# Defeating IMSI-Catchers Using Pseudonyms Can Enable A DDoS Attack: A Solution Exists

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

[1]University of Helsinki, Helsinki, Finland
{mohsin.khan, kimmo.u.jarvinen, valtteri.niemi}@helsinki.fi
[2] Huawei Technolgies
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. We have found one vulnerability in these solutions. The vulnerability enables an attacker to convince the home network (HN) to forget an old pseudonym of a legitimate 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 50 percent of the UEs out of service using a reasonably large botnet of mobile users. 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. Besides, an SN can behave maliciously in other even more fatal ways. We also discuss other practical issues of the usability of pseudonyms from charging and lawful interception point of view that appear to be ignored so far.

**Keywords:** 3GPP · IMSI-catchers · Pseudonym · Idenity · Privacy

## 1 Introduction

International mobile subscriber identity (IMSI) is the global identifier of a mobile phone subscriber. IMSI-catchers are devices that can create a list of IMSIs of the subscribers present in a certain geographical area. IMSI catching is an identity privacy problem. The problem has been known for long but still prevailing in all the 3GPP defined cellular networks (GSM, UMTS, LTE) for decades cite.

**How IMSI-catchers catch IMSI?** A subscriber's user equipment (UE) has to identify itself to the network before connecting. The identification message

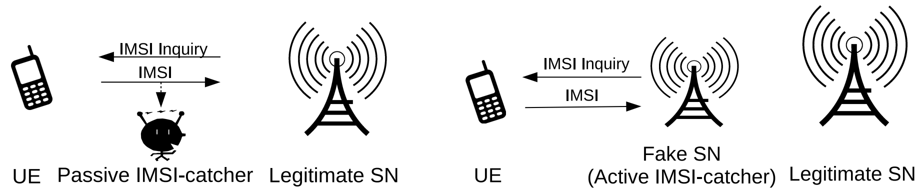


Fig. 1: IMSI-catcher

has to be sent in plain-text because the security of the network is based on symmetric key cryptography cite. In symmetric key cryptography, a secret key has to be shared before starting any encryption. The home network (HN) stores a secret key for every subscriber in the subscriber database. The secret keys are also securely stored in the respective subscriber identity module (SIM). However, the HN needs to know the identity of the subscriber to choose the right secret key to start encrypting or decrypting any message. So, when an unknown UE appears, the network makes an IMSI inquiry to the UE . Consequently, the UE has to sent the IMSI in plain-text cite.

A passive IMSI-catcher who is just listening to the radio channel can read the identification message. An active IMSI-catcher who sets up a fake base station and impersonates a legitimate serving network (SN), do an IMSI inquiry to all the UEs that try to connect. The UEs respond with their respective IMSIs in plain-text cite. See Figure 1.

**What an IMSI catcher can do with the caught IMSIs?** With the caught IMSIs, an IMSI catcher can monitor who are coming and leaving a certain geographical area cite. An IMSI catcher can also track the locations of a targeted individual cite. There are other range of more sophisticated active man in the middle attacks that start with catching the IMSI of a subscriber, e.g., attacking the confidentiality of user data by downgrading the air interface encryption cite. Now a days, all these advanced attackers are called IMSI-catchers. However, in this paper, we will limit our discussion to the attackers who only gathers a list of IMSIs.

**How available IMSI-catchers are in real life?**

**Current state of art in defeating IMSI-catchers**

**Our Contribution**

**Overview**

# 2 Background

**Identification in the existing networks**

**Authentication in the existing networks** we need to discuss the authentication mechanism because the pseudonym based approach uses the messages in the authentication protocol to piggyback the messages required to be sent across.

**How pseudonym based solution works**

# 3 Related Work

## 3.1 BVR Scheme

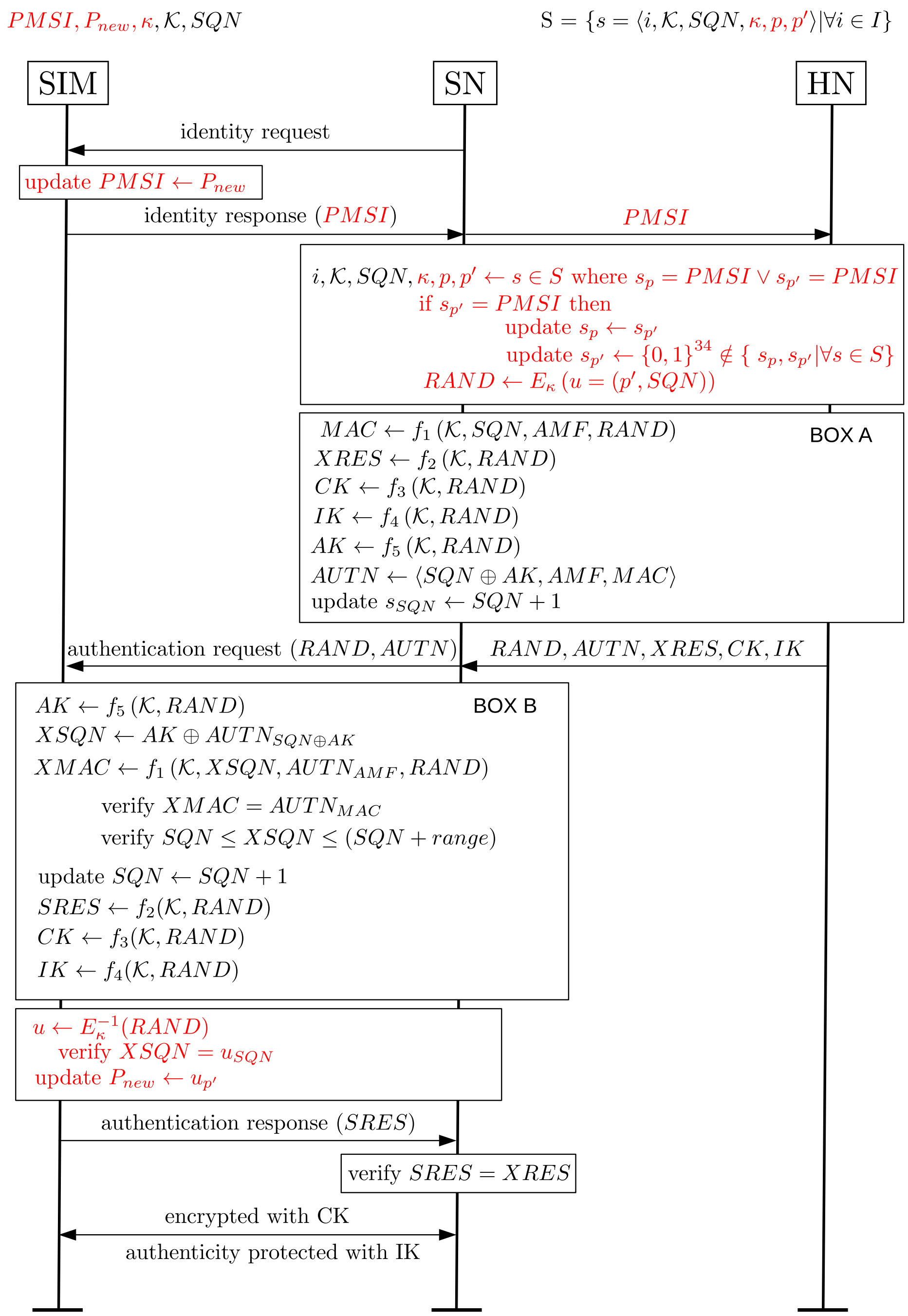


Fig. 2: BVR Solution

## 4 A Vulnerability of Pseudonym Based Solutions

The fundamental idea of all the pseudonym based solutions [1,2,3,4] are essentially the same. In all these solutions, when a certain old pseudonym is used by a user, the HN computes a new pseudonym, associate the new pseudonym to the respective IMSI and forget a certain old pseudonym. Forgetting an old pseudonym is important so that it can be reused. The vulnerability is: the HN forgets an old pseudonym of a legitimate UE without being confirmed that the legitimate UE has received the new one. This vulnerability can be exploited in the following way.

If a fake UE (FUE) identifies itself using a random pseudonym and if by chance, the random pseudonym is associated 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. This is a fatal damage to the identity of the UE, because all the successive authentications of the UE will fail.

This vulnerability can be exploited by a malicious mobile phone or an SN. We will use the BVR scheme to explain the vulnerability in detail, even though similar attacks can be mounted against the other schemes also.

**Exploiting the Vulnerability in BVR Scheme** In the BVR scheme, a subscriber $s$ has two pseudonyms $\left(s_p, s'_p\right)$ in the HN and two pseudonyms $(PMSI, P_{new})$ in the UE. In an ideal case, $PMSI = s_p, P_{new} = s_{p'}$.

The attack is mounted by an FUE. The FUE sends a random pseudonym $q_1$ to a legitimate SN. The legitimate SN forwards the pseudonym to the respective HN. If by chance, $q_1 = s_{p'}$, the HN forgets $s_p$ and sets $s_p \leftarrow s_{p'}$. The HN also generates an unused pseudonym $p''$ and sets $s_{p'} \leftarrow p^{''}$. As a result, in the HN, the current pseudonym-state for the subscriber $s$ is $(s_p = P_{new}, s_{p'} \neq PMSI, P_{new})$. At this stage, there is only one pseudonym present both at the UE and HN. See Figure 3.

The FUE sends another pseudonym $q_2$. If again by chance, $q_2 = s_{p'}$, then the HN again forgets $s_p$, sets $s_p \leftarrow s'_p$. HN also generates an unused pseudonym $p'''$ and sets $s_{p'} \leftarrow p'''$. Consequently, in HN, the current pseudonym-state of subscriber $s$ becomes $(s_p \neq PMSI, P_{new}, s_{p'} \neq PMSI, P_{new})$. If there were no pseudonyms sent by the HN to the legitimate UE while the attack was mounted, the pseudonym-state of the UE remains as $(PMSI, P_{new})$. So at this stage, there is no pseudonym present at both of the UE and HN sides. See Figure 3. The next time the user would need to authenticate itself to a network, the authentication will fail and hence be denied any service.

If there are $n$ number of subscribers in an HN, then the probability of the above attack being successful is $\frac{n}{10^{20}}$, which is apparently a tiny probability. However, in Section 5, we will show how this tiny probability can be exploited into a fatal DDoS attack.

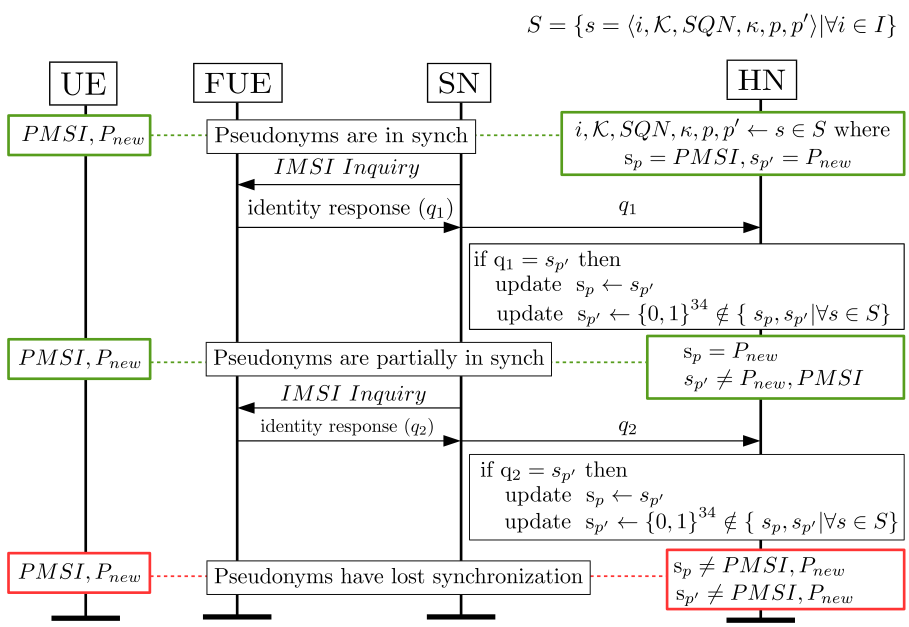


Fig. 3: A DoS Attack against the BVR scheme

## 5 The DDoS Attack Against the BVR Scheme

If the probability of success of the above attack to a targeted user is $\frac{1}{10^{20}}$. The probability of success of the attack to any user is $\frac{n}{10^{20}}$. This is a tiny probability, but by attacking many times, we can obtain a significant number of affected users. This can be achieved by deplyoing a botnet of mobile phones into a DDoS attack on the HN.

In the DDoS attack, the mobile bots send many pseudonyms to the targeted HN via a legitimate SN. The HN processes the pseudonyms as they arrive. 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 that are sent to the network 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 pecentage of affected users $E[u_a]$ is

$$E[u_a] = \left(1 - \left(1 - \frac{1}{10^{10}}\right)^m - m\left(\frac{1}{10^{10}}\right)\left(1 - \frac{1}{10^{10}}\right)^{(m-1)}\right) \times 100 \quad (1)$$

See Appendix **??** for the derivation. We have run a simulation of this attack and found that above model is fairly accurate. See Figure 4.

**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 pecentage of affected users $E[u_a]$ is

$$E[u_a] = \begin{cases} \frac{1}{10^{10}} \frac{m^2}{2 \cdot 10^{10}} \times 100, & \text{if } 0 < m \leq 10^{10} \\ \frac{1}{10^{10}}(2m - 10^{10} - \frac{m^2}{2 \cdot 10^{10}}) \times 100, & \text{if } 10^{10} < m \leq 2 \cdot 10^{10} \end{cases} \quad (2)$$

See Appendix **??** for the derivation. We have run a simulation of this attack and found that above model is fairly accurate. See Figure 4. 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.

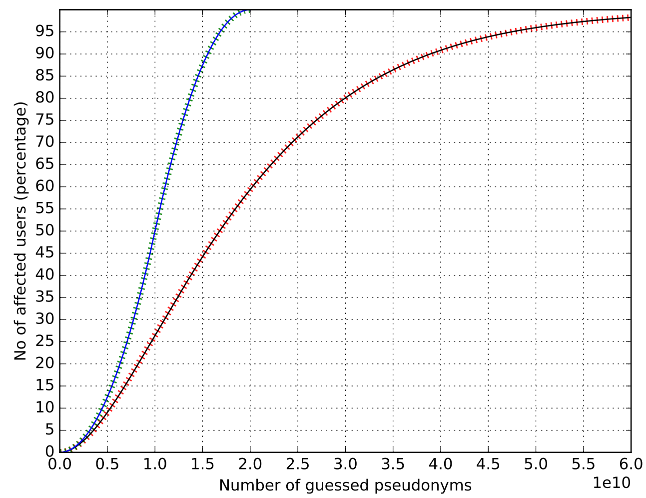


Fig. 4: Success Rate of the DDoS Attack. IMSI space is $10^{10}$. Number of subscribers in HN is $10^7$. The black and blue line presents the expected number of affected users in case of the with and without replacement attacks respectively. Under the black line, there are three red lines which represent the results of three simulations of with-replacement attack. Under the blue line, there are three green lines which represent the results of three simulations of without-replacement attack.

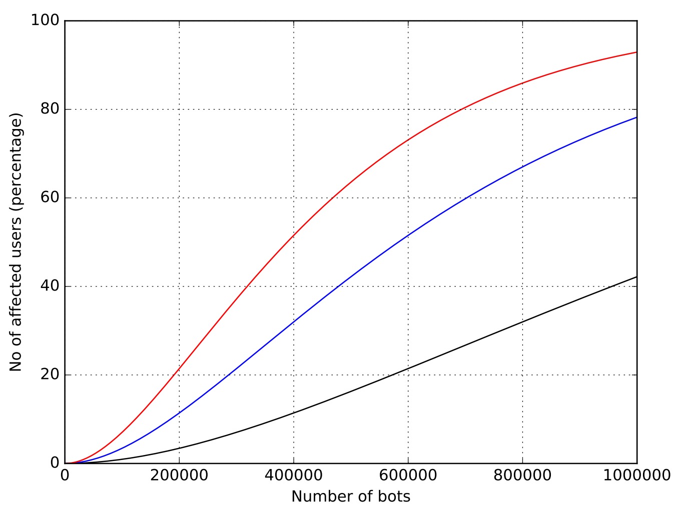


Fig. 5: Success Rate of the DDoS Attack in the case of with-replacement attack as $botnet_{size}$ grows. The black, blue and red lines represent the cases where the parameter $bot_{life}$ has the value of $2, 4$ and 6. In all the cases $AV_{latency} = 500$ milliseconds. The plot is drawn according to Equation 1.

### 5.1 How Fatal The DDoS Attack Can be In Practice

The intensity of the attack in practice will heavily depend on three parameters. The first parameter is the time a mobile bot needs to wait starting from sending a pseudonym to an HN (via SN) to when the RAND and AUTN is received from the HN (via SN). The second parameter is the size of a mobile botnet available to an attacker. The third parameter is the average time duration a mobile bot can be used in the attack before the power of the bot drains out. Let us denote this parameters as $AV_{latency}$,$botnet_{size}$, and $bot_{life}$.

According to a thesis conducted in Lund University in 2016 [5], the EPS AKA has the latency of 550 milliseconds even when the MME is far away (10,000 km) from the HN. The latency in this study is measured as the time from that the UE sends the Attach Request message to when the MME sends the Security Mode Command message. In our attack we do not need the MME and the bot to participate in the challenge and response based AKA protocol. It is sufficient for the bot to make the HN to respond to an AV request message. The bot ignores the RAND and AUTN sent by the SN. So, we can safely assume that it would take at most 500 milliseconds to send a pseudonym to the HN (via SN) and get the RAND and AUTN in response from the HN (via SN). Consequently we set the parameter $AV_{latency} = 500$ milliseconds.

Mobile botnets are on the rise cite. There have already been observed many mobile botnets cite. In 2015, it was reported in cite, that a mobile botnet of 650,000 mobile phones made an attack to a server. Researches cite suggest that these are only the early days of mobile botnets. The number of smartphones by cite year, is estimated to reach cite number. 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 $bot_{size} = 1$ million. Also, let us assume that the mobile bots used in our attack can be used for at least 2 hours on an average before the power of the bot drains out.

Under the above assumptions, our botnet can deploy 2 million bot-hours in the attack. This is equivalent to sending $1.44 \times 10^{10}$ pseudonyms to the network. In the with-replacement attack, by sending $1.44 \times 10^{10}$ pseudonyms, the attacker can kick around 40 percent of the users of the HN out of service. We believe, in the distrbuted without-replacement attack, the affected percentage of users would be between 40 and 80 percent. See Figure 5, it shows the percentage of affected users in the case of with-replacement attack as the size of the botnet grows.

## 6 A Solution To The DDoS Attack

The vulnerability of the pseudonym based solutions is that, the HN forgets an old pseudonym of a legitimate UE before being confirmed that the new pseudonym has been received by the legitimate UE. To mitigate the vulnerability, we look for a solution in which the HN will be acknowledged if the UE has received the new pseudonym. Untill the acknowledgement arrives, the HN will not forget the old pseudonym. But the question is, how the acknowledgement can be generated. We can not introduce a new message becasue we want our solution to be backward compatible with legacy SNs. We need to rely on the existing messages of 3G/4G networks.

There is a location update message that is sent by an SN to the HN after an AKA is successfully and positively run in between the SN and a UE cite. discuss that the location update message goes to a different entity in HN than the HSS, but it is okay. We design our solution by piggybacking this location update message as the desired acknowledgement. In an ideal case, if the location update message is sent by the SN to the HN, it is confirmed that the AKA has run positively and successfully. A successful and positive AKA run implies that the UE has received the new pseudonym. Using this location update message, we present a modified version of the BVR scheme as our solution.

### 6.1 Solution

In our solution, each subscriber $s$ keeps record of the IMSI $i$ and two pseudonyms $PMSI, P_{new}$. The HN keeps record of the IMSI $i$, three pseudonyms $s_p, s_{p'}$ and $s_{p''}$. We also introduce one binary flag $LUF_{p'}$ associated with every subscriber $s$ at the HN end. Along with the location udpate message, an SN also sends the pseudonym of the involved subscriber. $LUF_{p'}$ is set to 1 if the HN has already received a location update message for the pseudonym $s_{p'}$. The flags are set to 0 otherwise. In the begining of the life of a SIM card, it stores the IMSI and two pseudonyms $PMSI$ and $P_{new}$ where $PMSI = s_p, P_{new} = s_{p'}$

HN has to accept whenever the IMSI is sent. because all the SIMs will not be updated

The fundamental idea of the solution is: when a location update message arrives for $s_{p'}$, the HN sets $s_p \leftarrow s_{p'}, s_{p'} = s_{p''}$ and $s_{p''} = null$. But complexity arises in this solution when location update message is delayed, lost, or sent multiple times. Also in practice location update message for pseudonyms $s_p, s_{p'}, s_{p''}$ might arrive in different order because of the inherent characteristics of IP networks. To address this issue, we study the different states of the HN and decide what should be the action at a certain state when the HN receives a certain message. Figure **??** reresents the study. Taking a closer look at the state diagram, you can notice that state 3 is reached when the location update message arrives in an unexpected order. According to this study, we propose the solution as described in the Figure 8.

### 6.2 Analysis of the Solution

**why the solution is good** what happens in the error cases

**function** $g(q, s_i, s_p, s_{p'}, s_{p''}, f_{p'})$
  **if** $q = s_{p'} \wedge s_{p''} = null$ **then**
    **update** $s_{p''} \leftarrow \{0,1\}^{34} \notin \{t_p, t_{p'} | \forall t \in S\}$
  **end if**
  **if** $q = s_i$ **then return** $s_p$
  **if** $q = s_p$ **then return** $s_{p'}$
  **if** $q = s_{p'}$ **then return** $s_{p''}$
  **if** $q = s_{p''}$ **then return** $s_{p''}$
**end function**

Fig. 6: Function $g$

**function** $h(q, s_i, s_p, s_{p'}, s_{p''}, f_{p'})$
  **if** $q = s_i$ **then return** $false, false$
  **if** $q = s_p$ **then return** $false, false$
  **if** $s_{p''}! = null$ **then**
    **if** $q = s_{p'}$ **then return** $true, false$
    **if** $q = s_{p''}$ **then return** $true, true$
  **end if**
  **if** $s_{p''} = null \wedge f_{p'} = 0$ **then**
    **if** $q = s_{p'}$ **then return** $false, true$
  **end if**
  **if** $s_{p''} = null \wedge f_{p'} = 1$ **then**
    **if** $q = s_{p'}$ **then return** $false, false$
  **end if**
**end function**

Fig. 7: Function $h$

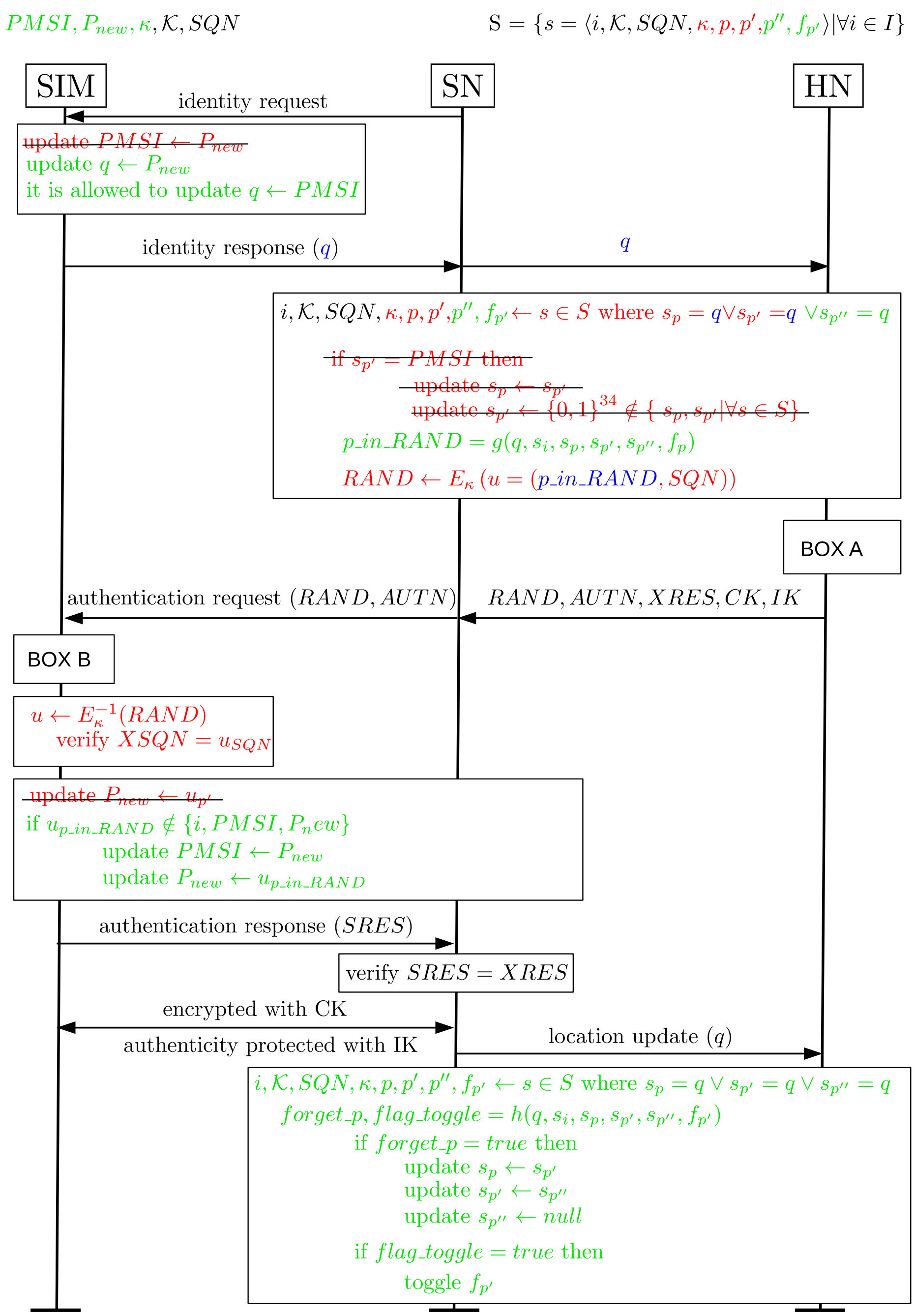


Fig. 8: Solution

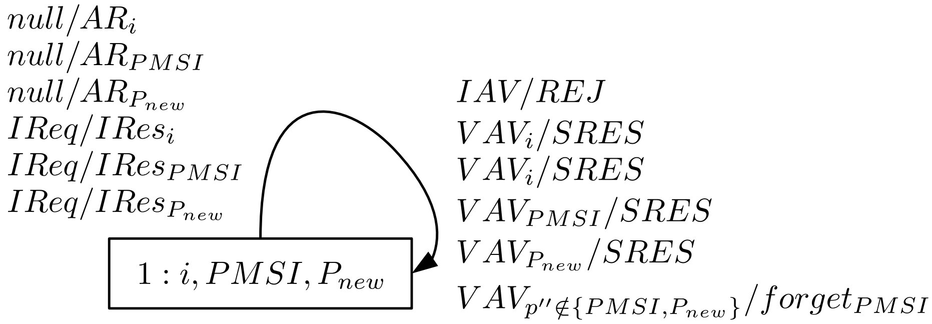


Fig. 9: State diagram of the solution for subscriber $s$ at the UE end. $AR_i$ = attach request using $i$. $IReq$ = identity request. $IRes_i$ = identity response with $i$. $IAV$ = Invalid $AV$. $VAV_i$ = Valid $AV$ that has pseudonym $i$ embeded in the $RAND$. $forget_{PMSI} = PMSI \leftarrow P_{new}, P_{new} \leftarrow p''$

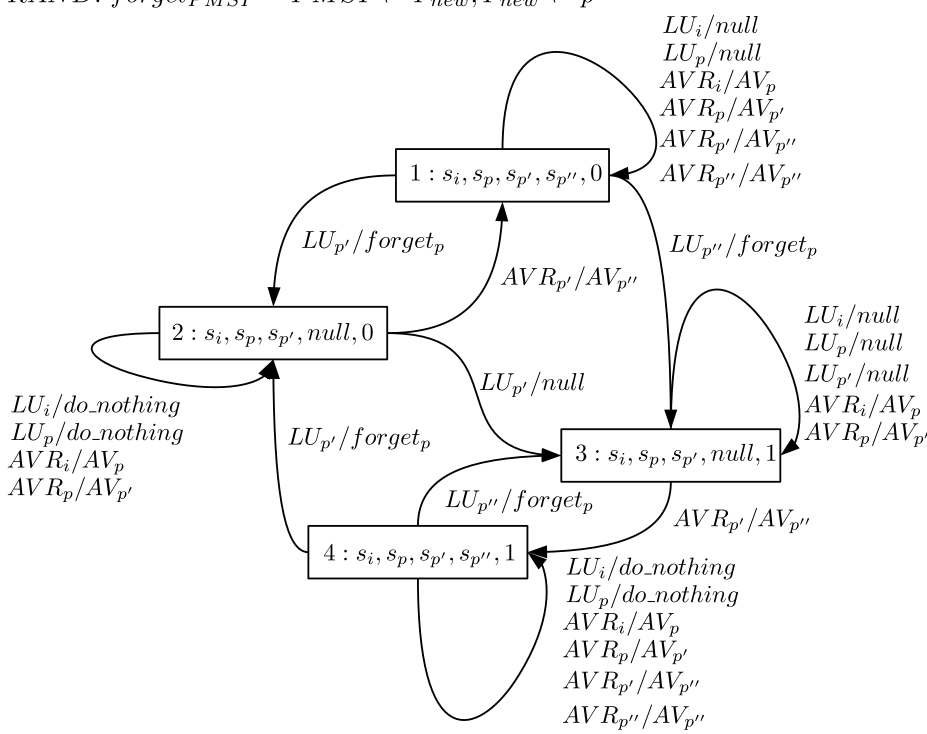


Fig. 10: State diagram of the solution for subscriber $s$ at the UE end. $AR_i$ = attach request using $i$. $IReq$ = identity request. $IRes_i$ = identity response with $i$. $IAV$ = Invalid $AV$. $VAV_i$ = Valid $AV$ that has pseudonym $i$ embeded in the $RAND$. $forget_{PMSI} = PMSI \leftarrow P_{new}, P_{new} \leftarrow p''$

## 7 SN is not a Potential Adversary Anymore

In principle, a malicious SN can still attack the HN by sending a fake location update message for a pseudonym $q$ that is in use by a legitimate subscrber $s$. We will show that the probability of success for such an attack is very low before the SN is detected and stopped. Besides an SN is in 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 losing an important business contract.

### 7.1 How a Malicious SN Could Attack

Without the presence of a malicious or buggy SN, the UE can be in one of the two cases showin in Figure 11. In an ideal situation, Case 1 is expected. In this section We will discuss the attack based on Case 1. However, the success probability of the attack in Case 2 is even smaller.

| Case 1 | Case 2 |
|---|---|
| $PMSI = s_p, P_{new} = s_{p'}$ | $PMSI = s_{p'}, P_{new} = s_{p''}$ |

Fig. 11: Values of $PMSI$ and $P_{new}$ in the absence of a malicious or buggy SN

Let us assume that a malicious SN has sent a fake location update message for a pseudonym $q$ to an HN. If by chance, the pseudonym $(q = s_{p'} \wedge s_{p''}! = null) \vee (q = s_{p''})$ for a legitimate subscriber $s$, then the HN forgets $s_p$. To avoid the conditions on $s_{p''}$ being null, the malicious SN might send $AVR_q$ so that $s_{p''}$ is set to a non-*null* value. Consequently, the attack consists of two consecutive messages. The malicious SN first sends $AVR_q$ and wait. After receiving the $AV$ from the HN, the malicious SN sends $LU_q$ to the HN. The attack has two phases:

**Phase 1** The malicious SN sends $AVR_{q_1}$ and $LU_{q_1}$. If by chance, $(q_1 = s_{p'}) \vee (q_1 = s_{p''})$, then the HN forgets $s_p$. At this stage the state of the subscriber at HN becomes $(s_i, s_p, s_{p'}, null, 0)$. The situation of the subscriber in the UE becomes $(PMSI \notin \{s_p, s_{p'}\}, P_{new} = s_p)$

**Phase 2** The malicious SN sends $AVR_{q_2}$ and $LU_{q_2}$. If by chance, $(q_2 = s_{p'})$, then the HN forgets $s_p$. At this stage the state of the subscriber at HN remains $(s_i, s_p, s_{p'}, null, 0)$. The situation of the subscriber in the UE becomes $(PMSI \notin \{s_p, s_{p'}\}, P_{new} \notin \{s_p, s_{p'}\})$

### 7.2 Probality of Success of the Attack

A malicious SN has to successfully guess two pseudonyms $q_1, q_2$ to affect a subscriber $s$. However, if the subscriber $s$ is currently connected to the malicious SN, $q_1$ does not need to be guessed. The SN can collect the $P_{new}$ of all the subscribers connected to it by making identity requests to the UEs. Then for each $P_{new}$, the malicious SN performs the Phase 1 of the attack discussed in Section 7.1.

However, the malicious SN has no way to know the new $s_{p'}$ the HN has set for a subscriber $s$ after the Phse 1 of the attack. Consequently the HN has to guess $q_2$ to mount the second phase of the attack. If the malicious SN guesses with replacement, the probability of one guess to be successful in the seond phase is $\frac{r}{10^10}$ where $r$ is the number of subscribers of the HN currently visiting the malicious SN. Figure 12 shows the expected number of affected subscribers as the number of guess grows. The expected number of affected subscribers are computed as $\left(1-\left(1-\frac{1}{10^{10}}\right)^m\right)$ where $m$ is the number of pseudonyms guessed. However, if the pseudonyms are guessed without replacement, the number of affected user will be a bit higher. But we believe it will still be very insignificant comparing with the number of pseudonyms have to be guessed. The malicious SN can be detected and be blocked far before it reaches guessing 1 million pseudonyms

However, the malicious SN can target the subscribers of an HN who are not even visiting the malicious SN. In that case the malicious SN has to guess both of the pseudonyms $q_1$ and $q_2$. If the pseudonyms are guessed without replacement then the expected number of affected subscribers would be as following:

$$E[u_a] = \begin{cases} \frac{1}{10^{10}} \frac{m^2}{2 \cdot 10^{10}} \times r, & \text{if } 0 < m \leq 10^{10} \\ \frac{1}{10^{10}} (2m - 10^{10} - \frac{m^2}{2 \cdot 10^{10}}) \times r, & \text{if } 10^{10} < m \leq 2 \cdot 10^{10} \end{cases} \tag{3}$$

Figure shows how it grows as $m$ grows with varied $r$

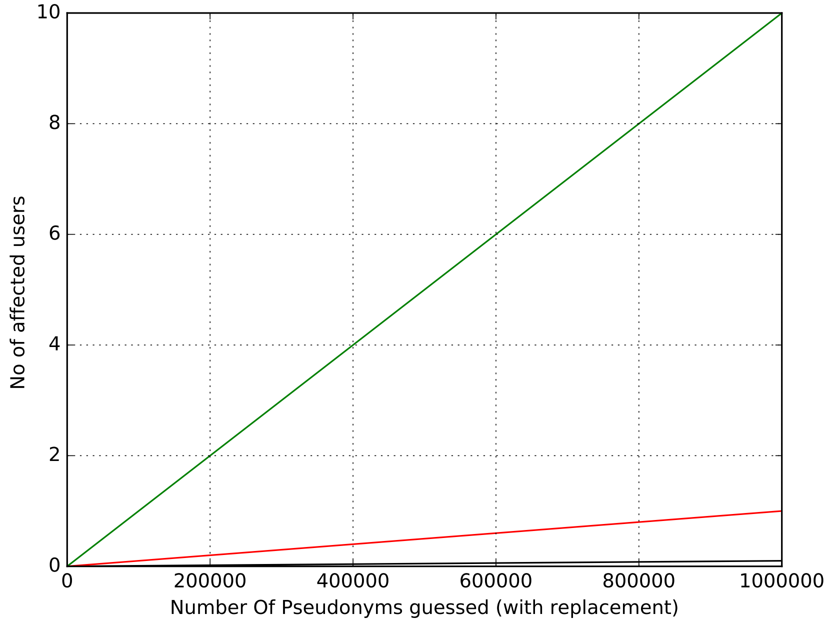


Fig. 12: Expected number of affected subscriber in the attack by SN. The attack is targeted to the subscribers who are visiting the SN

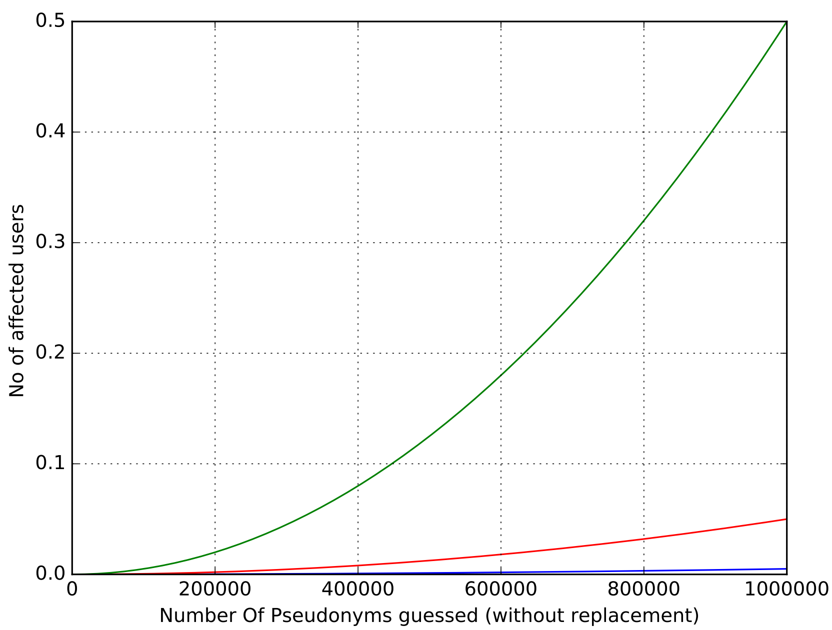


Fig. 13: Expected number of affected subscriber in the attack by SN. The attack is targeted to all subscribers of the HN

## 8 Usability of pseudonyms

## 9 Conclusion

**Acknowledgement.**

## References

1. 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)
2. 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)
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. 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)
5. Ahlström, M., Holmberg, S.: Prototype Implementation of a 5G Group-Based Authentication and Key Agreement Protocol. In: 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. (2016)

## References