Identity-Division Multiplexing Technique for Enhancing Privacy of Paging Procedure in LTE

Abdulrahman Muthana[1], and Abdulraqeb Al-Samei[2]
[1,2]Smart Security Solutions, Sana'a, Yemen
[1]**ab.muthana@smartsecurity-y.com**, [2]**abdu.alsamee@gmail.com**

ARTICLE INFO

Article History
Received 5 Jul 2019
Received in revised form 15 Aug 2019
Accepted 15 Aug 2019

Keywords:
LTE (Long Term Evolution), TMSI (Temporary Mobile Subscriber Identifier), identity-division multiplexing, linkability, traceability, paging

ABSTRACT

Despite efforts have been made by Long Term Evolution (LTE) toward enhancing privacy preserving capabilities, LTE is still vulnerable to privacy attacks. This paper evaluates the privacy issues of paging procedure in LTE and suggests a solution for enhancing the privacy of paging procedure in LTE. The solution introduces the Identity-Division Multiplexing (IDM) technique, in which the total sequence of temporary mobile subscriber identifiers M-TMSIs (2^{32} unique M-TMSI identities) is divided into a series of overlapping M-TMSIs ranges, each of which is allocated to one or more user equipment (UE). The solution guarantees the use of frequently changing unrelated TMSIs for identification; and thus, providing unlinkability and untraceability of the user. The solution provides an effective identity management that protects privacy of LTE users during paging process. The solution is formally verified using proVerif and proved to protect user privacy adequately.

I. INTRODUCTION

Long Term Evolution (LTE) cellular network technology [1] enhances the security of its predecessors: Global System Mobile Communication (GSM) and Universal Mobile Telecommunications System (UMTS) and offers a range of new security features.

LTE protects user identity privacy by allocating each user equipment (UE) various different temporary identities such as Global User Temporary Identifier (GUTI), temporary mobile subscriber identifier (TMSI), and cell radio network temporary identifier (C-RNTI) at different levels of LTE network architecture for different services. The UE can use these temporary identities instead of the International Mobile Subscriber Identifier (IMSI) to identity itself. This strategy reduces IMSI exposure risk and mitigates user identity privacy attacks.

Despite this strategy, LTE is still vulnerable to privacy attacks [2-6]. Temporary identifiers remain unchanged for amount of time sufficient for hacker to track the user and are transmitted in clear. For example, TMSI will not be changed within certain tracking area and that the paging messages are not encrypted [3].

This paper evaluates the privacy issues of paging procedure in LTE and also presents a solution for enhancing the paging privacy. The solution provides a high level of user unlinkability and anonymity within LTE cellular networks by using Identity-Division Multiplexing (IDM) technique, in which the total M-TMSIs sequence (2^{32} unique M-TMSI values) is divided into a series of overlapping ranges, each of which is allocated to one user equipment (UE). The solution guarantees the use of frequently changing unrelated temporary mobile subscriber identifiers (TMSI) for identification; and thus, providing unlinkability and untraceability of the user.

ISSN 2636-9680
eISSN 2682-9266

The proposed solution preserves privacy of user identity during paging procedure with minimal modifications at network architecture. The design strategy of the proposed solution aims at keeping LTE messaging system away as much as possible from the changes and modifications. We believe that this solution could be easily fit in current LTE cellular network architecture.

The main contribution of this paper is proposing a security solution that substantially enhances LTE capabilities in preserving user identity privacy during paging procedure. The privacy is preserved with minimal modifications on the network entities (i.e., Mobile Management Entity MME and UE) and with no modifications in the message system. The second contribution is an extensive theoretical study on privacy of paging procedure in LTE.

The rest of this paper is organized as follows: Section 2 reviews the related work and Section 3 describes privacy issues of paging procedure in LTE. Section 4 presents the proposed solution, Section 5 analyzes its security, and Section 6 concludes.

II. RELATED WORK

Many research works have discussed privacy in LTE and suggested solutions for protecting privacy in LTE. A number of researches have focused on privacy of user identity in LTE [2-13, 16, 19]. Paging and location privacy have also been highlighted in [3- 6, 15, 17,18, 20,21]. In this paper, we restrict ourselves to the closest related works (i.e., the research works addressing paging privacy issue in LTE).

Several researches investigate security issues of paging procedure in LTE. The research work [9] proposed encrypting the paging request using a shared session key, called as unlinkability key. The key is maintained for privacy preserving purpose only and is generated by applying a new one-way keyed function f to the long-term shared key KIMSI, and a random number rand included in the paging request. Furthermore, the solution requires that the encrypted request message should include a sequence number SQN and a random challenge chall.

The network stores the random challenge and checks it against the one received from the UE in the paging response. The aim of the sequence number SQN is to ensure freshness of the paging request and avoid replay attacks. A UE that receives a legitimate IMSI paging request should discard the request if the SQN is not in the correct range. The use of this procedure should still be kept minimal to avoid burdening the signaling communication with cryptographic operations. Each UE must decrypt all the received IMSI paging requests to determine whether it is the intended recipient or not.

The research work [21] studied the information leakage problem in the paging procedure and provided a solution by using a physical layer identification scheme. The scheme is a complementary and does not eliminate the need for enhanced privacy in the other signaling procedures. The scheme proposed a function that takes the UE's temporary ID as an input and has a tag as an outcome. During the paging period of a UE, instead of transmitting TMSI, the corresponding tag would be inserted. The scheme requires that no correlation should exist among the tags for different users. An interesting point is that in this case, the transmission power of the signal need not to be at such a level that the receiver could decode it. The receiver should only be able to detect the signal to be able to ensure if the user has been paged or not. This results in saving energy. The drawback of the scheme is that it requires changes in the physical layer procedure that would lead to changing the hardware, which might be costly.

The research work [3] suggested a solution to mitigating paging procedure privacy attacks through sending a hashed value of TMSI identifier allocated to the UE. The security solution requires that a random value like a nonce or a time stamp should be utilized as input to the hash function in order to change the hashed value of the pseudonym after each calculation.

III. PAGING PROCEDURE ISSUES IN LTE

The LTE network locates an idle UE using paging procedure in order to deliver a service to it (e.g, an incoming call, SMS message). The serving network MME locates an idle UE as per tracking area TA basis and sends the paging request message to every evolved eNodeB (eNB) within a particular tracking area. The transmitted paging request may contain the identity of one or more UEs. The UEs targeted by the paging request identified by temporary identities (TMSIs) [3].

Once a UE finds its TMSI identifier in the paging request message, it establishes a dedicated channel to allow the delivery of the service (responding to incoming call or receiving the SMS). It should be noted that TMSI is not changed within a certain tracking area TA and that the paging message are sent in clear text.

The possibility of initiating a paging request for a specific TMSI allows an attacker to check for the presence of a particular UE within a specific area. Assume that an attacker initiates several calls to a specific user within the user's tracking area and monitors the paging channel to obtain several sets of TMSIs that have been paged by the eNB. The attacker could reveal the identity of the concerned user by intersecting the sets of TMSIs identifiers.

IV. METHODOLOGY

The proposed solution replaces the fixed M-TMSI identifier with frequently changing M-TMSI identifiers selected from a range of M-TMSI identifiers. More than one UE can share M-TMSI values in one M-TMSI range. During paging process MME implements identity-division multiplexing IDM technique in order to select proper M-TMSI values for each UE. It is worthy to mention that, while it introduces the concept of identity ranges overlapping that allows the ranges of identities allocated to the UEs to be overlapped, the solution allows the MME to uniquely identify a specific UE using Identity-Division Multiplexing IDM technique and ensuring the unlinkability and the untracebility of the UE.

The MME first allocates a range of M-TMSI identifiers to the UE and delivers the range to it. We use *R* to refer to the M-TMSI range. Each range *R* is a pair(*S*, *L*), where *S* is 32bit value representing the starting point of *R* (i.e., the first M-TMSI value in *R*) while *L* is 16bit value representing the length of the range *R* (thus, the maximum length of *R* is 65536 values). However, it is up to the network operator to determine the length *L*. The UE, interprets the allocated range *R* as follows: *S* is the smallest M-TMSI value in *R* while the value (*S*+*L*) is the largest M-TMSI in *R*. The UE also understands that the valid M-TMSI used for paging UE should lie between *S* and *S*+*L*. Next, when the MME wishes to page the UE, it generates a random fresh M-TMSI value between *S* and *S*+*L*, embeds it within the paging request message, and transmits the message to the UE. Once the UE receives the paging request message, it checks whether the received M-TMSI lies between *S* and *S*+*L*. If the received M-TMSI is within the correct range, the UE responds to the request by initiating a service request procedure; otherwise it discards the request.

The proposed solution changes the way of creating and allocating M-TMSI identifiers to ensure subscribers unlinkability. It enhances the characteristics of M-TMSI identifiers allocated to UEs as follows: (1) each allocated M-TMSI is independent from other identifiers such as IMSI, GUTI, and from any previous allocated M-TMSIs. An attacker, who is monitoring the paging channel, cannot correlate the intercepted

M-TMSIs with each other nor correlate them with a particular UE, (2) the allocated M-TMSI is random and computationally unpredictable, (3) the allocated M-TMSI is used only once for paging a specific UE, (4) allocated M-TMSI is changed frequently, (5) allocated M-TMSI is not reused, (6) there are no collisions in the allocation areas, and (7) the concerned UE can easily check M-TMSIs in the paging message to find whether it is intended by the paging request or not.

The proposed method enhances the privacy of paging procedure in LTE and ensures the unlinkability of UEs with minimal modifications at the two network nodes (i.e., the MME and the UE) with no modifications on any other network node. It also considers the computational power and storage capabilities of both the MME and the UE. It introduces a negligible computation overhead at the UE and an affordable computation overhead at the MME. The method protects against paging-related attacks at a minimal cost and thus it can be easily integrated with the current mobile technology. Moreover, the proposed solution requires no changes on the messaging system. The boundary values (*S* and *L*) of M-TMSI range allocated to the UE can be included in the normal messages that serving network SN sends to the UE during communications between UE and SN

A. The MME

S	*L*	*STATUS*
S_1	L_1	1
...	...	…
Si	L_i	1
...	...	…
SK	*LK*	0
...	...	…
Sn	*Ln*	0

(a) M-pool

IMSI	*S*	*L*	*T*	*V*
IMSI_1	S_1	L_1	T_1	V_1
…	...	...	…	…
…	...	...	…	…
IMSIi	*Si*	*Li*	*Ti*	*Vi*
…	...	...	…	…
…	...	...	…	…
IMSI_K	S_K	L_K	T_K	V_K

(b) M-table

Fig. 1: (a) M-pool (b) M-table

The solution extends the MME storage with two tables: M-pool and M-table **(Fig. 1**). M-pool table stores a list of all available M-TMSI ranges from which MME can allocate ranges for the UEs in its service area while M-table stores information details of the ranges that are allocated to the UEs in the MME's service area. The M-pool table maintains three values for each M-TMSI range: *S*, *L* and *STATUS* values. *S* and *L* store respectively the start and the length of the range while *STATUS* is a binary value indicating whether the range is free for the use or not. The range with value 0 in *STATUS* is free for the use. The value 1 in *STATUS* indicates that the corresponding range is allocated.

Each record of M-table stores M-TMSI information details of one UE. Each record comprises a set of fields including: IMSI, *S*, *L*, *T* and *V*. The IMSI holds the IMSI identifier of the UE. *S* and *L* denote the start and the length of M-TMSI range allocated to the UE. *T* denotes the last M-TMSI value used by the MME for paging the UE while the *V* denotes the last M-TMSI value used by the UE for initiating service request procedure.

The proposed solution implements a set of algorithms at MME side:

*1) M-TMSI Ranges Creation Algorithm:*initializes M-pool table with the boundaries of all M-TMSI ranges available for the MME to use. **Fig. 2** shows the major steps of the algorithm and the details are as follows:

1. The total sequence of 2^{32} unique M-TMSI values is partitioned into a set of overlapping partitions each of which is a range *R*.

2. The boundaries of each range *R* are stored in *S* and *L* fields of a particular record at M-pool. The field *S* stores the first M-TMSI value in *R* whereas the field *L* stores the length of *R*. Initially, all created M-TMSI ranges are marked as free for the use (i.e., *STATUS* is set to 0 for all M-TMSI ranges).

The M-TMSI ranges creation algorithm ensures that every M-TMSI range overlaps with its previous range in some partition. This is done by selecting the starting point *S* of the next M-TMSI range from within current M-TMSI range. The algorithm computes the value of the starting point *S* of next range as the 2/3 of *L* of the current range (refer to step 10 at **Fig. 2**). By following this partitioning strategy, the boundaries of the consecutive ranges interleave with each other and each M-TMSI range has two types of partitions: shared and unshared partitions. Shared partition is basically a range of M-TMSI values that belong to two overlapping ranges while the unshared partition is a range of M-TMSI values that belong to one range only. The shared partition among two overlapping ranges say R_W and R_Z is denoted by $R_{W,Z}$; the unshared partition that belongs to range say R_X is denoted by R_X*.

```
Input: limits of range length min and max
1:     Let Avail ← $2^{32}$
2:     Let Stop ← 0
3:     Let $S$ ← 0
4:     While (Avail ≥ min) do
5:         generate a random $L$ ($min \le L \le max$)
6:         if (Avail < min ) then
7:             $L$= Avail
8:         end if
9:         $S$ = Stop;
10:        Stop = Stop + (2/3) $L$
11:        create an empty record at M-pool
12:        insert into the new record a tuple ($S$, $L$)
13:        Avail =Avail – $L$
14:    end while
```

Fig. 2: M-pool initialization algorithm

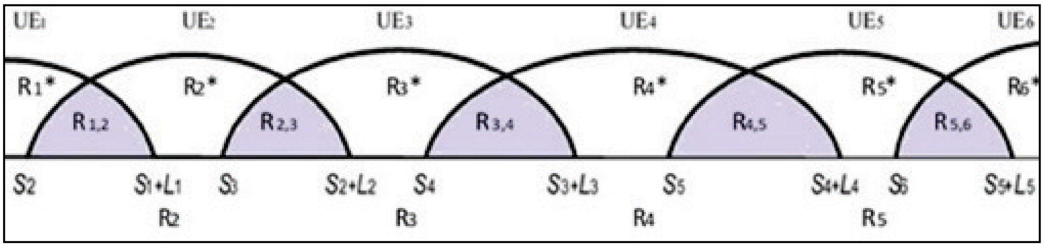

Fig. 3: Overlapping M-TMSI ranges

As **Fig. 3** shows, the M-TMSI range is divided into three partitions: two partitions shared with the previous and the next neighbor ranges respectively (in gray color) and one unshared partition (in white color). This partitioning strategy has several advantages: (1) It allows for controlled partitions overlapping, (2) MME is always able to uniquely identify a particular UE in its service area, (3) It is possible for MME to simultaneously identify two UEs using only one M-TMSI value. This can be done when two UEs, whose M-TMSI ranges overlapping with each other, are intended in the same paging message. The MME can select one T-MSI value from the shared range and include it in the paging message for both UEs.

2) M-TMSI Ranges Assignment Algorithm: selects a free M-TMSI range R from M-pool for the purpose of allocating it to the UE and delivers R information to the UE. **Fig. 4** shows the major steps of the allocation algorithm that MME follows for allocating M-TMSI range for a new UE that enters the MME's service area.

ISSN 2636-9680
eISSN 2682-9266

```
Input: UE's IMSI identifier ($IMSI_{UE}$)
Check if M-pool has free M-TMSI ranges if there exist free ranges then
    select a free range $i$ ($S$,$L$)
        set $STATUS(i)$ = 1
        allocate range $i$ ($S_i$, $L_i$) to UE
else
    select an arbitrary allocated M-TMSI
range $X$ ($S_x$, $L_x$) such that:
 ($S_x$, $L_x$) is allocated to $UE_X$
 AND TAL($UE_X$) ∩ TAL($UE$) = φ
      allocate range $X$ ($S_X$, $L_X$) to UE
End if
 create an entry at M-table and insert the
tuple ($IMSI_{UE}$, $S_{UE}$ $L_{UE}$, 0, 0) into it
```

Fig. 4: M-TMSI ranges allocation algorithm

1. The MME selects a fresh not-in-use M-TMSI range *R* from the M-pool, updates its *STATUS* value in M-pool to 1 and associates *R* with IMSI of the requesting UE. If M-pool has no free M-TMSI range to be allocated to the UE, the MME arbitrarily selects for the new UE an allocated M-TMSI range that would not cause M-TMSI collision in the tracking area the UE is in. The MME selects an M-TMSI range of any UE whose, Tracking Area List (TAL) does not overlap with the TAL of the concerned UE and reuses M-TMSI range for the concerned UE.
2. A new tuple with IMSI identifier, *S*, *L*, initial value of *T*, initial value of *R* is inserted into the M-table.

The information of the first allocated range *R* is delivered to the concerned UE in two stages. First, the length *L* is delivered during the attachment procedure included in the authentication vector AV, and then the starting value *S* of M-TMSI range is delivered during GUTI procedure.

-Delivery of *L value*: The main steps of the delivery of the length *L* of the allocated range *R* to the UE are shown in **Fig. 5** and explained below:

1. Upon receiving an attachment request issued by a UE, the MME allocates M-TMSI range *R* (*S, L*) to the UE.
2. The MME computes *L'* as a result of XORing *L* and first half of *S*, and forwards *L'* along with the attachment request to Home Subscriber Station HSS.
3. The HSS generates random token *RAND* and embeds *L'* into *RAND*. The calculation of authentication vector AV proceeds as in normal AKA (authentication and key agreement) procedure and transmitted to the MME.
4. The MME forwards the authentication request to the UE and completes with the UE the AKA procedure steps
5. If authentication succeeds, the UE extracts *L'* from *RAND* and get back *L* by XORing *L'* with first half of *S*, which is received in GUTI message.
6. Finally, the UE stores S and L for the purpose of paging procedure.

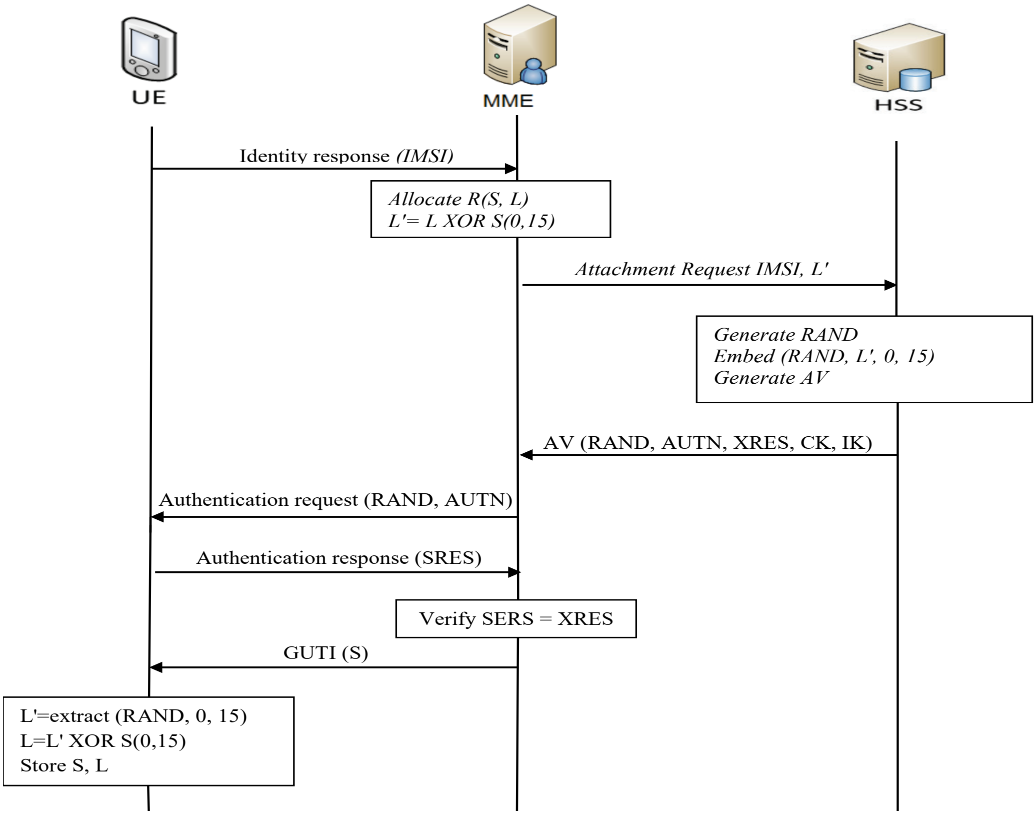

Fig. 5: The main steps of allocating and delivery of M-TMSI range to the UE

-Delivery of *S* value: the UE receives *L* value within *RAND* token during AKA procedure. The UE is then supplied with the starting point value *S* of the allocated M-TMSI range within the GUTI messages after a successful run of AKA procedure. Later, the UE may receive the *S* value included in the GUTI message in the following occasions:

- After Inter-MME handover request
- After a successful TAU request
- The serving network can be scheduled to send GUTI messages including *S* values to the UE at regular time intervals.
- The UE can be provided with the capability to request a fresh *S* value at arbitrary times.

3) M-TMSI Ranges De-Allocation Algorithm: the MME de-allocates the M-TMSI range R allocated to the UE by deleting R from M-table and frees up R if possible (i.e.; if R is non-sharable). The de-allocation algorithm is run whenever an existing UE is leaving the MME's service area. **Fig. 5** shows the main steps the MME runs in de-allocation algorithm and the steps details are given below:

1. Searches M-table for the UE’s entry that includes the range *R* using the IMSI of the concerned UE as a key and removes it.
2. Locates *R* at the M-table and verifies whether *R* is currently in use by another UE or not. If *R* is not found, the MME sets 0 value in the *STATUS* field corresponding to *R* in M-pool table and frees up *R*.

4) M-TMSI Ranges Re-Allocation Algorithm: replaces an M-TMSI range allocated to a UE with a different range. The Re-Allocation algorithm is run during the UE's movement within the service area when the UE moves into a new tracking

ISSN 2636-9680
eISSN 2682-9266

area which is not in the tracking area list TAL registered in the UE. MME can also run Re-Allocation algorithm at arbitrary time intervals. It is worth mentioning that the newly allocated range has the same length as the currently allocated R. This is because the MME sends only S to the UE during the Reallocation procedure. The major steps of the algorithm are:

1. The MME allocate a new range R whose L is the same as existing range R currently allocated to the UE.
2. The MME initiates GUTI relocation procedure and sends new S to the UE, which will replace its S with the newly received S and recalculate the boundary value (S+L).

5) Paging UE Algorithm: generates a fresh M-TMSI value and includes it in the paging request message transmitted to the UE. The MME runs the algorithm for page an idle UEs. **Fig. 6** demonstrates the major steps of the algorithm:

1. Searches M-table for the UE's entry using UE's IMSI identifier and generates a random fresh M-TMSI value M_{MME} such that: (i) M_{MME} is within the range R allocated to the UE, (ii) M_{MME} is different from the last M-TMSI value sent to the UE, and (iii) M_{MME} is different from the last M-TMSI received from the UE.
2. Proceeds with normal paging procedure steps with M_{MME} as M-TMSI value.

```
Input: UE's IMSI identifier
Search M-table using UE's MSI as a key
get $S$UE, $L$UE, $T$UE, and $V$UE of the UE
generate a fresh M-TMSI value($M$MME) such that:
$S\text{UE} \leq M\text{MME} \leq (S\text{UE} + L\text{UE})$ and
$M\text{MME} \neq T\text{UE}$ and
$M\text{MME} \neq V\text{UE}$
Update M-table at the UE's entry
    set $T\text{UE} = M\text{MME}$
embed $M$MME within the paging request message
proceeds with normal paging steps
```

Fig. 6: Algorithm for Paging UE

TABLE 1 shows the possible scenarios for paging three UEs with three overlapping ranges. It also shows the number of M-TMSI identities required for paging and the source ranges from which M-TMSI used in the paging can be selected. The following facts can be derived from the table:

- Two UEs having shared range can be identified in the same paging message using only one M-TMSI value selected from a shared range. For example, one M-TMSI value selected from shared range R1,2 can simultaneously identify UE1 and UE2 (Scenario 4(1)). Similarly, one M-TMSI value selected from shared range R1,2 can simultaneously identify UE2 and UE3 can be simultaneously identified using only one M-TMSI value selected from shared range R2,3 (Scenario 6(1)) .
- Three UEs (e.g., UE1, UE2, and UE3) allocated three consecutive ranges (e.g., R1, R2, and R3) can be targeted in the same paging message using two M-TMSI values selected from one range or two ranges or using three M-TMSI values selected from three ranges (Scenarios 7(1), 7(2), and7(3)).

***-UE Confusion Prevention Principle**: For any two UEs: UEx and UEz having overlapping ranges Rx and Rz with a shared range Rx,z. If, at any time, the MME wishes to page either UEx or UEz and uses M-TMSI identifier from Rx,z, then the unintended one will respond to the paging message thinking that it is intended. To prevent such situation, MME refrains from selecting M-TMSI identifier from Rx,z when paging either UEx or UEz. However, a selection from Rx,z can be made when both UEx and UEz are intended by the paging message. The principle could be relaxed and MME can select and use M-TMSI identifier from Rx,z for paging either UEx or UEz if and only if it is knows with certainty that UEx and UEz are in different tracking areas TAs.*

6) M-TMSI Validation Algorithm: upon receiving a service request initiated by the UE, the MME validates the M-TMSI value, M_{UE}, included in the service request. Depending on validation results the MME may respond to the request or not. If TRUE is returned from the validation algorithm after validating the request, the MME is assured that the request came from a legitimate UE and thus responds to the incoming request; if FALSE is returned the incoming request is discarded. **Fig. 7** shows the major steps of the algorithm to validate MUE value:

TABLE 1: Scenarios for paging 3 UEs {UE1,UE2,UE3} with 3 overlapping M-TMSI ranges {R1,R2,R3}

Scenario (case)	UEs intended by the paging message			No. of M-TMSIs required for Paging	No. of Ranges from which M-TMSI(s) can be selected	Possible Source Range(s) for selecting M-TMSI(s)
	UE1	UE2	UE3			
1	X			1	1	R1*
2		X		1	1	R2*
3			X	1	1	R3*
4 (1) (2)	X	X		1 2	1 2	R1,2 (R1*, R2*) **Or** (R1*, R1,2**)** **Or (**R1,2, R2*)
5 (1) (2)	X		X	2 2	1 2	(R1,2, R2,3) (R1,2, R3*) **Or** (R1*,R3*) **Or** (R1*, R2,3)
6 (1) (2)		X	X	1 2	1 2	R2,3 (R2*, R3*) **Or** (R2*, R2,3) **Or** (R2,3, R3*)
7 (1) (2) (3)	X	X	X	2 2 3	1 2 3	(R1,2 ,R2,3) (R1*,R2,3) **Or** (R1,2, R3*) R1*,R2*,R3*

1. Searches M-table looking for any M-TMSI range that contains the MUE value. If no M-TMSI range is found, then discards the request; otherwise proceeds with the next step.
2. Verify that MUE differs from the last M-TMSI sent to the UE, and from the last M-TMSI received from the UE.

```
Input: UE's IMSI and MUE included in the request
Output: TRUE or FALSE
Searches M-table for M-TMSI range where:
    $S\text{UE} \le M\text{UE} < 2/3(S\text{UE} + L\text{UE})$
If a particular range is found
    If $((M\text{UE} \neq T\text{UE}) \text{ and } (M\text{UE} \neq V\text{UE}))$
        return TRUE
    End if
End if
return FALSE
```

Fig. 7: Service request validation algorithm

B. The UE

The solution extends the UE with four 32 bit fields to store M-TMSI values: *S*UE, *L*UE, *T*UE and *V*UE. The *S*UE and *L*UE fields store respectively the values of the start and the length of the M-TMSI range supplied by the MME. The last M-TMSI value transmitted by the UE and the last M-TMSI value received by the UE are stored in *T*UE and the *V*UE fields respectively. For successful operation of the proposed solution, the functionalities of the UE with respect to GUTI relocation, paging, and service request procedures are modified.

The proposed solution implements a number of algorithms at UE side:

1) Receive Paging Request Message: Once a paging message request is received from the MME, the UE verifies that the incoming M-TMSI value MMME included within the paging request message is within the correct range and is also different from the TUE and VUE that are stored at the UE.

ISSN 2636-9680
eISSN 2682-9266

If so, the UE updates its V_{UE} to the newly arrived M_{MME} identity and initiates a service request; otherwise the paging request is ignored. **Fig. 8** presents the M-TMSI validation algorithm.

```
Input: paging request from MME including MMME
1:  if ( $S_{UE} \le M_{MME} \le S_{UE} + L_{UE}$ )
2:    if (($M_{MME} \ne T_{UE}$) and ( $M_{MME} \ne V_{UE}$))
3:        set $V_{UE} = M_{MME}$
4:        initiate a service request
5:    else ignore the paging request
6:    end if
7:  else ignore the paging request
8:  end if
```

Fig. 8: Paging request validation algorithm

2) Initiate a Service Request If the UE is confirmed that it is intended by the paging request message received from the MME, it initiate a service request. First, it UE first generates a random fresh M-TMSI value M_{UE}, embeds M_{UE} within the service request message, and updates TUE to M_{UE}. **Fig. 9** presents the steps of service request algorithm.

The condition in step 2 is to comply with MME Confusion Avoidance Principle. It allows the MME to uniquely identify the UE, while the conditions (in steps 3 and 4) are to ensure that the fresh M-TMSI is different from the last M-TMSIs values exchanged with the MME. The conditions (in steps 3 and 4) aim to eliminate the possibility of replay attack.

```
1:  Generate a fresh $M_{UE}$ value such that:
2:        $S_{UE} \le M_{UE} < (2/3)\ (S_{UE} + L_{UE})$,
3:           $M_{UE} \ne T_{UE}$, and
4:           $M_{UE} \ne V_{UE}$
5:  update $T_{UE} = M_{UE}$
6:  initiate service request
7:
```

Fig. 9: Service Request algorithm

***-MME Confusion Prevention Principle**: To prevent MME getting confused with the M-TMSI identifier M_{UE} transmitted by the paged UE, the UE must select a fresh M_{UE} that satisfies the following condition:*

$$S_{UE} \le M_{UE} < (2/3)\ (S_{UE} + L_{UE}) \tag{1}$$

S_{UE} is the starting point of the UE's M-TMSI range and L_{UE} is the range's length.

V. ANALYSIS AND RESULTS

LTE architecture assigns each UE a unique TMSI identifier in order to identify the user during the paging process. During the paging process, the MME includes the UE's TMSI within the paging request message and sends it to the UE. The security issue in using TMSI is that it remains for a period that is enough for an attacker to link the TMSI included in the paging requests with the user's permanent identity IMSI. Therefore, the existing LTE paging procedure is vulnerable to user linkability attack. The proposed solution enhances the characteristics of TMSI identifiers and their allocation procedure and improves the capabilities of LTE in preventing linkability attacks. Paging the UE with random TMSI identifier each time guarantees that an attacker cannot link the paging requests with the same user.

A. The key features

1) Minimal Computation Overhead: The majority of computation overhead is placed on the MME since its computation power is unlimited while a minimal computation is placed on the UE. We can claim that the overhead is negligible at both the MME and the UE.

2) Minimal system Impact: The solution does not change the messages and the messaging system, which makes it transparent to the intermediary networks.

3) Compatibility with LTE architecture: The solution can be easily integrated in the current LTE architecture with minimal modifications on the network parties.

B. Security analysis

This section analyzes the unlinkability and untraceability of the proposed solution.

1) User Unlinkability

The capability of an observer to linking between permanent identity and temporary identities of users is known as the linkability. The proposed solution mitigates the linkability attack that can be caused by using fixed TMSI in LTE paging procedure. The proposed solution provides unlinkability of LTE network subscribers by assigning each UE frequently changing M-TMSIs identifiers instead of a fixed M-TMSI that can be tracked and linked to a specific UE. Furthermore, since M-TMSI ranges overlap, it is possible for one M-TMSI from shared range used for identifying one UE to be safely reused for identifying another UE in some events (as discussed earlier in paging algorithm). Thus, ranges overlapping makes it harder for an adversary to track a specific UE.

2) User Untraceability

Traceability refers to the possibility of identifying past of identity requests and responses of the same subscriber. The proposed solution eliminates user traceability and protects user against tracking attack through enhancing the characteristics of, and the allocation procedures of, the pseudonyms (TMSIs). The allocation procedure of TMSI pseudonyms adopted by the presented solution prevents tracking of the user. The user is assigned a range of TMSIs and upon each request message, a random pseudonym TMSI is selected from within the range. Moreover, each pseudonym is utilized only once by respective network parties. Besides that, the same pseudonym can be reused by different UEs. This complicates the task of an observer to identify the requests and the responses that destined the same user as the M-TMSI exchanged in the network, from the observer's viewpoint are unrelated. As a result, the observer cannot identify the past identity requests and responses of the same user and cannot track the user.

VI. CONCLUSION

This paper presents a solution for enhancing the privacy of paging procedure in LTE network. It introduces identity ranges overlapping concept that allows the ranges of identities allocated to the UEs to be overlapped while it allows the MME to uniquely identify the UE using Identity-Division Multiplexing IDM technique. The UE is identified every time with a fresh M-TMSI identifier selected from a range of M-TMSI values allocated for the UE. The solution preserves paging procedure privacy through a secure identification system that allows a user to remain anonymous and be uniquely identified within the network and prevents attackers from being able to track the user. The solution is compatible with recent standards of LTE cellular network technology. The solution enhances the privacy of paging procedure in LTE and ensures user untraceability and unlinkability with minimal changes at the network and the UE and with low computation overhead on the part of the network and the UE.

VII. APPENDIX

The main result of this section is that the proposed solution indeed enhances the privacy of paging protocol in LTE and preserves unlinkability. The main idea of the proof is that an outside observer (attacker) sees no difference in the output of two runs of paging protocol that they differ only in user identifiers. The proVerif [14] is used for verifying that proposed solution enhances the privacy of paging protocol. The proof proceeds using the notion of observational equivalence.

```
Enhanced_Paging_Protocol:
event accept TMSI (bitstring, bitstring).
free net: channel.
free A:bitstring.
free B:bitstring.
(* constants *)
const PAGING_REQUEST:bitstring.
const PAGING_RSPONSE bitstring.
let UE(id: bitstring, out_tmsi: bitstring) =
   in( net, (=PAGING_REQEST, in_tmsi:bitstring));
   out(net, (PAGING_RSPONSE, out_tmsi)).
let MME(id: bitstring, in_tmsi:bitstring) =
      (*new in_tmsi: bitstring;*)
      out(net, (PAGING_REQUEST, in_tmsi));
      in(net,(=PAGING_RSPONSE,
out_tmsi:bitstring)).
process
      ((! (new out_tmsi1a: bitstring;  new out_tmsi1b:
bitstring;
      new in_tmsi1a: bitstring;   new in_tmsi1b:
bitstring;
(!  ((MME(choice[A,  B],  choice[in_tmsi1a,
in_tmsi1b])) | (UE(choice[A, B], choice[out_tmsi1a,
out_tmsi1b])))).
```

VIII. REFERENCES

[1] 3GPP, 3GPP System Architecture Evolution (SAE); Security architecture. 3GPP, TS 33.401, 2013.

[2] H. Choudhury, B. Roychoudhury and D. K. Saikia, “Enhancing user identity privacy in LTE”. In IEEE 11th International Conference on Security and Privacy in Computing and Communications (TrustCom), 2012. p. 949–957.

[3] H. Ghafghazi, A. El-Mougy, H. T. Mouftah, “Enhancing the privacy of LTE-based public safety networks”. In 13th Annual IEEE Workshop on Wireless Local Networks, Edmonton, Canada 2014.

[4] A. Bikos and N. Sklavos, “LTE/SAE security issues on 4g wireless networks”. IEEE Security and Privacy, 11(2):p. 55–62, 2013.

[5] N. Seddigh, B. Nandy, R. Makkar and J. F. Beaumont, “Security advances and challenges in 4g wireless networks”. In Eighth Annual International Conference on Privacy Security and Trust (PST), 2010. p. 62-71.

[6] I. Bilogrevic, M. Jadliwala and J. P. Hubaux, “Security and privacy in next generation mobile networks: LTE and femtocells”. In 2nd International Femtocell Workshop, Luton, UK. Citeseer, 2010.

[7] A. J. Bou, H. Chaouchi and M. Aoude, “Ensured confidentiality authentication and key agreement protocol for EPS”. In 3rd Symposium on Broadband Networks and Fast Internet, May 2012, 28-29.

[8] Li Xiehua, and Y, Wang, “Security Enhanced authentication and key agreement protocol for LTE/SAE network”, 2011, In 7th International Conference on Wireless Communications, Networking and Mobile Computing, IEEE.

[9] M. Arapinis, et al., “New privacy issues in mobile telephony: fix and verification”. In ACM Conference on Computer and Communications Security, 2012, p. 205–216.

[10] Z. Muxing, F. Yuguang, “Security analysis and enhancements of 3gpp authentication and key agreement protocol,” IEEE Trans, vol. 4, 2005.p. 734-742.

[11] G. M. Køien, “Mutual entity authentication for LTE”. In 7th International Wireless Communications and Mobile Computing Conference, 2011, IEEE.

[12] G. M. Køien, “Privacy enhanced mutual authentication in LTE”. In IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, 2013. p 614–621.

[13] F. Broek, R. Verdult and J. Ruiter, “Defeating IMSI catchers”. In CCS '15 Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015, ACM New York, NY, USA.

[14] B. Blanchet. “Proverif: Cryptographic protocol verifier in the formal model”. http://www.proverif.ens.fr/.

[15] K. Shubber for Wired magazine. “Tracking devices hidden in London’s recycling bins are stalking your smartphone”. http://www.wired.co.uk/news/ archive/2013-08/09/recycling-bins-are-watching-you. Last accessed May 2015.

[16] M. Arapinis, L. I. Mancini, E. Ritter, and M. Ryan. “Privacy through pseudonymity

in mobile telephony systems”. In NDSS, 2014.

[17] S. Datoo for The Guardian. “How tracking customers in-store will soon be the norm”. http://gu.com/p/3ym4v/sbl. Last accessed May 2015.

[18] N. Balasaheb, B. N. Gawande,"Hybrid model for location privacy in wireless ad-hoc networks", IJCNIS, vol.5, no.1, pp.14-23,.DOI: 10.5815/ijcnis.2013.01.02, 2013.

[19] A. Muthana, M. Saeed, “Analysis of user identity privacy in LTE and proposed solution”, I. J. Computer Network and Information Security, 1, 54-63 Published Online January 2017 in MECS (http://www.mecs-press.org/) DOI: 10.5815/ijcnis.2017.01.07, 2017.

[20] D. Forsberg, H. Leping, K. Tsuyoshi, and S. Alanara, “Enhancing security and privacy in 3gpp e-utran radio interface,” in Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on. IEEE, pp. 1–5.

[21] T. Tuan, and J. Baras, “Enhancing Privacy in LTE Paging System Using Physical Layer Identification”, Data Privacy Management and Autonomous Spontaneous Security. pp. 15-28. Springer Berlin Heidelberg, 2013.