On De-synchronization of User Pseudonyms in Mobile Network

Mohsin Khan[1(✉)], Kimmo Järvinen[1], Philip Ginzboorg[2,3], and Valtteri Niemi[1]

[1]University of Helsinki, Helsinki, Finland
{mohsin.khan, kimmo.u.jarvinen, valtteri.niemi}@helsinki.fi
[2] Huawei Technologies, Helsinki, Finland
[3] Aalto University, Espoo, Finland
philip.ginzboorg@huawei.com

Abstract. This paper is in the area of pseudonym-based enhancements of user identity privacy in mobile networks. Khan and Mitchell (2017) have found that in recently published pseudonym-based schemes an attacker can desynchronize the pseudonyms' state in the user equipment and in its home network. In this paper, we first show that by exploiting this vulnerability a botnet of mobile devices can kick out of service a large portion of the users of a mobile network. We characterize this novel DDoS attack analytically and confirm our analysis using a simulation. Second, we explain how to modify the pseudonym-based schemes in order to mitigate the DDoS attack. The proposed solution is simpler than that in Khan and Mitchell (2017). We also discuss aspects of pseudonym usage in mobile network from charging and regulatory point of view.

Keywords: 3GPP · IMSI catchers · Pseudonym · Identity · Privacy

1 Introduction

International mobile subscriber identity (IMSI) catchers are threats to the identity privacy of mobile users. Passive IMSI catchers are devices that observe the wireless traffic and store all the IMSIs observed. Active IMSI catchers are malicious devices that can trick a user equipment (UE) to reveal its IMSI. Protection against passive IMSI catchers has been in the cellular networks since the second generation (GSM). However, active IMSI catchers have persisted in all the cellular networks, namely, GSM, UMTS and LTE [1,2,3,4,5,6].

An active IMSI catcher impersonates a legitimate serving network (SN) and asks for the identity of all the UEs in its range. The UEs have no way to differentiate an IMSI catcher from a legitimate SN, hence reveal their IMSIs as if they were revealing to a legitimate SN.

A potentially simple and backward compatible solution approach is to use frequently-changing temporary identities for mobile users [7,8,3,9,10]. The idea is that, if a UE communicates with an active IMSI catcher, it reveals only its temporary identity. This prevents the IMSI catcher from associating the temporary identity with any user who is previously known. The temporary identities are called pseudonyms. Hence solutions using this approach are called pseudonym based solutions.

Borek, Verdult, and Ruiter [7] and Khan and Mitchell [8] described pseudonym based solutions where the pseudonyms have the same format as IMSIs. From now on we will refer these two schemes as BVR and KM15 schemes. As shown by Khan and Mitchell in [11], these solutions are prone to the loss of synchronization between the pseudonyms in the UE and the home network (HN) of the user. In the worst case, the synchronization is completely lost and there is not even one common pseudonym left in the UE and the HN. Hence all identification and authentication attempts will fail thereafter and the UE will go out of the service. There is a vulnerability in these solutions that can be exploited by an attacker to cause the loss of pseudonym synchronization. The attacker can be a malicious UE or a malicious SN.

In addition to identifying the above vulnerability, Khan and Mitchel [11] also proposed a solution. In the rest of the paper, we will refer to this solution as the KM17 scheme. Careful investigation into this scheme shows that a UE has to use one pseudonym at least twice before it can get a new pseudonym from the network. The authors also argue that their solution may be vulnerable to a pseudonym de-synchronization attack by a malicious SN. To address the issue of malicious SNs, they introduce an identity recovery procedure. But, this procedure adds complexity: the number of temporary identities per user increases from two to six. Moreover, as we will later explain, the recovery mechanism itself can be exploited by an IMSI catcher to track the mobile user.

Our Contribution: We propose a pseudonym based solution that builds on top of the BVR, KM15 and KM17 schemes. The following contributions are made:

1. Identify a DDoS attack against an entire HN when the BVR scheme is used.
2. Design a solution that corrects the weaknesses of the KM17 scheme and is simpler than that scheme.
3. Outline some practical concerns of using pseudonyms from billing and regulatory point of view.

2 Preliminaries

Conceptually, a cellular network can be divided into UE; the HN – where the mobile user has a subscription; and the SN – that is the network to which mobile device connects. The SN and HN are the same when the UE is not roaming. An SN or HN consist of many entities. In this paper we will not discuss those fine details. However, we need to know some details about the UE. A UE consists of two entities: a mobile equipment (ME) and a subscriber identity module (SIM). The SIM is known as universal subscriber module (USIM) in UMTS. SIM and USIM are smart cards which are portable across different MEs. In LTE, the USIM is an application in the universal integrated circuit card (UICC). In this paper, we will refer all of them as SIM for the sake of simplicity.

A user is identified by IMSI. IMSI is a string of 15 decimal digits. An IMSI is a concatenation of mobile country code (MCC), mobile network code (MNC) and mobile subscription identification number (MSIN). MCC is a string of 3 decimal digits. MNC is either 2 or 3 decimal digits and MSIN is 9 or 10. Pseudonyms discussed in this paper are strings of 15 decimal digits and thus are indistinguishable from IMSIs. In this paper we limit our discussion to only one HN. Consequently all the IMSIs or pseudonyms we discuss have the same MCC and MNC. When we talk about the IMSI space or the pseudonym space, we actually mean the MSIN space. We denote the size of this space by $\mathcal{M}$ and it is either 10^9 or 10^{10}. We will also use n to denote the total number of users subscribed with the HN.

In the cellular networks, the security is built on a pre-shared master key $\mathcal{K}$ between a user and its HN. The key $\mathcal{K}$ is stored in the SIM along with the IMSI. The HN maintains a map from IMSI to key $\mathcal{K}$ for all the users. The authentication mechanism used by an SN to authenticate a user is based on challenge and response. The key $\mathcal{K}$ is only known by the HN. Hence, the HN delegates the SN by sending the challenge and expected response. Pseudonyms are assigned to a user during the authentication. This requires certain changes in the authentication protocol. In UMTS and LTE, the authentication protocols are called UMTS AKA and LTE AKA respectively. Before we discuss the pseudonym based solutions, we present UMTS AKA and LTE AKA briefly.

2.1 UMTS/LTE AKA

We discuss UMTS and LTE AKA only very briefly in order to provide the required background. Details of UMTS AKA can be found in Clause 6.3 of 3GPP TS 33.102 [12] and LTE AKA can be found in Clause 6 of 3GPP TS 33.401 [13].

The UE identifies itself by sending the IMSI to the SN within an attach request or a response to an IMSI inquiry. Upon receiving the IMSI, the SN sends an authentication vector (AV) request to the HN for the IMSI. The HN finds the pre-shared key $\mathcal{K}$, randomly generates a challenge ($RAND$) and computes the expected response ($XRES$), as well as two keys CK and IK as functions of $\mathcal{K}$ and $RAND$. The HN also computes a string called $AUTN$ for the purpose of some cryptographic protections of the authentication protocol. HN forwards $RAND, AUTN, XRES, CK, IK$ to the SN that forwards the $RAND, AUTN$ to the UE. The UE verifies the $AUTN$, computes $SRES, CK, IK$ using the $RAND$ and key $\mathcal{K}$ and forwards $SRES$ to the SN. If $SRES$ and $XRES$ are the same strings, then the authentication is successful. The keys CK and IK are used for confidentiality and integrity protection thereafter. See Figure 1.

In LTE, upon receiving the AV request, the HN also computes another key K_{ASME}. Contrary to UMTS, the HN forwards $RAND, AUTN, XRES, K_{ASME}$ to the SN. The UE verifies the $AUTN$, computes $SRES, CK, IK, K_{ASME}$ using the $RAND$ and key $\mathcal{K}$ and forwards $SRES$ to the SN. If $SRES$ and $XRES$ are the same strings, then the authentication is successful. The key K_{ASME} is used

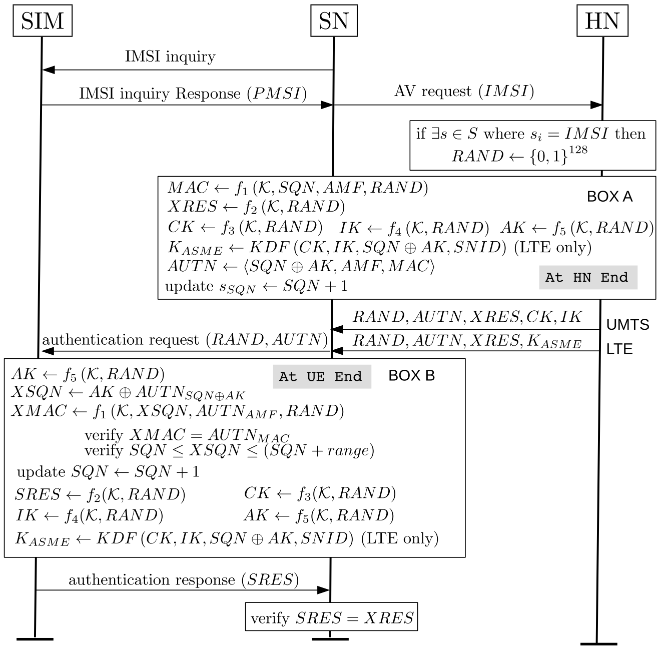

SIM
SN
HN
IMSI inquiry
IMSI inquiry Response ($PMSI$)
AV request ($IMSI$)
if $\exists s \in S$ where $s_i = IMSI$ then
$RAND \leftarrow \{0,1\}^{128}$
$MAC \leftarrow f_1\,(\mathcal{K}, SQN, AMF, RAND)$
BOX A
$XRES \leftarrow f_2\,(\mathcal{K}, RAND)$
$CK \leftarrow f_3\,(\mathcal{K}, RAND)$ $IK \leftarrow f_4\,(\mathcal{K}, RAND)$ $AK \leftarrow f_5\,(\mathcal{K}, RAND)$
$K_{ASME} \leftarrow KDF\,(CK, IK, SQN \oplus AK, SNID)$ (LTE only)
$AUTN \leftarrow \langle SQN \oplus AK, AMF, MAC \rangle$
At HN End
update $s_{SQN} \leftarrow SQN + 1$
$RAND, AUTN, XRES, CK, IK$
UMTS
authentication request ($RAND, AUTN$)
$RAND, AUTN, XRES, K_{ASME}$
LTE
$AK \leftarrow f_5\,(\mathcal{K}, RAND)$
At UE End
BOX B
$XSQN \leftarrow AK \oplus AUTN_{SQN \oplus AK}$
$XMAC \leftarrow f_1\,(\mathcal{K}, XSQN, AUTN_{AMF}, RAND)$
verify $XMAC = AUTN_{MAC}$
verify $SQN \leq XSQN \leq (SQN + range)$
update $SQN \leftarrow SQN + 1$
$SRES \leftarrow f_2(\mathcal{K}, RAND)$
$CK \leftarrow f_3(\mathcal{K}, RAND)$
$IK \leftarrow f_4(\mathcal{K}, RAND)$
$AK \leftarrow f_5(\mathcal{K}, RAND)$
$K_{ASME} \leftarrow KDF\,(CK, IK, SQN \oplus AK, SNID)$ (LTE only)
authentication response ($SRES$)
verify $SRES = XRES$

Fig. 1: UMTS/LTE AKA

to generate further keys for confidentiality and integrity protection. See Figure 1.

2.2 Location Update

3GPP TS 23.012 (Section 3.6.1.1) [14] specifies that, when a UE registers with a visitor location register (VLR), an entity in the SN, the VLR provides its address to the home location register (HLR), an entity in the HN. When a UE uses an IMSI/pseudonym for the first time, it is considered as a registration in the SN and, consequently the HN is informed with the address of the SN for the IMSI/pseudonym. We will refer to this location update (LU) message sent for IMSI/pseudonym x as $\texttt{LU}_x$ in this paper. We will use these LU messages in our solution.

3 Related Work

The BVR and KM15 schemes describe how pseudonyms can be introduced in the legacy networks. Also other proposals [3,9,10] were published in 2016 and 2017. All these proposals use essentially the same idea of using frequently changing pseudonyms recognized by the HN. The vulnerability identified in [11] is present in all these solutions. We will explain the DDoS attack and our solution in the context of the BVR scheme, but it applies to all existing pseudonym based solutions.

3.1 BVR Scheme

Along with the shared secret $\mathcal{K}$, every user shares another secret key κ with the HN. The SIM inside the UE stores two pseudonyms at any point of time, $(PMSI, P_{new})$. The SIM uses P_{new} the next time the UE receives an IMSI inquiry and keeps using P_{new} until it receives a new pseudonym. The HN also stores two pseudonyms (p, p') for every user at any point of time. In an ideal situation, $PMSI = p$ and $P_{new} = p'$.

The HN sends the next pseudonym encrypted by the key κ as a part of the random challenge $RAND$ used in AKA. Upon the successful completion of the AKA between the SN and the UE, the next pseudonym can be decrypted by the SIM. The BVR scheme builds on top of the UMTS/LTE AKA. Figure 2 shows the required changes. BOX A and BOX B in the figure refer to those operations in the same boxes in Figure 1.

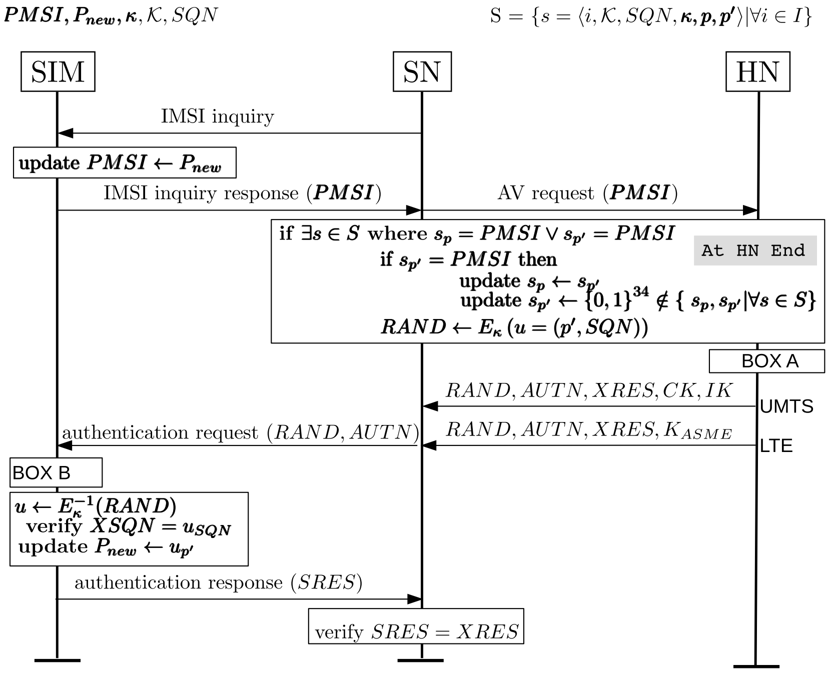

$\boldsymbol{PMSI}, \boldsymbol{P_{new}}, \boldsymbol{\kappa}, \mathcal{K}, SQN$
$S = \{s = \langle i, \mathcal{K}, SQN, \boldsymbol{\kappa}, \boldsymbol{p}, \boldsymbol{p'} \rangle | \forall i \in I\}$
SIM
SN
HN
IMSI inquiry
update $\boldsymbol{PMSI} \leftarrow \boldsymbol{P_{new}}$
IMSI inquiry response ($\boldsymbol{PMSI}$)
AV request ($\boldsymbol{PMSI}$)
if $\exists s \in S$ where $s_p = PMSI \vee s_{p'} = PMSI$
if $s_{p'} = PMSI$ then
update $s_p \leftarrow s_{p'}$
update $s_{p'} \leftarrow \{0,1\}^{34} \notin \{ s_p, s_{p'} | \forall s \in S\}$
$RAND \leftarrow E_\kappa (u = (p', SQN))$
At HN End
BOX A
$RAND, AUTN, XRES, CK, IK$
UMTS
$RAND, AUTN, XRES, K_{ASME}$
LTE
authentication request ($RAND, AUTN$)
BOX B
$u \leftarrow E_\kappa^{-1}(RAND)$
verify $XSQN = u_{SQN}$
update $P_{new} \leftarrow u_{p'}$
authentication response ($SRES$)
verify $SRES = XRES$

Fig. 2: The BVR Scheme

Whenever an AV request arrives for p', the HN forgets p. Forgetting an old pseudonym is important so that it can be reused. However, forgetting a pseudonym before being confirmed that p' has been received by the UE is a vulnerability as pointed in [11]. If a malicious UE identifies itself using a random pseudonym and if, by chance, the random pseudonym is associated with a legitimate UE, then the HN forgets an old pseudonym of this legitimate UE. In Section 4 we will show how this vulnerability can be exploited into a fatal DDoS attack.

3.2 KM17 Scheme

The KM17 scheme uses three pseudonyms at the HN end: $p_{past}, p_{current}$ and p_{future}. It also uses three recovery identities (RID) : RID_{past}, $RID_{current}$ and RID_{future}. The $\texttt{LU}_{p_{future}}$ message sent by an SN to an HN after registration of p_{future} is considered as the confirmation that p_{future}, RID_{future} have been delivered to the UE. Upon receiving $\texttt{LU}_{p_{future}}$, the HN forgets p_{past} and RID_{past} by setting $p_{past} \leftarrow p_{current}, p_{current} \leftarrow p_{future}, p_{future} \leftarrow null$ and after some other verifications sets $RID_{past} \leftarrow RID_{current}, p_{current} \leftarrow RID_{future}, RID_{future} \leftarrow null$. The HN always sends p_{future} as the next pseudonym embedded in the AV. If p_{future} is null, it generates a new one from the pool of unused pseudonyms.

Careful investigation of the KM17 scheme shows that a pseudonym has to be used at least twice before the UE can get a new pseudonym from the HN. The HN forgets p_{past} only when $\texttt{LU}_{p_{future}}$ arrives at HN. $\texttt{LU}_{p_{future}}$ would arrive only if p_{future} was used by the UE already at least once. Notice that $p_{current}$ that arrives after $\texttt{LU}_{p_{future}}$ is the same as p_{future} before $\texttt{LU}_{p_{future}}$ arrived. After the arrival of $\texttt{LU}_{p_{future}}$, p_{future} has become null. So, at this point, to get a new pseudonym, the UE has to use $p_{current}$. Consequently our claim follows. The use of the same pseudonym twice happens because the scheme does not forget p_{past} when $\texttt{LU}_{p_{current}}$ arrives. We take care of this issue in our solution.

The authors argue that the scheme is vulnerable to a malicious SN who tries to attack by sending fake LU messages. As a reactive measure, the authors propose a recovery process that enables a UE and the HN to get back in a synchronized state of pseudonyms. The recovery process uses a temporary recovery identity (RID). The HN sends the RID_{future} as a part of the $RAND$ in a similar way a pseudonym is sent. When a UE gets convinced that the pseudonym synchronization has been lost, the UE sends the RID piggybacked in the reject message $AUTS$. Based on the RID, the process can recover to a synchronized pseudonym state. Detail of the process can be found in [11]. However, an IMSI catcher can convince a UE that the synchronization has been lost and learn the RID of the UE. After learning the RID, the IMSI catcher can track the user using this RID. This is a severe problem because preventing such tracing is the reason for the use of pseudonyms in the first place.

However, one might argue that the RIDs can be changed as frequently as the pseudonyms. Note that forgetting an old RID is also triggered by the same $\texttt{LU}_{p_{future}}$ that triggers forgetting an old pseudonym. Consequently, synchronization of RIDs becomes as vulnerable as synchronization of pseudonyms, when a malicious SN sends fake LU messages. In the analysis of our solution in Section

6, we will show that a malicious SN can be detected very quickly and stopped before it can mount a meaningful attack.

4 Attack On BVR Scheme

The attack is mounted by a malicious UE. The attack has two phases.

Phase 1 A malicious UE sends an attach request using a random pseudonym q_1 to a legitimate SN. The legitimate SN sends an AV request for q_1 to the HN. If by chance, $q_1 = p'$ for a user s, the HN forgets p and sets $p \leftarrow p'$. The HN also generates an unused pseudonym p'' and sets $p' \leftarrow p^{''}$. As a result, in the HN, the current pseudonym state for the user s is $(p = P_{new}, p' \notin \{PMSI, P_{new}\})$.

Phase 2 The malicious UE sends another attach request using a random pseudonym q_2 to a legitimate SN. The legitimate SN sends an AV request for q_2 to the HN. If again by chance, $q_2 = p'$, then the HN again forgets p, sets $p \leftarrow p'$. HN also generates an unused pseudonym p''' and sets $p' \leftarrow p'''$. Consequently, the current pseudonym state of the user s is $\{PMSI, P_{new}\} \cap \{p, p'\} = \emptyset$; i.e. the user s and the HN become completely unsynchronized.

The next time the user would need to authenticate itself to a network, the authentication will fail and, hence, the user will be denied of any service. In this attack, it is assumed that the UE has not obtained a new pseudonym via a legitimate SN while the attack was mounted.

4.1 The DDoS Attack Against the BVR Scheme

The DDoS attack is mounted by a botnet of mobile devices. The mobile bots send many attach requests using different pseudonyms to legitimate SNs. The legitimate SNs in turn send AV requests for those pseudonyms to the HN. Let us assume that the total number of pseudonyms sent to the HN is a large integer m. In this case, a user s will be affected by the attack if there exists two integers $0 < x < y \leq m$ such that $q_x = p'$ and $q_y = p'$.

We have considered two different ways to mount this attack. In one way, the pseudonyms used in the attach requests are chosen randomly with replacement, which means the attack might sent one pseudonym more than once to the HN. In the other way, the pseudonyms are chosen without replacement.

With Replacement: In this case, after sending m pseudonyms to the HN, the expected portion of affected users $E[u_a]$ is

$$E[u_a] = 1 - \left(1 - \frac{1}{\mathcal{M}}\right)^m - m\left(\frac{1}{\mathcal{M}}\right)\left(1 - \frac{1}{\mathcal{M}}\right)^{(m-1)} \tag{1}$$

See the derivation in Appendix A. We have verified the accuracy of the above model via a simulation, see Figure 3.

Without Replacement: In this case the attacker runs two rounds of the attack. In the first round the attacker sends all the pseudonyms in the IMSI space without replacement, meaning that each pseudonym is sent exactly once.

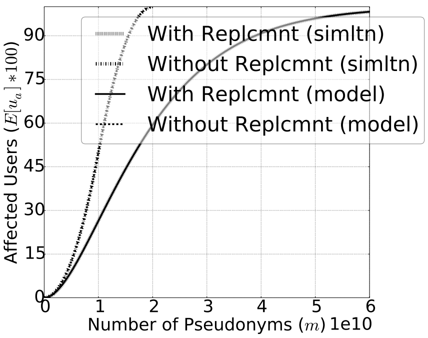


Fig. 3: DDoS Attack. $\mathcal{M} = 10^{10}, n = 10^7$. The model fits so well that it is difficult to distinguish the empirical lines from the model.

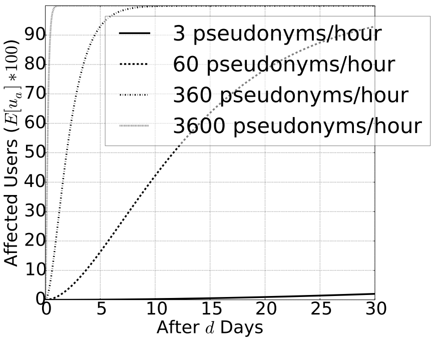


Fig. 4: DDoS Attack (with replacement), $\texttt{botnet}_{\texttt{size}} = 10^6$. Different lines represent the success rate as $\texttt{bot}_{\texttt{load}}$ varies.

Once the first round is completed, the attacker runs the attack for one more round. After sending m pseudonyms to the HN, the expected portion of affected users $E[u_a]$ is

$$E\big[u_a\big] = \begin{cases} \frac{m^2}{2 \cdot \mathcal{M}^2} & \text{if } 0 < m \leq \mathcal{M} \\ \frac{1}{\mathcal{M}}(2m - \mathcal{M} - \frac{m^2}{2\mathcal{M}}) & \text{if } \mathcal{M} < m \leq 2\mathcal{M} \end{cases} \tag{2}$$

See the derivation in Appendix B. We have verified the accuracy of the above model via a simulation, see Figure 3. Note that this is an estimation where the without-replacement attack is not a distributed attack. Rather the attack is mounted by only a single malicious UE. In the case of distributed and without replacement attack, the expected percentage of affected users will be less than what is shown in the plot unless the malicious UEs are very well synchronized. However, we believe that, a distributed without replacement attack will have a higher number of affected users than that of a distributed with replacement attack.

4.2 How Fatal is The DDoS Attack In Practice

The intensity of the attack will depend on the size of the botnet and the number of pseudonyms send by one bot in a unit time. We name these parameters as $\texttt{botnet}_{\texttt{size}}$ and $\texttt{bot}_{\texttt{load}}$ respectively. According to [15], the EPS AKA has the latency of 550 milliseconds. So, the peak value of $\texttt{bot}_{\texttt{load}}$ can safely be considered as 1 pseudonyms/second, i.e., 3600 pseudonyms/hour.

Mobile botnets are on the rise [16,17,18]. Many mobile botnets have already been observed, e.g., Geinimi [19], Zeus [20], AnserverBot [21], and DreamDroid [22]. A detailed survey of the state of mobile botnets can be found in [23]. In

2011, it was estimated that Dreamdroid was installed on 120,000 mobile devices [22]. In 2014, a mobile botnet of 650,000 mobile phones made an attack to a server [24]. It would not be surprising if we see a mobile botnet consisting tens of millions of mobile bots in the near future. However, for the discussion in this paper, we conservatively set the variable $\texttt{botnet}_{\texttt{size}} = 1$ million (10^6). Figure 4, shows how efficient a botnet of size 10^6 can be for varied values of $\texttt{bot}_{\texttt{load}}$.

5 Our Solution

In the HN, for a user s, our solution stores the IMSI i and three pseudonyms p, p', p''. In the SIM of the user s, two pseudonyms $PSMI, P_{new}$ are stored. In an ideal situation $PMSI = p, P_{new} = p'$. We build our solution on top of the BVR and KM17 scheme. The pseudonyms p, p', p'' can be compared with $p_{past}, p_{current}, p_{future}$ of KM17 scheme. However, unlike KM17 scheme, our solution uses $\texttt{LU}_{p'}$ to forget p. Let us assume that for a user s an AV request has arrived using the pseudonym p and the HN has responded with an AV by embedding p' in the $RAND$. When an LU for pseudonym p arrives, the HN considers it as a guarantee that pseudonym p' has been delivered to the UE of user s. Figure 5 presents our solution. The bold texts present the changes over UMTS/LTE AKA. BOX A and BOX B in the figure refer to those operations in the same boxes in Figure 1.

At HN side Whenever an AV request is received for a user s, using any of its identity, i.e., i, p, p' or p'', the HN responds with an AV that contains the pseudonym p'' in the $RAND$. If p'' is $null$ then an unused pseudonym is chosen and set as p''. When p'' is not null and LU message $\texttt{LU}_{p'}$ or $\texttt{LU}_{p''}$ arrives, the HN forgets p by setting $p \leftarrow p', p' \leftarrow p''$ and $p'' \leftarrow null$.

At UE side During the AKA, if MAC and $SEQN$ verification is successful, then the UE sends $SRES$ to the SN. Then the UE verifies if u_{SEQN} is the same as $XSEQN$ (see Figure 5). If this verification is also successful and $u_{p''} \notin \{PMSI, P_{new}\}$, then the UE sets $PMSI \leftarrow P_{new}$ and $P_{new} \leftarrow u_{p''}$. After a successful AKA, the old pseudonym $PMSI$ will still be used (in place of permanent identity IMSI) by the current SN, for example, in paging messages and in subsequent communications between the HN and the SN. The identity P_{new} comes into play in the next SN.

However, it is upon the freedom of the UE to identify itself either with $PMSI$ or P_{new} in an attach request or in response to an IMSI inquiry. If the consequent AKA fails many times after identifying with $PMSI$, the UE would identify itself using P_{new}.

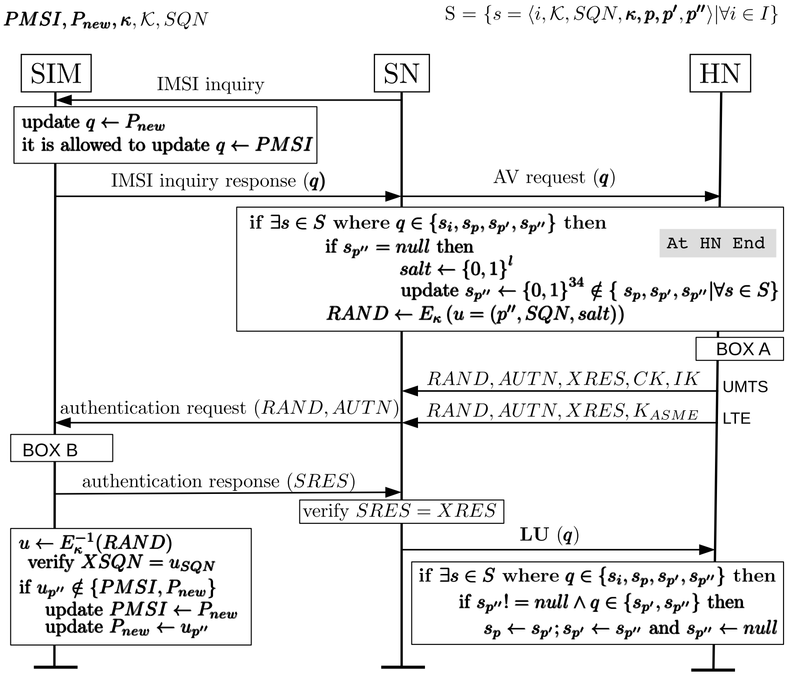

$PMSI, P_{new}, \kappa, \mathcal{K}, SQN$
$S = \{s = \langle i, \mathcal{K}, SQN, \kappa, p, p', p'' \rangle | \forall i \in I\}$
SIM
SN
HN
IMSI inquiry
update $q \leftarrow P_{new}$
it is allowed to update $q \leftarrow PMSI$
IMSI inquiry response (q)
AV request (q)
if $\exists s \in S$ where $q \in \{s_i, s_p, s_{p'}, s_{p''}\}$ then
if $s_{p''} = null$ then
$salt \leftarrow \{0,1\}^l$
update $s_{p''} \leftarrow \{0,1\}^{34} \notin \{ s_p, s_{p'}, s_{p''} | \forall s \in S\}$
$RAND \leftarrow E_\kappa(u = (p'', SQN, salt))$
At HN End
BOX A
$RAND, AUTN, XRES, CK, IK$
UMTS
authentication request ($RAND, AUTN$)
$RAND, AUTN, XRES, K_{ASME}$
LTE
BOX B
authentication response ($SRES$)
verify $SRES = XRES$
LU (q)
$u \leftarrow E_\kappa^{-1}(RAND)$
verify $XSQN = u_{SQN}$
if $u_{p''} \notin \{PMSI, P_{new}\}$
update $PMSI \leftarrow P_{new}$
update $P_{new} \leftarrow u_{p''}$
if $\exists s \in S$ where $q \in \{s_i, s_p, s_{p'}, s_{p''}\}$ then
if $s_{p''}! = null \land q \in \{s_{p'}, s_{p''}\}$ then
$s_p \leftarrow s_{p'}; s_{p'} \leftarrow s_{p''}$ and $s_{p''} \leftarrow null$

Fig. 5: Solution

6 Analysis of Our Solution

LU messages may be delayed, lost, or sent multiple times. Also in practice, the LU messages $\texttt{LU}_p, \texttt{LU}_{p'}, \texttt{LU}_{p''}$ might arrive in different order because of the inherent characteristics of IP networks. A malicious or faulty SN might send an LU message even when the corresponding AKA was failed or maybe not even run. To understand how our solution behaves in these unusual but possible situations, we analyze different categories of states a user s can be in the HN or the UE. We do this analysis based on the relevant variables and eventually construct a global state diagram of our solution. Based on the global state diagram, we show how our solution behaves in different situations.

6.1 State Diagrams

We divide all the possible states of a user s in HN in two categories. The first category is the one where p'' is not null and the second category is the one where p'' is null. Based on these two categories, we draw a state diagram as presented in Figure 6 (right side). The notation used below is explained in the lower part of Figure 6. Note that our solution is not sensitive to $\texttt{LU}_i$ and $\texttt{LU}_p$. Consequently only $\texttt{LU}_{p'}$ and $\texttt{LU}_{p''}$ are shown in the diagram.

On the other hand, the UE has only one kind of states as shown in Figure 6 (left side). It always has two pseudonyms $PMSI, P_{new}$. However, $PMSI$ and

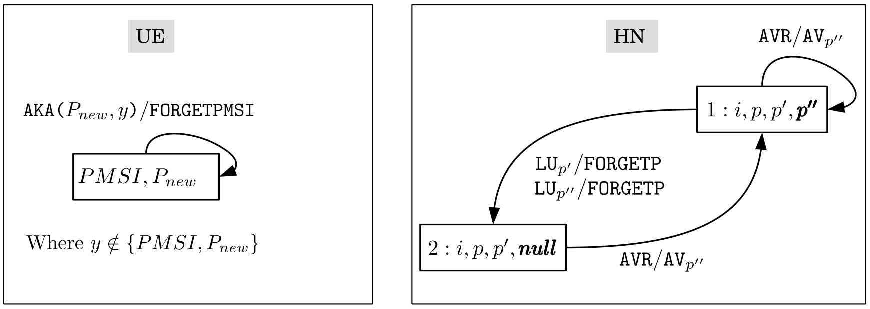
P_{new} may have different values. The values of $PMSI, P_{new}$ may change only when a successful AKA happens. For a user s, we have excluded the possibility of $\texttt{AKA}(x, y)$ where $x \notin \{PMSI, P_{new}\}$ because the UE will receive a wrong $RAND$ in that case. We also have excluded the possibility of $\texttt{AKA}(x, y)$ where $y \notin \{p, p', p''\}$. The reason for this exclusion is discussed in detail in Section 6. According to our solution, the UE does not do anything when $\texttt{AKA}(x, y)$ happens where $y \in \{PMSI, P_{new}\}$. However, even if $\texttt{AKA}(PMSI, y)$ happens where $y \notin \{PMSI, P_{new}\}$, the UE does not forget $PMSI$ because $PMSI$ would still be used by the SN, e.g., in paging messages. Consequently only $\texttt{AKA}(P_{new}, y)$ where $y \notin \{PMSI, P_{new}\}$ is shown in the diagram.

Next we merge the state diagrams of HN and UE into a global state diagram of our solution (Figure 7). The state of user's pseudonyms in the system can be described by whether $PMSI$ and P_{new} on the UE side are one of p, p', p'' or not, and by whether in the HN p'' has been allocated for user s or not. Based on this description, there can be $4 \cdot 4 \cdot 2 = 32$ pseudonyms-state of a user. But many of the states are never reachable. For example, it can never happen that $PMSI$ and P_{new} in the UE are the same because a UE forgets $PMSI$ only if the new pseudonym p'' is not in the set $\{PMSI, P_{new}\}$. All the inputs that can cause a transition of from one state to another in the state diagrams of HN and UE can also cause a transition from one state to another in the global state diagram (Figure 7).

In our solution, it is assumed that the initial state of a user in the system is $PMSI = p, P_{new} = p'$ on the UE side, and p'' has been allocated in the HN. Taking into account the possible transitions, we have found out that only 10 out of 32 possible states are reachable from this initial state. Those 10 states are illustrated in Figure 7. Note that neither the UE nor the HN has the knowledge in which state a user s is in the global diagram. All a UE knows are two pseudonyms $PMSI, P_{new}$ and the HN knows three pseudonyms p, p', p''.

For the limitation of space, we are not going to discuss all the states. Nevertheless, to assist the readers in understanding the diagram, let us take a closer look in few transitions. Let us consider that the user is currently at State 1. Since, p'' is already allocated, $\texttt{AVR}$ has no impact on this state. Consequently we do not mention $\texttt{AVR}$ in this state. Since p'' is allocated, it is possible that $\texttt{AKA}(P_{new}, p'')$ may run. If either one of these two AKAs happen, the UE forgets PMSI. Such an AKA run has no impact in HN until it receives the corresponding LU. Hence, the user moves to State 2 in the solution diagram where $PMSI = p', P_{new} = p'', p'' \neq null$. However, while at State 1, if either one of the LU messages $\texttt{LU}_{p'}, \texttt{LU}_{p'}$ arrives in the HN, the HN forgets p. Hence, the user goes to State 4 in the diagram where $PMSI \notin \{p, p', p''\}, P_{new} = p, p'' = null$. The AV request $\texttt{AVR}$ can cause a transition only when $p'' = null$, e.g., State 4.

Observe that the pseudonyms in UE and HN are (i) synchronized in states 1-3; (ii) partially unsynchronized in states 4, 5, 7, 8 and 9; (iii) completely unsynchronized in states 6 and 10, without any possibility of automatic recovery.


UE
AKA(P_{new}, y)/FORGETPMSI
$PMSI, P_{new}$
Where $y \notin \{PMSI, P_{new}\}$
HN
AVR/AV$_{p''}$
$1 : i, p, p', \boldsymbol{p''}$
LU$_{p'}$/FORGETP
LU$_{p''}$/FORGETP
$2 : i, p, p', \boldsymbol{null}$
AVR/AV$_{p''}$

AVR $=$ AV request has arrived using any of the identities i, p, p', p''

AV$_x$ $=$ AV sent to SN with x in $RAND$

LU$_x$ $=$ Location update for x has arrived

AKA(x, y) $=$ UE identified with x, the AKA is successful, y is the next pseudonym

$$\texttt{FORGETP} = \begin{cases} p \leftarrow p' \\ p' \leftarrow p'' \\ p'' \leftarrow null \end{cases} \qquad \texttt{FORGETPMSI} = \begin{cases} PMSI \leftarrow P_{new} \\ P_{new} \leftarrow y \end{cases}$$

Fig. 6: State diagrams of UE and HN

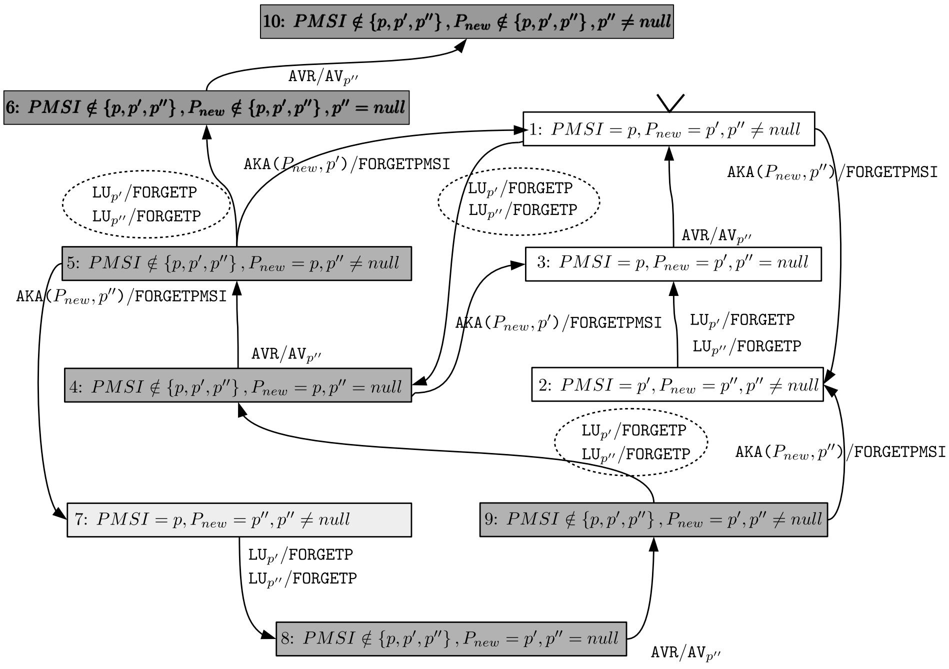

10: $\boldsymbol{PMSI \notin \{p, p', p''\}, P_{new} \notin \{p, p', p''\}, p'' \neq null}$
AVR/AV$_{p''}$
6: $\boldsymbol{PMSI \notin \{p, p', p''\}, P_{new} \notin \{p, p', p''\}, p'' = null}$
1: $PMSI = p, P_{new} = p', p'' \neq null$
AKA(P_{new}, p')/FORGETPMSI
AKA(P_{new}, p'')/FORGETPMSI
LU$_{p'}$/FORGETP
LU$_{p''}$/FORGETP
LU$_{p'}$/FORGETP
LU$_{p''}$/FORGETP
AVR/AV$_{p''}$
5: $PMSI \notin \{p, p', p''\}, P_{new} = p, p'' \neq null$
3: $PMSI = p, P_{new} = p', p'' = null$
AKA(P_{new}, p'')/FORGETPMSI
AKA(P_{new}, p')/FORGETPMSI
LU$_{p'}$/FORGETP
LU$_{p''}$/FORGETP
AVR/AV$_{p''}$
4: $PMSI \notin \{p, p', p''\}, P_{new} = p, p'' = null$
2: $PMSI = p', P_{new} = p'', p'' \neq null$
LU$_{p'}$/FORGETP
LU$_{p''}$/FORGETP
AKA(P_{new}, p'')/FORGETPMSI
7: $PMSI = p, P_{new} = p'', p'' \neq null$
9: $PMSI \notin \{p, p', p''\}, P_{new} = p', p'' \neq null$
LU$_{p'}$/FORGETP
LU$_{p''}$/FORGETP
8: $PMSI \notin \{p, p', p''\}, P_{new} = p', p'' = null$
AVR/AV$_{p''}$

Fig. 7: Global state diagram of our solution for a user $s \in S$.

6.2 Properties of Our solution

Behavior of our solution in unusual but possible cases If an AV request is responded by the HN but the corresponding AKA is failed or not even run, then the UE keep using the pseudonyms $PMSI$ or P_{new} in the upcoming AKA runs until an AKA succeeds. This can happen in global states 1, 5 or 9 (Figure 7), and in this situation the global state will remain the same.

If an AKA becomes successful but the corresponding LU message is not sent to the HN then the UE will not be able to get any new pseudonym in the successive AKA runs. This can happen in global states 2 or 7 (Figure 7).

If because of some internal error, the $PMSI$ goes out of sync, then consequent AKA where the UE is identified by $PMSI$ will fail. However, in this case $\texttt{AKA}(P_{new}, y)$ still may run where $y \notin \{PMSI, P_{new}\}$. Consequently the UE forgets $PMSI$ and gets back to sync.

If because of some internal error, the P_{new} goes out of sync, then consequent AKA where the UE is identified by P_{new} will fail. However, in this case $\texttt{AKA}(PMSI, y)$ still may run. Since the UE does not forget $PMSI$ after such an AKA, the UE does not update P_{new} even if $y \notin \{PMSI, P_{new}\}$. Consequently the UE will not be able to receive any new pseudonym at all. However, a pseudonym ($PMSI$ or P_{new}) going out of sync because of some internal error is an extremely rare event and can be compared with the case of corrupted SIM. A user can always go to the service center of the HN and get a new SIM. Another remedy of this problem can be to maintain a list of three pseudonyms at the UE end instead of two, i.e., when $\texttt{AKA}(PMSI, y), y \notin \{PMSI, P_{new}\}$ happens, the UE would store y as the third pseudonym even though it would not forget $PMSI$. We have not analyzed exhaustively what happens when such a third pseudonym is introduced in the solution and can be considered as a future work.

LU messages can arrive out of order. Receiving $\texttt{LU}_{p''}$ before receiving $\texttt{LU}_{p'}$ means the UE could not get a new pseudonym when it identified itself using p'' and ran the consequent AKA.

Receiving the LU messages multiple times for the same pseudonym may lead to the unsynchronized states 6 or 10 (Figure 7). But, as discussed later in this section, the probability of this is rather small.

Protection Against IMSI Catchers The pseudonyms are delivered to the UE encrypted by the pre-shared symmetric key κ. So, nobody except the UE can know the next pseudonym the UE will use. Hence an attacker, either active or passive, can not link a pseudonym with a previously known identity. In an ideal situation a UE uses one pseudonym in one successful AKA (notice the transitions `State 1` → `State 2` → `State 3` → `State 1` in Figure 7), which is unlike the KM17 scheme. In KM17 scheme, the UE has to use one pseudonym in two successful AKAs before it can obtain a new pseudonym (see our argument in Section 3.2). One pseudonym for one successful AKA essentially prevents an attacker to track a UE any longer than the attacker can track a UE using the TMSI or GUTI. However, the MCC and MNC part of the pseudonyms remains the same across all the pseudonyms used by a UE. Consequently if there are k many users with the same MCC and MNC in the geographical area of the UE, then our solution (like BVR and KM17) provides k anonymity of the user. Note that in a roaming situation k may be quite small.

Backward Compatibility The solution does not require any changes in the legacy SNs since no existing message format has been changed. The only changes are required in the HN and the SIM. Hence, once an HN implements the solution, any user having the upgraded SIM can enjoy the claimed identity privacy. The solution is still operable if the SIM is not updated even after the HN has implemented the solution. This is because, in our solution, the HN keep accepting the AV requests using the real IMSIs. The effect is, the UE will not be able to extract the new pseudonyms from the $RAND$. Otherwise everything else remains same and operable.

Our solution builds on top of UMTS/LTE AKA without introducing any new messages or changes in any existing messages. Hence solution will provide the claimed privacy in the presence of SNs from UMTS and LTE networks too.

A legacy SN may fetch multiple AVs from the HN for a single pseudonym x. Since $\texttt{AKA(}x, y\texttt{)}$ where $x = PMSI$ does not trigger any pseudonym update, let us consider the cases where $x = P_{new}$, i.e., $\texttt{AKA(}P_{new}, y\texttt{)}$. In such cases, if $y \notin \{PMSI, P_{new}\}$ then the UE forgets $PMSI$ and the P_{new} becomes the new $PMSI$. Consequently if the SN uses the pre-fetched AVs thereafter, it would be the $\texttt{AKA(}PMSI, y\texttt{)}$ cases – which has no impact on the pseudonym states of a user. Hence, our solution works smoothly even when an SN uses pre-fetched AVs for a pseudonym x unless some other user in the same SN is assigned with pseudonym x. Let us assume a user s_1 receives a new pseudonym x from the HN and sets $P_{new} \leftarrow x$. Then user s_1 uses $P_{new} = x$ in an SN where the SN already has a pre-fetched AV for the pseudonym x associated with user s_2 (forgotten by both HN and UE of s_2). In this case, user s_1's AKA will fail. But it is also very unlikely to happen. If it happens, the user s_1 can still run $\texttt{AKA(}PMSI, y\texttt{)}$. However, such AKAs will not enable the UE to receive a new pseudonym. One remedy of this problem can be that the UE may trigger an SQN resynchronization process if an AKA fails after identifying using P_{new}. By triggering such an SQN resynchronization process, the UE may force the SN to get a fresh AV from the HN.

Protection Against the DDos Attack: The DDoS attack is mounted by a botnet of mobile devices. The objective of the attack is to bring as many mobile users as possible to bring to State 6 of Figure 7. However, any path in the state diagram (Figure 7) that leads to State 6 involves at least one LU message. An SN will send an LU message for a pseudonym only if the corresponding AKA was successful. A mobile bot can not participate in a successful AKA with an SN using an arbitrary pseudonym. Hence, the attack does not work without an SN helping the botnet to do so.

Protection Against a Malicious or Faulty SN: In principle, a malicious SN can still attack the HN by sending a fake LU message for pseudonyms p', p'' that are associated with legitimate users. The target of the attacker would be to send a user to state 6 of the state diagram in Figure 7. We will show that the probability of success for such an attack is very low before the attack is detected and stopped. Besides an SN is in a roaming contract (it is a business contract) with an HN. The minimal harm the SN can cause to the HN before the attack is detected and stopped is not worth of risking the renewal of the contract.

Notice the paths that lead to state 6 in Figure 7. All such paths go via state 4. Let us assume that all the users of the HN are currently in State 1 or 9. This is a safe assumption because otherwise the attack would be even less likely to succeed. The malicious SN has to send fake $\texttt{LU}_{p'}$ or $\texttt{LU}_{p''}$ to reach State 4. This implies that the malicious SN needs to know either p' or p'' of a legitimate user s. The attack can be analyzed for two different situations. In one where the target users are currently not visiting the malicious SN. In other the target users are visiting the malicious SN.

Target Users are not Visiting the Malicious SN: The malicious SN can try to mount a DoS attack against an HN targeting the users who are not even visiting the SN. In that case, the malicious SN guesses $q = p''$ and send a LU for q to the SN. This brings the user to State 4. Then the malicious SN sends an AV request to the HN using q . This brings the user to State 5. Then the SN sends another LU for q. This time the user goes to State 6. So the attack is basically a sequence of LU message, AV request and another LU message using the same guessed pseudonym q. Let us consider that the SN starts from 0 and choose incrementally all the possible pseudonyms across the whole space and send the three messages to the HN using the chosen pseudonyms q. In that way after sending m triplets of messages to the HN, the expected number of affected users would be $\frac{nm}{\mathcal{M}}$.

Target Users are Visiting the Malicious SN: If the target users are currently visiting the malicious SN, it is easy to know the users' p' because the UE gives it to the SN. The malicious SN makes IMSI inquiries to all the UEs and hope that most of the UEs will respond with $P_{new} = p'$. Then the malicious SN sends LU messages for all the pseudonyms received as the respond of the IMSI inquiries. This brings many of the users to State 4. Once at State 4, the malicious SN can send an AV request to the HN using p' which will take the user from State 4 to State 5. However, once at State 5, neither the UE nor the malicious SN knows p', p''. So, to reach to State 6, the malicious SN has to guess one of p', p'' exhaustively. Suppose it starts from 0 and incrementally choose all the possible pseudonyms across the pseudonym space and send LU messages for the chosen pseudonyms to the HN. By doing so, after sending m LU messages to the HN, the expected number of users that will reach State 6 is at most $\frac{2rm}{\mathcal{M}}$, where r is the number of users currently visiting the malicious SN.

To summarize, a malicious SN can attack the users of an HN by sending to an HN LU messages with randomly chosen pseudonyms. But the HN can detect and counter the above attack: In order to desynchronize at least one user of the HN on the average, the malicious SN has to send more than $m \geq \mathcal{M}/n$ LU messages with randomly chosen pseudonyms. (Remember that n is the total number of users having the same MCC, MNC in the HN, and $\mathcal{M}$ is the size of MSIN space, which we assume is 10^{10}.) For example, if $n = 10^8$, then the number of those messages should be at least $10^{10}/10^8$, i.e. 100; and if $n = 10^6$, then the number of those messages should be at least 10^4.

But the malicious SN has a large, $1 - n/\mathcal{M}$ chance to guess the pseudonym wrongly in any given LU message; resulting in LU for a pseudonym not belonging to any user of the HN. Even a few such messages should make the HN realize that there is a de-synchronization attack. The HN could then apply countermeasures. For example, in the short term, HN could temporarily suspend processing any LU messages coming from the SN. This will make the UEs who are visiting the malicious SN more easy to track for the duration of suspension (because they will have to use the same pseudonym), but on the other hand, it will prevent de-synchronization of their pseudonyms. And in the long term, the HN may choose not to renew the roaming agreement with the operator of the SN.

Protection Against Replay Attack by an SN: A malicious SN may store an AV that it received from the HN with an intention to use later in an AKA with a UE. If the SN could do so, then a UE can be tricked to accept an old pseudonym which is already forgotten at HN. However, an SN can not do that. The pseudonyms are send to the UEs encrypted. No one including the SN knows the pseudonym before it is used by a UE. Consequently the malicious SN would know a valid AV for a UE identified by the pseudonym p only if the AV was obtained from the HN by making an AV request using the same pseudonym p. The next pseudonym embedded in such an AV can not be forgotten by either the UE or the HN. Hence a malicious SN can not make a replay attack to a targeted UE. However, the malicious SN can use an stored AV to run AKA with all the UE who are visiting the SN. In that case one user may get affected if the SQN in the AV is still fresh. This may imply that the valid range of SQN has to be small when a UE is in roaming.

Charging and LI: The HN has to keep track of the pseudonyms used by a particular user over time. This is needed, first, in order to bill the (right) user. Since the bills are typically settled once per month, the pseudonym usage records have to be retained for at least one month. Second, this is required for lawful surveillance of telecom traffic. The retention period of the records may be different in different countries, but in general, it will be longer than one month. For example, the European Union's Data Retention Directive [25] required to store call related data for a period of time between six and 24 months. The need of law enforcement agencies to know in real time the true identity of the mobile user can be met by the HN sending that identity to the SN during the connection establishment. But the handling of this identity would be a new feature in the legacy SN. When deploying pseudonym-based enhancements to user identity privacy, some parts in the SN may have to be upgraded to meet law enforcement needs.

Performance Overhead: A random choice of next pseudonym, the encryption of the pseudonym is the additional overhead in generating an AV. This should not be too much for an HN. The retention of the history of pseudonyms is also an additional overhead but does not impact the generation of AVs. In the SIM it adds one extra key and one decryption using the key. The SNs would see more users registering in the network because when a UE forgets a pseudonym, it does not inform the SN. However, the legacy SNs are familiar with cases where a UE suddenly powers off, and a UE forgetting an old pseudonym would just be treated as the UE has powered off.

Parameter Choice: The encryption used in encrypting the pseudonyms will not have forward secrecy since the same key κ is used all the time for encrypting pseudonyms. We have used 34 bits (as the BVR scheme) for next pseudonym in the $RAND$ instead of 4 bits per digit. This makes sense because the UE can convert it back into the required format. Besides, using less amount of bits for pseudonym encoding leaves room for the length of the *salt*, denoted by l be longer. If the cipher used for encrypting $u = (p'', SQN, salt)$ has block length 128, then embedding the same p'' in the successive $RAND$s can be randomized by the 48 bit long SQN and a 46 bit long *salt*. As the same p'' might be sent over successively many $RAND$s, the cipher used for encrypting u to generate $RAND$ should be immune against related plaintext attack [7]. AES is used in UMTS and LTE network for the implementation of authentication function and immune against related plaintext attack. So it would be enough to use AES for encrypting pseudonym also [7].

7 Conclusion

The need to maintain synchronized state between the UE and the home networks is one of the key issues in the design and implementation of pseudonym-based enhancements of user identity privacy in mobile networks. In this paper we have proposed a relatively simple design for a layer of pseudonyms between UE and home network that can withstand de-synchronization attacks. This gives hope that pseudonym-based solutions can be applied in commercial cellular networks. Topics for future work include formal verification of our scheme, and its implementation and testing using real UEs and mobile network elements.

Acknowledgement. We thank Markku Antikainen for the useful discussion to characterize the DDoS attack and his help to optimize the code of the simulation of the DDoS attack.

References

1. Samfat, D., Molva, R., Asokan, N.: Untraceability in Mobile Networks. In: Proceedings of the 1st Annual International Conference on Mobile Computing and Networking. MobiCom '95, New York, NY, USA, ACM (1995) 26–36
2. Strobel, D.: IMSI Catcher. `https://www.emsec.rub.de/media/crypto/attachments/files/2011/04/imsi_catcher.pdf` (July 2007) [It was available online at least until 14-July-2017].
3. Ginzboorg, P., Niemi, V.: Privacy of the long-term identities in cellular networks. In: Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications. MobiMedia '16, ICST (2016)

4. Soltani, A., Timberg, C.: Tech firm tries to pull back curtain on surveillance efforts in Washington. https://www.washingtonpost.com/world/national-security/researchers-try-to-pull-back-curtain-on-surveillance-efforts-in-washington/2014/09/17/f8c1f590-3e81-11e4-b03f-de718edeb92f_story.html?utm_term=.96e31aa4440b (September 2014) [Online; Was available until 14-July-2017].
5. Ney, P., Smith, J., Gabriel, C., Tadayoshi, K.: SeaGlass: Enabling City-Wide IMSI-Catcher Detection. In: Proceedings on Privacy Enhancing Technologies. PoPETs (2017)
6. Intelligence, P.E.: 3G UMTS IMSI Catcher. http://www.pki-electronic.com/products/interception-and-monitoring-systems/3g-umts-imsi-catcher/ [It was available online at least until 14-July-2017].
7. Van den Broek, F., Verdult, R., de Ruiter, J.: Defeating IMSI Catchers. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. CCS '15, ACM (2015)
8. Khan, M.S.A., Mitchell, C.J.: Improving Air Interface User Privacy in Mobile Telephony. In: Second International Conference, SSR 2015, Proceedings, Springer International Publishing (2015)
9. Norrman, K., Näslund, M., Dubrova, E.: Protecting IMSI and User Privacy in 5G Networks. In: Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications. MobiMedia'16, ICST (2016)
10. Muthana, A.A., Saeed, M.M.: Analysis of User Identity Privacy in LTE and Proposed Solution. In: International Journal of Computer Network and Information Security(IJCNIS), MECS Publisher (2017)
11. Khan, M., Mitchell, C.: Trashing IMSI Catchers in Mobile Networks. In: Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec 2017), Boston, USA, July 18-20, 2017, United States, Association for Computing Machinery (ACM) (May 2017)
12. 3GPP: 3GPP TS 33.102 V14.1.0 Security architecture (Release 14). https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2262 (March 2017)
13. 3GPP: 3GPP TS 33.401 V15.0.0 Security architecture (Release 15). https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2296 (June 2017)
14. 3GPP: 3GPP TS 23.012 V14.0.0 Location management procedures (Release 14). https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=735 (March 2017)
15. Ahlström, M., Holmberg, S.: Prototype Implementation of a 5G Group-Based Authentication and Key Agreement Protocol. http://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=8895975&fileOId=8895979 (2016)
16. Traynor, P., Lin, M., Ongtang, M., Rao, V., Jaeger, T., McDaniel, P., La Porta, T.: On cellular botnets: Measuring the impact of malicious devices on a cellular network core. In: Proceedings of the 16th ACM Conference on Computer and Communications Security. CCS '09, New York, NY, USA, ACM (2009) 223–234
17. Xiang, C., Binxing, F., Lihua, Y., Xiaoyi, L., Tianning, Z.: Andbot: Towards advanced mobile botnets. In: Proceedings of the 4th USENIX Conference on Large-scale Exploits and Emergent Threats. LEET'11, Berkeley, CA, USA, USENIX Association (2011) 11–11
18. Chen, W., Luo, X., Yin, C., Xiao, B., Au, M.H., Tang, Y.: Muse: Towards robust and stealthy mobile botnets via multiple message push services. In: Proceedings, Part I, of the 21st Australasian Conference on Information Security and Privacy - Volume 9722, New York, NY, USA, Springer-Verlag New York, Inc. (2016) 20–39

19. Linuxbsdos: Geinimi a Sophisticated New Android Trojan Found in Wild. http://linuxbsdos.com/2010/12/29/geinimi-sophisticated-new-android-trojan-found-in-wild/ (2010)
20. KrebsonSecurity: ZeuS Trojan for Google Android Spotted. https://krebsonsecurity.com/2011/07/zeus-trojan-for-google-android-spotted/ (2011)
21. Jiang, X.: Security Alert: AnserverBot, New Sophisticated Android Bot Found in Alternative Android Markets. https://www.csc2.ncsu.edu/faculty/xjiang4/AnserverBot/ (2011)
22. Tung, L.: Android Dreamdroid two: rise of laced apps. https://www.itnews.com.au/news/android-dreamdroid-two-rise-of-lacedapps-259147 (2011)
23. Karim, A., Shah, S.A.A., Salleh, R.B., Arif, M., Noor, R.M., Shamshirband, S.: Mobile botnet attacks - an Emerging Threat: Classification, Review and Open Issues. In: KSII Transactions on Internet and Information Systems - Vol. 9, no. 4. (2015) 1471–1492
24. Muncaster, P.: Chinese cops cuff 1,500 in fake base station spam raid. https://www.theregister.co.uk/2014/03/26/spam_text_china_clampdown_police/ (March 2014) [It was available online at least until 14-July-2017].
25. Wikipedia: Data Retention Directive. https://en.wikipedia.org/wiki/Data_Retention_Directive

Appendix A

Let us consider that there are $\mathcal{M}$ number of bins, each labeled by a pseudonym or IMSI. Choosing m random pseudonym with replacement and sending them in an attach request can be compared with the classic experiment of throwing m balls to $\mathcal{M}$ bins. The number of affected users by sending m number of attach requests is same as the number of bins that get two or more balls after throwing m balls to $\mathcal{M}$ bins. The probability that a particular bin gets no balls is $\left(1-\frac{1}{\mathcal{M}}\right)^m$. The probability that this bin get exactly one ball is $\binom{m}{1}\frac{1}{\mathcal{M}}\left(1-\frac{1}{\mathcal{M}}\right)^m$. Consequently, the probability that the bin will get two or more balls is:

$$1-\left(1-\frac{1}{\mathcal{M}}\right)^m-\binom{m}{1}\frac{1}{\mathcal{M}}\left(1-\frac{1}{\mathcal{M}}\right)^m$$

If n is the number of users in the HN, then by linearity of expectation, the expected number of affected user would be:

$$n\left(1-\left(1-\frac{1}{\mathcal{M}}\right)^m-m\frac{1}{\mathcal{M}}\left(1-\frac{1}{\mathcal{M}}\right)^m\right)$$

Consequently the expected portion of affected user would be:

$$E[u_a]=1-\left(1-\frac{1}{\mathcal{M}}\right)^m-m\frac{1}{\mathcal{M}}\left(1-\frac{1}{\mathcal{M}}\right)^m$$

Appendix B

In the without replacement attack, the attacker sends all the pseudonyms incrementally starting from 0 across the whole MSIN space. Let us consider that the pseudonym x is the first attack on a user's p'. Once the user's p' is attacked, HN updates $p \leftarrow p'$ and choose a new unused p' randomly. If the attacker sends total m number of pseudonyms to the HN, then the probability that the user's new p' will once again be attacked is: $\frac{m-x}{\mathcal{M}}$. If n is the total number of subscribers in the HN and $m \leq \mathcal{M}$ (first round) then expected number of subscribers affected after sending m pseudonyms is:

$$\begin{aligned}\frac{n}{\mathcal{M}}\int_0^m \frac{m-x}{\mathcal{M}}dx &= \frac{n}{\mathcal{M}^2}\int_0^m (m-x)\,dx\\ &= \frac{nm}{\mathcal{M}^2}\int_0^m dx - \frac{n}{\mathcal{M}^2}\int_0^m xdx\\ &= \frac{nm^2}{\mathcal{M}^2} - \frac{nm^2}{2\mathcal{M}^2}\\ &= \frac{nm^2}{2\mathcal{M}^2}\end{aligned}$$

Consequently, the expected portion of affected users would be $\frac{m^2}{2\mathcal{M}^2}$ where $m \leq \mathcal{M}$.

Let us now consider the case where $\mathcal{M} < m \leq 2\mathcal{M}$ (second round). The attacker again sends all the pseudonyms incrementally starting from 0. Choosing a pseudonym x will affect a user (who has not yet been affected) only if x is the pseudonym of a user who was attacked only once in the first round. The probability of that event is: $1 - \frac{x}{\mathcal{M}}$. So, after sending $m = \mathcal{M} + m'$ number of pseudonyms, the expected number of affected user would be:

$$\begin{aligned}&\frac{n\mathcal{M}^2}{2\mathcal{M}^2} + \frac{n}{\mathcal{M}}\int_0^{m'}\left(1-\frac{x}{\mathcal{M}}\right)dx\\ &= \frac{n}{2} + \frac{n}{\mathcal{M}}\int_0^{m'}\left(1-\frac{x}{\mathcal{M}}\right)dx\\ &= \frac{n}{2} + \frac{n}{\mathcal{M}^2}\int_0^{m'}(\mathcal{M}-x)\,dx\\ &= \frac{n}{2} + \frac{n}{\mathcal{M}^2}\int_0^{m'}\mathcal{M}dx - \frac{n}{\mathcal{M}^2}\int_0^{m'}xdx\\ &= \frac{n}{2} + \frac{nm'}{\mathcal{M}} - \frac{n}{\mathcal{M}^2}\frac{m'^2}{2}\\ &= \frac{n}{2} + \frac{n\,(m-\mathcal{M})}{\mathcal{M}} - \frac{n}{\mathcal{M}^2}\frac{(m-\mathcal{M})^2}{2}(\text{since } m = \mathcal{M}+m')\\ &= \frac{n}{\mathcal{M}}\left(2m - \mathcal{M} - \frac{m^2}{2\mathcal{M}}\right)\end{aligned}$$

Consequently, the expected portion of affected users would be $\frac{1}{\mathcal{M}}\left(2m - \mathcal{M} - \frac{m^2}{2\mathcal{M}}\right)$ where $\mathcal{M} < m \leq 2\mathcal{M}$.

Even though x in the above derivation is a discrete variable, we have used integration. So, the result we have found here is an approximation. We expect it to be a good approximation.