Authentication Protocol for IoT-Enabled LTE Network

NEETESH SAXENA, Georgia Institute of Technology
SANTIAGO GRIJALVA, Georgia Institute of Technology
NARENDRA S. CHAUDHARI, Indian Institute of Technology Indore

The Evolved Packet System-based Authentication and Key Agreement (*EPS-AKA*) protocol of the *LTE* network does not support *IoT* objects and has several security limitations including: transmission of the object's (user/device) identity and key set identifier in plaintext over the network, synchronization, large overhead, limited identity privacy, and security attack vulnerabilities. In this paper, we propose a new secure and efficient *AKA* protocol for the *LTE* network that supports secure and efficient communications among various *IoT* devices as well as among the users. Analysis shows that our protocol is secure, efficient, privacy-preserved, and reduces bandwidth consumption during authentication.

CCS Concepts:•**Security and privacy** → **Authentication;**

Additional Key Words and Phrases: *LTE*, *IoT*, Man-in-the-middle attack, Object-*ID* theft, Key-*ID* theft.

ACM Reference Format:
Neetesh Saxena, Santiago Grijalva, and Narendra S. Chaudhari. 2016. Authentication Protocol for IoT-Enabled LTE Network. *ACM Trans. Internet Technol.* x, x, Article xx (March 2016), 20 pages.
DOI: 0000001.0000001

1. INTRODUCTION

1.1. The Internet of Things and Long Term Evolution Network

The Internet of Things (*IoT*), a network connecting different objects including physical objects (things) has recently evolved by integrating the concept of Internet with various wired and wireless technologies, to ultimately control and manage different things in our environment [Jover 2015]. Examples of the Internet of Things include cloud-based systems, health-care monitoring, smart transportation, entertainment related applications, smart cities, machine-to-machine communications, and smart electricity grids. The Internet of Everything (*IoE*) that supports all the data generated and transmitted by these *IoT* objects, will ultimately revolutionize our society.

The *IoT* concept uses a unique identifier for each object in order to make the object available to other objects and to the applications related to the use and functionality of the objects. These objects are devices and users that have the ability to transmit information over the network for Device-to-Device communication (*D2D*), person-to-person communication, and person-to-device interaction. *D2D* communication enables direct communication between the devices. It is an exciting and innovative feature of the next generation cellular networks allowing traffic through any network infrastructure, which provides interconnections between the critical public safety network, the ubiquitous commercial network, and the users [Lin et al. 2014]. Very recently, *D2D* communication has been added to the Long Term Evolution (*LTE*) in *3GPP* Release-12

Author's addresses: N. Saxena and S. Grijalva, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, USA, 30332.
N. S. Chaudhari, Department of Computer Science and Engineering, Visvesvaraya National Institute of Technology, Nagpur, and also with Indian Institute of Technology, Indore, Simrol, 453552, India.

DOI: 0000001.0000001

as proximity services [Alam et al. 2014]. Today, we have 15 billion *IoT* devices, and a Cisco report predicts 50 billion devices by 2020 [Barbara V. Lundin 2015].

The *LTE*, a radio access technology, was evolved with several objectives that enable a *4G* (4^{th} Generation) heterogeneous network with high resource capacity, low cost at customer end, low latency, good quality of service, and good coverage across the wide area [Seddigh et al. 2010]. As a result, *LTE* is today one of the fastest growing wireless technologies. According to a report by Global Mobile Suppliers Association (*GSA*), *LTE* is being used in large communication systems worldwide with subscriber rates of over 130% annually [GMSA 2015]. Nevertheless, applying *IoT* in today's large distributed systems, such as smart grid, transportation, and telecommunication systems, faces many research challenges including handling a high volume of traffic, and providing different services to a large group of devices in a secure manner.

1.2. Security and Privacy of the IoT

Security and privacy are the main challenges in managing *IoT*-based services, particularly in systems including a very large number of devices. Because many large-scale systems are connected to the *LTE* network for high speed data communication, their data security and privacy preservation are crucial. In addition, critical infrastructure systems are the targets of sophisticated cyber-attacks over the communication network. Therefore, it is important to extend *LTE* authentication processes and make sure that the security mechanisms are scalable to a large number of *IoT* devices. Privacy and security attacks are the major concerns for the *IoT*-enabled *LTE* network. If the *IoT* object/entity's identity (*ID*) is revealed to an attacker, it can perform Man-in-the-Middle (*MITM*) as well as impersonation attacks. In fact, there are devices available in the market, such as International Mobile Subscriber identity (*IMSI*) catcher that can perform such attacks. The probability of facing identity or key theft over the *IoT*-enabled *LTE* network is also high, as some parts of the *LTE* network are still deployed with the support of *2G* and *3G* networks. Mobility may or may not be supported depending upon the applications of the *IoT*-enabled *LTE* network. For example, most smart grid applications, such as smart-metering and smart building, do not have a need for handover. On the other hand, an efficient and secure handover is required for the *IoT*-enabled smart transportation. In realistic scenarios, the entity's identity (sender as well as receiver) protection is necessary for service-driven applications that handle critical information and critical systems. Some of these applications are military services, health-care monitoring, content-based cloud services, location-based mobile services, smart cities-based services, home automation services, environment monitoring services, and smart grid sensing and control applications. Different types of connected devices interact remotely over the Internet with default or no access credentials, and the communication traffic takes place across different networks. In such scenarios, it is extremely important to protect the data, as the provider has practically no control once the data moves over the other networks.

Although the *IoT* service-operators are providing services based on their existing infrastructure, the existing authentication and encryption processes are still not sufficient in order to resist threats and attacks over the network. For example, Global System for Mobile Communication (*GSM*) networks are considered insecure over-the-air [Firoozjaei and Vahidi 2012]. Similarly, Universal Mobile Telecommunication System (*UMTS*) as well as *LTE* networks suffer from various security limitations [Zhang and Fang 2005]. Some of the *2G* shortcomings were addressed by *3G*, and the issues of *3G* are being addressed in the *4G* network. Various *GSM* Authentication and Key Agreement (*AKA*) variants [Firoozjaei and Vahidi 2012], [Fanian et al. 2010], [Lee et al. 2003], [Chang et al. 2003] and *UMTS-AKA* variants [Zhang and Fang 2005], [Tang and Wu 2008], [Lin et al. 2005], [Al-Saraireh and Yousef 2006] were presented in *2G* and

3G networks, respectively. Recently, the *LTE* research community has started to research the *IoT* objects' communications aspects. There are known security issues with the existing *LTE-AKA* protocol. Some of the vulnerabilities of *4G* security have been identified and addressed in [Aiash et al. 2010], [Park and Park 2007].

From the implementation point of view, we propose that the cellular system has a cyber-security layer on top of the communication network. This cyber-security layer includes critical modules, such as authentication, authorization, and encryption. However, embedded devices, such as mobile phones and smart meters, have limited computing resources. Therefore, it is strongly recommended to revisit authentication schemes in order to enable support for the *IoT* devices, while keeping the traffic privacy and information security preserved. The scheme must provide mutual authentication to mitigate security attacks for gaining control over the *IoT* network-based devices.

In the future generation network, the identity identifiers requirement for the users and the devices will be a great challenge. Some of the numbering and identification plans have been extended in order to support a large number of *IoT* devices. For instance, in *E.164*, *E.212* and *3GPP TS 23.003* [GSM Association 2014], the *ITU-T* defines the identifiers' structure in which communication module supports 15 digit directory numbers for subscribers and 3 digit mobile network codes. We believe that more digits are required than this proposal in order to provide various services over the Internet/cellular network. In extremely large and complex systems, such as smart electricity grids, cellular systems, and cloud-based systems, these identifiers are needed in very large quantities in order to support massive *IoT* devices and users. Therefore, the future generated frameworks and protocols must support a standard that can deal with billion of the devices as well as the users with different communication networks.

Some of the causes of fake identifies of the *IoT* devices in the network include:

— Sometimes, operators use substituted identities for *IoT* devices that have been dedicated to other devices in the network. This misleads other operators as well as the devices receiving information from the *IoT* devices.
— The identifier is not well protected within the device due to poor coding techniques.
— If the operators cannot distinguish legitimate identity from the malicious one, it may later assign a legitimate identity to the adversary's device.

Limited or non-existing regulations are available to verify the identity of the devices. As a result, adversaries can easily spread fake or malicious identities in the network and divert the communication for malicious purposes.

1.3. Research Problem

Previously, *2G AKA* and *3G AKA* protocols have been developed to support communication among various users and devices. However, these protocols do not fit well in the *4G* system due to two weaknesses:

(a) *4G* networks are heterogeneous networks, which are connected using wireless technologies and some unprotected wired parties of *2G/3G* networks via *IP*-based bone networks [Chlamtac et al. 2003]. This may lead to existing Internet attacks [Check Point Soft. Tech. 2013].
(b) Both, *2G* and *3G AKA* protocols do not provide mutual authentication between the wired parties, and many times they provide weak encryption between the base station and the mobile user.

The *4G* network has resolved the limitations of *2G/3G* cellular networks and proposed an Evolved Packet System-based Authentication and Key Agreement (*EPS-AKA*) protocol for the *LTE* network. However, this protocol has the following serious security drawbacks, which are crucial challenges for the *IoT*-enabled services over the network:

(a) In the *LTE* network, the identity of each object (device), e.g., *IMSI*, is sent from the User Entity (*UE*) to the Mobility Management Entity (*MME*) in clear text over-the-air interface that causes *MITM* and object-*ID* attacks [Tang et al. 2003], [Alquhayz et al. 2012].
(b) Passing clear text Key Set Identifier (*KSI*) of Access Security Management Entity (*ASME*), *i.e.*, KSI_{ASME}, (generated by each object) from the *MME* to the *UE* over the *LTE* network is another limitation. The *IMSI* and KSI_{ASME} protections are very crucial during communication over the network, as adversary $\mathcal{A}$ can misuse this information, which leads to object-*ID* theft and to key-*ID* theft.
(c) In the *EPS-AKA* protocol, both, the *UE* and the Home Subscriber Server (*HSS*) maintain a counter that causes a synchronization conflict.
(d) Communication and computation overheads generated by the existing *LTE* protocols are very large, and do not support *IoT* functionality over the network.
(e) The existing protocols are not secure enough against various security attacks.

In the future, it is expected that latest *LTE* technologies will provide a secure and efficient services to the *IoT* objects. Also, there may be situations in which the *IoT* devices behave in an abnormal way and increase network load, such as generating and sending fake traffic. These situations occur when a victim object cannot verify legitimate objects connected to it in the network. Adversary $\mathcal{A}$ can connect with the victim object and compromise its security, for instance by retrieving the user and key set identifier of the object, and disturbing the network traffic to change counters used for the objects' synchronization. Also, the proposed solution should be efficient in terms of overheads and execution time as compared to existing *LTE* protocols, and must be secured against attacks. Therefore, it is extremely important to provide mutual authentications in the *IoT*-enabled network.

1.4. Contribution

In this paper, we focus on the existing security and privacy problems present in the *LTE* network, and propose a novel secure and efficient protocol that is entirely based on the symmetric key cryptosystem. We propose symmetric key algorithms because they are about 1000 times faster than asymmetric key algorithms. The main contributions of the present work are as follows:

(1) The proposed protocol completely hides the actual identity of the object, *i.e.*, *IMSI*, during authentication over the network. It also restricts the key set identifier KSI_{ASME} to be transmitted over the network.
(2) Our protocol defeats object-*ID* theft, man-in-the-middle, impersonation, and key-*ID* theft attacks over the *LTE* network. Our protocol provides untraceability, forward privacy, and anonymity to various users and devices in the network.
(3) The protocol reduces 6.1%, 11.8%, 11.7%, and 13.4% of the bandwidth consumption during authentication between the *MME* and the *HSS* considering single authentication vector as compared to existing *LTE* protocols *EPS-AKA*, Kϕien's *AKA*, Purkhiabani's *AKA*, and Choudhury's *AKA*, respectively.
(4) The synchronization problem that occurs in *EPS-AKA* is resolved. Our protocol does not maintain any sequence number or counter, rather uses timestamps and *MAC*s to accept/reject each message.
(5) The communication overhead is also reduced by 6%, 18.5%, and 12.9% in comparison to the Kϕien's *AKA*, Purkhiabani's *AKA*, and Choudhury's *AKA* protocols, respectively. In other words, our protocol is able to solve the above mentioned security problems without increasing the bandwidth requirement and overhead.

Table I: Symbols and Abbreviations

Symbol	Definition	Size (*bits*)
eNB	Evolved node B	–
USIM	Universal subscriber identity module	–
IMSI/ID	International mobile subscriber identity	128
TID/GUTI	Temporary identity/global unique *TID*	128
DMSI	Dynamic mobile subscriber identity	166
CID	*MME/ASME EPS* context identity	48
SQN	Sequence number	48
TAI	Tracking area identity	64
SNID	Serving network identity	128
AV-req	Authentication vector request	8
NetType	Network type	3
PI	Protocol identifier	4
AMF	Authentication management field	48
RAND/MSR	Random number	128
AK/XAK	Anonymity key	128
CK/XCK	Cipher key	128
IK/XIK	Integrity key	128
SK/K	Secret key shared b/w *UE* and *HSS*	128
$KSI_{ASME}/XKSI_{ASME}$	Key set identifier for each K_{ASME}	3
ACK	Acknowledgement	3
K_{ASME}	*MME* intermediate key	256
K_{NASint}	Integrity key for *NAS* signaling	256
K_{NASenc}	Cipher key for *NAS* signaling	256
K_{eNB}	Intermediate key b/w *MME* and *UE*	256
K_{UPenc}	Cipher key for user plane	256
K_{RRCenc}	Cipher key for *RRC* signaling	256
K_{RRCint}	Integrity key for *RRC* signaling	256
MAC/XMAC	Message authentication code	64
RES/XRES	Response/expected response	64
T/Actcode	Timestamp/activation code for *USIM*	64

The remainder of this paper is organized as follows. Section 2 presents related research on security of *LTE* networks and its authentication protocols. Section 3 discusses authentication process in the *LTE* network in detail, while section 4 presents the security and privacy attack model. The novel *AKA* protocol for *IoT*-enabled *LTE* network is proposed and presented in section 5. Section 6 analyses the security of the protocol. Section 7 addresses protocol performance. Finally, the conclusions of the present work are summarized in Section 8. Table I represents various symbols and abbreviations used in the paper along with symbol sizes in bits. Table II describes the role of cryptographic functions used in the paper.

2. RELATED WORK

Various research works have been performed to make the *LTE* network more reliable, cost effective, and easy to connect and to access. Prevention against various threats and attacks as well as privacy preservation are the major concerns of the *LTE* network [Peng et al. 2012a]. It has been determined that in order to achieve privacy preservation and attacks resistance, the protocols and security algorithms used in *4G* must be significantly improved [Matos et al. 2007]. In the last five years, both symmetric key-

Table II: Cryptographic Functions Definition

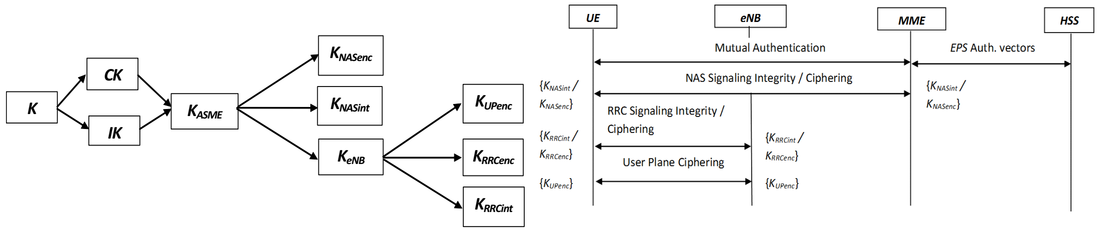
Function	Definition
$f_1()$	Function to generate *TID*
$f_2()$	Function to generate *MAC* / *XMAC*
$f_3()$	Function to generate *RES* / *XRES*
$f_4()$	Key generation function for *CK*
$f_5()$	Key generation function for *IK*
$f_6()$	Key generation function for *AK*
$f_7()$	Key generation function for K_{ASME}
$f'()$	Function to generate KSI_{ASME}
EK{}/*DK*{}	Functions to cipher/decipher message

based [Hadiji et al. 2009], [Kϕien 2011], [Gu and Gregory 2011] as well as asymmetric key-based [He et al. 2008], [Zheng et al. 2005] *AKA* protocols for the *LTE* network have been proposed by various researchers. The current focus is on *IoT* involving cellular network-based applications, such as smart grid, cloud computing, health monitoring, mobile banking, and mobile commerce supported [Peng et al. 2012b]. In the literature, different authentication protocols [Hadiji et al. 2009], [Kϕien 2011], [Gu and Gregory 2011], [He et al. 2008], [Zheng et al. 2005], [Choudhury et al. 2012] for the *LTE* network have been proposed. Vintila [Vintila et al. 2011] presents an outline of *LTE* security. Gu [Gu and Gregory 2011] and Choudhary [Choudhury et al. 2012] prevent *IMSI* to be sent over the network. Kϕien [Kϕien 2011] and Purkhiabani [Purkhiabani 2012] provide resistance against redirection attack, but do not defeat *MITM*, Object-*ID* theft, and key-*ID* theft attacks. Gu [Gu and Gregory 2011] does not propose any solution to these three attacks, while Choudhory [Choudhury et al. 2012] only addresses object-*ID* theft. Only Gu [Gu and Gregory 2011] has proposed solutions to the synchronization problem that appears between the *UE* and the *MME*/*HSS*. Out of these protocols, only [Choudhury et al. 2012] provides object identity protection over the network. However, this protocol generates a large communication overhead, and does not discuss prevention against attacks, such as *MITM*, replay, redirection, and impersonation. Additionally, the protocol suffers a synchronization problem related to the transmission of authentication related information. In summary, none of the above mentioned protocols is able to protect KSI_{ASME} as well as object's identity *IMSI* over the network.

3. AUTHENTICATION IN THE LTE NETWORK

In this section, we briefly outline the authentication process and the *EPS-AKA* protocol. The authentication in the *LTE* network generates different keys for Radio Resource Control (*RRC*) signaling, Non-Access Stratum (*NAS*) signaling, and User Plane (*UP*). Figure 1(a) and Figure 1(b) show the key generation process as well the authentication, integrity, and ciphering processes for *RRC* signaling, *NAS* signaling, and *UP*. The *SK* key, stored on the Universal Subscriber Identity Module (*USIM*) and the *HSS*, generates cipher key *CK*, integrity key *IK*, and a session key K_{ASME}, which then derives other cipher and integrity keys for all signaling planes as shown in Figure 1(a).

Figure 2 illustrates the *EPS-AKA* protocol, which starts by sending a service request with an attach request *NAS* message from the *UE* to the *MME*. The *MME* verifies the identity of the *UE* and asks to send its *IMSI* or Global Unique Temporary Identity (*GUTI*) using Tracking Area Update (*TAU*) procedure depending upon whether the *UE* is requesting first time or it is an existing user. The *MME* sends *GUTI* with *TAU* message to the old *MME* over the *S10* interface to extract the actual *IMSI* of the *UE*. If the *UE* was present earlier in a roaming *2G*/3G network, a new *MME* connects to the



(a) Key generation procedure. (b) Authentication, ciphering, and integrity flow.

Fig. 1: Authentication and key generation process in the *LTE* network.

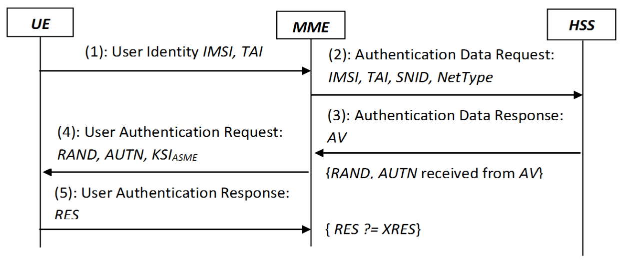


Fig. 2: *EPS-AKA* protocol of the *LTE* network.

$\{MAC = f_2(SQN, RAND, AMF)_{SK}, XRES = f_3(RAND)_{SK}, CK = f_4(RAND)_{SK}, IK = f_5(RAND)_{SK},$
$AK = f_6(RAND)_{SK}, K_{ASME} = f_7(SQN \oplus AK, SNID, CK, IK), AUTN = (SQN \oplus AK, AMF, MAC),$
$AV = (RAND, XRES, K_{ASME}, AUTN)\}$

old *MME* through Serving *GPRS* Gateway (*SGSN*) via *S3* interface to extract *IMSI* of the *UE*. Thereafter, the *MME* connects to the *HSS* via *S6a* interface and verifies *IMSI* of the *UE* through a diameter permission message. The *HSS* generates a new Authentication Vector (*AV*) set (or a single vector) and sends it to the *MME*. This *AV* consists of a random number *RAND*, Authentication Token (*AUTN*), signed response *XRES*, and access security management entity key K_{ASME}. Both, the *HSS* and the *UE* maintain a counter for synchronization purpose. A major improvement in *EPS-AKA* compared to *UMTS-AKA* is that the cipher key *CK* and the integrity key *IK* are never actually sent by the *HSS*. The *UE* sends signals to the *MME* with the type of access network it uses (for Evolved - Universal Terrestrial Radio Access Network (*E-UTRAN*), set *AMF* = 1). Thereafter, the *MME* sends *RAND* and *AUTN* to the *UE*, and waits for the response. The *MME* also sends a key set identifier *eKSI* to the *UE* through Evolved NodeB (*eNB*). There are two types of *eKSI*: one is KSI_{ASME} and other is KSI_{SGSN}. The KSI_{ASME} is used to indicate a native *EPS* security context, while the KSI_{SGSN} is used to indicate a mapping security context. On receiving, the *UE* computes *RES* and sends it to the *MME*. Then, the *MME* compares *RES* with the computed *XRES*. The *UE* gets authenticated only if both are same. The confidentiality and integrity protections in *NAS* signaling between the *UE* and the *MME* are provided by K_{NASenc} and K_{NASint} keys, respectively. The traffic confidentiality and integrity protections between the *UE* and the *eNB* are covered by K_{RRCenc} and K_{RRCint} keys, respectively, while *UP* data between the *UE* and the *eNB* (serving gateway-*SGW*) is protected by K_{UPenc} key. The *NAS* and *RRC* signaling integrity protections are provided by *AES* or *SNOW*.

4. SECURITY AND PRIVACY ATTACK MODEL

In this section, we discuss the security and privacy attack model for the *IoT*-enabled *LTE* network. We consider a communication scenario with legitimate users/devices as well as an adversary $\mathcal{A}$. If adversary $\mathcal{A}$ is able to retrieve the identity of a user/device by eavesdropping the network or by using a tracking device, $\mathcal{A}$ can perform the attack and harm the system on behalf of legitimate victim user/device. $\mathcal{A}$'s strength is defined by a set of oracles that it can access and be allowed to make queries. A weak adversary never corrupts the message whereas a destructive adversary may corrupt the message at any time. We consider a strong adversary that may corrupt the message at any time without destroying the message. Furthermore, the security issues of clear text transmission of *IMSI* and *KSI*$_{ASME}$ may result in various possible attacks. This security model is based on indistinguishability, where in the challenge phase the adversary is provided with two different messages. Later in the guess phase, $\mathcal{A}$ has to guess the correct message. $\mathcal{A}$ knows both messages in clear text, $\mathcal{A}$ is unable to identify which of the two messages was encrypted to produce a given ciphertext.

Definition 4.1. (*IND-CMA: Indistinguishability under Chosen Message Attack*): A protocol is *IND-CMA* secure if no adversary $\mathcal{A}$ can distinguish which of two messages msg_1 and msg_2 in time t was encrypted, and has no or negligible advantage.

$$\underset{k \leftarrow KG(1^\lambda)}{Pr}[\mathcal{A}(msg_1) = 1] - \underset{k \leftarrow KG(1^\lambda)}{Pr}[\mathcal{A}(msg_2) = 1] \leq \epsilon.$$

A man-in-the-middle attack can occur when a *UE* tries to connect to the *eNB*/*MME*, or in the case where the *UE* requires the transmission of *IMSI* during the initial request. In this attack, $\mathcal{A}$ puts itself in between the target object (user/device) and a genuine network, and captures, modifies, eavesdrops, spoofs signaling, and does data exchanges between both parties. This is only possible if the network is unencrypted. Unfortunately, most of the networks are either unencrypted or provide security with weak and vulnerable algorithms, such as *A5/1* and *A5/2*. $\mathcal{A}$ can intentionally re-send a previously used message as a fresh message to the *MME* or the *UE*, which results in a replay attack. Also, $\mathcal{A}$ may increase its signal strength to redirect and connect the legitimate user with a fake *MME* to perform redirection attack. Nowadays, due to the availability of phone number catcher in the market, it becomes easy to catch the *GSM*/*UMTS* phone number/*IMSI* over-the-air. The object-identity transmitted in clear text during the initial request procedure of *EPS-AKA* compromises the entire system. Creating security gap exploitation allowing an eavesdropper to track an object's location that results in an object-*ID* theft. $\mathcal{A}$ can send signaling and/or object data to the receiver in order to make the receiver believe that it is originating from a genuine source that leads to an impersonation attack. $\mathcal{A}$ can also capture subsequent session or derived keys based on the information from parent key and its *KSI*, which compromises key secrecy and causes key-*ID* theft.

On the other hand, revealing the user/device's identity to $\mathcal{A}$ results in privacy issues. For instance, $\mathcal{A}$ can easily track the behavior of the victim user/device, and perform unwanted activities, such as linking messages in order to extract information of the victim, sending fake traffic to the victim, and retrieving personal and location related information. In a computational environment, privacy properties are typically defined by means of games. We consider untraceability, forward privacy, and anonymity properties to be maintained by the *IoT*-enabled *LTE* network. In the challenge phase, we assume that $\mathcal{A}$ can eavesdrop communications and can also query all the messages in the system. Then $\mathcal{A}$ chooses a message msg_1 randomly from the set and makes a query. The game is over once $\mathcal{A}$ announces its guess of the selected message. Untraceability means that an adversary or others cannot trace actions performed by an object. The protocol satisfies untraceability if $\mathcal{A}$ cannot detect that the generated ciphertext message belongs to which one of the two known messages with probability higher than random guessing and also cannot learn at protocol level about the object to which the message belongs to. $\mathcal{A}$ cannot distinguish whether the two generated ciphertexts correspond to the same or two different messages, say msg_1 and msg_2, and belong to one or two different objects. We also consider forward and backward untraceability. Forward untraceability refers to $\mathcal{A}$ not being able to determine if at time t_{frw} ($t_{frw} > t$) a message belongs to a particular user/device. Similarly, backward untraceability means that $\mathcal{A}$ cannot determine if at time t_{brd} ($t_{brd} < t$) a message belonged to a particular user/device. We also model the forward privacy property of our system. It is similar to untraceability with the additional capability that $\mathcal{A}$ is able to break one of two messages and retrieves the information in the message. Clearly, $\mathcal{A}$ can trace the user/device. However, forward privacy is maintained if $\mathcal{A}$ is still unable to trace previous protocol sessions. Moreover, we also consider the anonymity property for our system, which refers to the fact that only the sender and the intended receiver can know the identity of actual object (user/device). There is a slight difference between user/device's anonymity and untraceability. Object anonymity means that except for the object and the home network, nobody including the serving network can retrieve the identity of the object, whereas untraceability means that except for the object and the home network, nobody including the serving network is able to identify previous protocol runs that involved the object.

5. PROPOSED PROTOCOL

This section presents a new authentication protocol for the *IoT*-enabled *LTE* network, which addresses the authentication shortcomings of the existing *LTE* network and supports *IoT* devices. The proposed protocol provides prevention against various security and privacy attacks for diverse types of *IoT* services. The purpose of the *eNB* node and other elements in our protocol are the same as in the *LTE* architecture. In this paper, we consider an identity of the object (mobile user equipment) as *IMSI*. This scenario can be further extended for other objects (devices). Below we describe the various parts that form our protocol: protocol setup, identity creation, protocol initialization, and protocol execution along with the corresponding protocol assumptions.

5.1. System Assumptions

We make the following assumptions commonly made in traditional cellular networks:

(1) The secret key *K* is stored in the Authentication Center (*AuC*)'s database as well as onto the *USIM* at the time of manufacturing, and is secure similar to the traditional cellular network. A session key *SK*, generated by *K*, is used at both ends.
(2) The *AuC* at the *HSS* is a trusted server that does not maliciously send the messages encrypted by the secret key of one user to others.

5.2. A New Generic Design of the Proposed Protocol

The proposed protocol, illustrated in Figure 3, is described in four parts as follows:
(1) **Protocol Setup:** Our protocol uses the $f_1()$, $f_2()$, $f_3()$, $f_4()$, $f_5()$, $f_7()$, $E/D\{\}$ functions, and a function to generate *Actcode*/*SIMcode*. Our protocol does not use *AK* key, which is generated by the $f_6()$ function. Therefore, we will not discuss the $f_6()$ function. We define the structure of various functions as follows:

f_1***() Function:*** It is a reversible symmetric encryption function, such as *AES-CTR* (*AES* with counter mode), where the plaintext and shared key generate the ciphertext, and then the ciphertext and same key are able to produce original plaintext. The *SK* key is available and stored at the *UE* as well as the *HSS*. This function is used to generate a temporary identity (*TID*) for each user and device.

f_2***() and*** f_3***() Functions:*** These two functions are used to generate the message authentication code *MAC* and the signed response *RES*, respectively. These functions can be implemented by a one-way *HMAC*, such as *HMACSHA1*. The *HMACSHA1* takes input as multiples of 512 *bits* with padding and *SK* key depending upon the type of signaling (*RRC*, *NAS*, or *UP*), and it generates 160 *bits* of hash code. Out of this, the first 64 *bits* are used as *MAC* and the next 64 *bits* represent the *RES*.

f_4***(),*** f_5***(), and*** f_7***() Functions:*** These are also one-way functions, such as *HMACSHA256*, which takes an input of 512 *bits* with padding and the *SK* key, and generates 256 *bits* of hash code. The first 128 *bits* are used as the *CK* key and the next 128 *bits* as the *IK* key. Furthermore, function f_7() is used to generate the K_{ASME} key. In order to generate this key, we implement f_7() as *HMACSHA256* with *CK* and *IK* keys that generates a K_{ASME} key of 256 *bits*. This key further generates different session keys (K_{NASint}, K_{NASenc}), (K_{RRCint}, K_{RRCenc}), and K_{UPenc} in different signaling planes.

$E/D\{\}$ ***Function:*** It is used to encrypt and decrypt transmitted messages over the network. The *AES-CTR* algorithm with K_{RRCenc}/K_{NASenc}/K_{UPenc} key is used to generate ciphers.

Actcode/SIMcode and KSI$_{ASME}$ ***Generation:*** We consider a very lightweight function to initially generate *Actcode/SIMcode* by using an *XOR* with left and right circular shift (*LCS/RCS*) of the bits of each *TID*. This function is strictly secret and operator-specific. We compute *KSI*$_{ASME}$ using $f^{'}()$ function implemented by $SHA256$.

All these functions generate the respective parameters with standard sizes that are supported by the current protocol security standards, and are being used in present traditional cellular networks.

(2) **Object (User/Device) Identity and Secrets Creation:** First, the *UE* generates a *TID* using function $f_1()$ (which is secret and shared only between the *UE* and the *HSS*) with *SK* key by passing in *IMSI* and timestamp T_1 as input (*TID* = $f_1(IMSI, T_1)_{SK}$). This $f_1()$ is a symmetric function, hence able to retrieve original *IMSI* when we pass *TID* and T_1 as inputs with *SK* key (*IMSI* = $f_1(TID, T_1)_{SK}$). *Actcode* is a one-time activation code that is sent to the *HSS*. The purpose of this code is to retrieve and verify *SIMcode*. It is assumed that a *SIMcode* is generated and stored onto the *USIM* and at *AuC* when a *USIM* card gets activated. This *SIMcode* is attached as a label to *SK* key at *AuC*. The generation of *Actcode* at *UE* and *SIMcode* at *HSS* are not publicly accessible and are secret in nature. This can be achieved as follows: at *UE*: *Actcode* = $LCS_n(T_1 \oplus SIMcode)$, and at *HSS*: *SIMcode* = $RCS_n(T_1 \oplus Actcode)$, where LCS_n = Left Circular Shift by *n*, RCS_n = Right Circular Shift by *n*, and *n* is the value of first digit of *TID* (in decimal, convert *TID* from bits to decimal and pick first leftmost decimal digit). After identifying the user/device in each session, the *HSS* sends a newly generated *Actcode* encrypted by $E\{\}/D\{\}$ to the respective *UE*. The *UE* uses *Actcode* in the next session while requesting to the *MME*/*HSS*. Thus, a new *Actcode* is used in each new session, which is secure from to be correctly retrieved by adversary $\mathcal{A}$.

(3) **Protocol Initialization:** In our protocol, the K_{ASME}, generated by the *HSS*, is sent and stored onto the *MME*. The advantage of doing this is that we do not need to execute the full *AKA* protocol again when re-synchronization of the *UE* is required. The keys, based on unique *KSI*$_{ASME}$, are generated only once. Hence, a new

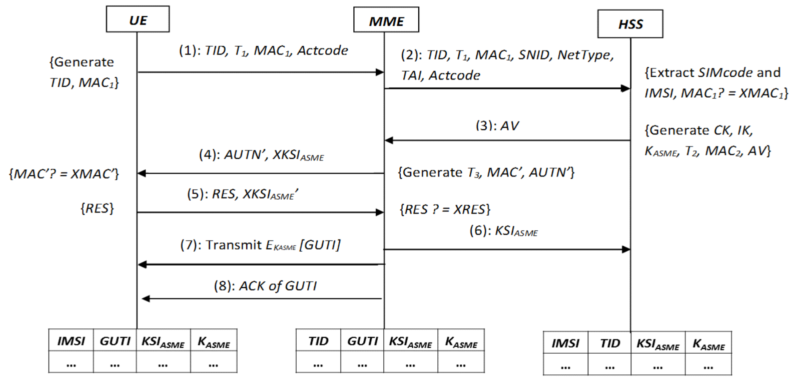


Fig. 3: Proposed *AKA* Protocol for the *IoT*-enabled *LTE* Network.

$\{Actcode = LCS_n(T_1 \oplus SIMcode), SIMcode = RCS_n(T_1 \oplus Actcode), TID = f_1(IMSI, T_1)_{SK}, MAC_1 = XMAC_1 = f_2(T_1, TID, TAI, Actcode)_{SK}, AV = (AUTN, K_{ASME}), AUTN = (T_2, AMF, MAC_2), MAC_2 = XMAC_2 = f_2(T_2, AMF)_{CK}, AUTN^{'} = (T_3, MAC^{'}, AMF), RES = XRES = f_3(T_3)_{K_{ASME}}, MAC^{'} = f_2(MAC_2, T_3, AMF, KSI_{ASME})_{K_{ASME}}, XMAC^{'} = f_2(f_2(T_2, AMF)_{CK}, T_3, AMF, KSI_{ASME})_{K_{ASME}}\}$

K_{ASME} is generated each time a request for *AV* initiates a new session. Therefore, the generation of new secret key no longer depends upon the previous keys and any other parameter. Each time, all the derived keys (K_{NASint}, K_{NASenc}), (K_{RRCint}, K_{RRCenc}), and K_{UPenc} are generated by K_{ASME}. For the initial session, a temporary *Actcode* is computed by the *UE*. For the subsequent sessions, the *HSS* can generate a new $Actcode$ using the following function, which is sent to the *UE* in cipher form: $Actcode_{new} = f^{'}(Actcode, Rand)_{SK}$, where *Rand* is a random number. $f^{'}()$ is implemented as *AES-CTR*. $\mathcal{A}$ cannot generate $Actcode_{new}$, as it does not know *SK* key and *Rand*.

(4) **Protocol Execution:** Initially, the *UE* sends a request (message (1) in Figure 3) to the *MME* for establishing a connection by transmitting *TID*, T_1, *Actcode*, and MAC_1, where *TID* is a temporary *ID*, T_1 is a timestamp, *Actcode* is a generated activation code for the *UE*, MAC_1 is a message authentication code ($MAC_1 = f_2(T_1, TID, TAI, Actcode)_{SK}$), and *SK* is a master key generated by *K*. The *MME* passes the received message to the *HSS* with Serving Network Identity (*SNID*) and Network Type (*NetType*), *i.e.*, *E-UTRAN* (message (2)). Upon receiving the message, the *HSS* first checks whether $T_1 < T_{current}$. If it is true, the *HSS* computes $XMAC_1 = f_2(T_1, TID, TAI, Actcode)_{SK}$ and compares it with the received MAC_1. If it holds, the *UE* is verified by the *HSS*. Then, the *HSS* computes *SIMcode* from the received *Actcode* and compares it with the stored value of *SIMcode*. If both *SIMcode* match, the *K* and *SK* keys corresponding to that *SIMcode* are retrieved. Thereafter, the *HSS* retrieves *IMSI* by $f_1()$ with *SK* key. If the computed *IMSI* is same as the stored *IMSI*, the *HSS* computes *AV*. Otherwise, the request is discarded and the connection is terminated. Afterwards, the *HSS* generates (T_2, K_{ASME}, *AMF*) and MAC_2 ($MAC_2 = f_2(T_2, AMF)_{CK}$) as a part of *AV* along with *CK* and *IK* keys, and sends *AV* to the *MME* (message (3)), where *AMF* is the Authentication Management Field. Upon receiving the message from the *HSS*, the *MME* generates (T_3, $MAC^{'}$, $AUTN^{'}$) and sends $AUTN^{'}$ to the *UE* along with a generated $XKSI_{ASME}$ (message (4)), where $MAC^{'} = f_2(MAC_2, T_3, AMF, KSI_{ASME})_{K_{ASME}}$. On receiving the message, the *UE* generates (*CK*, *IK*, KSI_{ASME}), computes $XMAC^{'} = f_2(f_2(T_2, AMF)_{CK}, T_3, AMF, KSI_{ASME})_{K_{ASME}}$, and

compares whether $MAC' \stackrel{?}{=} XMAC'$. If such condition does not hold, the connection is terminated. Otherwise, the *MME* and the *HSS* are verified by the *UE*. Then the *UE* computes signed response *RES*, generates a new $XKSI'_{ASME}$, and computes a new KSI_{ASME} by using $XKSI_{ASME}$ and $XKSI'_{ASME}$ as inputs to a predefined $f'()$ function shared between the *MME* and the *UE*. Thereafter, the *UE* transmits the message (5) to the *MME* with *RES* and $XKSI'_{ASME}$, where $RES = f_3(T_3, XKSI'_{ASME})_{K_{ASME}}$. Upon receiving message (5), the *MME* computes $XRES = f_3(T_3, XKSI'_{ASME})_{K_{ASME}}$ and compares it with the received *RES*. If both are equal, the *UE* is verified by the *MME*, and then the *MME* generates a new KSI_{ASME}, by putting in $XKSI_{ASME}$ and $XKSI'_{ASME}$ to the $f'()$. Hence, the *UE* and the *MME* have the same KSI_{ASME} without transmitting it over the network. Afterward, the *MME* transmits the actual KSI_{ASME} to the *HSS* (message (6)). All the *UE*, *MME*, and *HSS* store K_{ASME} and KSI_{ASME} in their memory/database. After completion of authentication, the *MME* sends message (7) to the *UE* with encrypted Global Unique Temporary Identity (*GUTI*), and finally the *UE* acknowledges the receipt of *GUTI* to the *MME* (message (8)).

6. SECURITY ANALYSIS OF THE PROPOSED PROTOCOL

In this section, we present security analysis of our protocol in terms of security goals and properties, resistance against attacks, and privacy properties. A flowchart describing differences between the *EPS-AKA* and proposed protocol is shown in Figure 4.

Mutual Authentication and No Synchronization Issue. *The proposed protocol provides mutual authentications between the UE-MME and the UE-HSS. Our protocol also solves synchronization issue in the LTE network.*

In our protocol, the *HSS* authenticates the *UE* by verifying MAC_1. To authenticate the *HSS*, the *UE* checks the received MAC' from the *MME*. If MAC' is equal to $XMAC'$, both, the *HSS* and the *MME* are authenticated by the *UE*. This process ensures mutual authentications between the *UE* and the *HSS*. Furthermore, the *MME* authenticates the *UE* by verifying *RES*. After receiving the message, the *MME* computes *XRES* and compares whether $RES \stackrel{?}{=} XRES$. The *UE* is authenticated, if this equality holds. The same procedure takes place in authenticating the *UE* when the *MME* receives *RES* while communicating with only *MME* (within a session).

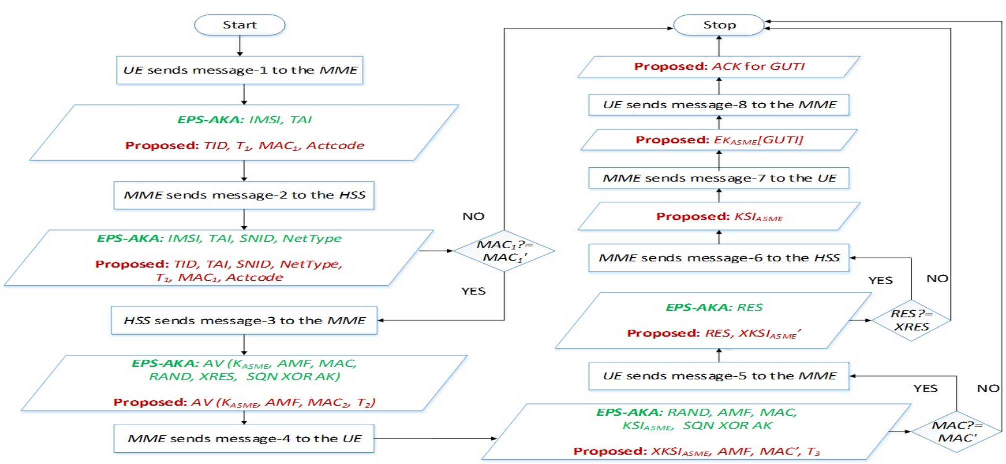


Fig. 4: Protocol flowchart: *EPS-AKA* (green) and proposed protocol actions (red).

Property 1. IND-CMA. *The proposed protocol maintains IND-CMA.*

In our protocol, the messages encrypted by the same session key generate different ciphertexts, as at least one of the input parameters (in plaintext) is always different. The *UE* generates *TID* as $f_1(\textit{IMSI}, T_1)_{SK}$, where T_1 changes each time the *UE* sends a request to the *MME*. Furthermore, the *UE* also generates a fresh *Actcode* and sends it to the *MME*. The *UE* and the *MME* generate a fresh K_{ASME} key for each session. Moreover, the *SK* and K_{ASME} keys are never sent over the network, hence $\mathcal{A}$ cannot retrieve these keys. Since we use *AES-CTR* as an encrypted algorithm, it encrypts the successive values of a counter with *AES*, and regurgitates the concatenation of the encrypted blocks. It is challenging for $\mathcal{A}$ to distinguish between such streams of equal lengths. Even if we provide a message and an encrypted oracle to adversary, he/she does not have any advantage to correctly guess any other messages. Therefore, *AES-CTR* is chosen message indistinguishable, even from random noise.

Property 2. Key Secrecy. *Key secrecy for session and derived keys is maintained.*

Any derived key, say Q_{key}, is generated by the session key K_{ASME} at the *UE* in the *LTE* network, which is also securely received by an honest serving *MME* network from the home *HSS* network. New session keys are derived by K_{ASME} key, and these keys are independent from each other in each session. Therefore, the Q_{key} key preserves one-session secrecy in the *IoT*-enabled *LTE* network. Our protocol is also able to solve the synchronization issue, as it does not use any sequence number. Timestamps are used in the protocol to prevent replay attack, not for synchronizing received *AV*.

Property 3. Key Theft, Key-identity Theft, and Attempts to Derive Keys. *Adversary $\mathcal{A}$ cannot extract the secret key SK, session key K_{ASME}, and key-ID (KSI$_{ASME}$) over the network. In fact, $\mathcal{A}$ will fail to retrieve SK or K_{ASME} key, even if it captures Actcode of a user/device sent over the network.*

Each K_{ASME} is generated by the *CK* and *IK* keys, and is generated at the *MME* and at the *UE*. Note that $\mathcal{A}$ cannot retrieve the *SK* and K_{ASME} keys, as they are never sent over the network. Moreover, if $\mathcal{A}$ retrieves few *Actcode*, it cannot derive any relation among them, as these *Actcode* are randomly generated each time. Moreover, each *Actcode* is sent exactly once in plaintext over the network from the *UE* to the *MME*. It will not work, if $\mathcal{A}$ uses previous generated *Actcode* in a new session. Furthermore, if $\mathcal{A}$ modifies *Actcode*, the computed $MAC_1^{'}$ at the *MME* will not match with the received MAC_1 at the *UE*. Hence, the connection will be terminated. *XKSI*$_{ASME}$ is sent over the network with the integrity protection. However, the actual *KSI*$_{ASME}$ is never sent over the network, which prevents the *LTE* network against Key-*ID* theft.

Property 4. Identity Privacy and Identity Theft. *Adversary $\mathcal{A}$ cannot trace the original identity of the UE. In fact, $\mathcal{A}$ will fail to recognize the actual user, even if it captures TID of a user/device.*

The privacy of each *UE* (identity) is well protected during authentication over the network. The *TID* is computed from the original *IMSI* as *TID* = $f_1(\textit{IMSI}, T_1)_{SK}$, where function $f_1()$ is reversible in nature for encryption and decryption. We implement function $f_1()$ as *AES-CTR* with a K_{ASME} key, which prevents the system against *MITM* attacks, as it is able to hide the actual identity of the user, and also no practical full attack has been found on *AES* until today. For all the authentication requests including the subsequent requests, a different *TID* is generated each time a user connects to the visiting *MME*. The *MME* and *HSS* flush out the *TID* from their memory, once a connection between the *UE* and the *MME* is terminated (either successful or invalid request abort). Hence, $\mathcal{A}$ cannot relate *TID* with the original *IMSI* of a user.

Property 5. Session Unlinkability, and Forward and Backward Untraceability. *Adversary $\mathcal{A}$ cannot link current session information with the previous sessions. Moreover, our protocol maintains perfect forward/backward untraceability.*

For each fresh request, the *UE* and the *MME* generate a fresh session key, temporary identity of the user/device, *Actcode*, K_{ASME} session key, and other derived keys. Therefore, $\mathcal{A}$ cannot retrieve any information based on linkability among various requests. Moreover, the K_{ASME} key is generated each time for a session. Hence, even compromising the current K_{ASME} will not allow $\mathcal{A}$ to generate future and past keys. In addition, if the past keys are compromised, they cannot be used for future sessions, as both ends generate a new session and derived keys in each session.

Property 6. Information Unlinkability and Confidentiality. *$\mathcal{A}$ cannot generate a SIMcode even if it receives Actcode considering generation function secret in nature.*

The generation of *Actcode* and *SIMcode* are secret in nature and are network operator-specific. If $\mathcal{A}$ finds *Actcode*, it cannot obtain *SIMcode*, as $\mathcal{A}$ does not know the generation function and cannot retrieve the *SK* key. Function $f_1()$ is used to generate $TID = f_1(IMSI, T_1)_{SK}$ and $IMSI = f_1(TID, T_1)_{SK}$. If $\mathcal{A}$ knows $f_1()$, it cannot forge the function, as $\mathcal{A}$ does not know *SK* key. Functions $f_2()$, $f_3()$, $f_4()$, and $f_5()$ are based on *HMAC* with hash, such as *HMACSHA1* and *HMACSHA256*. These functions are one-way in nature, and are considered secure till date. Therefore, it is not possible to break the security of these functions. $\mathcal{A}$ cannot retrieve any confidential information.

Property 7. Attack Resistance. *Our protocol defeats MITM, replay, redirection, and impersonation attacks between the UE and the MME. Furthermore, $\mathcal{A}$ can neither compromise message security nor retrieve any information by delaying messages.*

We justify that our protocol is secure against various attacks as follows:

(1) Replay Attack: Our protocol includes timestamps (T_1, T_2, T_3) with each transmitted message over the network. Therefore, the protocol is free from this attack.

(2) MITM Attack: Privacy preservation of *IMSI* and KSI_{ASME} helps to prevent the *LTE* network against *MITM* attack. In addition, message encryption with cipher keys using *AES-CTR* in different contexts defeats this attack. For an example, at the *MME*, *GUTI* is encrypted by K_{ASME}, and is sent it to the *UE*. Similarly, K_{NASenc}, K_{RRCenc}, and K_{UPenc} are used for ciphering in *NAS* signaling, *RRC* signaling, and *UP* planes existing between *UE-MME*, *UE-eNB*, and *UE-eNB*, respectively. Even if $\mathcal{A}$ captures *TID* and *Actcode*, it cannot derive the actual key K, as this key is securely stored at *AuC* similar to the traditional cellular network. It is impossible to retrieve K key, because it never participates, and only the session keys are used to perform operations.

(3) Redirection Attack: Our protocol uses *MAC* to maintain the integrity of *TAI* that prevents the network against redirection attack. This attack is easily possible when $\mathcal{A}$ gets the correct *UE*'s information. In our protocol, the *UE* includes *TAI* of *eNB* in MAC_1, and transmits MAC_1 to the *MME*. Authentication request is discarded, when the *HSS* fails to match the *TAI* sent by the *MME* and embedded in the $XMAC_1$.

(4) Object-identity Theft: In our protocol, the *UE* transmits a temporary object-identity *TID* instead of clear text *IMSI* during initial attach procedure over the *LTE* network that protects the object's information against object identity theft.

(5) Impersonation Attack: In our protocol, MAC_1 is computed at the *UE* and sent to the *HSS*. The *HSS* computes $XMAC_1$ using the *SK* key. Note that the *HSS* computes $XMAC_1$ that also includes the *TAI* of the *UE* received from the *MME*, whereas the *UE* computes MAC_1 based on actual *TAI*. In addition, $\mathcal{A}$ must reply with a valid response to the *MME* such that (*RES* = *XRES*) in order to impersonate the *UE*. However, it is not possible for $\mathcal{A}$ to have the correct *RES*. Similarly, the attempt to impersonate the *MME* will not be successful, as the *UE* verifies that it had not requested for an authentication, but still receives $AUTN'$ from the *MME*. If an adversary knows the initial working of the function to generate *Actcode* and later uses a code to impersonate a user, the server rejects the code, as the code can only be used once at the initial stage when the *USIM* card gets activated. Furthermore, the adversary cannot generate the

Table III: Summary of Various *LTE AKA* Protocols

Prevent Parameters	*EPS-AKA*	[Kϕien 2011]	[Gu 2011]	[Chou. 2012]	[Pukh. 2012]	Proposed *AKA*
IMSI over the network	No	No	Yes	Yes	No	Yes
KSI$_{ASME}$ over the network	No	No	No	No	No	Yes
Replay Attack	Yes	Yes	Yes	Yes	Yes	Yes
Redirection Attack	Yes	Yes	No	No	Yes	Yes
MITM Attack	No	No	No	No	No	Yes
User-*ID* Theft	No	No	No	Yes	No	Yes
Key-*ID* Theft	No	No	No	No	No	Yes
Synchronization	No	No	Yes	No	No	Yes
IoT Support	No	No	No	No	No	Yes

same code as a legitimate user for the subsequent authentications because the *SK* key and a random number *Rand* are unknown to $\mathcal{A}$, even if it knows the function.

(6) Message Modification: The integrity protection of the transmitted messages including message content and its threshold delivery in time, (*i.e.,* $T_{receive} \leq T_{generate} + T_{threshold}$) is maintained using *MAC* that helps to prevent the system against message tampering. We consider and define *MAC* functions for protecting message integrity. A *MAC* = $\{$*KeyGen*, *Comp*, *Verif*$\}$ is a triple (key generation, *MAC* computation, and *MAC* verification) of an algorithm with associated key space $\mathbb{K}$ and message space $\mathbb{M}$. The standard security notion for a randomized *MAC*: $\mathbb{K} \times \{0,1\}^* \rightarrow \{0,1\}^*$ is Unforgeability under Chosen Message and Chosen Verification Queries Attack (*UF-CMVA*).

Definition 6.1. We consider $Adv_{MAC}^{UF-CMVA}$ ($\mathcal{A}$, λ, Q_{Comp}, Q_{Verif}) as an advantage for adversary $\mathcal{A}$ in forging a message with a random key $k \leftarrow KG(1^\lambda)$, where $\mathcal{A}$ can make Q_{Comp} and Q_{Verif} queries.

Clearly, $\mathcal{A}$ cannot obtain such an advantage, as we use one-way *HMACSHA1* and *HMACSHA256*, which are hard to break. The protocol is also secure against narrow pipe attack related to *HMAC* functions. Narrow pipe attack can compromise *HMAC* function only with related key, whereas hash functions used with *HMAC* can be compromised with single and related keys. Our goal is to find a patch that does not affect the definition of hash function and could prevent *HMAC* against this attack. We propose a solution by appending an extra fixed bit (1 or 0) before an input message. We also use different initialization vector values for inner and outer hash functions in *HMAC* [Peyrin et al. 2012]. This process prevents *HMAC* against narrow pipe attack.

Table III summarizes several features of *EPS-AKA*, proposed protocol, and other existing protocols. The proposed protocol prevents *IMSI* and *KSI*$_{ASME}$ to be sent over the network, and provides resistance against attacks and synchronization solution.

Property 8. Untraceability, Forward Privacy, and Anonymity. *Adversary $\mathcal{A}$ cannot compromise privacy of the user/device, as our protocol maintains untraceability, forward privacy, and anonymity properties.*

Definition 6.2. (*Untraceability*): Our protocol satisfies untraceability if $\mathcal{A}$ cannot distinguish whether two *TID*s correspond to the same *UE* or two different *UE*.

$$\textit{Verif}(publicChannel).[(f_1(\textbf{\textit{IMSI}}_1, T_1)_{SK_1}, f_1(\textbf{\textit{IMSI}}_2, T_2)_{SK_2})|\textbf{\textit{TID}}_i|\textbf{\textit{UE}}|\textbf{\textit{HSS}}] \approx \textit{Verif}(publicChannel).[f_1(\textbf{\textit{IMSI}}_1, T_1)_{SK_1}|f_1(\textbf{\textit{IMSI}}_1, T_2)_{SK_1}|\textbf{\textit{TID}}_i|\textbf{\textit{UE}}|\textbf{\textit{HSS}}]$$

If adversary $\mathcal{A}$ retrieves a *TID* and makes a query from a random oracle to generate several *TID* from *IMSI* identities, $\mathcal{A}$ cannot conclude which *IMSI* matches with the retrieved *TID*. The reason is that our protocol implements *AES-CTR* with inputs of *IMSI* and a timestamp T_i to provide a unique *TID* each time. Furthermore, our protocol generates a new key for each session, and then derives various other keys for the *RRC* signaling, *NAS* signaling, and *UP* planes. Therefore, linkability to previous sessions is not possible. Also, by keeping the generated identities and messages during current session (say time t), $\mathcal{A}$ cannot determine and prove after t_{frd} time whether these messages belong to a particular object. Similarly, $\mathcal{A}$ cannot retrieve whether t_{brd} time ago, these messages were generated by a particular object, as each session's identities, keys, and messages are independent.

Definition 6.3. (*Forward Privacy*): Our protocol satisfies forward privacy if $\mathcal{A}$ is allowed to trace the user/device in current session, but it cannot trace the information related to the previous protocol sessions. In other words,

$$Verif(publicChannel).[(\textbf{\textit{IMSI}}_1 at\ \textbf{\textit{T}}_3, f_1(\textbf{\textit{IMSI}}_2, T_4)_{\textbf{\textit{SK}}_2})|\textbf{\textit{TID}}_i|\textbf{\textit{UE}}|\textbf{\textit{HSS}}] \approx$$
$$Verif(publicChannel).[f_1(\textbf{\textit{IMSI}}_1, T_1)_{\textbf{\textit{SK}}_1}|f_1(\textbf{\textit{IMSI}}_2, T_2)_{\textbf{\textit{SK}}_2}|\textbf{\textit{TID}}_i|\textbf{\textit{UE}}|\textbf{\textit{HSS}}];$$

We also consider forward privacy scenario, where even if $\mathcal{A}$ is given a breakable *IMSI*$_1$ at time T_3, $\mathcal{A}$ cannot trace *TID*$_1$ at time T_1 due to identity generation by *AES-CTR*, where $T_1 < T_2 < T_3 < T_4$. Furthermore, we quantify the anonymity provided by *TID* in terms of the advantage of $\mathcal{A}$ for correctly guessing the challenge bit.

Definition 6.4. (*IND-ANO: Indistinguishability under Anonymous Identity*): Our protocol is *IND-ANO* if no adversary $\mathcal{A}$ at time t can distinguish between two chosen identity *IMSI*$_1$ and *IMSI*$_2$ with negligible ϵ advantage.

$$Pr[\mathcal{A}(f_1(\textbf{\textit{IMSI}}_1, T_1)_{\textbf{\textit{SK}}_1})) = 1] - Pr[\mathcal{A}(f_1(\textbf{\textit{IMSI}}_2, T_2)_{\textbf{\textit{SK}}_2})) = 1] \leq \epsilon.$$

Our protocol maintains anonymity, as the actual identity is only known to the *UE* and the *HSS*. The serving network *MME* believes on the facts provided by the *HSS*.

7. PERFORMANCE ANALYSIS

This section provides performance analysis of our protocol in terms of storage overhead at the *UE*, *MME*, and *HSS*, bandwidth consumption, and efficiency of the system.

(1) Storage Overhead at UE, MME, and HSS: Our protocol generates a light storage overhead (see Figure 3) in order to successfully prevent clear text transmission of *IMSI* and *KSI*$_{ASME}$. The *KSI*$_{ASME}$ and K_{ASME} are stored to generate different subsequent keys for signaling and data channels. The *UE* and the *MME* store (*IMSI*, *GUTI*, *KSI*$_{ASME}$, K_{ASME}) and (*TID*, *GUTI*, *KSI*$_{ASME}$, K_{ASME}), respectively, whereas the *HSS* stores (*TID*, *IMSI*, *KSI*$_{ASME}$, K_{ASME}).

(2) Communication Overhead: We calculate the total transmitted messages size (*bits*) in order to evaluate communication overhead of our protocol considering single authentication vector. We assume that all the protocols are of the standard parameters and their sizes. The total number of bits in the messages of each *LTE AKA* protocol is as follows: *EPS-AKA*=1638 *bits*, Kϕien's *AKA*=1752 *bits*, Purkhiabani *et al.*'s *AKA*=2019 *bits*, Choudhury *et al.*'s *AKA*=1890 *bits*, and our *AKA* protocol=1647 *bits*. Note that the size of *ACK* is 3 *bits*. However, even if we make it 8 *bits*, our protocol is still better in terms of each point of discussion in the paper as compared to *EPS-AKA*. In fact, 3 *bits* leads to 8 different combinations, which is more than sufficient, as there is only one *ACK* needs to be sent during authentication. Total bandwidth used by our protocol during authentication as compared to the *EPS-AKA*, Kϕien's [Kϕien 2011], Purkhiabani's [Purkhiabani 2012], and Choudhury's [Choudhury et al. 2012] protocols are 100.5%, 94.0%, 81.5%, and 87.1%, respectively. Hence, our protocol reduces 6%, 18.5%, and 12.9% bandwidth utilization during authentication in comparison to

Table IV: Bandwidth Consumption by Various Protocols

Bandwidth Utilization (*bits*)	*EPS-AKA*	[Kϕien 2011]	[Purkhiabani 2012]	[Choudhury et al. 2012]	Proposed-*AKA*
Between *UE-MME*	627	676	944	794	697
Between *MME-HSS*	1011	1076	1075	1096	886

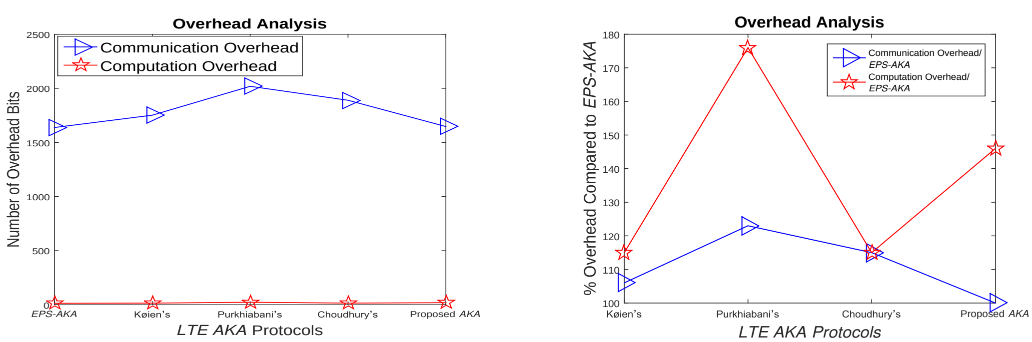
[Kϕien 2011], [Purkhiabani 2012], and [Choudhury et al. 2012] protocols respectively, except -0.5% for *EPS-AKA*. There is only 9-*bit* overhead difference between *EPS-AKA* and our protocol, which can be considered negligible.

(3) Computation Overhead: In order to evaluate the total computation overhead generated by each protocol with global values, all the computation functions are considered a unit value. This is a fair assumption [Zhang and Fang 2005], [Al-Saraireh and Yousef 2006], [Saxena and Chaudhari 2014], and the reason for choosing unit value is that it considers all the functions of various *AKA* protocols as global, and it assigns an equal weight without knowing their structures. We evaluate total computation overhead generated by each protocol as follows: *(1) EPS-AKA* protocol = 13, *(2) Kϕien's* protocol = 11, and with the generation of K_{ASME} by a standard function with *CK* and *IK* at the *UE* and the *MME*, total functions = 15, *(3) Purkhiabani's* protocol = 23, *(4) Hiten Choudhury's* protocol = 15, and *(5) Proposed* protocol = 19.

Figure 5(a) represents the communication and computation overheads of all the *LTE AKA* protocols, and Figure 5(b) compares them with respect to *EPS-AKA*. It shows that our protocol is efficient, as it generates lower communication overhead.

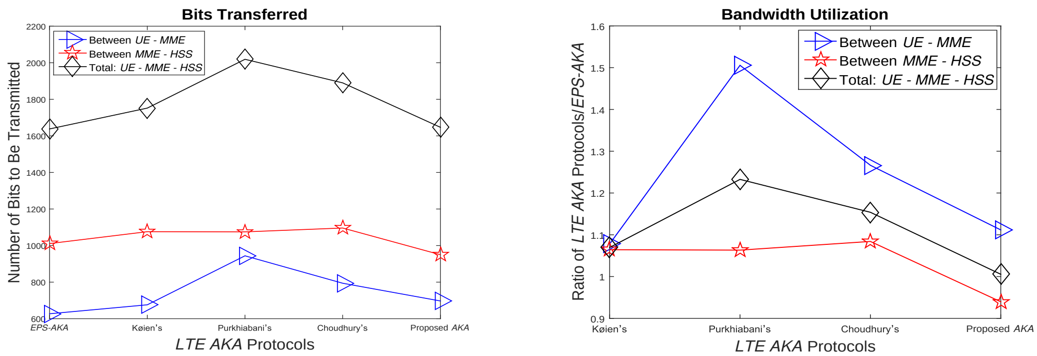
(4) Bandwidth Consumption: Our protocol is able to reduce bandwidth utilization between the *MME* and the *HSS*. From Table IV, it can be observed that for a single authentication, the protocol lowers (100-(950/1011)*100=) 6.1%, 11.8%, 11.7%, and 13.4% of the bandwidth utilization between *MME-HSS* as compared to the *EPS-AKA*, [Kϕien 2011], [Purkhiabani 2012], and [Choudhury et al. 2012] protocols, respectively.

Figure 6(a) illustrates the bandwidth consumption between *UE-MME*, *MME-HSS*, and *UE-HSS*, whereas Figure 6(b) shows the bandwidth utilization by other *LTE AKA* protocols with respect to *EPS-AKA*. The bandwidth utilization between the *UE* and the *MME* is slightly increased by our protocol about 11% and 3% in comparison to the *EPS-AKA* and Kϕien's [Kϕien 2011] protocols (while reduces by 26.2% and 12.3% with respect to the Purkhiabani's [Purkhiabani 2012] and Choudhury's [Choudhury et al. 2012] protocols). But we cannot declare this as a limitation, because overall in



(a) *LTE AKA* protocols. (b) *LTE AKA* protocols/*EPS-AKA*.

Fig. 5: Communication and computation overheads analysis.



(a) *LTE AKA* protocols. (b) *LTE AKA* / *EPS-AKA* protocols.

Fig. 6: Bandwidth consumption.

each authentication process, our protocol reduces 105 *bits* (1752-1647=105 *bits*) to be transmitted as compared to the Kϕien's [Kϕien 2011] protocol. In each authentication, our protocol lowers 61 *bits* and 126 *bits* with respect to the *EPS-AKA* and Kϕien's [Kϕien 2011] protocols. The bandwidth consumption by other protocols stated in the literature are not computed, as they do not clearly define their parameters.

7.1. Simulation of the Proposed Protocol

We simulate the proposed protocol with a client server paradigm, where the *UE* is a client and the *MME*/*HSS* are servers. All the simulated results are obtained on Core i3 processor, *2GB RAM*, 320*GB* hard disk, *Windows7*, and *J2ME* Wireless Tool Kit (*WTK*) with *JDK1.7*. We implement functions $f_4()$, $f_5()$, and $f_7()$ as *HMACSHA256*. Further, functions $f_2()$ and $f_3()$ are considered as *HMACSHA1*. However, all these functions are network operator-specific. The output of functions $f_4()$ and $f_5()$ are truncated to 128 *bits*, as these functions are used to generate 128 *bits* of *CK* and *IK* keys, respectively. Similarly, the output of functions $f_2()$ and $f_3()$ are also truncated to 64 *bits*, as $f_2()$ produces the output $MCA_1/MAC_2/MAC'$ of 64 *bits* in size, whereas $f_3()$ generates an output as *RES* of 64 *bits*. we use an extra fixed bit before the input message and also use different initialization vector values for the inner and outer hash functions in *HMAC*. This prevents *HMAC* against narrow pipe attack. The function to generate *Actcode*/*SIMcode* is an *XOR* function with *LCS*/*RCS*, whereas *AES* with counter mode (*AES-CTR*) is suitable for the encryption/decryption ($EK\{\}/DK\{\}$) [Saxena and Chaudhari 2014] with 256-*bit* K_{ASME} key. The output of $f_1()$ is *TID*/*IMSI* of 128 *bits*. Therefore, we implement $f_1()$ with 128 *bits* *SK* key similar to *AES* encryption (to get *TID*/*GUTI*) and decryption (to retrieve *IMSI*). The generation of KSI_{ASME} is implemented by *SHA256* hash function, which takes 20 milliseconds (*ms*) to compute.

Simulation results presented in Table V and Table VI are the average of 30 iterations for each output value. Here, the unit of time is *ms* and memory/space is measured in *bytes*. We calculate the transmission time for each message in our protocol with upload link speed at 11 *Mbps* and download link speed at 20 *Mbps* [EE 2014]. The Execution Time (*Ext*), Process *CPU* Time (*PCPU*), and Total Memory Usage (*TUM*) of functions $EK\{\}/DK\{\}$ and $f_1()$ can be observed from Table V. Similarly, Table VI shows the computation results of $f_2()/f_3()/f_4()/f_5()/f_7()$ and *Actcode*/*SIMcode*. Interestingly, we observed the same encryption/decryption time by *AES* with key sizes of 128 *bits* and 256 *bits* on *J2ME WTK*. *AES* took 8.8 *ms* for $EK\{\}$ and 9.1 *ms* for $DK\{\}$ [Saxena and Chaudhari 2014]. On average, the total execution time for our protocol = 3.23 *Sec*.

Table V: Computation Results of $f_1()$ and $EK\{\}/DK\{\}$

EK{} ExT (ms)	*TUM (bytes)*	*DK{} ExT (ms)*	*TUM (bytes)*
8.8	9329.6	9.1	4816

Table VI: Computation Results of $f_2()/f_3()/f_4()/f_5()/f_7()$/*Actcode*

Functions	*ExT (ms)*	*PCPUT (ms)*	*TUM (bytes)*
$f_2()/f_3()$ = *HMACSHA1*	221.60	296.40	15211840
$f_4()/f_5()/f_7()$ = *HMACSHA256*	273.41	296.40	15204024
Actcode/SIMcode	0.89	93.60	12968290

8. CONCLUSIONS

This paper has described a secure protocol for the *IoT*-enabled *LTE* network. This protocol presents several advantages with respect to the *LTE* network, and provides a secure and privacy-preserved environment that enables secure communications among the *IoT* objects (users and devices). Our protocol also prevents the need of synchronization unlike in the *EPS-AKA* protocol, as it does not use sequence number for the transmission of authentication information, instead uses *MAC* to verify information. Security analysis demonstrates that our protocol maintains collision-free *MAC*s, indistinguishability under chosen message attack, key secrecy and theft, identity privacy and theft, session unlinkability, and confidentiality. Our analysis indicates that the proposed protocol provides prevention against various attacks, such as man-in-the-middle attack (using *AES-CTR*), object-*ID* theft (by generating *TID*), key-*ID* theft (by generating secret KSI_{ASME}), replay attack (by using timestamps), impersonation attack (using *TID* and signed response *RES*), and redirection attack (using *TAI* and *MAC*). It also provides untraceability, forward privacy, and anonymity to the *IoT* objects (users/devices). Simulation results show that for a single authentication vector, our protocol utilizes lesser bandwidth (at least 6%) between the *MME* and the *HSS*, and lowers communication overhead (at least 6% except *EPS-AKA*) during authentication as compared to the existing protocols of the *LTE* network. The execution time of our protocol is 3.23 *Sec.*, which is a reasonable time frame in real time. Therefore, our protocol is suitable to use for the communications among various *IoT* objects in the *LTE* network. To the best of our knowledge, this is the first attempt of protecting the identities of key set identifier and the object over the network to make it *IoT*-enabled.

REFERENCES

Mehdi Aiash, Glenford Mapp, and Raphael Phan. 2010. Providing security in 4G systems: unveiling the challenges. In *Proceedings of the Advanced Int. Conf. in Telecomm.* IEEE, Barcelona, Spain, 439–444.

Jaafer Al-Saraireh and Sufian Yousef. 2006. A new authentication protocol for UMTS mobile networks. *EURASIP J. Wireless Communication and Networking* 2006, 2 (2006), 19–25.

Muhammad Alam, Du Yang, Jonathan Rodriguez, and Raed A. Abd-alhameed. 2014. Secure device-to-device communication in LTE-A. *IEEE Communications Magazine* 52, 4 (2014), 66–73.

Hani Alquhayz, Ali Al-Bayatti, and Amelia Platt. 2012. Security management system for 4G heterogeneous networks. In *Proceedings of the World Congress on Engg.* IEEE, London, UK, 1–5.

Barbara V. Lundin 2015. 50 Billion Connected IoT Devices by 2020. (2015). http://www.smartgridnews.com/story/50-billion-connected-iot-devices-2020/2015-04-21.

Chin-Chen Chang, Jung-San Lee, and YaFen Chang. 2003. Efficient authentication protocols of GSM. *Computer Communication* 28, 8 (2003), 921–928.

Check Point Soft. Tech. 2013. Next Generation Security for 3G and 4G LTE Networks. (2013). https://www.checkpoint.com/downloads/product-related/whitepapers/wp-ng-mobile-network-security.pdf.

Imrich Chlamtac, Marco Conti, and Jennifer N. Liu. 2003. Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks* 1, 1 (2003), 13–64.

Hiten Choudhury, Basav Roychoudhury, and Dilip K. Saikia. 2012. Enhancing user identity privacy in LTE. In *Proceedings of the TrustCom*. IEEE, Liverpool, UK, 949–957.

EE 2014. Double Speed 4G EE. (2014). Explore.ee.co.uk/ee-network/4Gee/doublespeed-4Gee.

Ali Fanian, Mehdi Berenjkoub, and T. Aaron Gulliver. 2010. A symmetric polynomial based mutual authentication protocol for GSM networks. In *Proceedings of the WCNC*. IEEE, Sydney, Australia, 1–6.

Mahdi D. Firoozjaei and Javad Vahidi. 2012. Implementing Geo-encryption in GSM Cellular Network. In *Proceedings of the 9th Inter. Conf. on Communi.* IEEE, Bucharest, Romania, 299–302.

GMSA 2015. Global mobile Suppliers Association. (2015). http://www.gsacom.com/gsm_3g/info_papers.php4.

GSM Association 2014. IoT Device Connection Efficiency Guidelines, Version 1.0. (2014). http://www.gsma.com/connectedliving/gsma-iot-device-connection-efficiency-guidelines.

Lili Gu and Mark A. Gregory. 2011. A green and secure authentication for 4th generation mobile network. In *Proceedings of the ITNAC*. IEEE, Melbourne, Australia, 1–7.

F. Hadiji, F. Zarai, and A. Kamoun. 2009. Authentication protocol in fourth generation wireless networks. In *Proceedings of the WOCN*. IEEE, Cairo, Egypt, 36–39.

Dake He, Jianbo Wang, and Yu Zheng. 2008. User authentication scheme on self-certified public key for next generation wireless network. In *Proceedings of the ISBAST*. IEEE, Islamabad, Pakistan, 1–8.

Roger P. Jover. 2015. *Security and Privacy in the Internet of Things (IoT): Models, Algorithms, and Implementations*. Taylor & Francis, 1–23.

Geir M. Kϕien. 2011. Mutual entity authentication for LTE. In *Proceedings of the 7th Intern. Conf. of Wireless Communication and Mobile Computing*. IEEE, Istanbul, Turkey, 689–694.

C. C. Lee, M. S. Hwang, and W. P. Yang. 2003. Extension of authentication protocol for GSM. *IEE Proc. of Comm.* 150, 2 (2003), 91–95.

Xingqin Lin, Jeffrey G. Andrews, Amitabh Ghosh, and Rapeepat Ratasuk. 2014. An overview of 3GPP device-to-device proximity services. *IEEE Communications Magazine* 52, 4 (2014), 40–48.

Yi-Bing Lin, Ming-Feng Chang, Meng-Ta Hsu, and Lin-Yi Wu. 2005. One-pass GPRS and IMS authentication Procedure for UMTS. *IEEE Journal of Selected Areas in Communication* 23, 6 (2005), 1233–1239.

Alfredo Matos, Susana Sargento, and Rui Aguiar. 2007. Embedding identity in mobile environments. In *Proceedings of the MobiArch*. ACM, Kyoto, Japan, 1–8.

Yongsuk Park and Taejoon Park. 2007. A survey of security threats on 4G networks. In *Proceedings of the Globecom*. IEEE, Washington, USA, 1–7.

Chunyi Peng, Chi-Yu Li, Guan-Hua Tu, Songwu Lu, and Lixia Zhang. 2012a. Mobile data charging: new attacks and countermeasures. In *Proceedings of the CCS*. ACM, Releigh, USA, 195–204.

Chunyi Peng, Guan-Hua Tu, Chi-Yu Li, and Songwu Lu. 2012b. Can we pay for what we get in 3G data access?. In *Proceedings of the MobiCom*. ACM, Istanbul, Turkey, 113–124.

Thomas Peyrin, Yu Sasaki, and Lei Wang. 2012. Generic related-key attacks for HMAC. In *Proceedings of the Advances in Cryptology - Asiacrypt*. Springer, Beijing, China, 580–597.

Masoumeh Purkhiabani. 2012. Enhanced authentication & key agreement procedure of next generation evolved mobile n/w. *International Journal of Information & Electrical Engineering* 2, 1 (2012), 69–77.

Neetesh Saxena and Narendra S. Chaudhari. 2014. SecureSMS: A secure SMS protocol for VAS and other applications. *Journal of Systems and Software* 90, 1 (2014), 138–150.

Nabil Seddigh, B. Nandy, R. Makkar, and J. F. Beaumont. 2010. Security advances and challenges in 4G wireless networks. In *Proceedings of the Privacy Security and Trust*. IEEE, Ottawa, Canada, 62–71.

Chunya Tang, David A. Naumann, and Susanne Wetzel. 2003. Analysis of authentication and key establishment in inter-generational mobile telephony. *IACR Cryptology* 1, 1 (2003), 1–70.

Caimu Tang and Dapeng O. Wu. 2008. An efficient mobile authentication scheme for wireless networks. *IEEE Transactions on Wireless Communication* 7, 4 (2008), 1408–1416.

Cristina E. Vintila, Victor V. Patriciu, and Ion Bica. 2011. Security analysis of LTE access network. In *Proceedings of the Inter. Conf. on Networks*. ACREO, St Maarten, Netherlands Antilles, Sweden, 29–34.

Muxiang Zhang and Yuguang Fang. 2005. Security analysis & enhancements of 3GPP authentication and key agreement protocol. *IEEE Trans. on Wireless Communication* 4, 2 (2005), 734–742.

Yu Zheng, Xiaohu Tang, and Hongxia Wang. 2005. AKA and authorization scheme for 4G mobile networks based on trusted mobile platform. In *Proceedings of the ICS*. IEEE, Bangkok, Thailand, 976–980.