS 31.102 V14.0.0 : Technical Specification; Technical Specification Group Core Network and Terminals; Characteristics of the Universal Subscriber Identity Module (USIM) application (Release 14)*. 3rd Generation Partnership Project (3GPP).

[10] 3rd Generation Partnership Project (3GPP) 2016. *3GPP TS 31.111 V14.0.0 : Technical Specification; Technical Specification Group Core Network and Terminals; Universal Subscriber Identity Module (USIM) Application Toolkit (USAT) (Release 14)*. 3rd Generation Partnership Project (3GPP).

[11] 3rd Generation Partnership Project (3GPP) 2016. *3GPP TS 33.102 V14.0.0 : Technical Specification; Technical Specification Group Services and System Aspects; 3G Security; Security architecture (Release 14)*. 3rd Generation Partnership Project (3GPP).

[12] 3rd Generation Partnership Project (3GPP) 2016. *3GPP TS 33.105 V13.0.0 : Technical Specification; Technical Specification Group Services and System Aspects; 3G Security; Cryptographic algorithm requirements (Release 13)*. 3rd Generation Partnership Project (3GPP).

[13] 3rd Generation Partnership Project (3GPP) 2016. *3GPP TS 35.206 V13.0.0 : Technical Specification; Technical Specification Group Services and System Aspects; 3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 2: Algorithm Specification (Release 13)*. 3rd Generation Partnership Project (3GPP).

[14] 3rd Generation Partnership Project (3GPP) 2016. *3GPP TS 35.231 V13.0.0 : Technical Specification; Technical Specification Group Services and System Aspects; Specification of the TUAK algorithm set: A second example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 1: Algorithm specification (Release 13)*. 3rd Generation Partnership Project (3GPP).

[15] Myrto Arapinis, Loretta Ilaria Mancini, Eike Ritter, and Mark Ryan. 2014. Privacy through Pseudonymity in Mobile Telephony Systems. In *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23–26, 2014*. The Internet Society. http://www.internetsociety.org/doc/privacy-through-pseudonymity-mobile-telephony-systems

[16] Myrto Arapinis, Loretta Ilaria Mancini, Eike Ritter, Mark Ryan, Nico Golde, Kevin Redon, and Ravishankar Borgaonkar. 2012. New privacy issues in mobile telephony: Fix and verification. In *The ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16–18, 2012*, Ting Yu, George Danezis, and Virgil D. Gligor (Eds.). ACM, 205–216. DOI:https://doi.org/10.1145/2382196.2382221

[17] Giuseppe Ateniese, Amir Herzberg, Hugo Krawczyk, and Gene Tsudik. 1999. Untraceable mobility or how to travel incognito. *Computer Networks* 31, 8 (1999), 871–884.

[18] Igor Bilogrevic, Murtuza Jadliwala, and Jean-Pierre Hubaux. 2010. Security Issues in Next Generation Mobile Networks: LTE and Femtocells. In 2nd International Femtocell Workshop, Luton, UK, June 21, 2010. (2010). Available at https://infoscience.epfl.ch/record/149153/files/secu-LTE-femtocells-BJH-final.pdf.

[19] Bruno Blanchet. 2001. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In *14th IEEE Computer Security Foundations Workshop (CSFW-14 2001), 11–13 June 2001, Cape Breton, Nova Scotia, Canada*. IEEE Computer Society, 82–96. DOI:https://doi.org/10.1109/CSFW.2001.930138

[20] Bruno Blanchet, Martín Abadi, and Cédric Fournet. 2008. Automated verification of selected equivalences for security protocols. *J. Log. Algebr. Program.* 75, 1 (2008), 3–51. DOI: https://doi.org/10.1016/j.jlap.2007.06.002

[21] Hiten Choudhury, Basav Roychoudhury, and Dilip K. Saikia. 2012. Enhancing User Identity Privacy in LTE. In *11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2012, Liverpool, United Kingdom, June 25–27, 2012*, Geyong Min, Yulei Wu, Lei (Chris) Liu, Xiaolong Jin, Stephen A. Jarvis, and Ahmed Yassin Al-Dubai (Eds.). IEEE Computer Society, 949–957. DOI:https://doi.org/10.1109/TrustCom.2012.148

[22] Christos K. Dimitriadis. 2007. Improving Mobile Core Network Security with Honeynets. *IEEE Security & Privacy* 5, 4 (2007), 40–47. DOI:https://doi.org/10.1109/MSP.2007.85

[23] Michael Dupré. 2004. Process to control a Subscriber Identity Module (SIM) in Mobile Phone System. US Patent Office, Available at https://www.google.com/patents/US6690930. (February 2004). US Patent 6690930 B1, Filing date — 25 May, 1999.

[24] Tobias Engel. 2008. Locating mobile phones using Signalling System 7. In 25th Chaos Communication Congress (25C3). (December 2008). Avaialable at https://events.ccc.de/congress/2008/Fahrplan/attachments/1262_25c3-locating-mobile-phones.pdf.

[25] Dan Forsberg, Günther Horn, Wolf-Dietrich Moeller, and Valtteri Niemi. 2010. *LTE Security*. John Wiley & Sons.

[26] Pierre-Alain Fouque, Cristina Onete, and Benjamin Richard. 2016. Achieving Better Privacy for the 3GPP AKA Protocol. *IACR Cryptology ePrint Archive* 2016 (2016), 480. http://eprint.iacr.org/2016/480

[27] Nico Golde, Kevin Redon, and Ravishankar Borgaonkar. 2012. Weaponizing Femtocells: The Effect of Rogue Devices on Mobile Telecommunications. In *19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, February 5–8, 2012*. The Internet Society. http://www.internetsociety.org/weaponizing-femtocells-effect-rogue-devices-mobile-telecommunications

[28] Dan Goodin. 2015. Low-cost IMSI catcher for 4G/LTE networks tracks phones precise locations. (October 2015). [Online] Available at https://arstechnica.com/civis/viewtopic.php?f=2&t=1298409.

[29] Chan-Kyu Han and Hyoung-Kee Choi. 2014. Security Analysis of Handover Key Management in 4G LTE/SAE Networks. *IEEE Trans. Mob. Comput.* 13, 2 (2014), 457–468. DOI:https://doi.org/10.1109/TMC.2012.242

[30] Amir Herzberg, Hugo Krawczyk, and Gene Tsudik. 1994. On Travelling Incognito. In *First Workshop on Mobile Computing Systems and Applications, WMCSA 1994, Santa Cruz, CA, USA, December 8–9, 1994*. IEEE Computer Society, 205–211. DOI:https://doi.org/10.1109/WMCSA.1994.29

[31] Mohammed Shafiul Alam Khan and Chris J. Mitchell. 2015. Improving Air Interface User Privacy in Mobile Telephony. In *Security Standardisation Research — Second International Conference, SSR 2015, Tokyo, Japan, December 15–16, 2015, Proceedings (Lecture Notes in Computer Science)*, Liqun Chen and Shin'ichiro Matsuo (Eds.), Vol. 9497. Springer, 165–184. DOI: https://doi.org/10.1007/978-3-319-27152-1_9

[32] Mohammed Shafiul Alam Khan and Chris J. Mitchell. 2016. Retrofitting Mutual Authentication to GSM Using RAND Hijacking. In *Security and Trust Management — 12th International Workshop, STM 2016, Heraklion, Crete, Greece, September 26–27, 2016, Proceedings (Lecture Notes in Computer Science)*, Gilles Barthe, Evangelos P. Markatos, and Pierangela Samarati (Eds.), Vol. 9871. Springer, 17–31. DOI: https://doi.org/10.1007/978-3-319-46598-2_2

[33] Geir M. Køien. 2013. Privacy enhanced mutual authentication in LTE. In *9th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob 2013, Lyon, France, October 7–9, 2013*. IEEE Computer Society, 614–621. DOI:https://doi.org/10.1109/WiMOB.2013.6673421

[34] Ming-Feng Lee, Nigel P. Smart, Bogdan Warinschi, and Gaven J. Watson. 2014. Anonymity guarantees of the UMTS/LTE authentication and connection protocol. *Int. J. Inf. Sec.* 13, 6 (2014), 513–527. DOI:https://doi.org/10.1007/s10207-014-0231-3

[35] Christopher Mitchell, Jason Brown, Liqun Chen, Dirk Goj, Dieter Gollmann, Yong-Fei Han, Nigel Jefferies, Michael Walker, and Dale Youngs. 1996. *LINK 3GS3 Technical Report 2: Security Mechanisms for Third Generation Systems*. Vodafone Ltd., GPT Ltd., and ISG, Royal Holloway, University of London. Available at http://www.chrismitchell.net/3GS3/TR2_pdf.ZIP.

[36] Karsten Nohl. 2014. Mobile Self-Defense. In 31st Chaos Communication Congress (31C3). (2014). Available at https://events.ccc.de/congress/2014/Fahrplan/system/attachments/2493/original/Mobile_Self_Defense-Karsten_Nohl-31C3-v1.pdf.

[37] Paulo S. Pagliusi. 2002. A Contemporary Foreword on GSM Security. In *Infrastructure Security, International Conference, InfraSec 2002, Bristol, UK, October 1–3, 2002, Proceedings (Lecture Notes in Computer Science)*, George I. Davida, Yair Frankel, and Owen Rees (Eds.), Vol. 2437. Springer, 129–144. DOI:https://doi.org/10.1007/3-540-45831-X_10

[38] Siddharth Prakash Rao, Silke Holtmanns, Ian Oliver, and Tuomas Aura. 2015. Unblocking Stolen Mobile Devices Using SS7-MAP Vulnerabilities: Exploiting the Relationship between IMEI and IMSI for EIR Access. In *2015 IEEE TrustCom/BigDataSE/ISPA, Helsinki, Finland, August 20-22, 2015, Volume 1*. IEEE, 1171–1176. DOI:https://doi.org/10.1109/Trustcom.2015.500

[39] Siddharth Prakash Rao, Ian Oliver, Silke Holtmanns, and Tuomas Aura. 2016. We know where you are!. In *8th International Conference on Cyber Conflict, CyCon 2016, Tallinn, Estonia, May 31 – June 3, 2016*, Nikolaos Pissanidis, Henry Roigas, and Matthijs Veenendaal (Eds.). IEEE, 277–293. DOI:https://doi.org/10.1109/CYCON.2016.7529440

[40] Didier Samfat and Refik Molva. 1994. A Method Providing Identity Privacy to Mobile Users During Authentication. In *First Workshop on Mobile Computing Systems and Applications, WMCSA 1994, Santa Cruz, CA, USA, December 8–9, 1994*. IEEE Computer Society, 196–199. DOI:https://doi.org/10.1109/WMCSA.1994.5

[41] Didier Samfat, Refik Molva, and N. Asokan. 1995. Untraceability in Mobile Networks. In *MOBICOM '95, Proceedings of the First Annual International Conference on Mobile Computing and Networking, Berkeley, CA, USA, November 13–15, 1995*, Baruch Awerbuch and Dan Duchamp (Eds.). ACM, 26–36. DOI: https://doi.org/10.1145/215530.215548

[42] Jeremy Scahill and Josh Begley. 2015. The great SIM heist—How spies stole the keys to the encryption castle. (February 2015). [Online] Available at https://theintercept.com/2015/02/19/great-simheist/.

[43] Ashkan Soltani and Craig Timberg. 2014. Tech firm tries to pull back curtain on surveillance efforts in Washington. (September 2014). [Online] Available at http://wapo.st/1qgzImt.

[44] Daehyun Strobel. 2007. *IMSI catcher*. Technical Report. Ruhr-Universität Bochum. Available at https://www.emsec.rub.de/media/crypto/attachments/files/2011/04/imsi_catcher.pdf.

[45] Keen Sung, Brian Neil Levine, and Marc Liberatore. 2014. Location Privacy without Carrier Cooperation. In *IEEE workshop on Mobile Security Technologies, MoST, San Jose, CA, USA, May 17, 2014*. IEEE.

[46] Chunyu Tang, David A. Naumann, and Susanne Wetzel. 2011. Symbolic Analysis for Security of Roaming Protocols in Mobile Networks [Extended Abstract]. In *Security and Privacy in Communication Networks — 7th International ICST Conference, SecureComm 2011, London, UK, September 7–9, 2011, Revised Selected Papers (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering)*, Muttukrishnan Rajarajan, Fred Piper, Haining Wang, and George Kesidis (Eds.), Vol. 96. Springer, 480–490. DOI: https://doi.org/10.1007/978-3-642-31909-9_29

[47] Telecommunication Standardisation Sector of ITU 2008. *ITU-T E.212—The international identification plan for public networks and subscriptions*. Telecommunication Standardisation Sector of ITU.

[48] The Aftenposten 2015. *Alt om mobilspionasje-saken*. The Aftenposten. [Online] Available at http://mm.aftenposten.no/mobilspionasje/.

[49] Joe-Kai Tsay and Stig Fr. Mjølsnes. 2012. A Vulnerability in the UMTS and LTE Authentication and Key Agreement Protocols. In *Computer Network Security — 6th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security, MMM-ACNS 2012, St. Petersburg, Russia, October 17-19, 2012. Proceedings (Lecture Notes in Computer Science)*, Igor V. Kotenko and Victor A. Skormin (Eds.), Vol. 7531. Springer, 65–76. DOI:https://doi.org/10.1007/978-3-642-33704-8_6

[50] Niemi Valtteri and Kaisa Nyberg. 2003. *UMTS Security*. John Wiley & Sons Limited.

[51] Fabian van den Broek, Roel Verdult, and Joeri de Ruiter. 2015. Defeating IMSI Catchers. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12–16, 2015*, Indrajit Ray, Ninghui Li, and Christopher Kruegel (Eds.). ACM, 340–351. DOI: https://doi.org/10.1145/2810103.2813615

[52] Vodafone. 2003. Cipher key separation for A/Gb security enhancements. 3GPP TSG SA WG3 Security. (July 15–18 2003). Document reference — S3-030463. Available at ftp://www.3gpp.org/tsg_sa/WG3_Security/TSGS3_29_SanFran/Docs/PDF/S3-030463.pdf.