International Journal of Computer Applications (0975 - 8887)
Volume 56 - No. 14, October 2012

A New Trust Model for Improved Identity Privacy in Cellular Networks

Hiten Choudhury
Dept. of Computer Science
St. Edmund's College
Shillong, Meghalaya, India

Basav Roychoudhury
Indian Institute of Management
Shillong, Meghalaya, India

Dilip Kr. Saikia
Dept. of Computer Science and Engineering
Tezpur University
Tezpur, Assam, India

ABSTRACT

Cellular networks have evolved through various generations, starting with 1G, followed by 2G and then by 3G, cellular networks have come a long way. A recent technology that has marked the beginning of 4G is Long Term Evolution (LTE). While transmission technologies, authentication mechanisms, confidentiality protection, etc., improved significantly through the generations, not much has improved with regards to the subscriber's identity privacy, and LTE is no exception. Much of this could be due to the trust model adopted in these networks. Introduction of sensitive services like mobile-banking, mobile-commerce, etc., has increased the importance of identity privacy by many folds. Identification of threats like location tracking and comprehensive profiling - where data about movement, usage, etc., of a subscriber is amassed and linked to his/her identity to explore various attacks - is quite alarming. In this paper, we propose a new trust model for strengthening identity privacy in cellular networks; it has an additional capacity to enhance interoperability among different cellular operators. We also propose a security extension that adopts this trust model to improve identity privacy and interoperability in LTE. A formal analysis of the extension proves that it meets its security goals.

General Terms:

Wireless networks, Security and privacy

Keywords:

Trust model, LTE, AKA, Identity privacy, Formal analysis, Cellular networks

1. INTRODUCTION

In all cellular networks, a common basic architectural framework is used; in which, three parties are involved viz. an User Equipment (UE), the Home Network (HN) and a Serving Network (SN). The UE that a subscriber owns is registered with the HN. The association between the UE and the HN is created from the moment the subscriber procures a Subscriber Identity Module (SIM) from the HN and fits it into his/her UE. The HN offers services to its registered UEs through SNs that are located within or outside its own service area. Communication between the UE and the SN happens through radio link, whereas communication between the SN and the HN happens through wired medium. The radio link is considered vulnerable to various kind of attacks as it is too open by nature for comfort of adversaries, whereas the wired link is considered secure [8].

In order to uniquely identify a subscriber for authentication, authorisation and billing purposes, the HN assigns a unique permanent identity called International Mobile Subscriber Identity ($IMSI$) [4] to the UE. The $IMSI$ is a valuable information that should not be accessible to anyone except the HN. Its compromise will expose the subscriber to threats like location tracking and comprehensive profiling - where data about movement, usage, etc., of a subscriber is amassed and linked to his/her identity to explore various attacks at a later time.

For the UE to access a particular service, it has to go through an Authentication and Key Agreement (AKA) procedure. During an AKA, the UE has to send a service request along with its identity to the SN. The SN in turn, obtains relevant authentication data from the HN by presenting the identity that it received from the UE. It then authenticates the UE using a challenge response mechanism [13].

Due to the current trust model adopted by cellular networks, there are occasions during identity presentation in the AKA procedure, when the $IMSI$ needs to be transmitted to the SN in clear text through the vulnerable radio path [10]. Moreover, the current trust model requires the SNs to be considered trustworthy, undermining the threat that a compromised third party SN may pose. Such requirement demands unconditional trust and thus limits interoperability.

In this paper, we propose a new trust model that has the potential to not only enhance identity privacy, but also to boost interoperability among cellular operators. We also propose a security extension that implements this trust model in LTE. A formal analysis of the extension is performed to prove that it meets its security goals. The rest of the article is organised as follows: in section 2, we discuss the current trust model; in section 3, we propose a new trust model; in section 4, we present the security architecture of LTE; in section 5, we discuss Evolved Packet System AKA (EPS-AKA) protocol: the AKA protocol adopted in LTE; in section 6, we discuss the identity privacy related vulnerabilities present in EPS-AKA; in section 7, we propose a new security extension for EPS-AKA; in section 8, we discuss the strong points of the proposed extension; in section 9, we perform a formal analysis of the proposed extension; finally, we conclude the paper in section 10.

2. CURRENT TRUST MODEL

In the current trust model, the following trust requirements with reference to the permanent identity of a subscriber exist.

UE → HN : As the UE is registered and has a direct service agreement with the HN, it is bound to fully trust the HN with its $IMSI$.

HN → SN : Since HN serves its subscribers through SNs, the HN confers full trust in the SN with regards to the $IMSI$ of a subscriber. For authentication, authorisation and billing

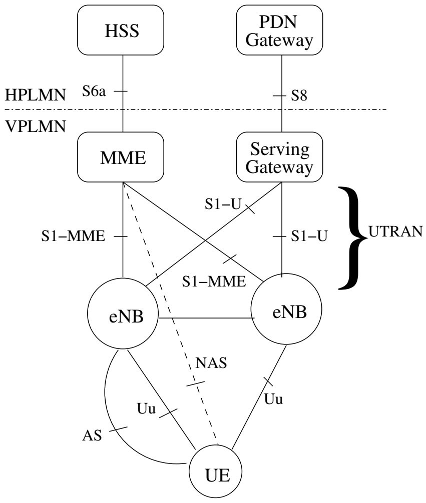

Fig. 1. Simplified security architecture of LTE.

purposes, the $IMSI$ is exchanged unabated between the HN and the SN.

UE → SN : This trust relation is a transitive outcome of the previous two trust relations; because of which, the UE fully trusts the SN with its $IMSI$ and it transmits its $IMSI$ immediately upon receiving a request from the SN.

From the above trust requirements, one can easily understand that even though the SN may belong to an untrusted third party cellular operator, the UE and the HN is required to confer unconditional trust on it. As a result, there exists several vulnerabilities, which an adversary may explore to compromise identity privacy of the subscriber. Moreover, such trust requirements are impractical in today's context when multiple cellular operators need to interoperate among each other to offer wider coverage to the subscribers. Roaming agreements with third party operators to provide service in a location where an operator has not set up its own infrastructure is a common practice. Thus, the need of the time is a paradigm shift that looks at the trust issues from a different perspective: a perspective that gives more emphasis on issues like privacy and interoperability.

3. PROPOSED TRUST MODEL

In this section, we propose a new trust model that is more flexible compared to the current trust model adopted by the cellular networks. In this, there is only one trust requirement, which is as follows:

UE → HN : The UE should trust only the HN with which it is registered and no one else. The $IMSI$ should not be shared with any third party and in no situation should leave the UE or the HN.

The above trust model will strengthen identity privacy, as the $IMSI$ is not shared with any one except the HN. It will also improve interoperability among cellular operators, since the requirement to trust the SN with respect to the permanent identity is totally relaxed. In order to adopt this model to provide improved identity privacy, an alternate mechanism for identity presentation, which can be used in all situations when an $IMSI$ is used otherwise, has to be formulated. To adopt this model even with respect to confidentiality and integrity protection of user data, end to end application layer based ciphering and integrity protection solutions like IPsec can be used [12]. With cellular networks gradually moving towards all-IP, such solutions will not be too hard to implement.

4. SECURITY ARCHITECTURE OF LTE

Fig. 1 depicts a simplified view of the roaming security architecture of an Evolved Universal Terrestrial Radio Access Network (E-UTRAN) [3] that serves as the core of LTE. In this, we show only the key elements associated with the AKA procedure used in LTE. Each and every user is registered with a Home public Land Mobile Network (HPLMN) (which, is the HN of the user) with their subscription and profile information stored in a Home Subscriber Server (HSS). In the Visiting Public Land Mobile Network (VPLM) (which, is the SN of the user), the User Equipment (UE) connects with an evolved NodeB (eNB) through the Uu interface, for attach, Tracking Area Update (TAU) and service requests. eNB is the new enhanced Base Transceiver Station (BTS) that provides the LTE air interface and performs radio resource management for the evolved access system. An eNB is connected with one or more Mobility Management Entities (MME) through the S1-MME interface. The MME is the key control node for the LTE access network and is responsible for authenticating the user by interacting with the HSS. For obtaining authentication data, the MME communicates with the HSS through the S6a interface. There are two layers of security between the UE and the E-UTRAN. The first layer is called the Access Stratum (AS) which protects the Radio Resource Control (RRC) plane signalling and the User Plane (UP) data between the UE and the eNB. The second layer is called the Non Access Stratum (NAS) which protects the control plane signalling between the UE and the MME. UE has access to packet data through the Packet Data Network Gateway (PDN-GW) via the Serving Gateway (S-GW).

5. EPS-AKA

EPS-AKA [2] is the AKA procedure used in LTE; it produces keying material forming a basis for UP, RRC, and NAS ciphering keys as well as RRC and NAS integrity protection keys. EPS-AKA during the initial connection and successive connections are as follows:

5.1 The Initial Connection

For the initial connection (when the subscriber switches on the UE for the first time), the UE transmits an attach request to the MME. Since the UE does not have a temporary identity at this moment, its $IMSI$ is included in this request. The EPS-AKA procedure during the initial connection is as follows:

(1) The MME invokes the procedure by requesting authentication data from the HSS. The request shall include the $IMSI$ and the SN/MME identity.

(2) Upon receipt of the request, the HSS assembles an Universal Mobile Telecommunication System - Authentication Vector ($UMTS\text{-}AV$) [1]. An $UMTS\text{-}AV$ contains a random part $RAND$, an authenticator token $AUTN$ used for authenticating the network to the UE, an expected response $XRES$, a 128-bit Integrity Key IK, and a 128-bit Cipher Key CK.

$$UMTS\text{-}AV = (RAND, AUTN, XRES, CK, IK) \tag{1}$$

The $AUTN$ contains a sequence number SQN used to indicate freshness of the AV. An $EPS\text{-}AV$ is then derived from $UMTS\text{-}AV$ by replacing CK and IK with a Key for Access Security Management Entity (K_{ASME}). To derive K_{ASME}, a Key Derivation Function (KDF) [2] is used that take the following input parameters: CK, IK and SN/MME identity. Thus,

$$K_{ASME} = KDF(CK, IK, MME\text{-}identity) \tag{2}$$

$$EPS\text{-}AV = (RAND, AUTN, XRES, K_{ASME}) \tag{3}$$

The HSS then sends $EPS\text{-}AV$ back to the MME.

(3) After receiving $EPS\text{-}AV$, the MME extracts $RAND$ and $AUTN$ from it and sends them to the UE as a challenge. A Key Set Identifier (KSI_{ASME}) is also send along with the challenge. The purpose of the KSI_{ASME} is to make it possible for the UE and the MME to identify a K_{ASME} without invoking the authentication procedure. This is used to allow re-use of the K_{ASME} during subsequent connections.

(4) At receipt of this message, the UE runs UMTS algorithm [1] to verifiy that $AUTN$ is correct and hereby authenticates the network. If $AUTN$ is incorrect, the UE rejects the authentication. If $AUTN$ is correct, the UE computes RES, IK and CK (using UMTS algorithm). It then derives the K_{ASME} from the newly computed IK and CK. The UE then responds back to the MME with a user authentication response message that includes the computed RES.

(5) Finally, MME checks whether RES is equal to $XRES$. If so, the authentication is successful. If not, the MME sends an authentication reject message towards the UE.

At the end of a successful EPS-AKA, a K_{ASME} is shared between UE and MME. A hierarchy of keys are then generated from the K_{ASME} [2] to be used for protection of the NAS and the AS. The MME then allocates a fresh temporary identifier called Globally Unique Temporary Identity ($GUTI$) [4] to the UE by initiating a $GUTI$ reallocation procedure through the NAS. During the $GUTI$ reallocation procedure, the MME sends $GUTI$ Reallocation Command to the UE and the UE returns $GUTI$ Reallocation Complete message to the MME. A new $GUTI$ shall be sent to the UE only after a successful activation of NAS security. A mapping between the $GUTI$ and the $IMSI$ of the UE is maintained at the MME. The purpose of the $GUTI$ is to provide an unambiguous identification of the UE, so that the subscriber's permanent identity (i.e., the $IMSI$) is not revealed.

5.2 Subsequent Connections

For subsequent connections (during attach requests, Tracking Area Updates (TAU) and service requests), identity presentation of the UE is accomplished by transmitting a $GUTI$ through the radio path. The KSI_{ASME} is also sent along with the request. Before transmitting, the UE integrity protects the request using NAS security. Upon receipt of the connection request, the MME identifies the corresponding K_{ASME} with the help of the received $GUTI$ and the KSI_{ASME}. The MME then checks the integrity protection of the message. If the integrity check succeeds, the MME, depending on the MME policy, may either decide to reuse K_{ASME} that was established during a previous AKA (without invoking a fresh authentication procedure) or may decide to go for a fresh EPS-AKA that will result in the establishment of a new K_{ASME}. In order to carry out a fresh EPS-AKA, the MME locates the $IMSI$ of the UE in its local database through the $IMSI\text{-}GUTI$ mapping and continues in the same manner as the initial connection (discussed above). In order to reuse a K_{ASME} a fresh set of keying material is derived form the K_{ASME}. Thus, the need to perform frequent AKA runs has been reduced in EPS through the use of a more elaborate key hierarchy. In particular, connection requests can be authenticated using a stored K_{ASME} without the need to perform a fresh AKA. Several successive connections may be secured through re-derived security contexts from the current K_{ASME}.

6. IDENTITY PRIVACY RELATED VULNERABILITIES IN EPS-AKA

In EPS-AKA, a $GUTI$ that is obtained in the previous connection (as explained in section 5.1) is transmitted instead of the $IMSI$ for identity presentation. The purpose of the $GUTI$ is to provide an unambiguous identification of the UE that does not reveal the subscriber's permanent identity (i.e., the $IMSI$). In spite of this security arrangement, there are occasions when the $IMSI$ may be transmitted in clear text. Some of the identity privacy related vulnerabilities in EPS-AKA are as follows:

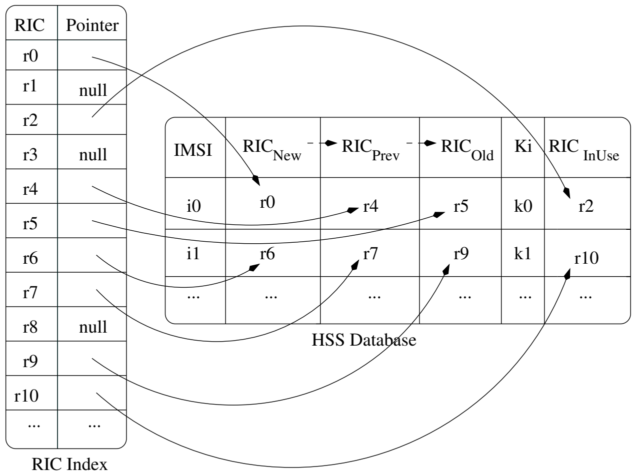

Fig. 2. RIC Index for the HSS's database.

—During the very first attach procedure the $IMSI$ has to be transmitted in clear text(TS 33.401 [2] section 5.1.1), since no $GUTI$ is available for identity presentation at this stage.

—The $IMSI$ has to be transmitted in clear text through the radio link as and when the MME requests for it. The MME has provision to make such a request when it cannot map the received $GUTI$ with the corresponding $IMSI$. For instance, whenever the UE visits a new MME and the new MME cannot acquire the $IMSI$ of the UE from the old MME. Such a recovery mechanism provides an opening for a fake MME to compromise a subscribers $IMSI$ [9].

—The SN, whose trustworthiness we question, has full knowledge about the $IMSIs$ of all subscribers to whom it provides services.

—The responsibility of creation and allocation of temporary identities (i.e., $GUTI$) are assigned to the MME, whose trustworthiness can itself be questioned.

Moreover, it is clearly evident that the trust model adopted in LTE is the current trust model discussed in section 2. Therefore, all the identity privacy related vulnerabilities and interoperability related drawbacks discussed in section 2 exists in LTE as well.

7. SECURITY EXTENSION FOR EPS-AKA

In this section, we propose a security extension for EPS-AKA to achieve improved identity privacy and enhanced interoperability in LTE. The extension protects the permanent identity of a subscriber in the radio path as well as in the wired path. Knowledge of the $IMSI$ of a subscriber is restricted only to the UE and the HE. In the extension, a Dynamic Mobile Subscriber Identity ($DMSI$) is transmitted by the UE instead of the $IMSI$. The role of the $DMSI$ is to randomise and mask the $IMSI$ so that an adversary having access to a particular $DMSI$ cannot link it with any subscriber or any previous communication. This extension is based on our work that was presented in [7]. For successful functioning of the security extension the following operator specific random number and functions are used:

RIC: Random number for Identity Confidentiality (RIC) is a random number that uniquely identifies a UE within a particular HE for an epoch of time. RIC is used to compute a $DMSI$ as follows:

$$DMSI = MCC \| MNC \| RIC \| ERIC \quad (4)$$

Where, MNC, MCC and ERIC stands for Mobile Network Code, Mobile Country Code and Encrypted RIC (explained later in equation 16) respectively. Size of RIC (b) in bit should be lesser than 128 bits and shall be determined by the operator depending on the subscriber base of the HE. A RIC of size b provides a pool of $n = 2^b$ unique RIC values. A fresh not-in-use RIC called RIC_{Fresh} is chosen every time a new $EPS\text{-}AV$ is generated at HSS. RIC_{Fresh} is then cryptographically embedded into the $RAND$ of $EPS\text{-}AV$. The resultant random number after embedding RIC into $RAND$ is called Embedded RAND ($ERAND$). Only UE having the knowledge of the long term shared key Ki is capable of extracting RIC from the $ERAND$. Multiple (m) $RICs$ comprising of the fresh and few previously generated $RICs$ ($RIC_{New}, RIC_{Prev}, RIC_{Old}$, etc.) are maintained at the HSS against a particular $IMSI$ in order to ensure robustness of the protocol even when an AV gets lost in transit (or due to some reason does not get utilised). Such an arrangement ensures that a mapping between the RIC that is currently stored at the UE and the corresponding $IMSI$ is always maintained at the HSS. An additional RIC called RIC_{InUse} is maintained at the HSS. RIC_{InUse} enables the MME to uniquely identify the UE as long as the later continue to stay within the former's service area.

f_i: This function returns a RIC that can be used to uniquely identify an UE. This is done by randomly selecting a not-in-use RIC from the ***RIC-Index***, the later being an index for the HSS's local database consisting of $n = 2^b$ unique RIC entries arranged in ascending order (Fig. 2), b being the number of bits in RIC. Each RIC entry in the ***RIC-Index*** has a pointer called ***IMSI-Pointer*** against it. A RIC that is already allotted to some UE, will have its ***IMSI-pointer*** pointing to that particular row in the HSS's database, which contains the $IMSI$ of the concerned UE. A null pointer against a particular RIC in the ***RIC-Index*** denotes that the particular RIC is (not-in-use) not allotted to any specific UE and is free to be used.

$$RIC = f_i(RIC\text{-}Index) \quad (5)$$

f_e: This function embeds RIC into $RAND$ to produce $ERAND$, using the long term shared secret key Ki as parameter.

$$ERAND = f_{e\,Ki}(RIC, RAND) \quad (6)$$

f_x: This function extracts RIC from $ERAND$, using the long term shared secret key Ki as parameter.

$$RIC = f_{x\,Ki}(ERAND) \quad (7)$$

Example algorithms for f_e and f_x are proposed in [6].

f_n: This function takes in a 128 bit $ERAND$ and the secret key Ki as parameter and encrypts a 32 bit RIC to produce a 128 bit output called Encrypted RIC ($ERIC$).

$$ERIC = f_{n\,Ki}(RIC, ERAND) \quad (8)$$

f_d: This function decrypts $ERIC$ by using Ki as parameter to produce RIC.

$$RIC = f_{d\,Ki}(ERIC) \quad (9)$$

f_s: This function stores a freshly generated RIC (RIC_{Fresh}) against a given $IMSI$ in the HSS's database. In order to make space for RIC_{Fresh}, the oldest RIC stored against the corresponding $IMSI$ is freed up. For example, for $m = 3$ with RIC_{New}, RIC_{Prev} and RIC_{Old} as the $RICs$ stored against the corresponding $IMSI$, the oldest RIC (i.e., RIC_{Old}) is returned to the pool of not-in-use $RICs$ by setting a null pointer against it in the ***RIC-Index***. RIC_{Old} is then replaced by RIC_{Prev}, and RIC_{Prev} is replaced RIC_{New}. Finally, RIC_{New} is replaced by RIC_{Fresh}. An entry in the ***RIC-Index*** against the ***IMSI-Pointer*** of RIC_{Fresh} is also made accordingly.

Table 1. Cryptographic Functions

Functions	*Details*
f_i	Returns a not-in-use RIC from the ***RIC-Index***.
f_e	Embeds RIC into $RAND$ to produce $ERAND$.
f_x	Extracts RIC from $ERAND$.
f_n	Encrypts RIC to produce $ERIC$.
f_d	Decrypts $ERIC$ to find RIC.
f_s	Stores a fresh RIC in the HSS's database.
f_m	Moves a specified RIC from its current location to the RIC_{InUse} field in the HSS's database.

$$RIC_{Old}.IMSI\text{-}Pointer = null \quad (10)$$

$$RIC_{Old} = RIC_{Prev} \quad (11)$$

$$RIC_{Prev} = RIC_{New} \quad (12)$$

$$RIC_{New} = RIC_{Fresh} \quad (13)$$

f_m: This function moves a given RIC (say RIC_p) from its current location to RIC_{InUse} in the HSS's database. When RIC_p is passed as parameter, f_m searches for it in the ***RIC-Index***. The ***IMSI-Pointer*** against RIC_p in the ***RIC-Index*** leads to the $IMSI$ to which RIC_p is allotted. The particular RIC field (ie. RIC_{InUse}, RIC_{New}, RIC_{Prev}, RIC_{Old}, etc.) in the HSS's database against which RIC_p is stored is then located. If RIC_p is found in the RIC_{InUse} field, no updation is required and the function quits. Otherwise, if RIC_p is found in a RIC field (say RIC_c) other than RIC_{InUse}, than all the RIC fields older than RIC_c are set to null. The RIC values in these older fields are then returned to the pool of not-in-use RIC by setting their ***IMSI-Pointers*** to null in the ***RIC-Index***. The value in RIC_c is then copied into RIC_{InUse}. And finally, RIC_c is also set to null. For example, for $n = 3$ with RIC_{New}, RIC_{Prev} and RIC_{Old} as the $RICs$ stored against the corresponding $IMSI$, if RIC_p is found in RIC_{New} then:

$$RIC_{Prev} = RIC_{old} = null \quad (14)$$

$$RIC_{Prev}.IMSI\text{-}Pointer = null \quad (15)$$

$$RIC_{Old}.IMSI\text{-}Pointer = null \quad (16)$$

$$RIC_{InUse} = RIC_{New} \quad (17)$$

$$RIC_{New} = null \quad (18)$$

A summary of all the cryptographic functions used in the extension is presented in table 1.

An $ERAND$ (Say $ERAND_{First}$) that has a unique RIC called RIC_{First} embedded into it, is stored in the USIM's flash memory in a field called $ERAND_{UE}$ before a subscriber procures it from the service provider. RIC_{First} is also stored at the HSS and an entry in the ***RIC-Index*** is made accordingly. RIC_{First} is meant for one time usage during the initial connection.

7.1 The Initial Connection

For the initial connection, the UE transmits an attach request that contains a $DMSI$ (instead of the $IMSI$, section 5.1) calculated according to the following steps.

$$RIC_r = f_{x_{Ki}}(ERAND_{UE}) \quad (19)$$

$$ERIC_r = f_{n_{Ki}}(RIC_r, ERAND_{UE}) \quad (20)$$

$$DMSI_r = MCC\|MNC\|RIC_r\|ERIC_r \quad (21)$$

(1.1) The MME initiates the authentication procedure by sending a request for a fresh $EPS\text{-}AV$ along with $DMSI_r$ to the HSS.

(1.2) On receiving the request, the HSS separates RIC_r from $DMSI_r$ and with the help of ***RIC-Index*** locates the corresponding $IMSI$ and the secret key Ki of the UE in its database. It then calculates:

$$RIC_d = f_{d_{Ki}}(ERIC_r) \quad (22)$$

and checks if $RIC_d = RIC_r$. If they do not match the request is rejected. If they match, it confirms that the request has been send by the concerned UE and not by a third party with malicious intentions; RIC_r is thus moved from its current position to RIC_{InUse} using f_m, and a fresh $EPS\text{-}AV$ ($EPS\text{-}AV_f$) is generated as explained in section 5.1.
HSS then computes

$$RIC_{Fresh} = f_i(RIC\text{-}Index) \quad (23)$$

RIC_{Fresh} is then cryptographically embedded into the $RAND$ ($RAND_f$) of $EPS\text{-}AV_f$ using f_e.

$$ERAND_f = f_e(RIC_{Fresh}, RAND_f) \quad (24)$$

A copy of RIC_{Fresh} is also stored using f_s at the HSS's database. After embedding RIC_{Fresh} into $RAND_f$ $EPS\text{-}AV_f$ looks like the following:

$$EPS\text{-}AV_f = (ERAND_f, XRES_f, CK_f, IK_f, AUTN_f) \quad (25)$$

From now on, $ERAND_f$ is used (instead of $RAND_f$) as the effective 128 bit random number for EPS-AKA related computations. Finally, HSS sends $EPS\text{-}AV_f$ along with $DMSI_r$ back to the MME.

(1.3) On receipt, MME continues the AKA procedure by extracting $ERAND_f$ and $AUTN_f$ from $EPS\text{-}AV_f$. $ERAND_f$ and $AUTN_f$ are then transmitted as a challenge towards the UE.

(1.4) The UE and the MME completes the remaining part of the AKA procedure following the same steps as explained in section 5.1. If the mutual authentication process is successful:
—UE saves the recent $ERAND_f$ that it has received from the MME as $ERAND_{UE}$.
—The MME stores $DMSI_r$ into the field meant for storing $IMSI$ in its local database. Let us call this field as $DMSI_{MME}$.
MME continues to uniquely identify the UE with $DMSI_{MME}$ till it receives a new $DMSI$ (during a successful AKA) from the UE.

(1.5) At the end of the AKA procedure, a shared K_{ASME} is established between the UE and the MME . The MME then allocates a new $GUTI$ to the UE by initiating a $GUTI$ reallocation procedure through the NAS (as explained in section 5.1). MME stores this $GUTI$ against $DMSI_{MME}$ in its local database. The $GUTI$ is also stored in the UE's memory (say in a field called $GUTI_{UE}$) for identity presentation during the next authentication.

7.2 Subsequent Connections

For all subsequent connections and the corresponding AKA procedure, the UE may present its identity in two different ways:

(i) *By transmitting a GUTI received in the previous AKA:* In this case, UE transmits the $GUTI$ stored in $GUTI_{UE}$. Out of the two options, this one is the preferred option since it reduces authentication latency (as the authentication procedure happens locally between the UE and the MME without involving the HE).

(ii) *By transmitting a fresh DMSI:* This option is less preferred and the UE may be forced to opt for this option when the UE roams into the area of a new MME or when the MME cannot identify the UE with its current $GUTI$.

The protocol flow for subsequent authentications through the transmission of a fresh $DMSI$ is same as that of the initial connection (i.e., step 1.1 through 1.5). The protocol flow for subsequent connections through the transmission of $GUTI$ is as follows:

(2.1) UE extracts $GUTI_{UE}$ from its memory and transmits it to the MME.

(2.2) Through this $GUTI$, MME identifies the corresponding $DMSI$ (ie. $DMSI_{MME}$ stored in its database). MME then sends a request for a fresh AV to the HSS along with $DMSI_{MME}$.

(2.3) After receiving the request, HSS extracts the embedded RIC (say RIC_r) from $DMSI_{MME}$. The ***IMSI-Pointer*** against RIC_r leads to the record (in the HSS's database) that contain details related with the corresponding $IMSI$ of the UE. The remaining portion of this step proceeds in the same manner as in step 1.2.

(2.4) The remaining part of the protocol flow is same as steps 1.3 through 1.5.

8. ACHIEVEMENTS OF THE EXTENSION

The key achievements of the proposed extension may be summarised as follows:

—*End to end identity privacy*: Knowledge of $IMSI$ is confined only to the UE and the HSS; it is never transmitted at any stage of the network.

—*Enhanced interoperability*: MME to MME as well as HSS/UE to MME trust relationship requirement with respect to the permanent identity (i.e. $IMSI$) is relaxed; thereby, enhancing interoperability among cellular operators.

—*No impact at the MME/SN*: As no extra computation is introduced at the MME, the adoption of the extension in EPS-AKA will be easy. The extension can be adopted with minor modifications at the UE and the HSS (without requiring any modification at the intermediary network that may even belong to a third party).

—*Reduced communication overhead:* The extension reduces two protocol messages. Unlike EPS-AKA, during handover, the extension does not need the current MME to communicate with the previous MME to acquire the permanent identity of the UE. Instead, a $DMSI$ is directly sent to the new MME.

9. FORMAL ANALYSIS

In this section, we perform a formal analysis of the proposed security extension through an enhanced BAN logic [5] called AUTLOG [11]. Through this analysis, the security goals described in the following subsection are proven to be achieved by the extension.

9.1 Security Goals

$IMSI$ should be a shared secret between the UE and the HN/HSS; it should not be disclosed to any third party including the SN/MME:

—G1: $UE\ believes\ UE \overset{IMSI}{\longleftrightarrow} HSS$
—G2: $UE\ believes\ \neg(MME\ sees\ IMSI)$

$GUTIs$ and $DMSIs$ are transmitted in lieu of the $IMSI$ of a UE. During every successful run of the protocol, if the UE receives a fresh $GUTI$ and a fresh $DMSI$, it can easily protect its $IMSI$ (section 7).

—G3: $UE\ believes\ UE\ has\ GUTI$
—G4: $UE\ believes\ fresh(GUTI)$
—G5: $UE\ believes\ UE\ has\ DMSI$
—G6: $UE\ believes\ fresh(DMSI)$

It should not be possible to link a $GUTI$ with the corresponding $DMSI$ and a $DMSI$ with the corresponding $IMSI$.

—G7: $UE\ believes\ \neg(GUTI \equiv DMSI)$
—G8: $UE\ believes\ \neg(DMSI \equiv IMSI)$

9.2 Prerequisites

UE recognises Ki and believes that it is a good key for communication with HSS:

$$UE\ has\ Ki \tag{26}$$

$$UE\ recognises\ Ki \tag{27}$$

$$UE\ believes\ HSS \overset{Ki}{\longleftrightarrow} UE \tag{28}$$

Since UE is capable of verifying freshness of SEQ, it believes in $SEQ's$ freshness.

$$UE\ believes\ fresh(SEQ) \tag{29}$$

UE regards the following messages as atomic messages

$$(X)_{UE} \equiv X \quad \forall X \in \{ERAND, GUTI\} \tag{30}$$

UE believes that it has not said $enc(Ki, SEQ, ERAND)$. Where enc is an encryption function.

$$UE\ believes\ \neg(UE\ said\ enc(Ki, SEQ, ERAND)) \tag{31}$$

As K_{ASME} is generated from $ERAND$ (section 5.1) - UE believes that if HSS says $ERAND$, it means that the cipher key (say NAS_{enc}) generated from K_{ASME} is a good key for encryption of the NAS.

$$UE\ believes\ (HSS\ says\ ERAND \longrightarrow HSS\ believes\ MME \overset{NAS_{enc}}{\longleftrightarrow} UE) \tag{32}$$

UE believes that HSS controls the freshness of RIC and that if HSS says $ERAND$ with a fresh SEQ, the RIC contained it in is also fresh.

$$UE\ believes\ HSS\ controls\ fresh(RIC) \tag{33}$$

$$UE\ believes\ (HSS\ says\ (SEQ, ERAND) \longrightarrow HSS\ believes\ fresh(RIC)) \tag{34}$$

$ERAND$ is an encrypted form of RIC. With knowledge of Ki, UE can easily extract RIC from $ERAND$. Thus, UE is able to identify $ERAND$ with $enc(Ki, RIC)$.

$$(ERAND)_{UE} \equiv enc(Ki, RIC) \tag{35}$$

UE has a unique $IMSI$ that constitutes of MCC, MNC and Mobile Subscriber Identification Number ($MSIN$).

$$UE\ believes\ UE\ has\ MCC \tag{36}$$

$$UE\ believes\ UE\ has\ MNC \tag{37}$$

UE believes that the MME has jurisdiction and belief regarding the freshness of $GUTI$.

$$UE\ believes\ MME\ controls\ fresh(GUTI) \tag{38}$$

$$UE\ believes\ MME\ believes\ fresh(GUTI)) \tag{39}$$

UE believes that HSS has jurisdiction and belief concerning the $IMSI$ as a shared secret between the UE and the HSS.

$$UE\ believes\ HSS\ controls\ HSS \overset{IMSI}{\longleftrightarrow} UE \tag{40}$$

$$UE\ believes\ HSS\ believes\ HSS \overset{IMSI}{\longleftrightarrow} UE \tag{41}$$

UE believes that HSS has jurisdiction on the fact that without access to the ***RIC-Index***, RIC cannot be derived or linked in any way with the corresponding $IMSI$:

$$UE\ believes\ HSS\ controls\ \neg(RIC \equiv IMSI) \tag{42}$$

$$UE\ believes\ (HSS\ says\ ERAND \longrightarrow HSS\ believes\ \neg(f_x(Ki, ERAND) \equiv MSIN)) \tag{43}$$

$$UE\ believes\ (HSS\ believes\ \neg(f_x(Ki, ERAND) \equiv MSIN) \longrightarrow HSS\ believes\ \neg(RIC \equiv MSIN)) \tag{44}$$

UE believes that MME controls and believes that without access to the MME's local database, $GUTI$ cannot be linked in any way with the corresponding $DMSI$.

$$UE\ believes\ MME\ controls\ \neg(GUTI \equiv DMSI) \tag{45}$$

$$UE\ believes\ (MME\ says\ enc(NAS_{enc}, GUTI) \longrightarrow MME\ believes\ \neg(GUTI \equiv DMSI)) \tag{46}$$

UE believes it has not said $enc(NAS_{enc}, GUTI)$ itself.

$$UE\ believes\ \neg(UE\ said\ enc(NAS_{enc}, GUTI)) \tag{47}$$

UE sees the following:

$$UE\ sees\ ERAND, \{SEQ\}_{enc(Ki, ERAND)}, enc(Ki, SEQ, ERAND) \tag{48}$$

$$UE\ sees\ enc(NAS_{enc}, GUTI) \tag{49}$$

9.3 Proving the security goals

$$48 \xrightarrow{H1} UE\ has\ ERAND \tag{50}$$

$$26, 50 \xrightarrow{H2} UE\ has\ (Ki, ERAND) \tag{51}$$

$$51 \xrightarrow{H3} UE\ has\ enc(Ki, ERAND) \tag{52}$$

$$48, 52 \xrightarrow{H1, H3} UE\ has\ SEQ \tag{53}$$

$$51, 53 \xrightarrow{H2,C3} (enc(Ki, SEQ, ERAND))_{UE} \equiv enc((Ki, SEQ, ERAND)_{UE}) \quad (54)$$

$$27 \xrightarrow{C1} (Ki, SEQ, ERAND)_{UE} \equiv (Ki_{UE}, SEQ_{UE}, ERAND_{UE}) \quad (55)$$

$$30, 55 \xrightarrow{E4} (Ki, SEQ, ERAND)_{UE} \equiv (Ki, SEQ, ERAND) \quad (56)$$

$$56 \xrightarrow{E3} enc((Ki, SEQ, ERAND)_{UE}) \equiv enc(Ki, SEQ, ERAND) \quad (57)$$

$$54, 57 \xrightarrow{E2} (enc(Ki, SEQ, ERAND))_{UE} \equiv enc(Ki, SEQ, ERAND) \quad (58)$$

$$48, 58 \xrightarrow{C} UE\ believes\ UE\ sees\ enc(Ki, SEQ, ERAND) \quad (59)$$

$$59, 28, 31 \xrightarrow{A1} UE\ believes\ HSS\ said\ (SEQ, ERAND) \quad (60)$$

$$29 \xrightarrow{F1} UE\ believes\ fresh(SEQ, ERAND) \quad (61)$$

$$60, 61 \xrightarrow{NV} UE\ believes\ HSS\ says\ (SEQ, ERAND) \quad (62)$$

$$32, 62 \xrightarrow{K} UE\ believes\ (HSS\ believes\ MME \xleftrightarrow{NAS_{enc}} UE) \quad (63)$$

$$62, 34 \xrightarrow{K} UE\ believes\ HSS\ believes\ fresh(RIC) \quad (64)$$

$$33, 64 \xrightarrow{J} UE\ believes\ fresh(RIC) \quad (65)$$

$$48, 35 \xrightarrow{C} UE\ believes\ UE\ sees\ enc(Ki, RIC) \quad (66)$$

$$66, 26 \xrightarrow{SE2} UE\ believes\ UE\ sees\ RIC \quad (67)$$

$$67 \xrightarrow{H1} UE\ believes\ UE\ has\ RIC \quad (68)$$

$$36, 37, 68 \xrightarrow{H2} UE\ believes\ UE\ has\ (MCC, MNC, RIC) \quad (69)$$

$$69 \xrightarrow{H3} \boxed{UE\ believes\ UE\ has\ DMSI}\ (\mathbf{G5}) \quad (70)$$

$$65 \xrightarrow{F1} \boxed{UE\ believes\ fresh(DMSI)}\ (\mathbf{G6}) \quad (71)$$

$$49, 52 \xrightarrow{SE2} UE\ sees\ GUTI \quad (72)$$

$$72, 30 \xrightarrow{C} UE\ believes\ UE\ sees\ GUTI \quad (73)$$

$$73 \xrightarrow{H1} \boxed{UE\ believes\ UE\ has\ GUTI}\ (\mathbf{G3}) \quad (74)$$

$$38, 39 \xrightarrow{J} \boxed{UE\ believes\ fresh\ (GUTI)}\ (\mathbf{G4}) \quad (75)$$

As proven in equation 70, 71, 74 and 75, the UE receives a pair of fresh $DMSI$ and $GUTI$ during every successful run of the AKA protocol. These new dynamic identities (as explained in section 7) may be used by the UE to identify itself (instead of the $IMSI$). Thus the following may be assumed about the UE:

$$UE\ believes\ UE\ controls\ \neg(MME\ sees\ IMSI) \quad (76)$$

$$UE\ believes\ \neg(MME\ sees\ IMSI) \quad (77)$$

$$77 \xrightarrow{K4} UE\ believes\ UE\ believes\ \neg(MME\ sees\ IMSI) \quad (78)$$

$$76, 78 \xrightarrow{J} \boxed{UE\ believes\ \neg(MME\ sees\ IMSI)}\ (\mathbf{G2}) \quad (79)$$

$$40, 41 \xrightarrow{J} \boxed{UE\ believes\ (HSS \xleftrightarrow{IMSI} UE)}\ (\mathbf{G1}) \quad (80)$$

$$62, 43 \xrightarrow{K} UE\ believes\ HSS\ believes\ \neg(f_x(Ki, ERAND) \equiv MSIN) \quad (81)$$

$$81, 44 \xrightarrow{MP} UE\ believes\ HSS\ believes\ \neg(RIC \equiv MSIN) \quad (82)$$

$$42, 82 \xrightarrow{J} UE\ believes\ \neg(RIC \equiv MSIN) \quad (83)$$

$$83 \xrightarrow{E4} UE\ believes\ \neg(MCC, MNC, RIC \equiv MCC, MNC, MSIN) \quad (84)$$

$$84 \xrightarrow{E3} \boxed{UE\ believes\ \neg(DMSI \equiv IMSI)}\ (\mathbf{G8}) \quad (85)$$

$$49, 63, 47 \xrightarrow{A1} UE\ believes\ MME\ said\ enc(NAS_{enc}, GUTI) \quad (86)$$

$$75 \xrightarrow{F2} UE\ believes\ fresh\ enc(NAS_{enc}, GUTI) \quad (87)$$

$$86, 87 \xrightarrow{NV} UE\ believes\ MME\ says\ enc(NAS_{enc}, GUTI) \quad (88)$$

$$88, 46 \xrightarrow{K} UE\ believes\ MME\ believes\ \neg(GUTI \equiv DMSI) \quad (89)$$

$$45, 89 \xrightarrow{J} \boxed{UE\ believes\ \neg(GUTI \equiv DMSI)}\ (\mathbf{G7}) \quad (90)$$

10. CONCLUSION

In conclusion, the research contributed to understanding the importance and the current status of subscriber's identity privacy in cellular network. It is observed that, even though a cell phone is now being used for various types of sensitive services, the status of identity privacy has not quite improved. The trust model that was being adopted may be blamed for this. With more and more operators taking a plunge into the competitive cellular market, interoperability is a key issue. A major factor that influence the ease at which interoperation may happen between cellular operators, depends on the flexibility of the trust model adopted by a cellular network. In this article, we put forward a trust model that may help in improving the status of identity privacy and as an additional benefit may make interoperability between cellular operators easier.

11. REFERENCES

[1] 3GPP. 3G Security;Security architecture. TS 33.102, 3rd Generation Partnership Project (3GPP), 2011.

[2] 3GPP. 3GPP System Architecture Evolution (SAE); Security architecture. TS 33.401, 3rd Generation Partnership Project (3GPP), 2011.

[3] 3GPP. General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network. TR 23.401, 3rd Generation Partnership Project (3GPP), 2011.

[4] 3GPP. Numbering, addressing and identification. TS 23.003, 3rd Generation Partnership Project (3GPP), 2011.

[5] M. Burrows, M. Abadi, and R.M. Needham. A logic of authentication. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 426(1871):233–271, 1989.

[6] H. Choudhury, B. Roychoudhury, and D.K. Saikia. Umts user identity confidentiality: An end-to-end solution. In *Wireless and Optical Communications Networks (WOCN), 2011 Eighth International Conference on*, pages 1–6. IEEE, 2011.

[7] H. Choudhury, B. Roychoudhury, and D.K. Saikia. Enhancing user identity privacy in lte. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pages 949–957. IEEE, 2012.

[8] G.M. Koien. An introduction to access security in umts. *Wireless Communications, IEEE*, 11(1):8–18, 2004.

[9] U. Meyer and S. Wetzel. A man-in-the-middle attack on umts. In *Proceedings of the 3rd ACM workshop on Wireless security*, pages 90–97. ACM, 2004.

[10] CB Sankaran. Network access security in next-generation 3gpp systems: A tutorial. *Communications Magazine, IEEE*, 47(2):84–91, 2009.

[11] G. Wedel and V. Kessler. Formal semantics for authentication logics. In *Computer Security ESORICS 96*, pages 219–241. Springer, 1996.

[12] C. Xenakis and L. Merakos. Ipsec-based end-to-end vpn deployment over umts. *Computer Communications*, 27(17):1693–1708, 2004.

[13] M. Zhang and Y. Fang. Security analysis and enhancements of 3gpp authentication and key agreement protocol. *Wireless Communications, IEEE Transactions on*, 4(2):734–742, 2005.