Defeating IMSI Catchers

Fabian van den Broek
Institute for Computing and
Information Sciences
Radboud University Nijmegen,
The Netherlands.
f.vandenbroek@cs.ru.nl

Roel Verdult
Institute for Computing and
Information Sciences
Radboud University Nijmegen,
The Netherlands.
rverdult@cs.ru.nl

Joeri de Ruiter
School of Computer Science
University of Birmingham,
United Kingdom
j.deruiter@cs.bham.ac.uk

ABSTRACT

IMSI catching is a problem on all generations of mobile telecommunication networks, i.e., 2G (GSM, GPRS), 3G (HDSPA, EDGE, UMTS) and 4G (LTE, LTE+). Currently, the SIM card of a mobile phone has to reveal its identity over an insecure plaintext transmission, before encryption is enabled. This identifier (the IMSI) can be intercepted by adversaries that mount a passive or active attack. Such identity exposure attacks are commonly referred to as 'IMSI catching'. Since the IMSI is uniquely identifying, unauthorized exposure can lead to various location privacy attacks. We propose a solution, which essentially replaces the IMSIs with changing pseudonyms that are only identifiable by the home network of the SIM's own network provider. Consequently, these pseudonyms are unlinkable by intermediate network providers and malicious adversaries, and therefore mitigate both passive and active attacks, which we also formally verified using ProVerif. Our solution is compatible with the current specifications of the mobile standards and therefore requires no change in the infrastructure or any of the already massively deployed network equipment. The proposed method only requires limited changes to the SIM and the authentication server, both of which are under control of the user's network provider. Therefore, any individual (virtual) provider that distributes SIM cards and controls its own authentication server can deploy a more privacy friendly mobile network that is resilient against IMSI catching attacks.

Categories and Subject Descriptors

C.2.0 [**Computer-communication networks**]: General—*Security and protection*; K.4.1 [**Computers and society**]: Public Policy Issue—*Privacy*

Keywords

3GPP; IMSI catching; privacy

CCS'15, October 12–16, 2015, Denver, Colorado, USA.

ACM 978-1-4503-3832-5/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2810103.2813615

1. INTRODUCTION

In cellular technologies it is unavoidable that the mobile telecommunication providers (which we will refer to as simply providers for the rest of this paper) need to know the approximate location of their users[1], in order to be able to route incoming traffic to the mast currently nearest to the user. This can lead to serious (location) privacy issues for the users, as their approximate location is continuously monitored. However, if we accept this issue, for instance because we trust the providers, then there are still many other privacy issues related to mobile telephony, such as apps sharing private data [2] or eavesdropping phone conversations [3]. One of the first practical privacy attacks against mobile phones was the so-called *IMSI catching attack*, and this attack persists even in today's mobile network standards. This term refers to the unique identifier present in every SIM[2] card, called the IMSI for International Mobile Subscriber Identifier. This identifier is transmitted in plain-text over the wireless network as (initial) identification and can therefore easily be intercepted.

IMSI Catching

IMSI catching was one of the first practical attacks on GSM, leading to the development of devices called *IMSI catchers*, which gather all IMSIs that are active in a geographic area. An IMSI catcher can achieve this in two different ways: passive and active. The passive way is by simply observing the wireless traffic and storing all IMSIs observed. For the more effective active attack a fake base station is set up, to which cell phones in the neighborhood will attempt to connect. The fake base station then simply commands each phone to identify itself. This way IMSIs can be retrieved at any time, while with the passive attack the attacker has to wait for phones to send out their IMSI. IMSI catchers are commercially available, though they are expensive and usually sold restrictively to government officials. However, in recent years cheap and precise enough equipment has become available that can be used to create an (active) IMSI catcher. Likewise, cheap base stations called femtocells are commercially available, several of which have been rooted,

[1] Actually, there have been proposals for a TOR-like network among mobile phones, in order to obscure the location of users for the network [1], but such a scheme seems impractical with regard to the reliability we have come to expect from the mobile network.

[2] For sake of simplicity we use the term SIM to refer to both the SIM (2G) and USIM (3G and 4G) applications and the physical smart card containing them.

making them into very cheap (below $100,-) IMSI catchers [4].

Over time, the commercial IMSI catchers were extended with a lot of additional functionality such as eavesdropping on wireless calls. However, they are still, rather euphemistically, called IMSI catchers. This leads to a lot of confusion on what is meant by an IMSI catching attack. For this article, we refer only to the gathering of IMSI numbers from the air waves (either passive or active) as IMSI catching.

Recent news stories uncovered widespread use of unregulated IMSI catchers. In an article from The Washington Post researchers found 18 IMSI catchers in Washington D.C. within two days [5]. These IMSI catchers were present at airfields and embassies and could be detected since they actually ran active Man-in-the-Middle attacks, possibly to eavesdrop on mobile connections. So, traditional IMSI catchers, as discussed in this article, might be even more widespread. The FCC started an internal task force to study the use of IMSI catchers by criminals and foreign intelligence agencies [6], indicating that these attacks are considered a serious issue.

Location privacy

IMSI catching attacks mostly relate to the issue of location privacy, as the transmission of your IMSI reveals your approximate location. Location privacy attacks attempt to link an identity to a location. By keeping one of these (identity or location) fixed and trying to recover the other, we identify two different goals of location privacy attacks:

Retrieving identities at a location (monitoring) A list of caught IMSIs can reveal who came at what time within, for example, the near vicinity of a specific building, or was present at a certain rally. Such monitoring is also used for commercial goals such as customer monitoring. We have seen that shop keepers already collect WiFi signals from phones to determine statistics such as returning customers or the effectiveness of their shop front [7]. IMSI catching could be used not only to invade privacy but also for actual physical attacks. Consider, as (dramatic) example, automated terrorist attacks that trigger bombs to explode when high-value targets come in range of an IMSI catcher [8, 9].

Retrieving a person's location (tracking) Recovering a person's geographic movements can reveal a lot about what they do and who they meet. As IMSI catching requires the operation of rogue cell towers or passive listening antennas in the vicinity of the victim(s), it seems most useful for monitoring attacks. In GSM a cell tower can service an area up to 34 kilometers in diameter, so the vicinity is in the order of several kilometers. For actual tracking over a larger area, an attacker would need a mobile setup or a network of antennas. This is not unthinkable, for example the city of London initiated a project where trash cans monitor WiFi signals of mobile phones to profile people's behavior in order to send them targeted advertisements [10].

Traditionally, IMSI catching is associated with a more hybrid attack where police forces use IMSI catching to recover information about the mobile subscription of a target [11]. They follow a target and gather a list of active IMSI numbers in several unrelated crowds on independent locations and intersect the recovered sets of IMSI numbers. The legitimacy of this method is debatable, especially since they often transmit signals from fake cell-towers which interfere with genuine cell-towers. Furthermore, this technique seems to be used unregulated by several entities, such as intelligence agencies and malicious adversaries [5, 12].

Contribution

This paper demonstrates that the weak protection against unauthorized identification of SIM cards in mobile networks is not a necessity, as is often claimed. Furthermore, we propose a solution which defeats the IMSI catching attacks and increases the credibility of the mutual authentication with the home network. The latter has significant impact against man-in-the-middle attacks such as presented in [13], and provides additional security to 2G networks which currently only support a unilateral authentication procedure. Our solution was formally verified using ProVerif and does not interfere with the workings of the networks as they are defined today, and is in fact backwards compatible with current implementations. The only party that would need to make a change would be the mobile providers, as they provision the user with an IMSI and the SIM that contains it and are also the only party in control of the authentication server, where the IMSIs are linked to the authentication parameters (keys and algorithms). Every provider can independently decide whether to implement this solution, and as our solution changes nothing in the message definitions a change happens transparently for any intermediate providers.

Overview

We continue with some background information about mobile networks in Section 2. This section not only discusses identification in mobile networks, but also looks at authentication, since our solution requires a change of a parameter of one of the authentication messages. Section 3 describes our solution against IMSI catching for the current technology and provides the general idea of our solution. Section 4 describes our solution modified for the older, but still heavily used 2G technology. We formally verify our solution in Section 5, as well as analyze the effectiveness and consequences of implementing it. Finally, we review related work in Section 6 and draw conclusions in Section 7.

2. BACKGROUND

IMSI catching is an issue in what is often called the 3GPP or GSM family of cellular technology; world-wide this is by-far the most popular of the mobile telecommunication systems. This family of algorithms started in the early 90s with the introduction of GSM, which was then 2nd generation mobile technology. Currently, we are seeing the deployment of the 4G networks (LTE and LTE+). Within each generation there have been incremental improvements mostly to the up- and down-link speed (e.g. HSDPA+ over UMTS in 3G), but the protocols themselves only receive real changes with the move to a newer generation. As previously stated, the IMSI catching problem exists in all generations. This following sections give some background into specifics of the 3GPP networks that are relevant for the IMSI catching dilemma. In Section 2.4 we describe 3G authentication in more detail, both as an example and because we use the authentication messages in our solution.

2.1 Identification within 3GPP networks

Cell towers in mobile networks identify themselves by broadcasting identifiers. Mobile phones pick up these signals and decide whether to connect to a network or not. This decision is based on data from the SIM, which instructs the phone to look out for certain networks by both frequency and identifier. When a mobile phone connects to a network, it first requests a channel to exchange information on with the cell tower. On this channel the cell tower can always request the SIM's identity. Identification is performed after a simple command from the cell tower to a mobile phone. This command, *Identity request*, specifies a specific identifier (IMSI, TMSI, IMEI or IMEI(SV), see Section 2.2), to which the phone responds with a so-called *Identity response* containing the requested identifier [14, 15]. Authentication can only take place after identification, because the authentication is based on a symmetric key shared between the SIM and provider.

Interestingly, the specifications of the mobile standards acknowledge the problems of IMSI catching. In [16] several security goals for mobile networks are stated, among which are confidentiality of the IMSI (user identity confidentiality), user location confidentiality and user untraceability (Section 5.1.1 of [16]). The same document acknowledges the breach of user identity confidentiality introduced with the request identification message (Section 6.2 of [16]), though no breach of the location privacy issues is mentioned here. The specifications further mandate that a SIM does not answer Identity request messages asking for any identifier, other then the IMSI, when no encryption context is yet established (Section 4.4.4.2 of [15]). This, of course, would not prevent IMSI catching, but does prevent the leaking of the other identifiers to IMSI catchers. However, our experiments show that all of the current 3G or 4G enabled phones and SIM cards we tested also transmit the TMSI and IMEI unprotected when requested.

2.2 3GPP identifiers

While this article is mostly concerned with protecting the IMSI, many more identifiers exist in the 3GPP networks. We discuss the most important of these briefly below.

International Mobile Subscriber Identifier (IMSI)
The IMSI is the main identifier in 3GPP networks and belongs to one specific SIM card. It is a 15 digit number where the first three digits identify the home country (MCC, Mobile Country Code), the following two or three digits identify the home network (MNC, Mobile Network Code). The remaining nine or ten digits identify the specific user/SIM within the provider's database.

Temporary Mobile Subscriber Identifier (TMSI)
The TMSI is introduced to protect against traceability of users. The TMSI is a temporary pseudonym provided to the mobile device by the network, to use instead of the IMSI, essentially masking the IMSI against passive attacks. The TMSI is only valid within a certain geographical area. When a mobile phone moves to another area it initiates a location update procedure, which should provide it with a new TMSI. The time a single TMSI remains valid is configurable by the serving network. Since all communication with the mobile phone should be based on the TMSI[3], phones are traceable via the TMSI during a validity period. TMSIs do not provide adequate protection against IMSI catching attacks though, since a cell tower can always request a phone's IMSI. TMSIs are therefore easily defeated by active IMSI catching attacks. Furthermore, research shows that in practice TMSIs remain valid for far too long and are re-used over different areas [17], making them even usable in passive IMSI catching attacks.

International Mobile Equipment Identifier (IMEI)
The IMEI is a 15 digit number that identifies the mobile device itself. It is included to make black-listing of stolen phones possible. There is a closely related alternative to the IMEI, often referred to as IMEI(SV), which is one digit longer and also identifies the software version running on the phone.

Other identifiers
There are several other ways to identify a mobile device based on its transmissions, for instance the phone's answer to authentication requests. Because these requests are answered based on a shared secret key, the same challenge always invokes the same response. There are also several ways to identify a mobile device outside of the 3GPP protocols. Examples of these include the MAC address of the WiFi or Bluetooth adapter.

2.3 Authentication within 3GPP networks

For all 3GPP systems the customer's SIM card shares a (set of) secret key(s) with the authentication server of his provider. Any authentication and encryption of messages is performed with temporary keys derived from these shared secret keys. The link between a SIM's unique identifier (IMSI) and its secret keys is made either through a diversified key solution or a simple look-up table. Once the SIM has been identified, the network can look-up the accompanying secret key and initiate authentication.

It is important to note that the party authenticating the mobile device does not need to be the user's own provider. A mobile device can be 'roaming,' i.e. using the network of another provider. We call the network with which the mobile device is currently connected the *serving network*, and the network of the user's provider the *home network* (see Figure 1).

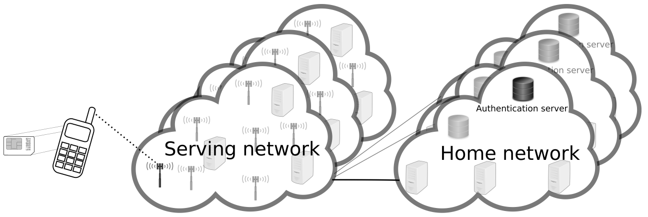

Figure 1: Schematic overview of the division between the serving network and the home network.

The serving network starts a challenge-response protocol with the SIM, which authenticates the SIM and establishes (a) sessions key(s). When using 3G or 4G technologies, the challenge sent by the serving network also authenticates the home network, for 2G technologies there is only SIM-only

[3] Phones can also still be paged using their IMSI numbers, and listening on paging channels shows that this occurs frequently.

authentication. Since only the authentication server within the home network and the SIM know the secret key, the serving network is not capable of authenticating (to) the SIM. Obviously, it would be unacceptable for the different providers to share the secret keys among each other. Therefore, all 3GPP protocols have the serving network contact the SIM's home network to request authentication parameters for the IMSI. This is possible because part of the IMSI identifies the home network. The authentication parameters contain a challenge, the associated response, resulting session key(s) and, in case of 3G and 4G technology, an authentication token. This authentication token is used for the SIM to be able to authenticate the home network and is essentially a sequence number shared between the SIM and home network and a MAC over this sequence number and the challenge. The serving network forwards the challenge and authentication token to the SIM, which verifies the validity of the authentication token and responds to the challenge. The SIM also uses the random challenge to compute the session key(s) and forwards these to the phone. The serving network compares the response from the SIM with the response from the home network. If they are equal, then the SIM is authenticated and both sides of the wireless interface share a (set of) session key(s). In Section 2.4 the authentication protocol for the 3G technology is discussed in more detail.

This set-up leads to an interesting a-symmetry in the authentication, whereby the *serving* network authenticates the SIM and (in case of 3G and 4G) the SIM authenticates the *home* network. It is also curious that for establishing the session key(s), no input or freshness of the SIM is required.

2.4 3G Authentication and key establishment

Since our solution against IMSI catching requires a change in the authentication procedure we will now take a detailed look at authentication in 3G networks. This so-called Authentication and Key Agreement (AKA) protocol provides both mutual authentication, between SIM and home network, and establishes session keys. The AKA protocol for 4G is almost identical, with only one additional parameter used to diversify the session keys. A difference that has no impact for this paper.[4] This AKA protocol is defined in [16] and we present an overview in Figure 2. The bit lengths of important variables are summarized in Table 1.

To be more precise we first introduce the variables and functions used in this paper to formalize the authentication protocol. The set of all available $IMSI$ numbers is denoted by I and is available to the home network of the provider. The home network stores the properties of a SIM card, $IMSI$ (i), secret key ($\mathcal{K}$) and sequence number (SQN), as a tuple $s = \langle i, \mathcal{K}, SQN \rangle$ for each $i \in I$ in the set of all available SIM cards S. To simplify the notation, we use subscript on a tuple to denote a single element from the corresponding tuple, e.g. s_i denotes the $IMSI$ number of the SIM card s. Encryption with key k is denoted by $E_k()$ and decryption by $E_k^{-1}()$. Consequently, generation of a MAC with key k is denoted by $M_k()$. We do not specify a specific algorithm, but several standardized cryptographic primitives and methods are suitable for encryption [18, 19, 20] and generating MACs [21, 22, 23]. The AKA protocol relies on five encryption functions referred to as f_1 to f_5. The implementation of these functions is provider-specific and not fully standardized. However, the standard that defines the security architecture [16] suggest that the provider may use the example algorithm implementation set, defined in [24], for authentication and key generation. This implementation set is based on Rijndael, combined with different provider codes that get XORed with the random challenge, for each of the five algorithms. We assume in this paper that the operator uses functions with similar characteristics as those proposed in [24].

The SIM has at some point identified itself before the authentication starts. The serving network can then request authentication parameters from the home network. The home network computes the authentication parameters, which consist of a freshly generated random which acts as a challenge ($RAND$), the corresponding response ($SRES$), confidentiality key (CK), integrity key (IK), anonymity key (AK) and an authorization token ($AUTN$). The AUTN token, is the authorization proof by the home network. It consists of the sequence number XORed with the anonymity key, the Authentication Management Field (AMF) and a MAC over SQN, AMF and $RAND$. The sequence number protects against re-play attacks, the AMF is for the provider to use, for instance to signal a specific algorithm suite, or set a time validity for a key and the MAC authenticates this message as coming from the home network. The authentication parameters are then transmitted to the serving network. The challenge $RAND$ and the AUTN token are forwarded to the SIM.

The SIM, upon reception of an authentication request, first retrieves the sequence number ($XSQN$), by computing the anonymity key and XORing this with the sequence number from the $AUTN$ token. Then the SIM verifies the MAC over the authentication token. If the authentication token proves genuine, the SIM verifies that the sequence number from the network ($XSQN$) is higher than its own sequence number (SQN). If the received sequence number is lower, or too high, then the SIM responds with an error message and a genuine network will then start a re-synchronization setup. By how much the sequence numbers can deviate from each other is a setting chosen by the home network.

If the sequence number falls within the range of allowed sequence numbers, the SIM computes the confidentiality key, integrity key and response. The response is transmitted back to the serving network and the two keys are stored in the phone. The serving network compares the response of the SIM ($SRES$) with the response of the home network ($XRES$), and when found correct, the network can order the use of integrity protection and ciphering.

The AKA protocol had been formally verified using enhanced BAN logic and shown to provide both authentication and confidentiality [26]. However, using ProVerif, a location privacy attack was found by Arapinis et al. [27]. Replay attacks of the authentication token are prevented by the sequence number, but replaying this token will break location privacy, as the SIM responds differently to an out of sync message than to an incorrect MAC message. Of course, IMSI catching is an even simpler way of breaking location privacy.

[4] In 4G AKA the serving network is not trusted to verify the response to the challenge, as opposed to 2G and 3G. However, this difference also has no influence on the rest of the paper, as the serving network still requests the IMSI to find the home network.

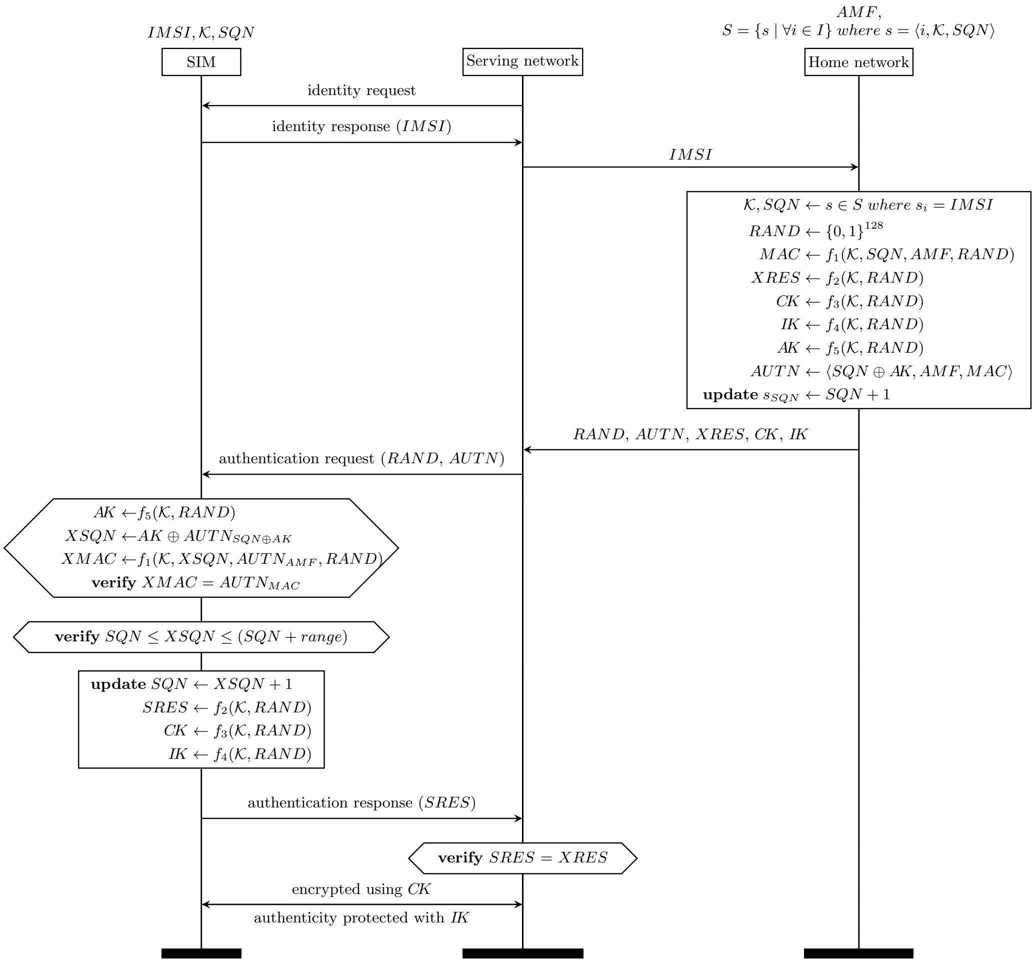
$IMSI, \mathcal{K}, SQN$
$AMF,$
$S = \{s \mid \forall i \in I\}$ where $s = \langle i, \mathcal{K}, SQN \rangle$
SIM
Serving network
Home network
identity request
identity response ($IMSI$)
$IMSI$
$\mathcal{K}, SQN \leftarrow s \in S$ where $s_i = IMSI$
$RAND \leftarrow \{0,1\}^{128}$
$MAC \leftarrow f_1(\mathcal{K}, SQN, AMF, RAND)$
$XRES \leftarrow f_2(\mathcal{K}, RAND)$
$CK \leftarrow f_3(\mathcal{K}, RAND)$
$IK \leftarrow f_4(\mathcal{K}, RAND)$
$AK \leftarrow f_5(\mathcal{K}, RAND)$
$AUTN \leftarrow \langle SQN \oplus AK, AMF, MAC \rangle$
update $s_{SQN} \leftarrow SQN + 1$
$RAND$, $AUTN$, $XRES$, CK, IK
authentication request ($RAND$, $AUTN$)
$AK \leftarrow f_5(\mathcal{K}, RAND)$
$XSQN \leftarrow AK \oplus AUTN_{SQN \oplus AK}$
$XMAC \leftarrow f_1(\mathcal{K}, XSQN, AUTN_{AMF}, RAND)$
verify $XMAC = AUTN_{MAC}$
verify $SQN \leq XSQN \leq (SQN + range)$
update $SQN \leftarrow XSQN + 1$
$SRES \leftarrow f_2(\mathcal{K}, RAND)$
$CK \leftarrow f_3(\mathcal{K}, RAND)$
$IK \leftarrow f_4(\mathcal{K}, RAND)$
authentication response ($SRES$)
verify $SRES = XRES$
encrypted using CK
authenticity protected with IK

Figure 2: A schematic representation of successful SIM authentication in 3G networks [25]

Table 1: Bit length overview, starred lines are newly added in the presented solution.

variable	bit length
$IMSI$	60 (bcd encoded)
$PMSI$	34 $\star$
$\mathcal{K}$	64 (2G) / 128 (3 and 4G)
$RAND$	128
SQN	62 $\star$ (2G) / 48 (3 and 4G)
MAC	64 (3 and 4G)
M_κ	32 $\star$ (2G)

3. SOLUTION FOR 3G/4G

The underlying weakness enabling IMSI catching attacks is that the authentication is based on symmetric cryptography. The use of a shared secret key, means a SIM has to be identified before it can be authenticated. Identification prior to (mutual) authentication is a problem that crops up in other systems as well, such as in e-passports [28] and RFID tags [29]. However, common solutions to this issue do not help for the IMSI catching case. For instance randomizing the IMSI is no solution, because the IMSI needs to be identifying for the provider in order to provide cellular service. Other solutions, such as encrypting the identifier with the public key of the home network, would require changes in the messages between the phone and network, as the resulting ciphertext would not fit inside the currently defined identity response messages. Furthermore, some additional randomness would need to be added to every encryption to ensure that the SIM does not simply use another long term identifier (the encryption of the IMSI) instead of the IMSI. The space for the IMSI in identity response messages leaves too little room to add the randomness to the encryption without changing the message size. Since it is unrealistic to expect such changes to core message sizes being implemented in the current mobile phone technology, we present a solution that works within the current implementations.

We propose a solution where the IMSI is replaced with a changing pseudonym that only the SIM's home network can link to the SIM's identity. This hiding of the IMSI is done without changing any of the system messages, thus making it transparent to the serving network. This allows our solution to be deployed by providers, on an individual basis, on top of the currently available 3GPP networks.

During authentication, the authentication server supplies the user's SIM with a random new IMSI, which we refer to as *Pseudo Mobile Subscriber Identifier* (PMSI). The SIM uses the new PMSI the next time it is requested to reveal its IMSI. As discussed in Section 2.4, the user's provider op-

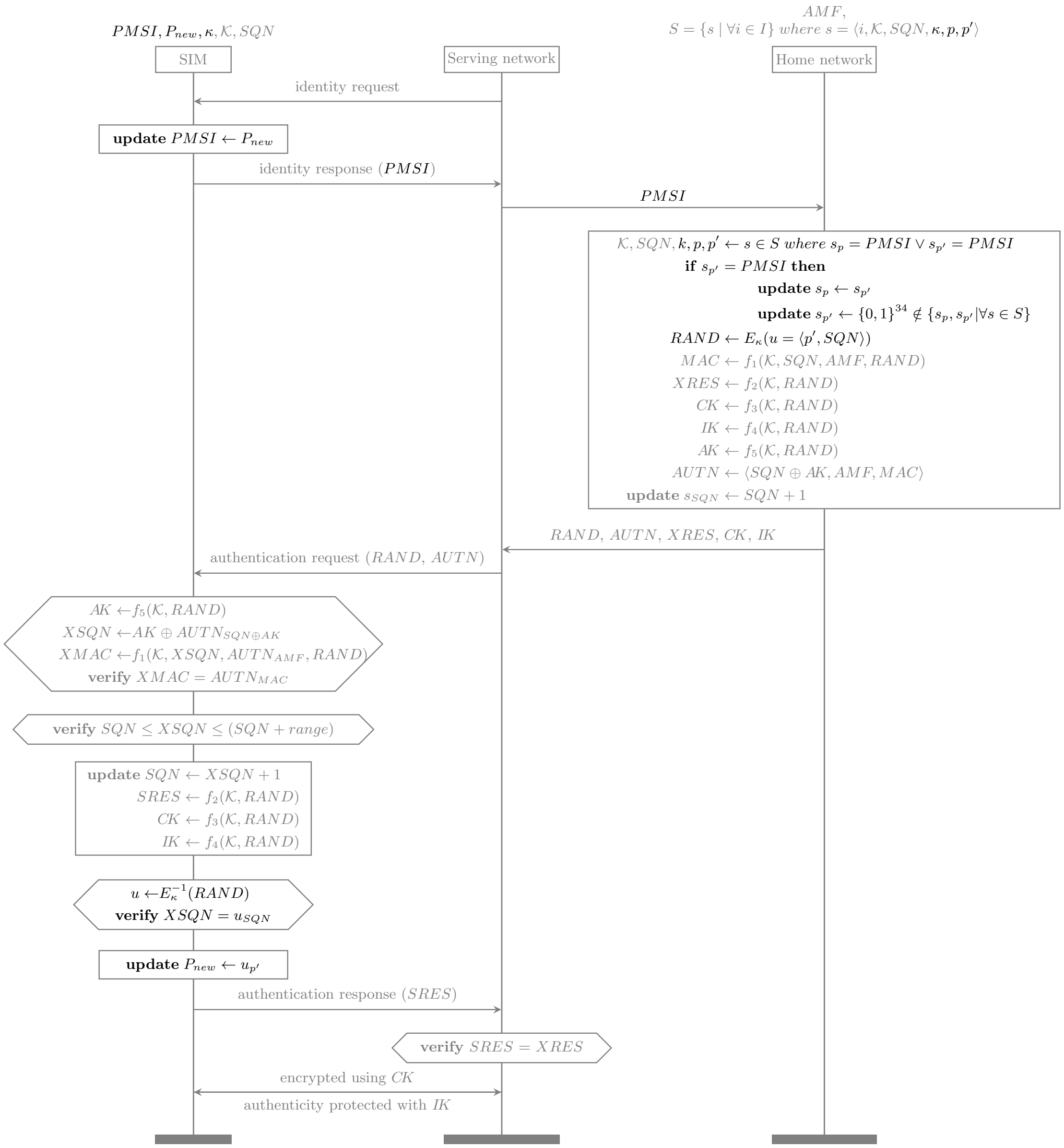
$PMSI, P_{new}, \kappa, \mathcal{K}, SQN$
$AMF,$
$S = \{s \mid \forall i \in I\}\ where\ s = \langle i, \mathcal{K}, SQN, \kappa, p, p' \rangle$
SIM
Serving network
Home network
identity request
update $PMSI \leftarrow P_{new}$
identity response ($PMSI$)
$PMSI$
$\mathcal{K}, SQN, k, p, p' \leftarrow s \in S\ where\ s_p = PMSI \vee s_{p'} = PMSI$
if $s_{p'} = PMSI$ **then**
update $s_p \leftarrow s_{p'}$
update $s_{p'} \leftarrow \{0,1\}^{34} \notin \{s_p, s_{p'} | \forall s \in S\}$
$RAND \leftarrow E_\kappa(u = \langle p', SQN \rangle)$
$MAC \leftarrow f_1(\mathcal{K}, SQN, AMF, RAND)$
$XRES \leftarrow f_2(\mathcal{K}, RAND)$
$CK \leftarrow f_3(\mathcal{K}, RAND)$
$IK \leftarrow f_4(\mathcal{K}, RAND)$
$AK \leftarrow f_5(\mathcal{K}, RAND)$
$AUTN \leftarrow \langle SQN \oplus AK, AMF, MAC \rangle$
update $s_{SQN} \leftarrow SQN + 1$
$RAND$, $AUTN$, $XRES$, CK, IK
authentication request ($RAND$, $AUTN$)
$AK \leftarrow f_5(\mathcal{K}, RAND)$
$XSQN \leftarrow AK \oplus AUTN_{SQN \oplus AK}$
$XMAC \leftarrow f_1(\mathcal{K}, XSQN, AUTN_{AMF}, RAND)$
verify $XMAC = AUTN_{MAC}$
verify $SQN \leq XSQN \leq (SQN + range)$
update $SQN \leftarrow XSQN + 1$
$SRES \leftarrow f_2(\mathcal{K}, RAND)$
$CK \leftarrow f_3(\mathcal{K}, RAND)$
$IK \leftarrow f_4(\mathcal{K}, RAND)$
$u \leftarrow E_\kappa^{-1}(RAND)$
verify $XSQN = u_{SQN}$
update $P_{new} \leftarrow u_{p'}$
authentication response ($SRES$)
verify $SRES = XRES$
encrypted using CK
authenticity protected with IK

Figure 3: Solution proposed for 3G and 4G compatible authentication protocols. The black text shows our additions to the standard protocols.

erates an authentication server which generates a random challenge-response pair and the corresponding session keys. We propose to use the random challenge ($RAND$) to provide the SIM with the PMSI. The PMSI has to be encrypted in a semantically secure way, in order to keep it confidential between SIM and authentication server. The resulting ciphertext should be sufficiently random and unpredictable to still serve as the challenge.

The changes we make in the authentication protocol for 3G are illustrated in Figure 3, though these changes can be applied the same way to the authentication in the other generations of mobile networks. Comparing Figures 2 and 3 shows that no changes are made in the messages that are transmitted, but only in the end points, i.e. the SIM and home network. This makes our solution compatible with all current implementations of the different generations of mobile networks. The changes needed for the authentication protocol used for 3G and 4G are discussed in detail in the next section, the changes for the 2G authentication protocol are discussed in Section 4.

3.1 Authentication server

In our solution, authentication servers have to be extended to store three additional values for each SIM: the new shared secret key κ and the two PMSI values p and p'. Here p is used to store the PMSI value the SIM s is currently using and p' stores the new PMSI value that the authentication server designates as the successor PMSI for that SIM. A provider implementing this solution would change the normal routine of its authentication server, when composing an authentication request for PMSI as follows:

1. *Validate if PMSI is known by the home network*
$$\exists s \in S,\ s_p = PMSI \vee s_{p'} = PMSI$$
2. *Update the PMSI when the successor $s_{p'}$ was used*
```
if $s_{p'} = PMSI$ then
    $s_p \leftarrow s_{p'}$
    $s_{p'} \leftarrow \{0,1\}^{34} \notin \{s_p, s_{p'} | \forall s \in S\}$
```
3. *Compute challenge $RAND$ by encrypting $s_{p'}$ and SQN*
$$RAND \leftarrow E_\kappa(u), \textbf{ where } u = \langle s_{p'}, SQN\rangle$$
4. *Compute other authentication parameters: MAC, $XRES$, CK, IK, AK and $AUTN$*
5. *Increase sequence number SQN and update $s \in S$*
$$s_{SQN} \leftarrow SQN + 1$$
6. *Transmit authentication parameters to serving network*

Steps 1 and 2 are new for our solution, while the computation of $RAND$ in step 3, was changed from generating a random number in the standard procedure. The other steps remain unchanged. The check in steps 1 and 2 has to be done for each message parameterized with an PMSI arriving at the authentication server, such as the location update message[5], essentially using this as confirmation that the SIM card is now using the pending PMSI $s_{p'}$ as replacement for the previous PMSI s_p. Because of efficiency, serving networks often request multiple authentication parameters for a SIM at the same time. So, to keep a check on the number of pending PMSIs, the authentication server will keep sending the same new PMSI number (with a different sequence number) to the SIM, as long as a SIM identifies itself with the same PMSI. This means the authentication server needs to keep a running record of at most two PMSI-numbers per SIM card.

3.2 SIM card

The SIM is extended to store the new shared secret key κ next to two PMSIs; the currently active PMSI ($PMSI$) and the future PMSI (P_{new}). Upon receiving the challenge, after normal verification steps (e.g. verifying the MAC and the sequence number) a SIM can decrypt the challenge and verify if the sequence number from the decrypted challenge (u_{SQN}) is equal to the sequence number in the authentication token (SQN). If so, the SIM retrieves the new PMSI (P_{new}). Any future IMSI identification request can then be answered with the newly received PMSI, instead of the previous PMSI by updating $PMSI$ with P_{new}. It is possible for $PMSI$ and P_{new} to temporarily have the same value, if $PMSI$ is updated to its successor (P_{new}) and no new PMSI has yet been received.

Additionally, the SIM could have a policy which determines the amount of time it will wait for a new identity request, so it can refresh its PMSI, before forcing a refresh itself. Forcing a PMSI update is performed by, for instance, signing on to the serving network as a freshly arrived SIM with the new PMSI.

To support our solution a SIM card's handling of authentication requests should be changed to include steps 5 and 6 in the following procedure:

1. *Perform existing authentication steps to recover $XSQN$*
2. *Use existing authentication procedure to verify MAC*
3. *Verify sequence number and update SQN*
$$SQN \leftarrow XSQN + 1$$
4. *Compute CK, IK and SRES*
5. *Decrypt $RAND$ and verify sequence number u_{SQN}*
$$u \leftarrow E_\kappa^{-1}(RAND)$$
$$\textbf{verify } XSQN = u_{SQN}$$
6. *Update the future $PMSI$ P_{new} to the supplied successor $u_{p'}$*
$$P_{new} \leftarrow u_{p'}$$

4. SOLUTION FOR 2G

In 2G authentication, the SIM authenticates itself to the network, but the network does not authenticate itself to the SIM. Figure 4 shows the 2G AKA protocol, with our changes to hinder IMSI catching highlighted. These changes are primarily meant to prevent IMSI catching attacks by supplying the SIM with new PMSI numbers through the challenge, same as before. However, as a side effect our solution gives the SIM the capabilities to verify if a challenge presented to him actually came from his home network.

Since standard 2G authentication has no sequence number, this has to be added for our solution. Compared to 3G and 4G, the set of acceptable SQN values for the SIM has to be much larger as there is no separate check on the SQN before the PMSI is retrieved from the challenge. Furthermore, in 3G/4G their exist protocols to sync SQN when the SIM and home network get out of sync. In the 2G environment we do not have the room to implement a syncing protocol within the current specifications. An attacker could therefore attempt to let the network issue many different challenges for the same SIM, thereby increasing its SQN value, hoping to get the network out of sync with the SIM. The SIM thus has to accept a much larger set of SQN values, e.g. every u_{SQN} value higher than its current value for SQN, which still prevents replay attacks.

Since in 2G there is no guarantee that the challenge presented to the SIM is authentic, our solution requires an additional integrity check on the encrypted PMSI. Otherwise, an attacker could act as a base station and transmit a random number as a challenge. In turn the SIM would decrypt this, and if the SQN value is higher than the current value, it would accept the first part of the decoded challenge as the new PMSI and the SIM will start to identify itself with a number unknown to the home network.

A cryptographic MAC (M_κ), computed by the home network, counters a Denial-of-Service attack which aims to desynchronize the SQN numbers known by the SIM and the home network. Such an authenticity check makes it very difficult

[5] Technically the location update is directed to a logically separate entity: the Home Location Register. However in practice this entity is always combined with the authentication server.

for an attacker to forge a valid *authentication request* without knowledge of the secret key κ. This essentially introduces an authentication of the home network to the SIM card, which is not available in the default 2G authentication protocol.

5. ANALYSIS

The previous sections presented our solution against IMSI catching, in this section we analyze the effect of the proposed solution. It provides new pseudonyms to the SIM to be used instead of the IMSI. The pseudonyms are provided in a confidential manner. An attacker, either active or passive, is unable to learn said pseudonym before it is used, as long as the attacker does not know the secret key κ. This provides unlinkability between consecutive pseudonyms. Furthermore, this protocol changes nothing in the messages as they are currently defined for 3GPP mobile telephony. The change is transparent for the serving network and the challenge used to transmit the PMSI should still be random due to the encryption and fresh due to the increasing sequence number.

In the case of 2G there is an extra benefit to our approach, as it adds a message authentication to the challenge. This does not prevent a Man-in-the-Middle attack, whereby an attacker simply passes on the challenge (though without learning the PMSI it contains), but it does prevent the replaying of challenges or the insertion of false challenges. Essentially achieving the same level of network authentication as in the standard 3G and 4G AKA algorithms.

The presented approach does not completely remove IMSI catching as an attack. After all, a SIM receives a new PMSI when authenticating, and will only start using it on the next identity request message. So, after switching to a new PMSI, a SIM will keep using that same PMSI for some time. Which means traceability remains until the switch to a new PMSI is made. In practice though, this remaining traceability mostly coincides with the already existing traceability of the TMSI. Our solution even alleviates the problems of TMSIs – too long validity periods over multiple geographical areas – as switching PMSIs, will automatically refresh the TMSI, since the SIM will then appear as a new SIM to the serving network.

The country and home network code of the IMSI will need to remain intact for our PMSI numbers, because these are needed to route messages to the user's home network. This means, that even using our PMSI pseudonyms, there are still some privacy issues, since a SIM will still reveal its home country and home network, when transmitting their PMSI. In essence the use of a PMSI provides k-anonymity to users [30], where k is the size of the group of expected users in this geographic area who have the same home country and home network. In most countries there are only a small number of mobile providers operational, they will mostly have fairly large consumer bases and thus a large k can be expected [31]. In other words, this solution provides stronger anonymity to users from a provider with a large consumer base, within their home country. If those users use their phone abroad, their traceability will likely increase dramatically, as there will be few other users transmitting an IMSI starting with the same country code.

The sequence number is added to the encryption for three reasons: (I) it provides semantically secure encryption, which is important because the same PMSI can be encrypted into several challenges, (II) it prevents replay attacks, and (III) it provides a way for the SIM to check the integrity of the decrypted data (in 3G and 4G), effectively preventing fake challenge attacks.

The use of the sequence number SQN suggests the possibility of the SIM and authentication server getting out-of-sync. However, in the 3G and 4G case the sequence number is already verified before the decryption of the challenge. Furthermore, there is a protocol already in place to re-sync the sequence number. In the 2G case the sequence number is only needed to prevent replay attacks, so the set of acceptable sequence numbers can be big enough to assure that the home network's SQN does not get out-of-sync with the SIM's. An attacker could attempt to start many fake sign-on sessions with a victim's current PMSI in a different cell, to force the home networks SQN out-of-sync with the SIM. However, this can be detected by the home network (many incoming requests for authentication parameters without a location update following it), so this can be counteracted by delaying the handing out of authentication parameters, when such an attack is detected.

In the original protocol for 2G networks no sequence numbers are used during authentication. Even worse, there is no mutual authentication in 2G networks, which means an active attacker can simply insert authentication challenges for the phone. Our approach has the added benefit of preventing such attacks, as both replay attacks on challenges or insertion of fake challenges can be detected (due to the sequence number and MAC).

As it is not possible for an attacker to create a correct PMSI update without the secret key κ, and the sequence numbers cannot be forced out-of-sync, there is no increased risk for a Denial-of-Service attack. A DoS attack could still be used to prevent the SIM from getting new PMSIs. However, this is essentially the same as simply preventing all service to the user, and as soon as the SIM connects to a genuine network, this protocol provides it with a new PMSI.

When the secret key κ is compromised, an adversary is able to track users in the future. Moreover, our method does not provide perfect forward secrecy [32] and lacks protection against analysis of historical recordings of previous $PMSI$ updates if the secret key κ is known. However, recovery of key κ requires considerable computational power, which can most-likely better be utilized to attack other cryptographic primitives used in 2G and 3G protocols.

Even with our solution in place, users might still have to take additional measures to make tracking harder, e.g. by disabling their WiFi and Bluetooth services, as these are also uniquely identifying.

Finally, our approach only protects from IMSI catching and not eavesdropping or Man-in-the-Middle attacks, which are also often referred to as IMSI catching. A powerful adversary might be able to forward and relay messages between the victim's phone and the genuine home network, while mounting various well-known cryptographic attacks on 2G networks [3, 33, 34, 35, 36]. In such scenario, we consider the security to be compromised since all 2G communication traffic can be observed, and location privacy attacks can be re-introduced through identification based on the contents of data transmissions. However, our solution hinders these attacks and others, such as sending malicious messages over

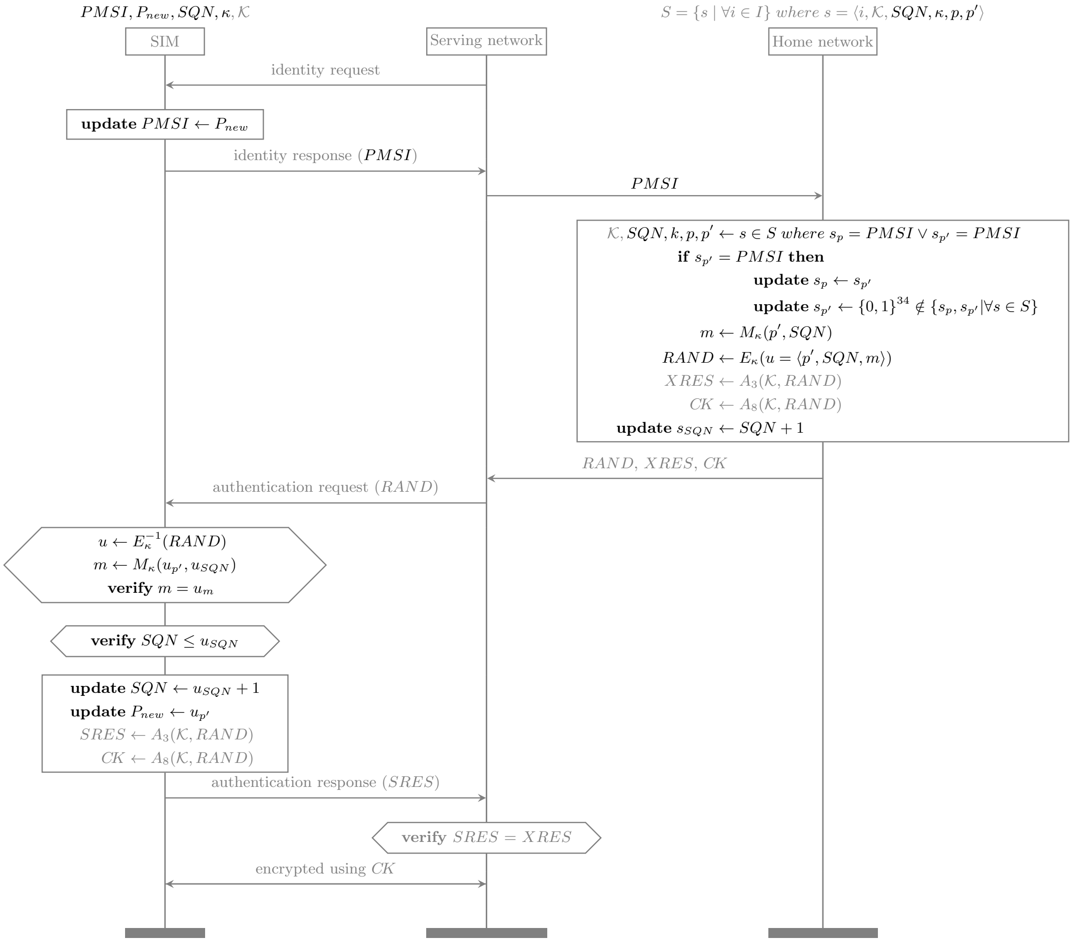
$PMSI, P_{new}, SQN, \kappa, \mathcal{K}$
$S = \{s \mid \forall i \in I\}\ where\ s = \langle i, \mathcal{K}, SQN, \kappa, p, p' \rangle$
SIM
Serving network
Home network
identity request
update $PMSI \leftarrow P_{new}$
identity response ($PMSI$)
$PMSI$
$\mathcal{K}, SQN, k, p, p' \leftarrow s \in S\ where\ s_p = PMSI \vee s_{p'} = PMSI$
if $s_{p'} = PMSI$ **then**
update $s_p \leftarrow s_{p'}$
update $s_{p'} \leftarrow \{0,1\}^{34} \notin \{s_p, s_{p'} | \forall s \in S\}$
$m \leftarrow M_\kappa(p', SQN)$
$RAND \leftarrow E_\kappa(u = \langle p', SQN, m \rangle)$
$XRES \leftarrow A_3(\mathcal{K}, RAND)$
$CK \leftarrow A_8(\mathcal{K}, RAND)$
update $s_{SQN} \leftarrow SQN + 1$
$RAND$, $XRES$, CK
authentication request ($RAND$)
$u \leftarrow E_\kappa^{-1}(RAND)$
$m \leftarrow M_\kappa(u_{p'}, u_{SQN})$
verify $m = u_m$
verify $SQN \leq u_{SQN}$
update $SQN \leftarrow u_{SQN} + 1$
update $P_{new} \leftarrow u_{p'}$
$SRES \leftarrow A_3(\mathcal{K}, RAND)$
$CK \leftarrow A_8(\mathcal{K}, RAND)$
authentication response ($SRES$)
verify $SRES = XRES$
encrypted using CK

Figure 4: Solution proposed for 2G compatible authentication protocols. The black text shows our additions to the standard protocols.

the air, as these can no longer rely on the IMSI to identify their target.

5.1 Parameter choice

As was explained in Section 2.2 the IMSI is a 15 digit number, containing a three digit country code and a maximum of three digits for the home network code. This means there can be at most 10^{10} different IMSIs per provider. Therefore, we need at most 34 bits for the PMSI in a challenge. Phones send the IMSI in BCD encoding (4 bits per digit), but there is no reason to encode a PMSI encrypted inside the $RAND$ in such an inefficient way. As the challenge is 16 bytes for all generations of mobile networks, this leaves us 94 bits to use for the counter SQN, which is 48 bit, in the case of 3G/4G. In the case of 2G, the remaining 94 bits would need to accommodate both the counter SQN and the MAC (M_κ). This means the 2G case can accommodate a SQN of 62 bits while allowing a M_κ of 32 bits. Table 1 provides a short overview of the bit lengths of different variables.

We introduced a new shared key for the encryption of the PMSIs: κ. While the existing shared key $\mathcal{K}$ could be reused for κ, we do not recommend this. It is good security design to use different keys for different functions. In fact, while we use κ for both encryption and MAC generation in the 2G solution, it is again good design to use two separate keys for this. The eventual choice for κ is naturally dependent on the choice of encryption and MAC scheme and as such it is not included in Table 1.

Since asking the authentication server for multiple challenges for an IMSI will give several encryptions of the same PMSI, with only an increased counter, the cipher used for encryption should be secure against related plaintext attacks. For the 3G and 4G networks, the AES blockcipher is advised for the implementation of the authentication functions [24]. AES is secure under related plaintext attacks, so simply reusing AES here would be enough. For 2G networks, the authentication functions are called A3/A8 and can be chosen by the provider. The suggested algorithm, called COMP128, is secret and proprietary. At least the first version is known to be weak [37]. Also in the 2G case, the choice for a the MAC algorithm (M_κ) is not trivial, with a 32 bit output and a 96 bit input. One option would be to reuse 2G's authentication algorithm A3, which takes a 128 bit challenge and computes a 32 bit response, as M_κ. So, a provider would have to see whether their implementation for A3/A8 would be secure for both the encryption of the PMSIs (so, related plaintexts) and the MAC generation. Then again, our fix demands an update of the SIM cards anyway, so updating the 2G authentication algorithms could be added at minimal cost.

5.2 Roll-out scenario & overhead

We discussed our solution with a major telecom provider, both in terms of feasibility and possible roll-out scenarios. They considered our solution as a possible fix to prevent (long term) tracking of customers.

Swapping all SIMs is a costly operation for a provider. The SIMs themselves are quite cheap, but the process of handing them out is costly. However, many SIM cards currently out in the field can be updated remotely through Over-The-Air (OTA) commands. Not all SIMs in the field that support such updates, accept updates to the authentication procedure, but we could not obtain the numbers of updateable SIMs. Still, since the presented solution is backwards compatible – a non-updated SIM will simply answer the challenge as it always did, without ever switching to the new PMSI – there are no big issues in rolling out our improvement to only a set of SIMs.

The overhead introduced by the additional computations in our solutions is expected to be very small for both the home network and SIM. We introduce one MAC generation and verification (only for 2G) and a symmetric encryption and decryption operation. Even for SIM cards with limited computational power this should not present a problem, as this is functionality that is already used in the original protocols.

Our solution introduces a small overhead for the serving network, because a SIM that switches to a new PMSI will look like a completely new phone to the serving network. Naturally, the SIM cannot sign off with the old PMSI first, as this would defeat the purpose of unlinkability. Networks are used to phones that do not sign off properly, and occasionally check whether phones registered in their area are still present. Wide adoption of our solution is likely to increase these numbers. The exact influence of this overhead is hard to estimate without field trials, though serving networks are already resilient against high number of fast moving SIMs, making it unlikely that our change would significantly impact their workings.

5.3 Formal verification

To verify our solution formally we make use of ProVerif, an automatic cryptographic protocol verifier [38]. This tool is used to check for secrecy and authentication properties of security protocols, but can also be used to check privacy related properties [39]. The usual Dolev-Yao attacker model is assumed, where the attacker has complete control over the network but cannot break cryptography [40]. The analysis by ProVerif is sound but not complete, i.e. ProVerif returns no false positives (claiming that a property holds, though an attack still exists), but it might find invalid attacks. We modeled the original protocols and our proposed modified versions for both 2G and 3G/4G in the typed applied pi-calculus.[6] For these models we check whether an attacker can link different sessions of the AKA protocol that belong to the same user. This is done by starting two sessions by different users. After this, a third sessions is started and the attacker has to distinguish which of the two users started it. For the original protocols for 2G and 3G/4G, no unlinkability is provided and a valid attack is returned by ProVerif. Unlinkability is however proved by ProVerif for our solutions for these mobile technology generations. In addition to unlinkability, we also check authentication between the SIM and home network for 2G. As expected this fails for the original protocol, but holds when our solution is used. These proofs give confidence that our solutions for both the 2G and 3G/4G protocols actually introduce protection against IMSI catching – an attacker is no longer able to determine which sessions belong to which users – while keeping the original functionality of the protocols (authentication) intact.

6. RELATED WORK

We present a solution against probably the oldest practical attack against 3GPP networks: IMSI catching. We know of no other work describing a solution against IMSI catching, though there is a lot of work regarding privacy issues in mobile telephony in general.

Dabrowski et al. [41] listed several indicators for the possible presence of IMSI catchers. They also created both a network of stationary measurement devices and an Android app, each capable of detecting IMSI catchers. Karsten Nohl and others similarly introduced an Android app capable of detecting IMSI catchers [42]. This app warns the user for the likely presence of IMSI catchers, where the term IMSI catchers refers to the more inclusive meaning of the word and also includes eavesdropping. Warning users of these attacks is very valuable and provides a basis for our claims that these attacks are prevalent, but it does not prevent such attacks as our solution does.

While not addressing IMSI catching, there has been research into other location privacy issues caused by the 3GPP protocols. Arapinis et al. [27] used ProVerif to formally verify the 3G specifications. This revealed two new privacy issues; linkability of the IMSI to the TMSI using paging of mobile phones and a traceability attack that was detailed in Section 2.4. They also present solutions for both attacks; encrypting the IMSI in a paging command with a shared session key, and encrypting the response of a failed authentication request with a public key of the provider. Interestingly, our solution would diminish the effects of the IMSI - TMSI linkability, as a PMSI refresh will appear to the serving network as a new SIM arriving, which causes the assigning of a new TMSI. Our presented solution therefore negates the need for encrypting the IMSI. The solution for the traceability attack is still required whether or not our solution is implemented.

Hahn et al. [43] suggest a different solution for the Arapinis traceability attack. The response of a failed authentication request is essentially encrypted with the new symmetric session key instead of the public key solution offered by Arapinis et al. This solution might be more efficient, though the consequences of switching to the session key provided by a re-played challenge are not deeply explored.

In other work Arapinis et al. looked specifically at the TMSI reallocation protocol [17], both formally and experimentally. Both the specifications and common implementations were found to be having problems leading to privacy attacks.

[6] The models are available at http://www.cs.bham.ac.uk/~deruitej/

These privacy attacks stem from possible linkability between different TMSIs or recovering the link between an IMSI and a TMSI. These attacks mostly have even simpler counter measures than our solution against IMSI catching. However, again the implementation of our solution would also prevent these TMSI attacks. If the SIM changes its IMSI (PMSI), the TMSI will get changed as well.

7. CONCLUSION

We present the first solution against IMSI catching attacks that fits within the current standards, making the change transparent for intermediate networks and backwards compatible. We propose the introduction of changing pseudonyms (PMSIs), to use for identification. This solution can be deployed within the current architecture by an individual provider, as it controls the only two entities that need adapting: the SIM cards providing the IMSI and the authentication server within the home network. Additionally, this solution provides the SIM with a way to verify whether a given challenge was generated by its home network, adding a form of mutual authentication to the traditionally weak SIM-only authentication of 2G networks. Using the protocol verifier ProVerif, we verified that our solution indeed provides unlinkability between succeeding pseudonyms without harming the original verification.

In essence we bring the effectiveness of IMSI catching down to the effectiveness of TMSI catching: learning a temporary id, which is unlinkable to future temporary ids. This prevents long term tracking of individuals, as well as tracing individuals returning to specific locations. Essentially, our proposal provides k-anonymity for users [30], where k is the expected number of users from a specific home network provider and country in a specific location.

Our solution need not interfere with lawful interception uses of IMSI catching, as authorities can still go to a provider with a caught PMSI together with the time of the catching. Based on this information the provider should be able to retrieve the corresponding account from its logs. Our solution, however, does interfere with unlawful IMSI catching.

For future cellular communication standards (5G and on) the issue of IMSI catching could be easily tackled using a-symmetric cryptography, which we could not use because the increase in resulting ciphertext sizes would not fit inside the current message specifications. However, it is still unclear whether the newer standards will introduce such a fix and even if they do, eventual roll-out is still far away. Even worse, the current roll-out of second, third and fourth generation cellular communication will not simply be replaced, and will likely remain functional for the foreseeable future. The solution presented in this paper could remedy the issue of IMSI catching in the current systems. Hopefully, our fix will contribute to finally solve the privacy and traceability attacks present in over 25 years of 3GPP protocols.

8. REFERENCES

[1] Sebastian Kay Belle, Oliver Haase, and Marcel Waldvogel. Callforge: Call anonymity in cellular networks. Technical report, 2010.

[2] Sangmin Lee, Edmund L Wong, Deepak Goel, Mike Dahlin, and Vitaly Shmatikov. πbox: A platform for privacy-preserving apps. In *NSDI*, 2013.

[3] Elad Barkan, Eli Biham, and Nathan Keller. Instant ciphertext-only cryptanalysis of GSM encrypted communication. In *CRYPTO 2003*, volume 2729/2003. Springer Berlin / Heidelberg, 2003.

[4] Fabian van den Broek and Ronny Wichers Schreur. Femtocell Security in Theory and Practice. In *Secure IT Systems*, volume 8208 of *LNCS*, pages 183–198. Springer Berlin Heidelberg, 2013.

[5] Craig Timberg for The Washington Post. Tech firm tries to pull back curtain on surveillance efforts in washington. http://wapo.st/1qgzImt. Last accessed May 2015.

[6] Craig Timberg for The Washington Post. Feds to study illegal use of spy gear. http://www.washingtonpost.com/blogs/the-switch/wp/2014/08/11/feds-to-study-illegal-use-of-spy-gear/. Last accessed May 2015.

[7] Siraj Datoo for The Guardian. How tracking customers in-store will soon be the norm. http://gu.com/p/3ym4v/sbl. Last accessed May 2015.

[8] Stuart Owen Goldman, Richard E Krock, Karl F Rauscher, and James Philip Runyon. Mobile forced premature detonation of improvised explosive devices via wireless phone signaling. US Patent 7552670, June 30 2009.

[9] Michael Böck. Simulation chamber and method for setting off explosive charges contained in freight in a controlled manner. US Patent 14345697, September 19 2012.

[10] Kadhim Shubber for Wired magazine. Tracking devices hidden in London's recycling bins are stalking your smartphone. http://www.wired.co.uk/news/archive/2013-08/09/recycling-bins-are-watching-you. Last accessed May 2015.

[11] Daehyun Strobel. IMSI catcher. *Chair for Communication Security, Ruhr-Universität Bochum*, page 14, 2007.

[12] Nico Golde, Kevin Redon, and Ravishankar Borgaonkar. Weaponizing femtocells: the effect of roque devices on mobile telecommunication. In *NDSS 2012*. The Internet Society, 2012.

[13] Ulrike Meyer and Susanne Wetzel. A man-in-the-middle attack on UMTS. In *The 3rd ACM workshop on Wireless security*. ACM, 2004.

[14] ETSI. *Digital cellular telecommunications system (Phase 2+); Mobile radio interface layer 3 specification*, 1998. EN 300 940 / GSM 04.08.

[15] ETSI. *Universal Mobile Telecommunications System (UMTS);LTE;Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3*, 2015. 3GPP TS 24.301 version 12.7.0 Release 12.

[16] ETSI. *Digital cellular telecommunications system (Phase 2+);UMTS;LTE;3G security;Security architecture*, 2013. 3GPP TS 33.102 version 11.5.0 Release 11.

[17] Myrto Arapinis, Loretta Ilaria Mancini, Eike Ritter, and Mark Ryan. Privacy through pseudonymity in mobile telephony systems. In *NDSS*, 2014.

[18] PUB FIPS. Advanced encryption standard (AES). *National Institute for Standards and Technology (NIST)*, 197(1), 2001.

[19] Andrey Bogdanov, Lars R Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew JB Robshaw, Yannick Seurin, and Charlotte Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In *CHES 2007*, volume 4727 of *LNCS*. Springer-Verlag, 2007.

[20] Christophe De Canniere, Orr Dunkelman, and Miroslav Knežević. KATAN and KTANTAN - A family of small and efficient hardware-oriented block ciphers. In *CHES 2009*, volume 5747 of *LNCS*. Springer-Verlag, 2009.

[21] Morris Dworkin. Recommendation for block cipher modes of operation: The CMAC mode for authentication. *NIST Special Publication (800-38B)*, 38B:1–25, 2005.

[22] PUB FIPS. The keyed-hash message authentication code (hmac). *National Institute for Standards and Technology (NIST)*, 1:1–13, 2008.

[23] PUB FIPS. Secure hash algorithm-3 (SHA-3) standard: Permutation-based hash and extendable-output functions. *National Institute for Standards and Technology (NIST)*, 202(0), 2014.

[24] ETSI. *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*, 2014. (3GPP TS 35.206 version 12.0.0 Release 12).

[25] ETSI. *Digital cellular telecommunications system (Phase 2+);Universal Mobile Telecommunications System (UMTS);LTE;Mobile radio interface Layer 3 specification;Core network protocols; Stage 3*, 2015. (3GPP TS 24.008 version 12.8.0 Release 12).

[26] ETSI. *Universal Mobile Telecommunications System (UMTS); Formal Analysis of the 3G Authentication Protocol*, 2001. 3GPP TR 33.902 version 4.0.0, Release 4.

[27] Myrto Arapinis, Loretta Mancini, Eike Ritter, Mark Ryan, Nico Golde, Kevin Redon, and Ravishankar Borgaonkar. New privacy issues in mobile telephony: fix and verification. In *CCS '12*. ACM, 2012.

[28] Jaap-Henk Hoepman, Engelbert Hubbers, Bart Jacobs, Martijn Oostdijk, and Ronny Wichers Schreur. Crossing borders: Security and privacy issues of the european e-passport. In *Advances in Information and Computer Security*, volume 4266 of *LNCS*, pages 152–167. Springer Berlin Heidelberg, 2006.

[29] Yong Ki Lee, Lejla Batina, and Ingrid Verbauwhede. Privacy challenges in rfid systems. In *The Internet of Things*. Springer New York, 2010.

[30] Latanya Sweeney. K-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[31] TSB - Telecommunication Standardization Bureau of ITU. *Mobile Network Codes (MNC) for the international identification plan for public networks and subscriptions*, 2014. (According to Recommendation ITU-T E.212 (05/2008))(Postition on 15 July 2014).

[32] David P Jablon. Strong password-only authenticated key exchange. *ACM SIGCOMM Computer Communication Review*, 26(5):5–26, 1996.

[33] Elad Barkan and Eli Biham. *Conditional Estimators: An Effective Attack on A5/1*, pages 1–19. Springer Berlin / Heidelberg, 2005.

[34] Alex Biryukov, Adi Shamir, and David Wagner. Real time cryptanalysis of A5/1 on a PC. In *Fast Software Encryption (FSE 2000)*, pages 1–18. Springer Berlin / Heidelberg, 2000.

[35] Marc Briceno, Ian Goldberg, and David Wagner. A pedagogical implementation of the GSM A5/1 and A5/2 "voice privacy" encryption algorithms. http://cryptome.org/gsm-a512.htm (originally on www.scard.org), 1999. Last accessed April 2014.

[36] Fabian van den Broek. Eavesdropping on GSM: state-of-affairs. In *5th Benelux Workshop on Information and System Security (WISSec 2010)*, November 2010.

[37] Ian Goldberg and Marc Briceno. GSM cloning. http://www.isaac.cs.berkeley.edu/isaac/gsm.html.

[38] Bruno Blanchet. An efficient cryptographic protocol verifier based on Prolog rules. In *14th Computer Security Foundations Workshop*. IEEE, 2001.

[39] Bruno Blanchet, Martín Abadi, and Cédric Fournet. Automated Verification of Selected Equivalences for Security Protocols. In *20th IEEE Symposium on Logic in Computer Science (LICS 2005)*. IEEE, 2005.

[40] Danny Dolev and Andrew C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

[41] Adrian Dabrowski, Nicola Pianta, Thomas Klepp, Martin Mulazzani, and Edgar Weippl. Imsi-catch me if you can: Imsi-catcher-catchers. In *Proceedings of the 30th Annual Computer Security Applications Conference*, pages 246–255. ACM, 2014.

[42] SR Labs. Snoopsnitch website. https://opensource.srlabs.de/projects/snoopsnitch. Last accessed February 2014.

[43] Changhee Hahn, Hyunsoo Kwon, Daeyoung Kim, Kyungtae Kang, and Junbeom Hur. A privacy threat in 4th generation mobile telephony and its countermeasure. In *Wireless Algorithms, Systems, and Applications*, volume 8491 of *LNCS*. Springer, 2014.