Subscription Identifier Privacy in 5G Systems

Enrique Cobo Jiménez*
School of Electrical Engineering
Royal Institute of Technology (KTH)
Stockholm, Sweden
ejcj@kth.se

Prajwol Kumar Nakarmi, Mats Näslund, Karl Norrman
Security Research
Ericsson AB
Stockholm, Sweden
{prajwol.kumar.nakarmi, mats.naslund, karl.norrman}@ericsson.com

***Abstract*—Privacy is a main concern for mobile network users, and there are many proposed enhancements for the protection of the long-term subscription identifier. Some enhancements require asymmetric key operations, which increase both processing requirements and protocol message sizes. To the best of our knowledge, there has been no practical implementation feasibility study of these enhancements using commodity mobile devices. Neither is it clear whether the enhancements are sufficient. This paper highlights privacy weaknesses, when the long-term subscription identifier is used in Paging procedures, and proposes new ways to resolve these. Further, the paper evaluates an Android implementation of one of the enhancements, which includes the asymmetric scheme Elliptic Curve Integrated Encryption Scheme (ECIES). We conclude that it is feasible to implement asymmetric encryption methods for the long-term subscription identifier and that the highlighted privacy weaknesses can be efficiently countered. This removes another set of obstacles for realizing the protection in mobile network standards.**

Keywords—5G; Privacy; IMSI; Paging; ECIES; Android

I. INTRODUCTION

There are growing societal concerns for user privacy due to various electronic fingerprints left by increased use of ICT. Mobile networks are of special concern due to the relative ease of gaining access to the communication medium (radio signals) and the support for mobility, which comes with an attached downside of potentially permitting tracking of subscribers' locations and geographical movements. For these reasons, already GSM introduced a basic subscription identifier protection mechanism, enabling the use of short-term subscription identifiers, taking the place of the fixed, long-term subscription identifiers, i.e., International Mobile Subscriber Identity (IMSI) when possible. There are however situations when the mobile device, known as the User Equipment (UE), is simply forced to use the IMSI (e.g., in Initial Network Attach requests, which takes place before the subscriber has been authenticated and before a secure link has been established). This potential shortcoming was studied both when UMTS standardization started, and again when LTE was to be introduced. For various reasons, however, the complexity of adding protection was judged not to be motivated by the severity of the threats.

Therefore, the use of short-term subscriber identifiers provided partial protection against passive attacks on privacy, but active attacks on privacy (such as the ones mentioned in [1]) were still left completely unaddressed. With increased off-the-shelf availability of technology such as Software Defined Radio, those active attacks have become a major concern, with a number of attack vectors recently exploited [2].

As we are now moving into the era of fifth generation (5G) networks, attention to the privacy issues around long-term subscription identifier exposure is again in focus, and various enhancements are being discussed [3].

One type of enhancements proposed in [3] encrypts the subscriber's IMSI in the Initial Network Attach requests sent in the uplink (from the UE to the network) by the use of asymmetric cryptography. One particular enhancement (described in Clause 5.7.4.15 of [3]), proposes to use ECIES without a Message Authentication Code (MAC). That variant of ECIES is denoted by ECIES* in the sequel. Enhancements of the type to which ECIES* belongs are however insufficient because there are occasions when the IMSI may also be transferred in the downlink Paging message (from network to UE) [4, 5].

The standardization body governing the standards for mobile networks is 3GPP [6]. Based on the technical reports so far produced by 3GPP for 5G, we expect 5G procedures to have similarities to LTE procedures. For example, there will be procedures for Attach and Paging. Therefore, the rest of this paper uses the architecture of LTE and show how LTE can be enhanced for privacy in a way that is useful also in a 5G context.

In this paper, we make two main contributions. First, we evaluate the feasibility of implementing IMSI encryption using ECIES* in Android-based devices. Second, we analyze privacy aspects of the Paging procedure in detail and investigate the possibility of a new protection mechanism that protects the IMSI in both Initial Attach and Paging procedures. This turns out to be challenging due to a not previously examined bandwidth limitation, which we however are able to overcome.

The paper is organized as follows. Section II discusses relevant background in mobile network architecture, ECIES, use of ECIES* to encrypt the IMSI, and the existing Paging procedure. Section III presents the evaluation of ECIES* in Android-Based devices. Section IV discusses our proposals for privacy enhancements to the Paging procedure. Section V puts the paper in the context of related work. Section VI concludes the paper.

* The first author was with Ericsson Research during this work

II. BACKGROUND

In this section, we first briefly introduce a LTE network and how it is expected to relate to a 5G network. Then we describe the ECIES scheme and its variation (ECIES*), and how the latter can be used in 3GPP for the encryption of the IMSI in the Initial Attach procedure for 5G. Finally, we outline the Paging procedure.

A. Network architecture

A simplified LTE network is shown in Fig. 1. The UE is wirelessly connected to a serving network via an E-UTRAN Node B (eNB) which belongs to the Radio Access Network (RAN). An eNB is connected to the Core Network (CN) entity called Mobile Management Entity (MME), which acts as security and mobility anchor. The MME is responsible for authenticating the UE, for which purpose it fetches necessary information from a Home Subscriber Server (HSS). The HSS is a CN entity in the home network that contains a database of all its subscribers. On the UE side, the credential needed for the authentication is stored in the Universal Subscriber Identity Module (USIM).

B. Elliptic Curve Integrated Encryption Scheme (ECIES)

ECIES is a type of asymmetric encryption scheme based on Elliptic Curve Cryptography (ECC). ECIES is constituent of different functions like Key Generation (KG), Key Agreement (KA), Key Derivation (KD), Hash, Encryption, and Message Authentication Code (MAC). The encryption function in ECIES is symmetric; however, the key used in the encryption is generated using the ECC public/private key pairs of the communicating parties.

We refer to the Standards for Efficient Cryptography Group (SECG) specification, SEC1 [7], for detailed description of ECIES. Also, the authors of [8] have nicely summarized the functions mentioned above according to various standards, viz. ANSI X9.63 [9], ISO/IEC 18033-2 [10], together with SECG SEC1 [7].

C. Using ECIES* to Encrypt the IMSI

The 3GPP Technical Report (TR) 33.899 [3] contains solution proposals to encrypt the IMSI, using asymmetric cryptography, in the uplink network Attach request. To address the potential bandwidth issue concerning the size of the encrypted IMSI, there is also a proposal in which the encrypted IMSI is used only on rare occasions, and normally short-term identifiers (called Pseudonyms) assigned to the UE by the HSS are used. Fig. 2 illustrates a simplified Initial Attach procedure using encrypted IMSI, using the term Pseudonym as optional for generalization.

Further, for the encryption of the IMSI, there is a proposal in [3] to use ECIES*, a variant of ECIES. The use of ECIES* stands out as a very attractive solution for encrypting the IMSI because it is efficient, very flexible, and there is also a proof of security (holding under certain assumptions) [7]. The encryption and decryption process of that proposal, performed at the UE and HSS side respectively, are illustrated in Fig. 3. Examples of implementing various functions are Elliptic Curve

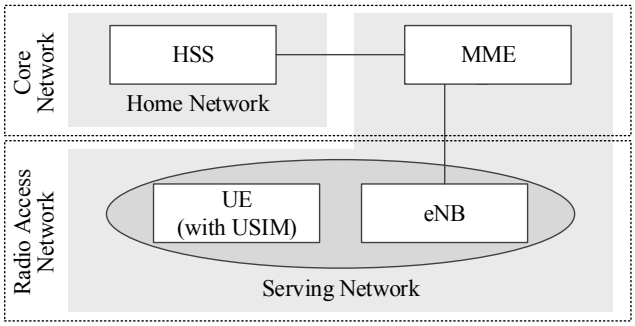

Fig. 1. Simplified LTE network architecture (illustrating a roaming scenario)

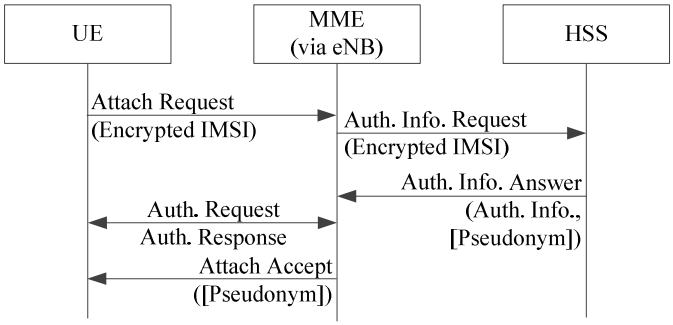

Fig. 2. Simplified Initial Attach procedure using encrypted IMSI

Diffie-Hellman (ECDH) for KA, X9.63-KDF with SHA-2 for KD, and AES in Counter mode for Encryption.

The format of the encrypted IMSI in that proposal includes the Mobile Country Code (MCC) and the Mobile Network Code (MNC) in clear-text, the encrypted Mobile Subscription Identification Number (MSIN), and the UE's ECC ephemeral public key, required for decryption. That proposal preserves the size and format of the MSIN even after encryption and allows to easily locate the UE's home network due to MCC and MNC being unencrypted.

The home network's ECC public key is static and pre-provisioned at the UE. Therefore, the UE is able to send the encrypted IMSI in the very first Attach request. The UE's ECC public/private key pairs are ephemeral and freshly generated by the UE, every time the IMSI needs to be encrypted. It means that the derived symmetric encryption key is also ephemeral such that different keys are used to encrypt the same IMSI at different times. Since the UE's ephemeral keys are not bound to the UE's identity, an integrity tag does not serve any purpose and therefore, that proposal does not use the MAC function present in normal ECIES (hence the scheme is named ECIES*).

When a 256-bit elliptic curve with point compression is used, the size of the ephemeral public key is 256 bits, plus sign indication if needed. The IMSI is 15 digits or 60 bits maximum in Binary-Coded Decimal (BCD) encoding (MCC/MNC is 24 bits, and the MSIN is 40 bits, both being maximum values). Therefore, the total size of the encrypted IMSI when using that proposal is in the order of 320 bits, using a length-preserving encryption scheme for the MSIN part. We will refer to above format and size of the encrypted IMSI later in this paper.

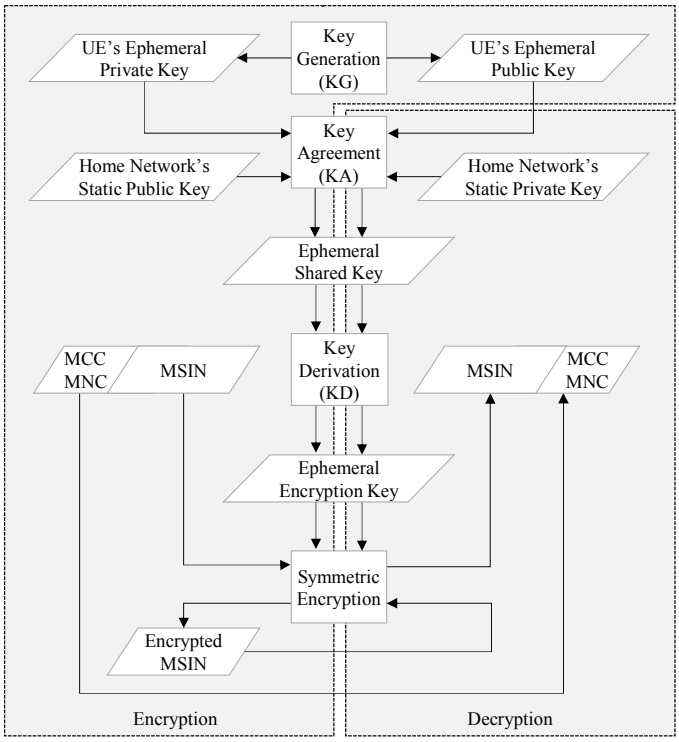

Fig. 3. Using ECIES* for encrypting and decrypting IMSI

D. Paging Procedure

A Paging procedure is a network-initiated procedure, used to inform UEs of incoming services, new or updated system information, and public warnings. Fig. 4 illustrates a simplified Paging procedure where the MME initiates the Paging procedure by sending the UE's Paging identifier to the eNB; the eNB sends the Paging message over the air; and the UE responds to the Paging by sending an appropriate message to the MME via eNB.

The Paging procedure for the 5G systems is expected to be similar to the one in the existing LTE systems with some improvements, e.g., the RAN itself being able to initiate the Paging procedure. The rest of this section discusses some technical details of the Paging procedure regarding LTE systems [11, 12, 13]. However, only those aspects relevant to protecting the identifier are discussed in this section,

The UE checks for a Paging message every ~0.32 seconds to ~2.56 seconds, depending on the value of the Discontinuous Reception (DRX) cycle [12]. The radio frame and the subframe which may contain the Paging message are respectively called the Paging Frame (PF) and the Paging Occasion (PO). Table I shows some selected Information Elements (IE) contained in the Paging message. Among other IEs are the UE Identity Index and the UE Paging Identity, sent from MME to the eNB in the S1 Application Protocol (S1AP) Paging message. The UE Identity Index is used at the eNB to calculate the PF and is defined as (*UE Identity Index = IMSI mod 1024*). The UE Paging Identity is either the SAE-Temporary Mobile Subscriber Identity (S-TMSI), which is a short-term identifier, or the IMSI, which is a long-term identifier.

The eNB can Page several UEs in a batch fashion, i.e., a single Radio Resource Control (RRC) Paging message can contain up to 16 UE identifiers. The maximum sizes of the S-TMSI and the IMSI used in the Paging message are 40 bits and 84 bits respectively. Therefore, the maximum number of bits representing the UE identifiers that can fit in a single RRC Paging message is 1344 (*16 * 84 = 1344*). The size of a whole RRC Paging message is dependent upon the maximum downlink transport block size (TBS) (e.g., the UEs with downlink categories 0 and M1 have maximum TBS of 1000 bits [14]). However, we will use the size of 1344 bits as a reference later in this paper.

In normal scenarios, Paging with the short-term S-TMSI identifier is sufficient. However, Paging with the IMSI still can occur [5], to cover the following abnormal cases when the MME has lost the UE's context (related to security, mobility management), and:

- there is mobile terminating Packet Switched (PS) service that associates the UE with the UE's IMSI;
- the MME provides interoperability with legacy networks, and there is mobile terminating Circuit Switched (CS) service that associates the UE with the UE's IMSI.

When the UE is Paged using the S-TMSI, the UE responds by sending a service request to the MME using the same S-TMSI. Similarly, when the UE is Paged using the IMSI, the UE responds by sending an Attach request to the MME using the same IMSI. In the rest of this paper, unless explicitly clarified, we will use the term Paging to refer to the above mentioned IMSI based Paging.

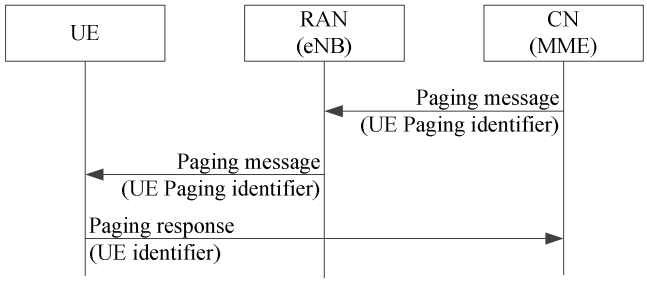

Fig. 4. Simplified Paging procedure initiated by the Core Network (CN)

TABLE I. SELECTED INFORMATION ELEMENTS OF PAGING MESSAGE IN LTE

Protocol	**Information Element**	**Value**
S1AP (MME-eNB)	UE Identity Index	0 to 1023
	UE Paging Identity	S-TMSI or IMSI
RRC (eNB-UE)	Paging Record List	Paging Record (**up to 16**), each containing UE Identity (either S-TMSI or IMSI)

III. Evaluation of ECIES* in Android-Based Devices

This section describes the evaluation of an implementation of the proposed enhancement discussed in Section II.C (Fig. 3) that uses ECIES* to encrypt the IMSI. We ran experiments on Android-based devices to evaluate the proposed enhancement. Our experiments used our standalone test application that calculates the time taken for each of the KG, KA, KD and symmetric encryption ECIES* functions.

In Section III.A, we discuss which Android-based devices the experiments were run on, which crypto libraries were used, and the implementation of the test application. In Section III.B, we then analyze the results from the experiments concerning overhead on computation time and bandwidth.

A. Experiment Setup

1) Targeted Android Devices

We targeted our experiments on four Android-based devices with varying computational capacities, listed in Table II. Those devices more or less represent the types available in the market (not in terms of vendor, but in terms of computational capacities). We will refer to the CPU specifications later in the paper when referring to these devices.

2) The ECIES Functions and Crypto Libraries

The various functions used in the ECIES* scheme (viz., KG, KA, KD and symmetric encryption), discussed in Sections II.B and II.C, have to be instantiated by concrete functions. We tested two elliptic curves in our experiments for use in KG and KA. The first is NIST P-256 [15], which is recommended by the National Institute of Standards and Technology [16]. The second is Curve25519 [17], which has become the most popular alternative to NIST P-256. For KA and KD, we chose Elliptic Curve Diffie-Hellman (ECDH) and X9.63-KDF with SHA-2 respectively, according to the SECG SEC 1 specification of ECIES [7]. For the symmetric Encryption, we evaluated both methods defined in SECG SEC1, the first being AES in Counter mode (where the MSIN is XORed with the output of encrypting a pre-agreed counter with the ephemeral key [7]), and the second being XOR (where the MSIN is XORed directly with the ephemeral key [7]). None of these encryption methods expands the length of the MSIN.

We chose two popular crypto libraries for our experiments, i.e., OpenSSL v1.1.0c [18] and Nettle v3.3 [19], which include all the implementations mentioned above we needed.

We cross-compiled both OpenSSL and Nettle using the arm-linux-gnueabi toolchain, available as {gcc,binutils}-arm-linux-gnueabi packages on Ubuntu operating systems, for use in Android-based devices, using as a host machine Ubuntu v14.04 (kernel v4.4) in our case. For using Nettle, we also had to cross-compile the dependent library GMP v6.1.1 [20].

TABLE II. Android Based Devices Used in Experiments

CPU spec.	Android version	Market name
2.2 GHz Qualcomm MSM8974	5.1.1 Lollipop	Sony Xperia Z1 Compact
1.7 GHz MediaTek MT6592	4.4.4 KitKat	Aquaris E10 (Tablet)
1.3 GHz MediaTek MT6582	5.0 Lollipop	Aquaris E5 HD
1.0 GHz Qualcomm QSD8250	2.3.7 Gingerbread	Nexus 1

3) Test Application

We thus wrote two ECIES* test application in C, one using OpenSSL and another using Nettle. For each library, the application calculated the time taken for each of the KG, KA, KD and symmetric encryption ECIES* functions.

We ran the test application 10K times for each library on each device and calculated the average computation time taken for completing each ECIES* functions. We chose to run the test application only 10K to reduce the total experiment time, after verifying that running it more, e.g. 100K times, was not necessary because the results had no noticeable difference.

To record the time, we used the function `clock_gettime()` with the flag `CLOCK_PROCESS_CPUTIME_ID`, provided by `time.h`, which returns the per-process time from the CPU.

About using the OpenSSL library, we used the EVP*() functions (so called high-level API) instead of the EC*() functions (so called low-level API) because the latter was not compatible with Curve25519. We verified that there was no noticeable difference in term of delay for NIST P-256 curve while using either of the APIs.

We cross-compiled the test application as above using the arm-linux-gnueabi toolchain on Ubuntu, and executed the application in Android using the Android Debug Bridge (ADB) shell [21].

B. Experiment Results

1) Computational Overhead

In this section, we discuss the computational overhead due to using ECIES*, in terms of time taken to execute KG and KA (ECDH). The time taken by KD (X9.63-KDF with SHA-2) and symmetric encryption (AES/XOR) were negligible (KD, AES, and XOR were at least ~100, ~150, and ~300 times faster, compared to KG) and do not worth considering, therefore are skipped. Also, the standard deviation for KG plus KA was at most 0.4 ms for all the implementations and targeted devices.

Table III and Table IV (summarized in Fig. 5) list the average overhead in milliseconds, as discussed in Section III.A.3 when using NIST P-256 and Curve25519 respectively. The rows list the target devices regarding the CPU specifications (see Table II). The OpenSSL and Nettle columns refer to using the respective libraries. In the eBACS column, we list the values as reported by the ECRYPT Benchmarking of Cryptographic Systems [22], for the sake of comparison. Note that for eBACS, we list the median value scaled by the CPU frequency with the closest reference to each device, i.e., Qualcomm Snapdragon S3 for Qualcomm (Snapdragon) MSM8974 and QSD8250, and Broadcomm Cortex A7 for MediaTek (Cortex A7) MT6592 and MT6582.

We observed that the Nettle implementation of NIST P-256 performed the best with an overhead of just 2.95 ms (compared to 2.73 ms in eBACS) in the device with 2.2 GHz CPU. However, the OpenSSL implementation of Curve25519 performed the best with the overhead of just 1.61 ms

TABLE III. NIST P-256 COMPARISON, IN MS

CPU spec.	OpenSSL		Nettle		eBACS	
	KG	KA	KG	KA	KG	KA
2.2 GHz Qualcomm MSM8974	4.62	4.69	1.39	1.56	0.62	2.11
1.7 GHz MediaTek MT6592	6.37	6.50	2.36	2.94	1.01	3.43
1.3 GHz MediaTek MT6582	8.73	8.90	3.22	4.03	1.33	4.49
1.0 GHz Qualcomm QSD8250	10.28	10.47	3.16	4.38	1.37	4.65

TABLE IV: CURVE25519 COMPARISON, IN MS

CPU spec.	OpenSSL		Nettle		eBACS	
	KG	KA	KG	KA	KG	KA
2.2 GHz Qualcomm MSM8974	0.43	1.18	1.31	1.24	0.19	0.19
1.7 GHz MediaTek MT6592	0.97	2.77	2.10	2.23	0.55	0.54
1.3 GHz MediaTek MT6582	1.34	3.79	2.87	3.06	0.72	0.71
1.0 GHz Qualcomm QSD8250	1.27	3.51	2.93	3.51	0.42	0.41

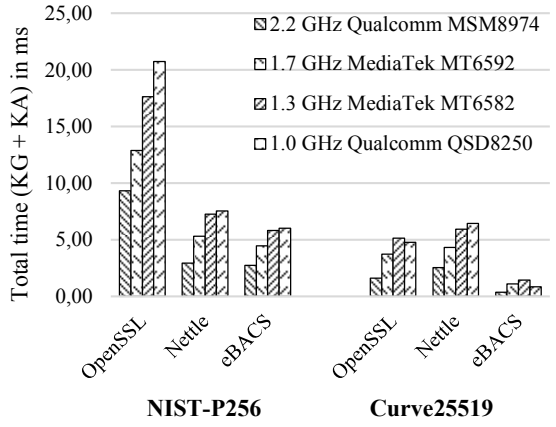

Fig. 5. Comparison of total execution time (KG + KA) for ECIES encryption on Android-based devices.

(compared to 0.38 ms in eBACS) in the same device. Since the typical call setup times are in the order of some seconds (referring to everyday experience when making a call), we can conclude that the additional overhead introduced by the ECIES* scheme for encrypting the IMSI at the UE side is acceptable. It is also worth noting that test applications were run as user space applications, whereas in real life the encryption of IMSI might be done in hardware and therefore be even faster.

On the network side, the HSS needs to decrypt the encrypted IMSI (see Fig. 1 and Fig. 3). However, for decrypting a single encrypted IMSI, the additional overhead at the HSS side is negligible, first because of much more computational resources than UE, and secondly because the HSS needs to run only the KA function before each decryption (KG is not required because the HSS's public key is static).

2) Bandwidth Overhead

The theoretical bandwidth overhead measured as the size of the encrypted IMSI, due to using ECIES*, was mentioned in Section II.C to be in the order of 320 bits when using 256-bit elliptic curve key. Compared to transmitting 60 bits clear-text IMSI in LTE, the increase in overhead is 260 bits. It matches with the output of our practical implementation. It is already discussed in [3] that sending 320 bits over the air in Attach request message is not a problem. The reason is that, in terms of LTE, the radio bearer carrying the Attach request message has larger TBS than 320 bits (e.g., the UEs with uplink categories 0 and M1 have maximum TBS of 1000 bits [14]) and supports segmentation.

However, to see that the overhead in bandwidth can increase drastically based on the encryption scheme chosen, and become problematic, we also give a theoretical example of using the traditional RSA [23] scheme instead. The corresponding key length in the RSA scheme that is equivalent to the 256-bit key in the ECIES scheme for the same security level is at least 2048 bits [7, 24]. So roughly speaking, the overhead on bandwidth due to RSA would be in the order of 2000 bits.

IV. PRIVACY ENHANCEMENTS TO PAGING

As discussed in Section II.D, in the current LTE systems, the UE's clear-text IMSI is revealed over the air during the Paging, which is a privacy issue. In Section IV.A, we discuss two variants of privacy enhancing identifiers in the downlink Paging message. These variants extend on the proposed enhancement described in the Section II.C.

Another privacy issue in the current system is that the same identifier is used in both the downlink Paging and the uplink response message that reveals, to an eavesdropper over the air, that the UE with that identifier is present in the observed area. In Section IV.B, we discuss the use of privacy enhancing identifier in the uplink message in response to the Paging.

Yet another privacy issue in the current system is that some bits of the IMSI are still revealed even during the Paging based on S-TMSI, because the UE Identity Index (*IMSI mod 1024*) is used to calculate the PF. In Section IV.C, we discuss the use of privacy enhancing UE Identity Index.

Finally, Section IV.D discusses other practical aspects, i.e., Paging identifier information element and handling recovery.

A. Privacy Enhancing UE Identifier in Downlink Paging

1) Using a Pseudonym Assigned to the UE by the HSS

Referring to Section II.C and Fig. 2, this variant is relevant to the case when the shown Pseudonym is in use. The Pseudonym is assigned to the UE by the HSS via the MME. The UE uses that Pseudonym instead of the clear-text IMSI to identify itself to the HSS the next time.

In this variant, we propose that the MME uses the same Pseudonym also as the UE's Paging identifier in a next Paging (variant 1 in Fig. 6). The size and format of the Pseudonym are supposed to be the same as the clear-text IMSI. Therefore, using the Pseudonym does not affect the current maximum

number of identifiers that can be sent in a single Paging message, i.e., the maximum is still 16 IMSIs.

2) Using Parts of the Encrypted IMSI that was sent by the UE in the Earlier Attach Request

Referring to Section II.C and Fig. 2, this variant is relevant regardless of whether the Pseudonym is in use or not. The UE sends the encrypted IMSI to the HSS via the MME. We point to the fact that both the MME and the UE know the value of UE's encrypted IMSI. Therefore, in principle, the MME could use the same encrypted IMSI, sent by the UE in the earlier Attach request message, as the UE's Paging identifier in a next Paging. However, the problem with doing so is that the number of UE's Paging identifiers that can fit in a single RRC Paging message will decrease because of the large size of the encrypted IMSI, which is around 320 bits. Since a single RRC Paging message can contain up to 1344 bits for UE's Paging identifier (Section II.D), only three encrypted IMSIs (instead of 16 clear-text IMSIs) can fit into a single RRC Paging message. This problem becomes worse with RSA encryption because not even one encrypted IMSI (> 2000 bits) fits into a single RRC Paging message (1344 bits).

In order to solve the above problem, in this variant, we propose to use only some parts of the encrypted IMSI value (e.g., some LSBs) instead of the full value as the UE's Paging identifier (variant 2 in Fig. 6). A potential issue is that there could be multiple UEs whose encrypted IMSI values have a common part (i.e., two different encrypted IMSIs could have the same LSBs that were chosen as Paging identifier and there could be identifier collision). However, we point out that it is sufficient for the UE's Paging identifier to be unique when considered together with the Tracking Area (TA) and the PF/PO where the Paging message is sent. The reason is that a Paging message sent in one combination of TA/PF/PO is not present in another combination.

Since the MME is responsible for sending the Paging message, the MME is also in able to know the corresponding TA/PF/PO. Therefore, the MME can dynamically adjust the size of the UE's Paging identifier to avoid the collision in the particular TA/PF/PO combination. If there is no collision, then it is possible to use fewer parts (e.g., LSBs or MSBs) to uniquely address the UE, and if there is a collision, then more bits could be used.

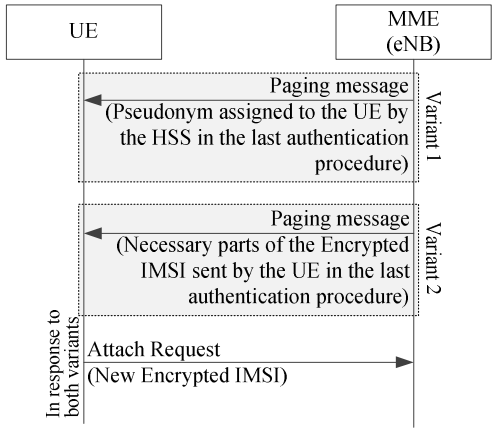

Fig. 6. Simplified Paging with privacy enhancements

To that end, the format of the encrypted IMSI as described in Section II.C is as follows:

Encrypted IMSI = MCC || MNC || Encrypted MSIN || Ephemeral UE Public Key

When there is no collision between any pair of the encrypted MSINs, then the MME can omit the Ephemeral UE Public Key in the Paging. Doing so does not affect the current maximum number of identifiers that can be sent in a single Paging message, i.e., 16. However, in the rare case in which there is a collision, then the MME can include more bits of the Ephemeral UE Public Key part to ensure that the value uniquely identifies the UE within the TA/PF/PO where the Paging is sent. The MME also has the additional opportunity to exclude the MCC and MNC to make space for more bits of the Ephemeral-UE-Public-Key if necessary and when possible, e.g., to page non-roaming UEs.

B. Privacy Enhancing UE Identifier in Uplink Attach Request in Response to the Paging

We propose that when the UE attaches to the network in response to the Paging, the UE always uses a new encrypted IMSI as its identifier (Fig. 6).

Note that in the first variant mentioned in Section IV.A.1, the Pseudonym was primarily intended for use in the next Attach request requiring a long-term identifier. Therefore, in principle, the UE could use the same Pseudonym in Attach request as the one used in the Paging. However, one privacy issue in doing so is the following. By correlating the identifier in the Paging and the Attach request, it is revealed that the UE that was Paged is currently present in the observed area. We point out that if the UE actively chooses to use a new encrypted IMSI in the response to Paging, the above-mentioned correlation privacy threat is thwarted.

C. Privacy Enhancing UE Identity Index

The UE Identity Index used for determining the PF and PO is calculated today in LTE as

UE Identity Index = IMSI mod 1024

Referring to Section IV.B and Fig. 6, we propose that the UE Identity Index is instead calculated based on the Pseudonym for variant 1 and the encrypted IMSI for variant 2. Doing so enhances the privacy, because both the Pseudonym and the encrypted IMSI are short-term identifiers compared to the long-term IMSI. This means that the UE Identity Index will also be short-term and no information on the IMSI is revealed over the air through the PF/PO.

Note that the UE Identity Index above mentioned could be used for both the IMSI based Paging and the S-TMSI based Paging (see background on Paging procedure in the Section II.D).

D. Other Aspects

1) Paging Identifier Information Element

For the UE to be able to distinguish the type of Paging identifier, new types of identifiers may be introduced for the Pseudonym and the encrypted IMSI. An example of encoding

these identifiers in the syntax of the PagingUE-Identity message [12] is shown below:

```
PagingUE-Identity ::= CHOICE {
  s-TMSI    S-TMSI,     -- existing
  imsi      IMSI,       -- existing
  pseudo    IMSI,       -- new IE for Pseudonym
  e-imsi    BIT STRING, -- new IE for encrypted IMSI
}
```

For the Pseudonym, the same type as for the IMSI could be used. For the encrypted IMSI, a variable length type could be used, e.g., BIT STRING when not limiting the size or BIT STRING (SIZE (320)) when limiting the size to, say, 320.

2) Handling Recovery

There may be concerns that when the UE loses the last assigned Pseudonym or the last used encrypted IMSI, the MME would not be able to reach the UE. We point that the UE losing its last assigned Pseudonym or the last used encrypted IMSI is a worst case scenario, i.e., rare. Therefore, if it happens, there could already be other serious consequences, e.g., UE's hardware failure, UE not able to make phone calls, UE not able to connect to the Internet, among others. Nevertheless, the recovery depends upon the UE itself re-attaching to the network using a freshly encrypted IMSI.

V. Related Work

Already when UMTS standardization started, some solutions for increased subscription identifier privacy were discussed, based on, e.g., public key encryption, one-time pseudonyms, and pre-shared (group) keys [25, 26, 27]. Unfortunately, none was adopted due to a number of questions related to, e.g., its effectiveness and efficiency (considering CPU capacities of the era), or the integration of the scheme into the communication protocols [28]. Again, as LTE standardization started, the issue was raised, and the proposals from UMTS standardization were revisited [29]. However, once again the decision was not to introduce any countermeasures.

In the ongoing 3GPP TR 33.899 [3], the 5G solution proposals discussed for the confidentiality protection of the IMSI include using encrypted IMSI based on ECIES (without MAC), Key-Policy Attribute-Based Encryption (KP-ABE), Identity-Based Encryption (IBE), and RSA. There are also solution proposals that either use only Pseudonym, or both Pseudonym and encrypted IMSI.

In [30], authors presented a solution that normally uses a Pseudonym, and an encrypted IMSI for rare cases, e.g., recovery. In [31, 32], authors presented solutions in which a Pseudonym is sent to the UE by using existing fields of the authentication procedure.

Regarding IMSI-based Paging, the 3GPP contribution [4] discussed remaining vulnerabilities due to the potential need to still use IMSI in Paging messages. It suggested a method in which the home network provides the serving network with special Paging identifiers, to be used to address the UE during IMSI-based Paging. To determine whether a Paging message is relevant (i.e., intended for it), the UEs need to decrypt Paging identifiers received in broadcast Paging. The Paging identifiers are encrypted by the HSS using a special key to this purpose, shared between the HSS and each USIM. To ensure that the transmitted Paging identifier is different every time, the solution implements a freshness mechanism based on sequence numbers, which requires a re-synchronization procedure to be used in case the USIM and HSS get out of synchronization.

In [34], the authors proposed to use symmetric cryptography based on a shared session key to preserving the privacy of IMSI during the IMSI based Paging. The network derives the shared session key by using the subscriber specific long-term key and a nonce. Then, the IMSI, a challenge number, and a sequence number are encrypted using the shared session key; and they are included in the Paging message along with the nonce. On the UE side, the same session key is derived, and the IMSI, the challenge number and the sequence number are decrypted. If the IMSI belongs to the UE and the sequence number is valid compared to the local value, the UE responds to the Paging by sending the received challenge number.

The discussion of the Paging Frame number revealing some bits of the long-term subscription identifier (IMSI) in LTE systems was presented in the 3GPP contribution [35]. Another contribution [36] discussed a privacy enhancing calculation of the Paging Hyper Frame (for extended DRX in LTE systems) using the hash of the short-term subscription identifier (S-TMSI). The Hyper Frame number is 1024 times larger than the normal Frame number. The contribution [36] acknowledged that the use of S-TMSI suffers from the fact that the UE becomes unreachable when the MME does not have the S-TMSI for the UE.

VI. Conclusion

In this paper, we presented the result of our practical evaluation of using ECIES (without MAC) for encrypting the IMSI in 5G systems. We used two crypto libraries, OpenSSL and Nettle, and tested the performance in four Android-based devices. We observed that the additional overhead of encrypting the IMSI is well acceptable, in terms of both computation time and bandwidth. Therefore, there are not technical impediments for the future 5G systems to adopt the mechanism of encrypting IMSI and thus thwart the threat of IMSI catching attack, both in its active and passive versions.

We also presented our analysis and proposed enhancements for Paging procedure to protect the privacy of IMSI not only in the uplink but also in the downlink. We highlighted privacy weaknesses in the current Paging procedure and presented proposed improvements, which are attractive for being adopted in future 5G systems.

References

[1] A. Shaik, R. Borgaonkar, N. Asokan, V. Niemi, and J.P. Seifertet, "Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems," [Online]. Available: https://arxiv.org/pdf/1510.07563.pdf, accessed 22 November 2016.

[2] A. Bakke Foss, Aftonbladet (2015, Jun, 26), *New report: Clear signs of mobile surveillance in Oslo, despite denial from Police Security Service* [Online]. Available: http://www.aftenposten.no/norge/New-report-

Clear-signs-of-mobile-surveillance-in-Oslo-despite-denial-from-Police-Security-Service-61149b.html, accessed 4 January 2017.

[3] 3GPP Technical Report 33.899, "Study on the security aspects of the next generation system" [Online]. Available: http://www.3gpp.org/DynaReport/33899.htm, accessed 29 December 2016.

[4] 3GPP Discussion Document S3-99314, "Enhanced User Identity Confidentiality and IMSI-Paging" [Online]. Available: http://www.3gpp.org/ftp/tsg_sa/wg3_security/TSGS3_06_9910/docs/S3-99314Mannesmann,%20Cleartext%20IMSI-paging.doc, accessed 29 December 2016.

[5] 3GPP Technical Specification 23.007, "Restoration procedures" [Online]. Available: http://www.3gpp.org/DynaReport/23007.htm, accessed 29 December 2016.

[6] *3GPP, the Third Generation Partnership Project* [Online]. Available: http://www.3gpp.org, accessed 29 December 2016.

[7] ECC Standards for Efficient Cryptography Group, SECG SEC specification, "Elliptic Curve Cryptography, SEC 1," version 2, 2009.

[8] V. Gayoso, L. Hernández, and C. Sánchez, "A Survey of the Elliptic Curve Integrated Encryption Scheme," in Journal of Computer Science and Engineering, Volume 2, 2010, pp. 7-13.

[9] American National Standards Institute (ANSI), "Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography," X9.63, 2001.

[10] International Organization for Standardization/International Electrotechnical Comission (ISO/IEC), "Information Technology – Security Techniques – Encryption Algorithms – Part 2: Assymmetric Cyphers," 18033-2, 2006.

[11] 3GPP Technical Specification 36.304, "Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) procedures in idle mode" [Online]. Available: http://www.3gpp.org/DynaReport/36304.htm accessed 29 December 2016.

[12] 3GPP Technical Specification 36.331, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification" [Online]. Available: http://www.3gpp.org/DynaReport/36304.htm, accessed 29 December 2016.

[13] 3GPP Technical Specification 24.301, "Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS)" [Online]. Available: http://www.3gpp.org/DynaReport/24301.htm, accessed 29 December 2016.

[14] 3GPP Technical Specification 36.306, "Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio access capabilities" [Online]. Available: http://www.3gpp.org/DynaReport/36306.htm, accessed 29 December 2016.

[15] National Institute of Standards and Technology, Federal Information Processing Standards Publication, "Digital Signature Standard (DSS)".

[16] *NIST, National Institute of Standards and Technology* [Online]. Available: https://www.nist.gov. Accessed 27 December 2016.

[17] A. Langley, M. Hamburg, and S. Turner, RFC 7748 "Elliptic Curves for Security," 2016 [Online]. Available: https://www.ietf.org/rfc/rfc7748.txt, accessed 27 December 2016.

[18] *OpenSSL, cryptography and SSL open source Toolkit* [Online]. Available: https://www.openssl.org, accessed 12 December 2016.

[19] N. Möller (editor), *Nettle, a low-level cryptographic library* [Online]. Available: https://www.lysator.liu.se/~nisse/nettle/, accessed 12 December 2016.

[20] *GMP, The GNU Multiple Precision Arithmetic Library* [Online]. Available: https://gmplib.org, accessed 12 December 2016.

[21] *Android Debug Bridge,* [Online]. Available: https://developer.android.com/studio/command-line/adb.html, accessed 13 January 2017.

[22] D. J. Bernstein and T. Lange (editors), *eBACS: ECRYPT Benchmarking of Cryptographic Systems* [Online]. Available: https://bench.cr.yp.to, accessed 19 December 2016.

[23] K. Moriarty, B. Kaliski, J. Jonsson, and A. Rusch, RFC 8017 "PKCS #1: RSA Cryptography Specifications," Version 2.2, 2016 [Online]. Available: https://www.ietf.org/rfc/rfc8017.txt, accessed 27 December 2016.

[24] National Institute of Standards and Technology, NIST Special Publication 800-57 (Rev. 4), "Recomendation for Key Management, Part 1: General."

[25] A. Herzberg, H. Krawczyk, and G. Tsudik, "On Travelling Incognito," in First Workshop on Mobile Computing Systems and Applications (WMCSA 1994), Santa Cruz, USA, 1994, pp. 205-211.

[26] 3GPP Discussion Document S3-99067, "Enhanced User identity Confidentiality" [Online]. Available: http://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_02/docs/S3-99067.zip, accessed 25 January 2016.

[27] 3GPP Discussion Document S3-99360, "Enhanced User Identity Confidentiality" [Online]. Available: http://www.3gpp.org/ftp/tsg_sa/wg3_security/TSGS3_07/docs/s3-99360_enhanced%20UI%20conf.zip, accessed 25 January 2016.

[28] 3GPP Discussion Document S3-00268, "Removal of enhanced user identity confidentiality" [Online]. Available: http://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_12_Stockholm/Docs/PDF/S3-000268.PDF, accessed 25 January 2016.

[29] 3GPP Technical Report 33.821, "Rationale and track of security decisions in Long Term Evolution (LTE) RAN / 3GPP System Architecture Evolution (SAE)" [Online]. Available: http://www.3gpp.org/DynaReport/33821.htm, accessed 29 December 2016.

[30] K. Norrman, M. Näslund, and E. Dubrova, "Protecting IMSI and Subscriber Privacy in 5G Networks," in Proceedings of 9th EAI International Conference on Mobile Multimedia Communications (MobiMedia 2016), Xi'An, China, 2016.

[31] F. van den Broek, R. Verdult, and J. de Ruiter, "Defeating IMSI Catchers," in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS 2015), New York, USA, 2015, pp. 340-351.

[32] M. S. A. Khan and C. Mitchell,"Improving Air Interface User Privacy in Mobile Telephony," in Proceedings of second Security Standardisation Research conference (SSR 2015), Tokyo, Japan, 2015, pp. 165-184.

[33] *ECRYPT: European Network of Excellence in Cryptology* [Online]. Available: http://www.ecrypt.eu.org, accessed 9 January 2017.

[34] M. Arapinis, L. Mancini, E. Ritter, M. Ryan, N. Golde, K. Redon, and R. Borgaonkar, "New Privacy Issues in Mobile Telephony: Fix and Verification," in Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS 2012), New York, USA, 2012, pp. 205-216.

[35] 3GPP Discussion Document S3-161153, "Reply to LS on eDRX paging timing calculation and security concern" [Online]. Available: http://3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_84_Chennai/Docs/S3-161153.zip, accessed 25 January 2016.

[36] 3GPP Discussion Document R2-165935, "LS on eDRX Paging Hyper-frame and PTW_Start Calculation" [Online]. Available: http://3gpp.org/ftp/tsg_ran/WG2_RL2/TSGR2_95/Docs/R2-165935.zip, accessed 25 January 2016.