Implementing pseudonym based approach to secure the long-term-identity privacy of mobile users using 5G enabled mobile-phones, home-networks along with legacy serving-networks and USIMs

Khan, Niemi, Akman
Department of Computer Science
University Of Helsinki
Helsinki, Finland

November 10, 2016

Abstract

To secure the privacy of long term identity of mobile phone users, recently two different approaches have been published. One is the use of pseudonyms and the other is the use of public-key cryptography where the public-private key pair is owned by the home network (HN). In this article, the challenges which will come forward while implementing those two approaches are discussed. In some cases, plausible solutions are proposed.

Introduction

An IMSI (Internation Mobile Subscriber Identity) which is the long term identity of a mobile phone subscriber is a string composed of 15 decimal digits. It has three parts: mobile country code (MCC), mobile network code(MNC), and mobile subscriber identification number (MSIN) where

$$IMSI = MCC||MNC||MSIN$$

Pseudonym and public-key cryptography, in both of the approaches, the MCC and MNC are sent across the mobile network in clear text and focus on the privacy of MSIN part only. We will also focus on the same principle in this article and will mean the privacy of MSIN when we mention about the privacy of IMSI. Note that there are only 10 decimal digits represented by 40 binary bits available to represent an MSIN (cite).

The Pseudonym scheme

Goals:

1. Privacy of the long-term-identity from any outsider and also from honest but curious insider by using a temporary identity of a user recognized by HN. This temporary identity is also called a pseudonym.

2. The scheme has to be compatible with the legacy USIMs and serving networks (SN) so that the scheme can be brought into use only by modifying the mobile equipment (ME) and HN.

3. Privacy of the pseudonym even from the SN. That is, the SN involved in an AKA will not know the pseudonym of a user which the user will use in another AKA when he moves to another SN.

The scheme:

1. Every user having an IMSI is given a pseudonym P by the HN initially.

2. When the user identifies itself with P to the SN, the SN forwards P to the HN

3. In response, along with the authentication vector (AV) and pseudonym P, the home subscription server (HSS) in the HN sends a new pseudonym $P^{'}$ to the SN encrypted by the secret key K or by some other secret key P_{pseudo} derived from K.

4. The SN runs the AKA with UE based on the AV it received from the SN. It also sends the encrypted new pseudonym $P^{'}$ to the UE.

5. If the AKA is successful then the ME will be able to decrypt it and obtain $P^{'}$. The old pseudonym P will be continued in use instead of IMSI until the UE tries to connect via another SN.

6. When the UE tries to connect to the network via a new SN, the new SN has no knowledge about P. At this point instead of sending the IMSI, the UE identifies itself with $P^{'}$ which leads into a new AKA run in the similar fashion mentioned in step 3

7. Whenever a new AKA is run, the HN generates a new pseudonym $P^{''}$ in the similar fashion mention in step 3, replaces P with $P^{'}$ and $P^{'}$ with $P^{''}$

The implementation issues:

Here we describe the issues of the above scheme that we found while implementing it:

1. How the HN assigns the first pseudonym P to a user is an implementation issue with the mentioned scheme. A pseudonym can't be assigned to a USIM because the scheme needs to be compatible with legacy USIMs to achieve the goal 2. The scheme also can not assign a pseudonym to an ME because a user is not identified by the ME but by the USIM. Besides, many USIMs can be inserted in in and ME over it's lifetime.

2. The pseudonym $P^{'}$ can not be encrypted by the master key K or some other key derived from K. Because in that case, the decryption has to happen in USIM as the ME doesn't know the master key K. But the scheme needs to be compatible with the legacy USIM and such decryption is not a functionality of the legacy USIMs. Also this secret key needs to be unknown by the SN to achieve the goal 3.

3. This scheme requires the modification in the legacy SNs, because the SN needs to be able to receive the encrypted pseudonyms and send them to the UEs. This is not acceptable because the scheme needs to be compatible with legacy SNs according to goal 2.

4. Another issue with this scheme is, every user is associated with an IMSI and two pseudonyms. These 3 distinct IMSI looking strings for one user

create pressure in the space of valid IMSIs or pseudonyms. The size of this space is only 10^{10}. This is because in the standard, IMSI is defined to be composed of decimal digits (cite) and there might have the existence of SNs which strictly check on this constraint to validate an IMSI.

5. This is already mentioned in the original paper that the pseudonyms have to look like the original IMSI so that the scheme becomes compatible with the legacy SNs. But it doesn't discuss how to generate such pseudonyms which are uniformly distributed in the pseudonym space.

6. It needs to be ensured that the newly generated pseudonym is not already in use as an IMSI or a pseudonym for a different user in the HN.

7. Another important issue is, how to ensure the synchronization of the pseudonyms in use for a user in between the ME, SN and HSS. Apparently, the HSS needs to know that the UE has been successful in the AKA before it updates the entry of that user with the new pseudonym. Indeed, otherwise the HN and UE will have different pseudonyms if the AKA fails. To ensure it, the SNs has to acknowledge the HNs about the result of the AKA. This is not the case with the legacy SNs. However, even if an acknowledgement mechanism is introduced in the upcoming 5G networks, the acknowledgment itself might be lost even when the AKA is successful. Based on this weakness we can mount an active attack to make the pseudonyms in HN and ME inconsistent by setting up a fake base station. Here we describe the attack:

   (a) The attacker sets up a fake base station impersonating a legitimate SN and asks for the long term identity of the user

   (b) The user responds to the query by sending it's pseudonym

   (c) The attacker sends the received pseudonym to a legitimate SN leading to run an AKA in between the legitimate SN and the attacker. This AKA will cause the HN of the victim to update the entry of the victim with a new pseudonym. On the other hand, the victim has no knowledge about this AKA or the new pseudonym that the HN has updated the victim's entry with in the HSS. As a result, the victim will not update the pseudonym

in the ME. Which means the synchronization of the pseudonyms in between ME and HN is lost and all the consequent AKA will fail even when the victim tries to connect with a legitimate SN

8. Note that, in this scheme the current SN knows the pseudonym P that the user used during the last AKA and the SN also keeps it stored and use in error messages of a failed AKA. This pseudonym P is also used to page for the UE. It is possible that the pseudonym P of a user U_1 in the old SN (SN_1) will be kept stored and not deleted even when the user has moved to another SN (SN_2). Now if U_1 participates in an AKA in SN_2 then the user's pseudonym will be updated in HN and SN_2. Now if some other user U_2 comes in SN_1 and run an AKA, there is a tiny probability that user U_2 will be assigned with the pseudonym P by the HN. This will cause the SN_1 to have more than one entries with one pseudonym even though there is no such conflict in the HN.

9. (The scheme might have some issue with the handover process from one eNodeB to another eNodeB, or from one MME to another MME. This issue needs to be checked closely)

Potential Solutions

Solution to issue 1:

Apparently there is no solution to the issue 1 that uses a pseudonym and still works with the legacy USIMs. This leaves the only option of using the IMSI for the very first time. Fortunately using IMSI in such a manner still gives us the benefit of using a pseudonym. Because, using this IMSI, only for the very first time, and never after, makes the IMSI effectively a pseudonym itself. Besides, in a situation of such a limited use of the IMSI, an attacker interested in the IMSI of a target user has to eavesdrop on the target's AKA messages exactly at the very first time the target connects to the network.

Solution to issue 2:

To solve the issue 2, the secret key used for encrypting the new pseudonym $P^{'}$ can be CK, or IK, or generated from either one or both of them. This

will work if the SN is of LTE. However, this solution fails to achieve the goal 3 if the SN is of UMTS. Because CK and IK are known by an SN of UMTS.

Solution to issue 3:

One solution to the issue 3 could be to send the encrypted pseudonym as part of the RAND of AV. But there is a chicken-egg problem here. The keys CK and IK are derived from RAND. So, if the RAND includes the encrypted pseudonym then we need to know the CK and IK first. That is, we can't generate CK and IK without knowing RAND and we can't have RAND without knowing CK and IK. Figure 1 shows the situation pictorially. To Solve the chicken-egg problem, we propose to generate the pseudonym $P^{'}$ separately both in HSS and ME end by encrypting the RAND which will be sent from HSS to SN as part of the AV in anyway. Our idea is to use keyed message authentication code (MAC) of the RAND to use as the pseudonym $P^{'}$.

Figure 1: The chicken-egg problem

Solution to issue 5 and 6:

To solve the issues 5 and 6, we have a proposal here:

1. Choose $RAND$ in the HSS using a random number gnerator. (Use the same random number generator that is used in the existing HSSs)
2. Generate CK and IK from the RAND. (Use the same mechanism which is used in LTE)

3. Generate a message authentication code (MAC) of $RAND$ using a key generated from CK and IK (An appropriate HMAC has to be chosen)

4. Lets say the MAC is $M = a_1 \cdots a_n$ where a_i is the i-th bit of the code

5. Let us represent M as a sequence of m hexadecimal digits, that is $M = a_1 \cdots a_n = h_1 \cdots h_m$ where $m = n/4$ and h_i is the hexadecimal value represented by the bit string

$$a_{(i-1)*4+1}a_{(i-1)*4+2}a_{(i-1)*4+3}a_{(i-1)*4+4}$$

6. Consider a function $f : \mathbb{Z}_{15} \times \mathbb{Z}_2 \to \mathbb{Z}_{10}$ where

$$f(h_i, a_n) = \begin{cases} h_i, & \text{if } 0 \leq h_i \leq 9 \\ 0 + a_n, & \text{if } h_i = A \\ 2 + a_n, & \text{if } h_i = B \\ 4 + a_n, & \text{if } h_i = C \\ 6 + a_n, & \text{if } h_i = D \\ 8 + a_n, & \text{if } h_i = E \end{cases}$$

7. For every i, if h_i is not equal to F, compute $d_i = f(h_i, a_n)$ until 10 d_is are obtained. If more or equal to $(m - 9)$ h_is are F, then go back to step 1.

8. The pseudonym is constructed by concatenating the 10 d_is in the same sequence they were were obtained.

9. If the resultant pseudonym is already used as a pseudonym of IMSI, then go back to step 1 (An appropriate data structure needs to be chosen to make this search less expensive)

One issue with the above approach is, the HSS might end up in generating shorter or duplicate pseudonyms in consecutively many attempts. To minimize the probability of generating duplicate pseudonyms we need to ensure the uniform distribution of the pseudonyms and show that the HSS will be able to generate a valid pseudonym within an acceptable number of attempts.

- **Distribution of Pseudonyms:** We assume that the distribution of $RAND, CK, IK$ and the digests of the HMAC used to generate the

pseudonyms are uniformly distributed. If the above assumption is true then M is uniformly distributed. Which means $P(h_i = h) = 1/16$. Then $P(d_i = d)$ is equal to either $1/16 + (1/16) * P(a_n = 0)$ or $1/16 + (1/16) * P(a_n = 1)$. Now as M is chosen from a uniform distribution, $P(a_n = 0) = P(a_n = 1) = 1/2$ which makes the distribution of d_is uniform and consequently the distribution of pseudonyms is uniform.

- **How Many Attempts the HSS needs:** There are two situations in which the HSS might end up in generating an invalid pseudonym. One is, more or equal to $(m - 9)$ hexadecimal digits h_i extracted from M are F. Another is, the resultant pseudonym is already in use. The probability of the first case is

$$\sum_{i=m-9}^{m} \binom{m}{i} (1/16)^i (15/16)^{m-i}$$

$$= \sum_{i=m-9}^{m} \binom{m}{i} \frac{15^{m-i}}{16^m}$$

It can be shown that when m is reasonably large this probability is an extremely small number that can be safely ignored. (I will come up with some upper bound of this probability if Valtteri thinks it is required).

The probability for the second case depends on the number of subscribers present in the HSS associated with a valid IMSI. Let us consider this number to be N. Then there are at most $3N$ IMSIs or pseudonyms generated and in use at a given time. At this point of time, if a pseudonym is chosen from a uniform distribution, then the probability of the choice to collide with an existing IMSI or pseudonym is $3N/10^{10} \approx 3N/2^{33.2}$. Now if we set $N = 2^r$, then the probability of such a conflict is $3/2^{33.2-r}$. And consequently the probability that the HSS will need q number of attempts to generate a pseudonym that doesn't conflict with any existing is $(\frac{3}{2^{33.2-r}})^{q-1}$. This probability can become significantly high for an operator having more than 500 million subscribers. Nevertheless, by introducing multiple MNC for a single operator, it might become possible to reduce such probability

to a usable level. (we can draw a graph to show how the probability increase/deccrease for different values of q and r)

Solution to issue 7

A simple solution is that the user will send the IMSI when the AKA is failed consecutively for a configured number of times. However such a solution will bring back the similar IMSI catcher attack which is possible with the existing mobile networks.

Another solution is to keep track of last two pseudonyms P_i and P_{i+1} and generate a new pseudonym $P^{'}$only when the HN receives P_{i+1} from a legitimate SN. If the HN receives P_i then it responds with the same AV that was used while generating the pseudonym P_{i+1}. One issue with this solution is, the HN can be forced to send the same AV in successively many failed AKAs. (We need to check closely if this is bad or not)

To mitigate the issue of reusing the same AV, another solution would be to use the approach of sending the new generated pseudonym by encrypting and embedding in the RAND. Such an approach will solve the issue of reusing the same AV but of course will bring back the chicken-egg problem for the key used for the encryption of the pseudonym. One solution of the chicken-egg problem is to run the AKA successively twice druing the first time the user connects to the network. During the first AKA, the user sends the IMSI, and after the AKA is completed, the HN and ME have CK,IK and the SN and ME have the TMSI. During the second AKA, the user sends the TMSI. During this AKA the HN generates a new pseudonym and a new pair of CK, IK. The HN encrypts the generated pseudonym with old CK,IK and embed it in the RAND. In this case also the HN maintains the history of last two generated pseudonyms P_i and P_{i+1}. The HN generates a new pseudonym only when it receives P_{i+1}. The pseudonym can be generated in the same way as described above. The encryption of the pseudonyms can be done by 64-bit KASUMI. The original pseudonym is of 40 bits. So there is an opportunity of adding 24-bit long salt with the pseudonym before encryption. After the encryption, we will get 64-bit ciphertext. But the RAND is 128 bit. So there is 64 more bits left to chose a random value. Consequently, the RAND used in successive failed AKAs are quite different

from each other. (check numerically, how much different they are from each other. Now, the keys used for encrypting the pseudonyms can change over the time as more and more AKA take place. One choice is to use the CK,IK which were generated during the AKA when P_i was generated, to encrypt pseudonym P_{i+1}. Another choice is to use the same CK,IK always which were generate during the very first AKA. (Both of the possibilities have to be closely evaluated with their pros and cons.) However, in both cases, the CK and IK has to be converted into a key usable by KASUMI. As KASUMI uses keys of size 128 bits, and both CK and IK are of 128 bits, it is easy to generate a key for KASUMI by doing bitwise XOR in between CK and IK. The 64-bit long encrypted pseudonym can become the first or last 64 bits of the RAND. The position of the pseudonym bits in the 128-bit RAND is a public information, hence it is sufficient to embed the encrypted pseudonym in the RAND in an arbitrarily simple manner.