Pseudonym Based Solutions to Defeat IMSI Catchers Can Enable A DoS Attack

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. IMSI catchers are still in existence in all the 3GPP defined networks. Pseudonym based solutions to defeat IMSI catchers have been published in the recent years. In these solutions, we have found one vulnerability, that enables an attacker to convince the home network (HN) to forget an old pseudonym of a legitimate user equipment (UE) without any participation of the legitimate UE. A malicious UE or an SN can exploit this vulnerability to kick a legitimate UE out of service. We show that, exploiting this vulnerability, a novel DDoS attack can be mounted against an entire HN. The attack can send around 50 percent of the UEs out of service using a reasonably large botnet of mobile devices. We justify our claim by an analytical argument backed by a simulation. We present a solution to fight against the DDoS attack by using the location update message sent by an SN to an HN. We argue that our solution is immune to the the DDoS attack, protects the identity privacy, and remains backward compatible. In principle, a malicious SN can still mount a DoS attack against our solution. However, we argue that the SN can not gain anything meaningful before the DoS attack is detected and stopped. We also discuss other practical issues of the usability of pseudonyms from charging and lawful interception point of view.

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

1 Introduction

International mobile subscriber identity (IMSI) cathers 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].

IMSI Catching The network a UE has a subscription with is called the home network (HN). The network a UE visits and gets services from is called serving network (SN). In an ideal situation, a UE has to identify and authenticate itself to an SN before receiving any services from it. In cellular networks the encryption key in a UE is generated using the pre-shared symmetric key during authentication [7]. So, before authentication, neither a UE nor the SN/HN knows the key to use for encryption or decryption. Consequently, the identity of the UE has to be sent in plaintext to the SN. This enables an active IMSI catcher to play its trick.

The trick an IMSI catcher play against the UEs is that, it impersonates a legitimate SN and ask for the identity of all the UEs in the range of the IMSI catcher. 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.

An IMSI catcher can exploit the knowledge of caught IMSIs to monitor and track the physical location of a mobile user [8,9]. Please note that the term "IMSI catcher" is also used in a wider meaning, referring to extended attacks, including man-in-the-middle type of attacks or just spamming [10,11]. In this paper we limit our discussion only to prevent the IMSI catchers from catching the IMSIs (identities) of the users.

Different kind of solutions to defeat IMSI catchers have been proposed over the years. In addition to protect privacy, a desirable property of the solution is backward compatibility, i.e., it should protect the identity privacy even in the presence of a legacy SN. This is because, if the solution to defeat IMSI catchers works only in the latest generation of cellular network (e.g., 5G), then an attacker may be able to mount a downgrade attack.

Pseudonym Based Solutions A potentially simple and backward compatible approach is to use frequently-changing temporary identities for mobile users [12,13,3,14,15]. The idea is, even if an IMSI catcher play its trick, only the temporary identity of a user would be revealed. So, the IMSI catcher would not be able to associate the temporary identity with any user who is previously known. The temporary identities are called pseudonyms, hence the solutions use this approach are called pseudonym based solutions.

In 2015, Borek, Verdult, and Ruiter [12] and Khan and Mitchell [13], described solutions based on pseudonyms that have the same format as IMSIs. From now on we will refer these two schemes as BVR and KM15 schemes. These solutions are sensitive to the loss of synchronization between the pseudonyms in the UE and the HN. In the worst loss of synchronization case, there is not even one pseudonym left in the UE that the HN accepts. Hence all the identification and authentication attempt would fail thereafter and the UE would 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 serving network (SN).

In 2017, Khan and Mitchel [16] identified the loss of synchronization problem caused by a malicious UE and proposed a solution. In the rest of the paper we will refer to this solution as 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 is not immune to a pseudonym desynchronization 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 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 those in BVR, KM15 and KM17 schemes. The following contributions are made:

1. We show that a DoS attack can be mounted against an entire HN (when the BVR scheme is used) using the vulnerability identified in [16]. We calculate the expected success rate of the attack and argue that the attack can be fatal in practice.
2. We have identified weaknesses in KM17 scheme
3. We show how the synchronization of pseudonyms can be handled in a simple manner (also when there are DoS attacks), with three pseudonyms per user instead of six. In our solution, a UE can get a new pseudonym after using an old pseudonym only once in a successful AKA unlike the KM17 scheme. The KM17 scheme uses one pseudonym in two successful AKA runs.
4. Using probabilistic analysis, we show that a malicious SN can not mount any meaningful pseudonym desynchronization attack against our solution.
5. We discuss some practical concerns of using pseudonyms instead of IMSIs from billing and lawful interception point of view and suggest solutions.

2 Preliminaries

From a high level, a cellular network consists of 3 elements. A UE, SN and HN. The SN and HN are the same network when the user 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 detail about a 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 or USIM, they 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, for the sake of simplicity we will refer all of them as SIM.

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 of 2 to 3 decimal digits and MSIN is of 9 to 10 decimal digits. Pseudonyms discussed in this paper are IMSI looking strings. 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 IMSI space or pseudonym space, we actually mean the MSIN space. We denote the size of this space by $\mathcal{M}$. The value of $\mathcal{M}$ can be either 10^9 or 10^{10}. We will also use n to denote the total number of users subscribed with an 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 the key $\mathcal{K}$ is only known by the HN, hence the HN delegates the SN by sending the challenge and expected response. We will soon describe the authentication mechanism in UMTS and LTE briefly. But before that, we mention the use of temporary mobile subscription identity (TMSI) and globally unique temporary UE identity (GUTI).

Once a user is authenticated by the SN, the user is given a temporary identity by the SN with confidentiality protection. The temporary identity is used by the SN and UE thereafter to identify the user so that anyone listening to the radio traffic can not identify the user. The temporary identity is known as TMSI in GSM and UMTS networks, and as GUTI in LTE network. The use of TMSI and GUTI can not defeat the IMSI catchers because the SN can still make an IMSI inquiry to a UE. Hence an IMSI catcher impersonating a legitimate SN can also make an IMSI inquiry.

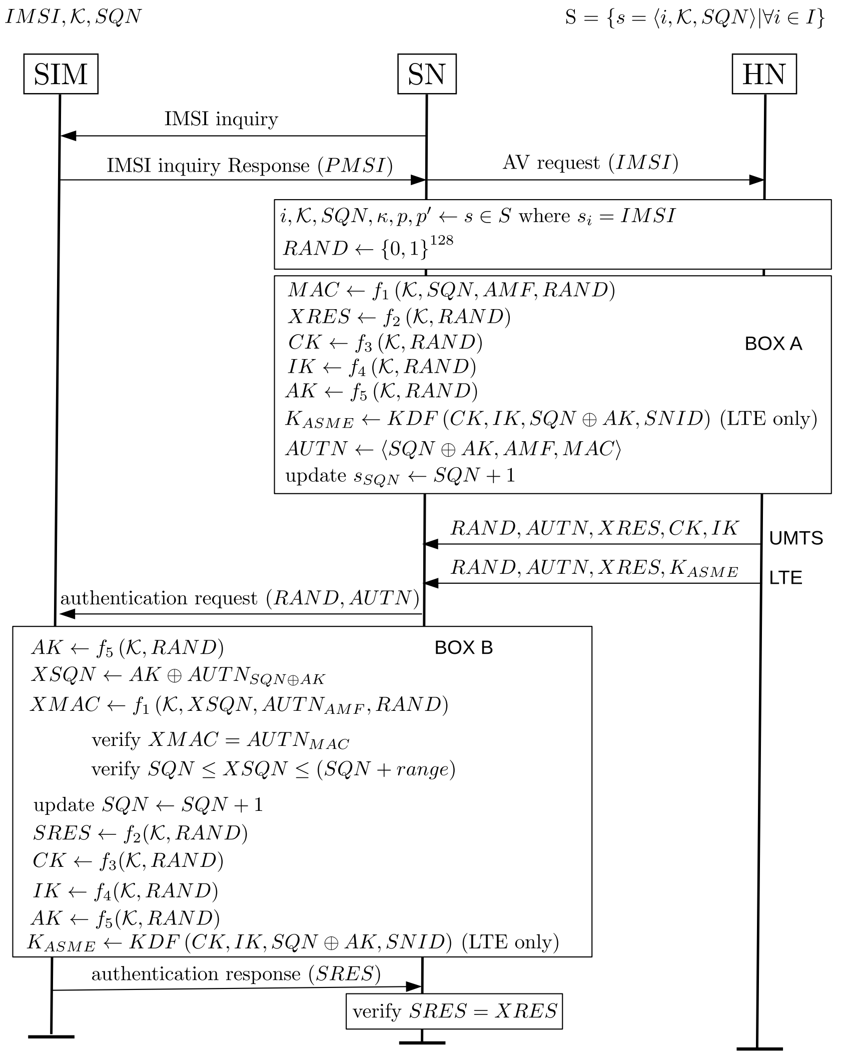
TMSI or GUTI is assigned to a UE by the SN. They are assigned with confidentiality protection after the authentication is successful and necessary security context has been set up. However, pseudonyms are assigned by the HN and no security context is set up between a UE and an HN after authentication. Hence, the idea is to assign the pseudonym to a user during the authentication. This requires to make certain changes in the authentication protocol. In UMTS and LTE, the authentication protocols are called UMTS AKA and LTE AKA respectively. LTE AKA has a very little difference compared with UMTS AKA. 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 so that we can make a smooth transition to the discussion of pseudonym based solution. Details of UMTS AKA can be found in clause 6.3 of 3GPP TS 33.102 [17] and LTE AKA can be found in clause 6 of 3GPP TS 33.401 [7].

The UE identifies itself using the IMSI to the SN while sending an attach request or responding to an IMSI inquiry. Upon receiving the IMSI, an SN sends an authentication vector (AV) request to the HN for the IMSI. The HN finds the pre-shared key $\mathcal{K}$, randomly generate a challenge ($RAND$) and compute the expected response ($XRES$), two keys CK and IK as function 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. SN 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}. And unlike 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 to generate further keys for confidentiality and integrity protection. See Figure 1.



$IMSI, \mathcal{K}, SQN$
$\mathrm{S} = \{s = \langle i, \mathcal{K}, SQN\rangle | \forall i \in I\}$
SIM
SN
HN
IMSI inquiry
IMSI inquiry Response ($PMSI$)
AV request ($IMSI$)
$i, \mathcal{K}, SQN, \kappa, p, p' \leftarrow s \in S$ where $s_i = IMSI$
$RAND \leftarrow \{0,1\}^{128}$
$MAC \leftarrow f_1(\mathcal{K}, SQN, AMF, RAND)$
$XRES \leftarrow f_2(\mathcal{K}, RAND)$
$CK \leftarrow f_3(\mathcal{K}, RAND)$
BOX A
$IK \leftarrow f_4(\mathcal{K}, RAND)$
$AK \leftarrow f_5(\mathcal{K}, RAND)$
$K_{ASME} \leftarrow KDF(CK, IK, SQN \oplus AK, SNID)$ (LTE only)
$AUTN \leftarrow \langle SQN \oplus AK, AMF, MAC\rangle$
update $s_{SQN} \leftarrow SQN + 1$
$RAND, AUTN, XRES, CK, IK$
UMTS
$RAND, AUTN, XRES, K_{ASME}$
LTE
authentication request ($RAND, AUTN$)
$AK \leftarrow f_5(\mathcal{K}, RAND)$
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

2.2 Location Update

It is specified in 3GPP TS 23.012 (Section 3.6.1.1) [18], 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 message sent for IMSI/pseudonym x as LU_x in this paper. We will use these LU messages in our solution.

3 Related Work

The BVR and KM schemes describe how the use of HN recongnized pseudonym can be introduced in the legacy networks. Following the BVR and KM15 schemes in 2015, there have been few other proposals [3,14,15] published in 2016 and 2017. All these proposals use essentially the same idea of using frequently changing pseudonyms recongnized by the HN. The vulnerability identified in [16] is present in all these solutions. So, for simplicity and limitation of space, we choose and explain only one of these schemes briefly and present our attack and solution in the context of the chosen scheme. We choose the BVR scheme.

3.1 BVR Scheme

They pseudonym used in this scheme is called pseudo mobile subscriber identifier (PMSI). Besides 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 keep using P_{new} until it receives a new pseudonym. The HN also stores two pseudonyms (p, p') for every subscriber at ay 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 and positive 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. Comparing Figure 1 and 2 shows that no changes are made in the messages that are transmitted, but only in the end points, i.e., the SIM and the HN. Since both of the HN and the SIM are maintained by same entity, the scheme is transparent to the legacy SNs.

Vulnerability in BVR Scheme Note that, whenever an AV request arrives for p', the HN forgets p. Forgetting an old pseudonym is important so that it can be reused. But forgetting before being confirmed that p' has been received by the UE is a vulnerability as pointed in [16]. If a fake UE (FUE) identifies itself using a random pseudonym and if by chance, the random pseudonym is associated

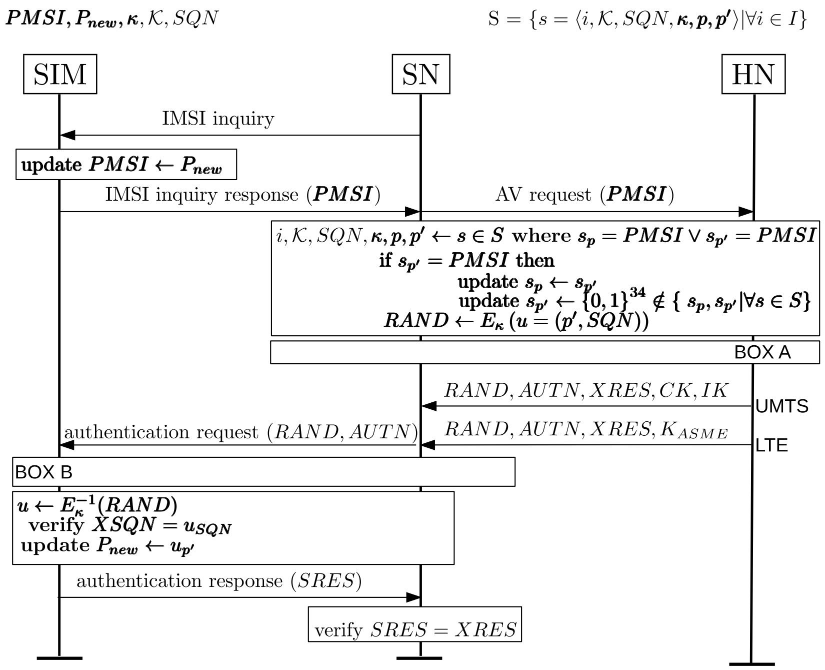

$\boldsymbol{PMSI}, \boldsymbol{P_{new}}, \boldsymbol{\kappa}, \mathcal{K}, SQN$
$S = \{s = \langle i, \mathcal{K}, SQN, \boldsymbol{\kappa}, \boldsymbol{p}, \boldsymbol{p'}\rangle | \forall i \in I\}$
SIM
SN
HN
IMSI inquiry
update $\boldsymbol{PMSI} \leftarrow \boldsymbol{P_{new}}$
IMSI inquiry response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

Fig. 2: The BVR Scheme

with a legitimate UE, the HN forgets an old pseudonym for the legitimate UE. The network also computes a new pseudonym which the legitimate UE has no knowledge of. If the network remembers k number of pseudonyms before forgetting any, the FUE needs to make the attack k times so that the network forgets all the pseudonyms that the legitimate user possesses. So, in the case of BVR scheme, the FUE has to send two pseudonyms. This is a fatal damage to the identity of the UE, because all the successive authentications of the UE will fail. In Section 4 we will show how this vulnerability can be exploited into a fatal DoS attack.

3.2 KM17 Scheme

In KM17, the authors have used the LU messages sent by an SN to an HN after registration of a new IMSI as the confirmation that the UE has received p'. The scheme uses three pseudonyms in the HN instead of two. It also uses three recovery identities (RID), that we will discuss in the following section.

Weaknesses in KM17 Scheme Careful investigation in KM17 scheme shows that a pseudonym has to be used at least two times before the UE can get a new pseudonym from the HN. The scheme maintains three pseudonyms $p_{past}, p_{current}$ and p_{future} at the HN end. The HN always embeds encrypted p_{future} (generates a new one if p_{future} is null) in the RAND. The HN forgets p_{past} only when $\mathrm{LU}_{p_{future}}$ arrives at HN. $\mathrm{LU}_{p_{future}}$ would arrive only if p_{future} was used by the UE already at least once. After receiving $\mathrm{LU}_{p_{future}}$, the HN updates $p_{past} \leftarrow p_{current}$, $p_{current} \leftarrow p_{future}$ and $p_{future} \leftarrow null$. Now, the UE has to use $p_{current}$ to get a new pseudonym from the HN. Notice that the $p_{current}$ after the location update is same as the p_{future} before the location update arrived. Consequently, our claim follows.

The authors argue that the scheme is not immune to 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 scheme uses temporary recovery identity (RID). The HN sends the RID 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 [16]. However, an IMSI catcher can convince a UE that the synchronization has been lost and learn the RID of the UE. Now the IMSI catcher can track the user using this RID instead of IMSI. This argument shows that the pseudonyms used in this scheme are as good as frequent the RIDs are changed.

However, one might argue that the RIDs can be changed as frequent as the pseudonyms are changed. Note that, forgetting an old RID is also triggered by the same location update message that triggers forgetting an old pseudonym. Consequently, synchronization of RIDs become as vulnerable as synchronization of pseudonyms, when a malicious SN sends fake location update message.

4 Attack On BVR Scheme

The attack is mounted by an FUE. The attack has two phases.

Phase 1 An FUE 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, $q_1 = p'$, 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 subscriber s is $(p = P_{new}, p' \notin \{PMSI, P_{new}\})$.

Phase 2 The FUE sends another attach request using a random pseudonym q_2 to a legitimate SN. The legitimate SN sends a 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 subscriber s is $\{PMSI, P_{new}\} \cap \{p, p'\} = \emptyset$.

The next time the user would need to authenticate itself to a network, the authentication will fail and hence be denied 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 sends AV request for those pseudonyms to the HN. Let us assume, 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 $q_x = s_{p'}$ and $q_y = s_{p'}$.

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, which means the attack send one pseudonym only once.

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

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

See Appendix **??** for the derivation. We have run a simulation of this attack and found that above model is fairly accurate. See Figure 3.

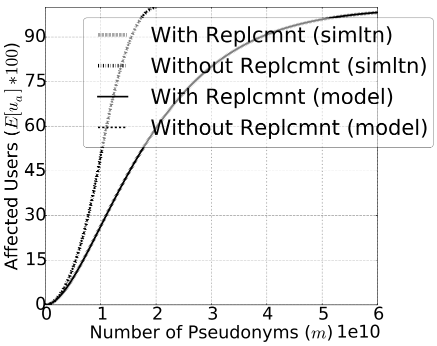


Fig. 3: DDoS Attack. $\mathcal{M} = 10^{10}, n = 10^7$. The model fits so well that it is difficult to distinguish the empirical lines from the model.

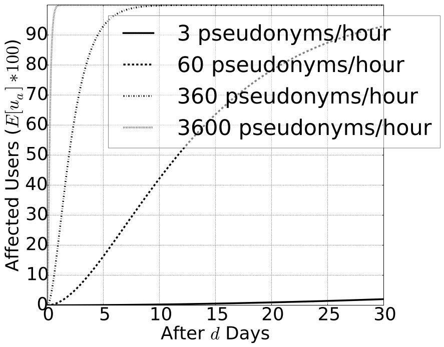


Fig. 4: DDoS Attack (with replacement), $\texttt{botnet}_{\texttt{size}} = 10^6$. Different lines represent the success rate as $\texttt{bot}_{\texttt{load}}$ varies.

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, means each pseudonym is sent exactly once. Once the first round is completed, the attacker runs the attack for one more round. However, after sending m number of pseudonyms to the HN, the expected portion of affected users $E[u_a]$ is

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

See Appendix **??** for the derivation. We have run a simulation of this attack and found that above model is fairly accurate. 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 FUE. In the case of distributed and without replacement attack, the expected perecentage of affected users will be less than what is shown in the plot. However, we believe that, the distributed and without replacement attack will have higher number of affected users than that of distributed with replacement attack.

4.2 How Fatal The DDoS Attack Can be In Practice

The intensity of the attack in practice will depend on two parameters. The first one is the size of the botnet, we name as $\texttt{botnet}_{\texttt{size}}$. The second one is the number of pseudonyms a bot sends per unit time in the attach requests, we name as $\texttt{bot}_{\texttt{load}}$. According to a thesis conducted in Lund University in 2016 [19], the EPS AKA has the latency of 550 milliseconds even when the MME is far away (10,000 km) from the HN. 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 [20,21,22]. Many mobile botnets have already been observed, e.g., Geinimi [23], Zeus [24], AnserverBot [25], DreamDroid [26]. A detailed survey of the state of mobile botnets can be found in [27]. In 2011, it was estimated that Dreamdroid was installed on 120,000 mobile devices [26]. In 2014, a mobile botnet of 650,000 mobile phones made an attack to a server [11]. It would not be surprising if we see a mobile botnet consisting tens of millions of mobile bots in near futrue. However, for the discussion of this paper, we conservatively set the variable $\texttt{botnet}_{\texttt{size}} = 1$ million (10^6). See Figure 4, it shows how efficient a botnet of size 10^6 can be for varied values of $\texttt{bot}_{\texttt{load}}$.

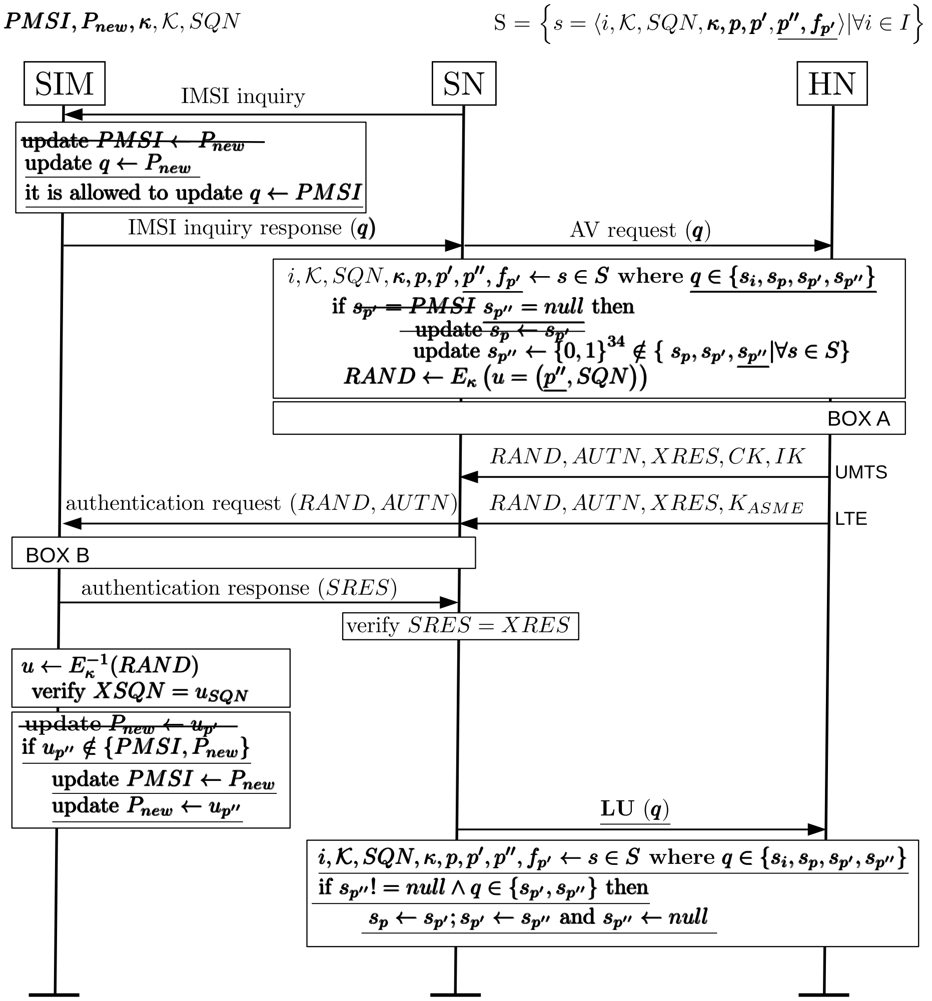

$\boldsymbol{PMSI}, \boldsymbol{P_{new}}, \boldsymbol{\kappa}, \mathcal{K}, SQN$
$\mathrm{S} = \left\{ s = \langle i, \mathcal{K}, SQN, \boldsymbol{\kappa}, \boldsymbol{p}, \boldsymbol{p'}, \underline{\boldsymbol{p''}, \boldsymbol{f_{p'}}} \rangle | \forall i \in I \right\}$
SIM
SN
HN
IMSI inquiry
~~update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

Fig. 5: Solution

5 Our Solution

As in the KM17 scheme, our solution also uses the LU message sent by an SN to the HN after the registration of a new IMSI/pseudonym i.e., after the AKA using a new IMSI/pseudonym. Let us assume that for a subscriber $s \in S$ 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 the guarantee that pseudonym p' has been delivered to the UE of user s.

In the HN, for a user $s \in S$, 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 scheme. Figure 5 presents our solution. The bold texts present the changes over UMTS/LTE AKA. The bold stricken out texts present the parts that we have discarded from the BVR scheme. The bold and underlined texts present the parts that we have added.

At HN side Whenever an AV request is received for a subscriber 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 $\text{LU}_{p'}$ or $\text{LU}_{p''}$ arrives, the HN forgets p by setting $p \leftarrow p', p' \leftarrow p''$ and $p'' \leftarrow null$.

At UE side If MAC and $SEQN$ verification is successful then the UE sends the SRES to the SN. Then the UE verifies if u_{SEQN} is the same as the $XSEQN$. If this verification is also successful and $u_{p''} \notin \{PMSI, P_{new}\}$, then the UE sets $PMSI \leftarrow P_{new}$ and $P_{new} \leftarrow u_{p''}$.

6 Analysis of Our Solution

Our solution needs only three temporary identities comparing with the KM17 scheme which needs 6 temporary identities. However, complexity arises in our solution when the LU message is delayed, lost, or sent multiple times. Also in practice, the LU messages $\text{LU}_p, \text{LU}_{p'}, \text{LU}_{p''}$ might arrive in different order because of the inherent characteristics of IP networks. A malicious or buggy SN might send a 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 study different categories of states a user s can be in the HN or the UE, based on the relevant variables.

We categorize 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. On the other hand, the UE has only one state. It always have two pseudonyms $PMSI, P_{new}$. However, $PMSI$ and P_{new} may have different values. There are four options these values can be chosen from: p, p', p'' and $\notin \{p, p', p''\}$.

Based on successful AKA in between the UE and the SN, the values of $PMSI, P_{new}$ may change in the UE. On the other hand, the values of p, p', p'' may change due to the arrival of an AV request or a LU message at HN. The action taken by an HN upon receiving an AV request or an LU message depends on the value of p'' (null or not null). Based on the different values $PMSI, P_{new}, p''$ can obtain, our solution can have 32 different states. Many of these states are never reachable. Figure 7 shows the state diagram of our solution with 10 reachable states out of the total 32. Notations in Figure 6 are also applicable in Figure 7. In Figure 7, we have excluded the possibility of running an $\texttt{AKA}(x, y)$ where $x \in \left\{p, p', p^{''}\right\}, y \notin \left\{p, p', p^{''}\right\}$. In Section 6.5, we argue why it is not feasible even when a malicious SN tries to mount a replay attack. Note that neither the UE nor the HN has the knowledge in which state a user s is in the solution. All a UE knows are two pseudonyms $PMSI, P_{new}$ and the HN knows three pseudonyms p, p', p''.

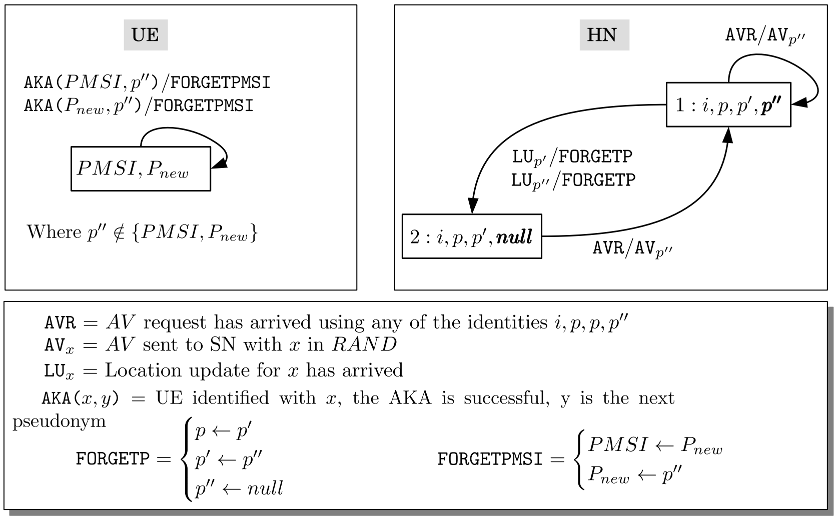

UE
$\texttt{AKA}(PMSI, p'')$/FORGETPMSI
$\texttt{AKA}(P_{new}, p'')$/FORGETPMSI
$PMSI, P_{new}$
Where $p'' \notin \{PMSI, P_{new}\}$
HN
$\texttt{AVR}/\texttt{AV}_{p''}$
$1 : i, p, p', \boldsymbol{p''}$
$\texttt{LU}_{p'}$/FORGETP
$\texttt{LU}_{p''}$/FORGETP
$2 : i, p, p', \boldsymbol{null}$
$\texttt{AVR}/\texttt{AV}_{p''}$
AVR $= AV$ request has arrived using any of the identities i, p, p, p''
$\texttt{AV}_x = AV$ sent to SN with x in $RAND$
$\texttt{LU}_x =$ Location update for x has arrived
$\texttt{AKA}(x, y) =$ UE identified with x, the AKA is successful, y is the next pseudonym
$\texttt{FORGETP} = \begin{cases} p \leftarrow p' \\ p' \leftarrow p'' \\ p'' \leftarrow null \end{cases}$
$\texttt{FORGETPMSI} = \begin{cases} PMSI \leftarrow P_{new} \\ P_{new} \leftarrow p'' \end{cases}$

Fig. 6: State diagrams of UE and HN

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. See State $1, 5$ and 9 in Figure 7. 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. See State 2 and 7. $LU_{p'}$ and $LU_{p''}$ gets the same treatment in the HN, so it does not matter in which order they arrive. But receiving $LU_{p''}$ before receiving $LU_{p'}$ means the UE could not get a new pseudonym when it identified

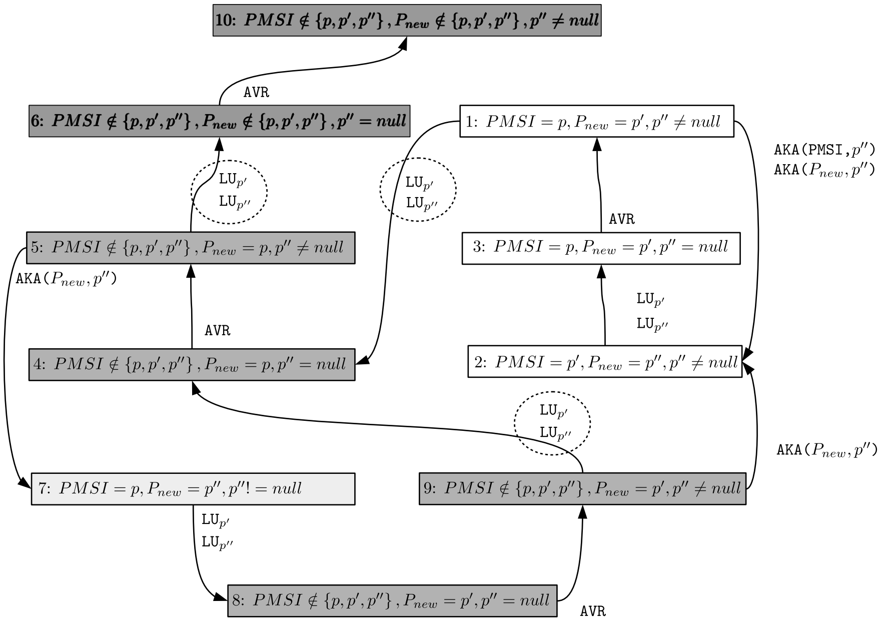

10: $PMSI \notin \{p,p',p''\}, P_{new} \notin \{p,p',p''\}, p'' \neq null$
AVR
6: $PMSI \notin \{p,p',p''\}, P_{new} \notin \{p,p',p''\}, p'' = null$
1: $PMSI = p, P_{new} = p', p'' \neq null$
AKA(PMSI,p'')
AKA(P_{new}, p'')
$\mathtt{LU}_{p'}$
$\mathtt{LU}_{p''}$
$\mathtt{LU}_{p'}$
$\mathtt{LU}_{p''}$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

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

itself using p'' and ran the consequent AKA. Receiving the LU messages multiple times for the same pseudonym can be fatal with very small probability. We discuss this case in detail in Section 6.4.

6.1 Protection Against IMSI Catcher

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 attakcer 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 anonimity of the user. Note that in a roaming situation k may be quite small.

6.2 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. However, our solution does not provide the privacy when the SN is from GSM.

A legacy SN may fetch multiple AVs from the HN for a single pseudonym p. This is not a problem because all those AVs will have the same next pseudonym embedded in them. So, once one of those AVs are used in a successful AKA, the pseudonym of the UE will change and the rest of the fetched AVs in the SN will never be used in any other AKA unless some other user in the same SN is assigned with pseudonym p.

Let us assume a user s_1 receives a new pseudonym p from the HN and sets $P_{new} \leftarrow p$. Then user s_1 uses $P_{new} = p$ in an SN where the SN already has a pre-fetched AV for the pseudonym p 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 because of the SQN synchronization. Even if it happens, the user s_1 can still use its $PMSI$ and get a new pseudonym.

6.3 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.

6.4 Protection Against a Malicious or Buggy 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 $\mathrm{LU}_{p'}$ or $\mathrm{LU}_{p''}$ to reach State 4. This implies that the malicious SN needs to know either p' or p'' of a legitimate subscriber s. The situation can be analyzed for two different situations. In one where the target users are currently visiting the malicious SN. In other the target users are not visiting the malicious SN.

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'. 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 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 p', p'' exhaustively. The best it can do is to start 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 $\frac{2rm}{\mathcal{M}}$, where r is the number of users currently visiting the malicious SN. Figure 8 shows the expected number of affected subscribers as m grows for different values of r.

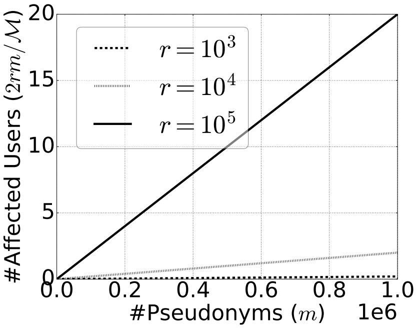


Fig. 8: Expected number of affected subscriber in the attack by SN. The attack is targeted to the subscribers who are visiting the SN.

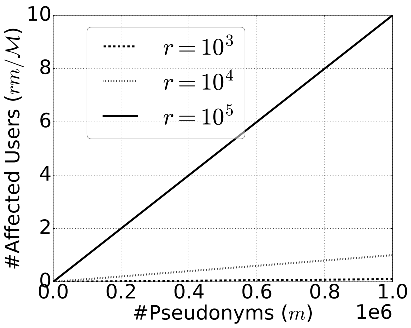


Fig. 9: Expected number of affected subscriber in the attack by SN. The attack is targeted to all subscribers of the HN.

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 $\frac{nm}{\mathcal{M}}$. Figure 9 shows the expected number of affected subscribers as m grows where n is the total number users of the HN.

6.5 Protection Against Replay Attack by 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.

We see that after sending 1 million fake LUs, the malicious SN can affect only at most 20 users. And far before sedning 1 million fake LUs, the attack will be detected by the HN and the malicious SN will be isolated. Such a faint attack is not worth for the malicious SN to risk the renewal of the roaming contract with the HN.

6.6 Charging and Lawful Interception

As the pseudonyms are frequently changing and one pseudonym may be used by many users over the time, the blling and charging system of HN needs to keep track of the time a pseudonym is used by a particular user. This implies that the HN needs to remember the history of the use of pseudonyms for a while so that two mobile operators who have a roaming contract can settle their bills. Usually the roaming partners settle their bills once in a month. This implies that the pseudonym history has to be retained for at least a month. On the other hand, for lawful interception (LI) purpose the history might needed to be remembered for quite a longer time, depends on the law of the land. The retention of the pseudonym history might become an overhead for the HN. However, it is just a mapping from the pseudoynms to the pairs $(IMSI, \text{date})$ and should not be a deal breaker for using pseudonyms.

6.7 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.

6.8 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 [12]. 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 [12].

7 Conclusion

1. discuss the solution where the UE will be asked if it want to share the IMSI.
2. forward secrecy
3. me based solution
4. formal verification

Acknowledgement.

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. 3GPP: 3GPP TS 33.401 V15.0.0 Security architecture (Release 15). https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2296 (June 2017)
8. Miller, J.: City of london calls halt to smartphone tracking bins. http://www.bbc.com/news/technology-23665490 (August 2013) [It was available online at least until 14-July-2017].
9. Goldman, S., Krock, R., Rauscher, K., Runyon, J.: Mobile forced premature detonation of improvised explosive devices via wireless phone signaling. http://www.google.com/patents/US20070234892 (October 2007) US Patent App. 11/233,198.
10. Dabrowski, A., Pianta, N., Klepp, T., Mulazzani, M., Weippl, E.: Imsi-catch me if you can: Imsi-catcher-catchers. In: Proceedings of the 30th Annual Computer Security Applications Conference. ACSAC '14, New York, NY, USA, ACM (2014) 246–255
11. 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].
12. 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)
13. 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)
14. 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)
15. 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)

16. 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)
17. 3GPP: 3GPP TS 33.102 V14.1.0 Security architecture (Release 14). https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2262 (March 2017)
18. 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)
19. 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)
20. 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
21. 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
22. 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
23. 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)
24. KrebsonSecurity: ZeuS Trojan for Google Android Spotted. https://krebsonsecurity.com/2011/07/zeus-trojan-for-google-android-spotted/ (2011)
25. Jiang, X.: Security Alert: AnserverBot, New Sophisticated Android Bot Found in Alternative Android Markets. https://www.csc2.ncsu.edu/faculty/xjiang4/AnserverBot/ (2011)
26. Tung, L.: Android Dreamdroid two: rise of laced apps. https://www.itnews.com.au/news/android-dreamdroid-two-rise-of-lacedapps-259147 (2011)
27. 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

References