
End-to-End Mobile Communication Security Testbed Using Open Source Applications in Virtual Environment

K. Jijo George¹, A. Sivabalan¹, T. Prabhu¹ and Anand R. Prasad²

¹*NEC India Pvt. Ltd. Chennai, India*

²*NEC Corporation. Tokyo, Japan*

E-mail: {k.george; sivabalan.arumugam; prabhu.t}@necindia.in; anand@bq.jp.nec.com

Received 20 January 2015;

Accepted 01 May 2015

Abstract

In this paper we present an end-to-end mobile communication testbed that utilizes various open source projects. The testbed consists of Global System for Mobiles (GSM), General Packet Radio Service (GPRS) and System Architecture Evolution/Long Term Evolution (SAE/LTE) elements implemented on a virtual platform. Our goal is to utilize the testbed to perform security analysis. We used virtualization to get flexibility and scalability in implementation. So as to prove the usability of the testbed, we reported some of the test results in this paper. These tests are mainly related to security. The test results prove that the testbed functions properly.

Keywords: LTE, Amarisoft, Testbed, Security, OpenBTS, OpenBSC, OsmoSGSN, OpenGGSN, OpenIMS.

1 Introduction

Mobile network complexity has increased with time due to the coexistence of multiple technologies like GSM, GPRS, Universal Mobile Telecommunications System (UMTS) and SAE/LTE. Telecom service providers are trying to accommodate the existing customer service (GSM, i.e. voice only) and also

Journal of ICT, Vol. 3, 67–90.

doi: 10.13052/jicts2245-800X.314

© 2015 River Publishers. All rights reserved.

trying to provide cutting edge services such as Live TV, video conference etc. without discontinuing the old services. Coexistence of technologies provides more avenues to attack the network by making use of the weaknesses existing in the old generation network. Thus a detailed security analysis of the mobile network is necessary. For this purpose we have developed an end-to-end mobile communication security testbed using open source components on a virtualized platform.

There are a number of projects which have shown the feasibility of using open source components and low cost hardware platform to develop GSM testbed and SAE/LTE testbed. In one such project, an Open Base Transceiver Station (OpenBTS) ported on a common PC with Software Defined Radio (SDR) hardware was used to create a GSM network [1, 2]. Similar to OpenBTS implementation there are other projects which make use of open source for implementing a cellular network that can be operated at low cost, such as private networks in rural deployments, remote areas, or developing countries [3, 4]. During 2011, Anand *et al.* did OpenBTS experiments in the USA based laboratory environment. These experiments accelerated research on OpenBTS for rural deployment, in collaboration with LinkNet/UNZA [4]. In their paper, Anand *et al.* expand on the rationale for OpenBTS, and describe the technical performance that can be expected in a mixed traffic environment, using traffic patterns observed in Macha. Kretchmer *et al.* evaluated the Quality of Service (QoS) of OpenBTS mobile calls across a multi-hop wireless testbed that carries typical Internet traffic [5]. Similar to GSM experimentation there are few projects focused on SAE/LTE technology. One such project is the Amarisoft LTE 100 which is a low-cost SAE/LTE base station running on a Personal Computer(PC) [6], another project is the Open Source Long-Term Evolution Deployment (OSLD) which is a project aimed at design and development of a complete open source SAE/LTE stack [7].

Based on the above literature we see that all of the above projects were either for setting up GSM or SAE/LTE testbeds independently for performance analysis, application trial and QoS experiments. Not been much work has been done on developing an end-to-end mobile communication testbed which houses 2G, 2.5G and SAE/LTE network elements. Thus we have developed an end-to-end mobile communication testbed consisting of 2G, 2.5G and SAE/LTE mobile network elements using open source components on a virtual platform.

The organization of the paper is as follows. In Section 2, we give an overview of the standard 3rd Generation Partnership Project (3GPP) architecture and introduce the concept of virtualization. We present the role of

hypervisor in virtualization, the types of hypervisor and how it helps in improving the scalability of our testbed. In Section 3, we discuss the testbed architecture. We mapped the testbed to the standard 3GPP architecture as defined in TS 23.002 [8]. We also explain the open source projects and open components that have been integrated to set up the testbed architecture. In Section 4, we discuss sample threat scenarios and test cases together with the results. Conclusion and future work are discussed in Section 5.

2 Background

The 3GPP architecture has continuously evolved over the years with the changes in the technology. The motivation for the paper is to incorporate the different generations of mobile technology elements on a common testbed to study the security implications. In this section we discuss the key elements of 3GPP architecture and different types of virtual platforms that can be used to create a testbed. Use of virtualization allows us to run different implementation over the same platform without porting the code.

2.1 3GPP Standard Architecture

3GPP scope is to develop and maintain specifications on mobile communications system such as GSM, GPRS, Enhanced Data-rate for GSM Evolution (EDGE), IP Multimedia Subsystem (IMS) and SAE/LTE [9, 10]. Standard architecture of 3GPP is as depicted in the Figure 1. Broadly it constitutes of Core Network (CN) and Access Network (AN) Elements [8]. A brief description of each category is given below:

2.1.1 Core network

The CN elements can be seen as the basic platform for all communication related services provided to the user. The key functionality of the core network elements include switching of calls, routing of packet data, management of user data, authentication and other services. The CN is constituted of a Circuit Switched (CS) domain and a Packet Switched (PS) domain (which includes GPRS and Evolved Packet Core (EPC)). A “CS connection” is a connection for which dedicated network resources are allocated at the connection establishment and released at the connection release. A “PS connection” transports the user information using autonomous concatenation of bits called packets: each packet can be routed independently from the previous one.

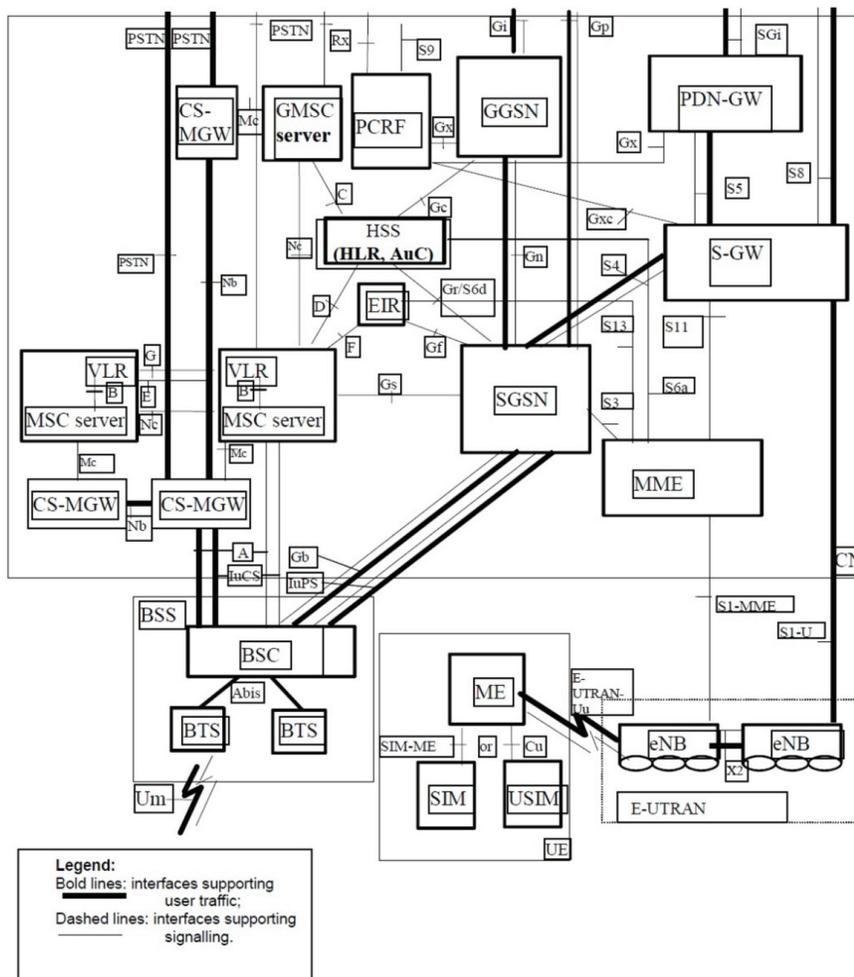


Figure 1 3GPP standard architecture [9].

CN elements common to all technologies GSM, GPRS and SAE/LTE [8]: Home Subscriber Server (HSS): The HSS is the master database which contains the subscription-related information. Function of HSS includes mobility management, call and/or session establishment support, user security information generation, access authorization, service authorization support, etc. There are two subsets of the HSS – Home Location Register (HLR) is a subset of the HSS that enables subscriber access to the CS and PS Domain services and to support roaming to legacy GSM/UMTS CS Domain networks.

Authentication Centre (AuC) is a subset of the HSS that stores an identity key for each mobile subscriber registered with the associated HLR. This key is used to generate security data for each mobile subscriber.

Equipment Identity Register (EIR): An EIR in the GSM system is the logical entity which is responsible for storing the International Mobile Equipment Identities (IMEIs), used in the GSM EDGE Radio Access Network (GERAN)/Universal Terrestrial Radio Access Network (UTRAN)/Evolved Universal Terrestrial Radio Access Network (E-UTRAN) system.

Element specific to PS and CS domains (GSM and GPRS): *Visitor Location Register (VLR)* is the element in a CN which stores the location information of a User Equipment (UE) when it moves out of its home location, the VLR and the HLR exchange information to allow the proper handling of CS calls involving the MS.

Element specific to CS domain (GSM): *Mobile-services Switching Centre (MSC)* is an exchange, which performs all the circuit switching services and signaling functions for mobile stations located in a geographical area designated as the MSC area. MSC acts as the interface between the radio system and the fixed networks.

Elements in PS domain (GPRS): *Gateway GPRS Support Node (GGSN)* is responsible for the inter-networking between the GPRS network and external packet switched networks, like the internet and X.25 networks. GGSN converts incoming data traffic from UE (via the SGSN) and forwards it to the relevant network, and vice versa.

Serving GPRS Support Node (SGSN) is responsible for the delivery of data packets from/to the mobile stations within its geographical service area. Functions of SGSN include packet routing, packet transfer, the mobility management (attach/detach and location management), logical link management, authentication and the charging functions. The location register of the SGSN stores location information (e.g., current cell, current VLR) and user profiles (e.g., International Mobile Subscriber Identity (IMSI), address(es) used in the packet data network) of all GPRS users registered with it; as defined in TS 23.016 [11] and TS 23.060 [12].

Elements in SAE/LTE: *Mobility Management Entity (MME)* – MME is the control plane entity within EPS supporting functions like Non-Access Stratum (NAS) signaling and security, inter CN node signaling for mobility between 3GPP access networks, PDN Gateway (PGW) and Serving Gateway (SGW) selection, roaming, authentication, bearer management functions, etc. MME is responsible for authenticating user towards the HSS. Its duties include authorization of UE to Public Land Mobile Network

(PLMN) and enforcing UE roaming restrictions if any. MME is also the termination point of ciphering and integrity protection for NAS signaling. Lawful Interception (LI) of signaling is also managed and supported by the MME.

Serving Gateway (SGW) – The SGW is the gateway which terminates the interface towards E-UTRAN for user plane. SGW is responsible for data transfer in terms of all packets across user plane. Its duties include taking care of mobility interface to other networks such as 2G/3G.

PDN Gateway (PGW) – The PDN GW is the gateway which terminates the SGI interface towards the PDN. PGW is responsible to act as an “anchor” for mobility between 3GPP and non-3GPP technologies. PGW acts as the point of entry/exit of traffic for the UE. The PGW manages policy enforcement, packet filtering for users, charging support and LI.

2.1.2 Access Network

The AN elements are the radio interface part of the architecture. Three different types of access network are used by the CN: the AN include GERAN (also called Base Station Subsystem (BSS)), UTRAN (also called Radio Network Controller (RNS)) and E-UTRAN. The MSC can connect to one of the following Access Network type or to both of them: BSS, RNS. The MME and SGW connect to the E-UTRAN.

Access Network Element for GSM we have *Base Station System (BSS)* which is the system of base station equipment (transceivers, controllers, etc.) that is responsible for communicating with UEs in a given area. It constitutes of a Base Station Controller (BSC) with the function to control one or more BTS and a Base Transceiver Station (BTS) is a network element which serves one cell.

Access Network elements for E-UTRAN(SAE/LTE): *E-UTRAN Node B (eNB)* is a logical network element which serves one or more.

E-UTRAN cells: It acts as the radio interface for the SAE/LTE network. An eNB hosts the following functions: Radio Resource Management, Dynamic allocation of resources to UEs in both uplink and downlink, IP header compression and encryption of user data stream, routing of User Plane data towards Serving Gateway. The Evolved UTRAN (E-UTRAN) consists of eNBs, providing the E-UTRA user plane (Packet Data Convergence Protocol (PDCP)/Radio Link Control (RLC)/Medium Access Control (MAC)/Physical Layer (PHY)) and control plane Radio Resource Control (RRC) terminations towards the UE. The eNBs can be interconnected with each other by means of the X2 interface.

2.2 Virtual Environments

Virtualization provides the ability to run multiple operating systems on a single physical system and share the underlying hardware resources. It brings significant cost of ownership and manageability benefits. The virtualization layer sits on top of a software or firmware called hypervisor acting as the intermediary between the physical hardware and the virtual machines (running guest OSs) as shown in Figure 2.

Benefits of virtualization includes: the isolation of virtual machines and the hardware-independence that results from the virtualization process, and also reduce the hardware cost, optimization of workloads, IT flexibility and responsiveness. Virtual machines are highly portable, and can be moved or copied to any industry-standard hardware platform, regardless of the make or model. Thus, virtualization facilitates adaptive IT resource management, and greater responsiveness to changing business conditions [13].

Virtualization, involves a shift in thinking from physical to logical, as it improves resource utilization by treating physical resources as pools from which virtual resources can be dynamically allocated. The following section talks about hypervisor and its role in implementing virtualization.

2.2.1 Hypervisor

Hypervisor is a software or firmware component that can help to virtualize the system resources [14]. The term hypervisor call refers to the

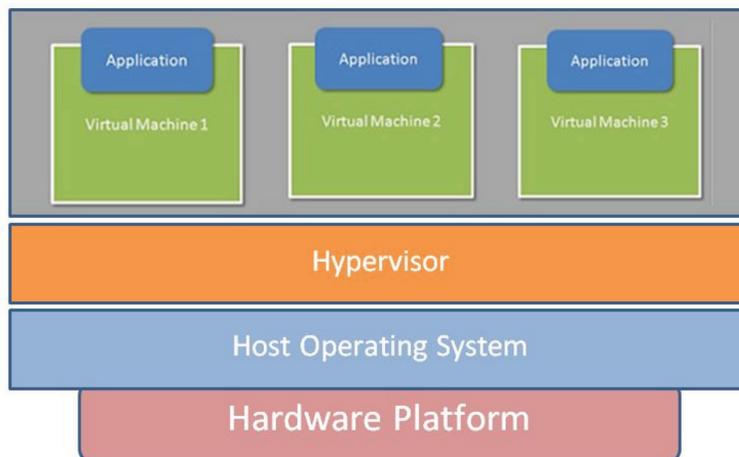


Figure 2 VM architecture.

para-virtualization interface, by which a guest operating system accesses services directly from the higher-level control program. Each operating system appears to have the hosts processor, memory, and resources to itself. In fact, the hypervisor is controlling the host processor and resources, distributing what is needed to each operating system in turn and ensuring that the guest operating systems (virtual machines) are unable to disrupt each other [15].

2.2.2 Hypervisor Classifications

Hypervisor can be classified into two types [13] as shown in Figure 3:

Type 1 hypervisor Bare Metal/Native Hypervisor: Software systems that run directly on the hosts software as a hardware control and guest operating system monitor. A guest operating system thus runs on another level above the hypervisor. This is the classic implementation of virtual machine architectures. Examples of Type 1 hypervisor include VMware ESXi, Citrix XenServer and Microsoft Hyper-V hypervisor.

Type 2 hypervisor Embedded/Host Hypervisor: Software applications that run within a conventional operating system environment. Considering the hypervisor layer being a distinct software layer, guest operating systems thus run at the third level above the hardware. As is done in the case of VMWare's Workstation, Oracle VM Virtualbox.

In this testbed, we used Type 2 hypervisor (i.e. Oracle Virtualbox) because the Oracle Virtualbox is a free cross-platform desktop virtualization tool having a host based hypervisor which runs on top of the host operating system (Fedora 17 and Ubuntu 12.04).

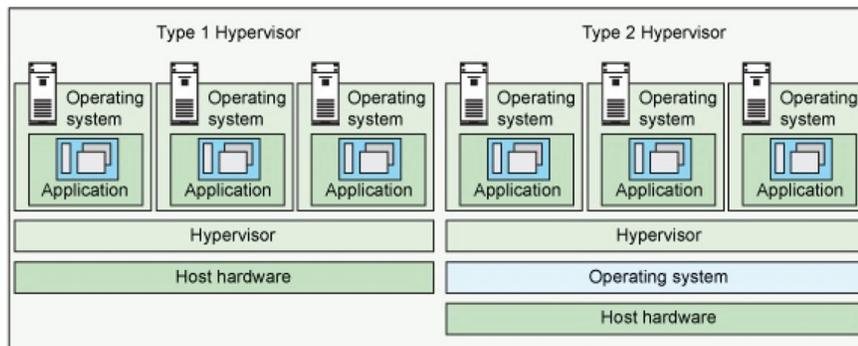


Figure 3 Types of hypervisor.

3 Testbed Architecture and Implementation

In this section we present the testbed architecture and the open source implementations used in the testbed. The details about all of the open source projects are discussed in the subsequent sections.

3.1 Testbed Architecture

Figure 4 shows the overall testbed architecture where we mapped the elements and interfaces of 3GPP standard network architecture, to create an end-to-end testbed in such a way that we can communicate between two different mobile communication technologies.

This testbed has four major group of network elements: i. GSM Network elements, ii. GPRS Network elements, iii. SAE/LTE Network elements, and iv. IMS Network elements.

The GSM network consists of two major network elements, they are: the AN consisting of BTS, and the CN consisting of HSS and MSC as defined in the TS 23.002 [8]. Our testbed architecture makes use of open source projects available for each category of GSM elements which is shown in the Figure 4. We use OpenBTS project which acts as the transceiver and the OpenBSC project which runs in the (Network in the Box) NITB mode it includes the functionality of BSC, MSC, HLR, AuC and HSS. In NITB mode only the Gb interface is exposed for external connection [16, 17].

The GPRS network has two network elements which are implemented in the testbed. In case of GPRS we identified OpenGGSN and OsmoSGSN

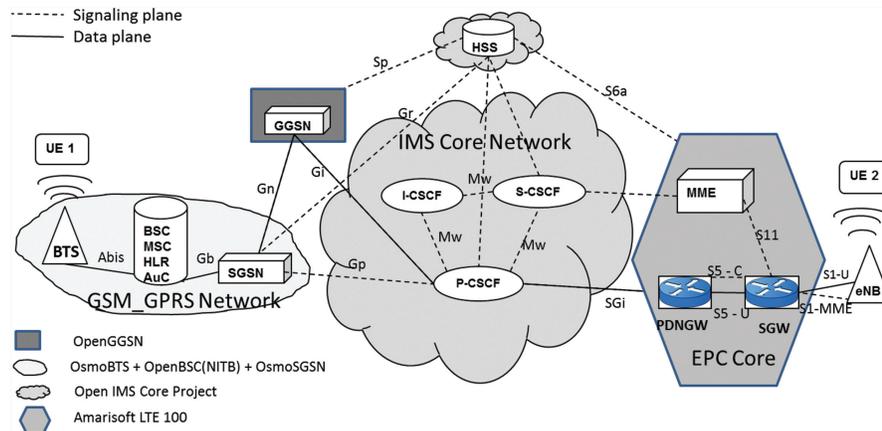


Figure 4 Testbed architecture.

as open source projects to implement the GPRS core network in our testbed. OsmoSGSN is a part of the OpenBSC project under Osmocom and implements an SGSN. OpenGGSN is an open source implementation of GGSN CN element. It connects to OsmoSGSN over the Gn interface and the Gi interface is connected to OpenIMS which acts as the PDN.

The System Architecture Evolution (SAE) is the core network of SAE/LTE. It is an evolved form of its legacy GPRS Core Network. There are no open source projects available for implementing SAE/LTE network elements so we used LTE Amarisoft 100 as the EPC in our testbed.

IMS was developed as an all-IP system designed to assist mobile operators deliver next generation interactive and interoperable services [18]. For our testbed we have used IMS as the core to interlink the two communicating networks. For this purpose we identified OpenIMS as an open source IMS project.

3.2 Open Source Projects and Tools

We have carried out detailed survey of open source projects related to mobile network elements and mapped the relevant projects to the core 3GPP architecture. The various open source and commercial components that we used to setup our testbed providing an end-to-end connectivity between two different mobile communication technologies such as OpenBSC, OsmoSGSN, OpenGGSN, OpenIMS, Amarisoft LTE are given below [16–30].

3.2.1 OpenBSC

OpenBSC is an open source implementation of the BSC features of a GSM network it also includes support for mobility management and authentication and intra-BSC handover, SMS and voice calls. GPRS and EDGE support are possible if combined with OsmoSGSN and OpenGGSN as has been shown in the testbed. It can work as a pure BSC or as a full network in a box.

OpenBSC-NITB mode includes functionality normally performed by the following elements of a GSM network: BSC, MSC, HLR, AuC, VLR, EIR. NITB mode of OpenBSC also implements the A-bis protocol as defined in the GSM TS 08.5× [19–22] and TS 12.21 [23] for communicating with the BTS. It implements a minimal subset of the BSC, MSC and HLR. It does not implement any of the interfaces (like the A and B interfaces) between the higher order GSM network elements.

3.2.2 OsmoSGSN

OsmoSGSN is an open source implementation of the SGSN [24]. It implements the GPRS Mobility Management (GMM) and SM (Session Management). The SGSN connects via the Gb-Interface to the BSS and it connects via the Gn to OpenGGSN. Presently no authentication is done, i.e. the GPRS network will simply allow every IMSI to attach to it as far as it has the same MCC/MNC as the network.

3.2.3 OpenGGSN

OpenGGSN is a Gateway GPRS Support Node (GGSN). The GGSN is a small application which is provided in order to test and demonstrate the use of gtplib. It is fully compliant to the 3GPP standards, but lacks functionality such as charging and management [25]. The project also developed an SGSN emulator suitable GPRS core network testing. OpenGGSN was developed and tested using Redhat 8.0 and 9.0.

3.2.4 Amarisoft LTE

Amarisoft LTE 100 is a LTE Base Station running on a PC, with the configuration mentioned in Section 3.3.2 on the SAE/LTE PC Hardware Requirements. LTEENB, a module of Amarisoft, allows building a real SAE/LTE base station using a standard PC and a low cost software radio front-end. All the physical layer and protocol layer processing is done in real time inside the PC, so no dedicated SAE/LTE hardware is necessary. It can be used to set up an entire SAE/LTE open network.

LTE-MME is a MME implementation [26]. It has a built-in SGW, PGW and HSS. It can be used with the Amarisoft LTE eNodeB.

LTE-ENB is a LTE base station (eNodeB) implemented entirely in software and running on a PC [27]. The PC generates a baseband signal which is sent to a radio front end doing the digital to analog conversion. The reverse is done for the reception. LTE-ENB interfaces with a SAE/LTE Core Network through the standard S1 interface. In particular, the Amarisoft Core Network software (LTEMME) can easily be connected to it to build a highly configurable LTE test network. Some of the implemented features are as follows: implements SAE/LTE release 8 with Frequency Division Duplexing (FDD) configuration, bandwidth ranges from 1.4 to 20 MHz, runs in real time on a standard PC using Linux, Core Network emulation is implemented so that no LTE network infrastructure is needed to use the base station, IP traffic is redirected to a Linux virtual network interface, supports test USIM cards using the standard XOR authentication algorithm, flexible configuration system to

support various SAE/LTE parameters. It implements the LTE PHY, MAC, RLC, PDCP, RRC and NAS layers.

3.2.5 OpenIMS

The IMS is an architectural framework for delivering IP multimedia services [28]. The Open Source IMS Core project was developed by the Fraunhofer Institute FOKUS. Its purpose is to provide an IMS core reference implementation for IMS technology testing, IMS application development and prototyping. It's not meant for commercial product development activities. The Open Source IMS Core consists of Call Session Control Functions (CSCFs), the central routing elements for any IMS signaling, and a Home Subscriber Server (HSS) to manage user profiles and associated routing rules [29]. The central components of the Open Source IMS Core project are the Open IMS CSCFs (Proxy, Interrogating, and Serving) which were developed at FOKUS as extensions to the SIP Express Router (SER). As a basic implementation for HSS signaling compatible with SER there is the FOKUS Home Subscriber Server (FHoSS) which is also part of the Open Source IMS Core project [30].

3.3 Testbed Implementation

Figure 5 shows the implementation view of the end-to-end testbed discussed in the previous section. This implementation mainly consists of two generations of mobile network elements i.e. 2G/2.5G and SAE/LTE. We used mainly open source projects, operating systems, open source hypervisor and low cost open source hardware platforms.

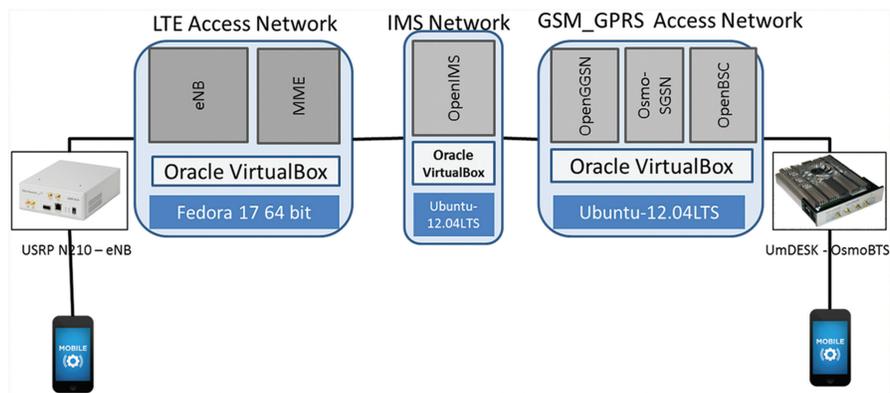


Figure 5 End-to-end testbed architecture.

The GSM-GPRS implementation involves a small base transceiver station utilizing OpenBTS, OpenBSC, OpenGGSN and OsmoSGSN software [31]. The Linux- based software application configures the UmTRAY to present a GSM air interface to standard GSM phones and also interfaces with the OpenIMS on the back-end acting as PDN.

For the SAE/LTE implementation we configured the Universal Software Radio Peripheral (USRP) hardware using Linux based USRP Hardware Drivers (UHD). We ran the LTE-MME and LTE-ENB on the PC and the radio signal trans-reception carried out on the USRP. Similar to GSM-GPRS we connected the SAE/LTE core network to the same OpenIMS network thus enabling the end-to-end connectivity between GSM-GPRS and SAE/LTE network.

We do not have the spectrum license to perform experiments in the licensed spectrum, so we directly connected the UE to the BS hardware via RF cables, which ensured that there is no RF radiation due to this experiment. The following section explains in detail about the hardware and software used in this implementation.

3.3.1 GSM/GPRS Testbed

We set up the OpenBSC-GPRS testbed using a number of open source hardware and software elements, as given below:

Hardware

- UmTRAY as an SDR to act as GSM BTS.
- A Fast PC: Ubuntu 12.04 LTS OS, 1 Gigabit Ethernet ports, 2 GB of RAM, 1 GB of hard disk space.

Software

- Oracle Virtualbox 4.3.12 as hypervisor.
- Open source projects on Base Station Controller (OpenBSC).
- Open Source BTS software OsmoBTS.
- Open source transceiver OsmoTRX for GSM Layer 1 implementation.
- Open source packet control unit (OsmoPCU) RLC and MAC layers of the GPRS Um (radio) interface on the MS-facing side, as well as the Gb Interface (NS, BSSGP) on the SGSN-side.
- OpenGGSN an Open Source GGSN Implementation.
- Off the shelf mobile phones with GSM SIM cards as client devices.

3.3.2 LTE Testbed

We set up the SAE/LTE Access network using the Amarisoft LTE 100, other hardware and software, as mentioned below:

Hardware

- A Fast PC
 - A quad core Intel Core i7 CPU (Nehalem or later).
 - 2 Gigabit Ethernet ports.
 - 2 GB of RAM.
 - 1 GB of hard disk space.
- Radio front end.
 - USRP N200 or N210 from Ettus Research with the SBX daughter-board[32].
 - Antennas for the intended LTE frequencies or cables and attenuators to connect to a UE.
- LTE UE compatible with LTE release 8 FDD.
- Test USIM cards.

Software

- A 64 bit Linux Fedora 17 [33].
- Oracle Virtualbox 4.3.12 as hypervisor.
- UHD drivers ($\times 86$ 64 target) from Ettus Research [28].
- The Amarisoft LTE Core Network.

4 Sample Tests and Results

To showcase the usability of this testbed (i.e. security testing), in this section we discuss sample threat scenarios and test cases together with the results. The test cases, given in following sub-sections, are only examples to prove the usability of the testbed, other test cases can be tried as well. The definitions of some of the terminology used in the test cases are given below.

Theft of service: In this form of attack a malicious user gains access to a legitimate user's services. The attacker can then use these services while the user pays for them without realizing it being misused.

Information Disclosure Vulnerability: Information Disclosure is a loophole in a network where the information may be leaked outside the network. This can lead to crucial information being made available to an attacker.

Integrity Check: Integrity is one of the basic principles of security, which deals with maintaining the accuracy and consistency of the data being communicated. The loss of integrity can be caused by someone gaining access to the information in transit wherein he can modify the data and forward it without the end users realizing it.

4.1 Test Case I – Unauthorized Access and Theft of Service

- *Scenario:* The UE is assigned an IP address dynamically by the MME from an IP address range when setting up the Packet Data Protocol (PDP) context. The UE then accesses the services using the given IP address. In this test case we manually assigned an IP in the same subnet to our network interface and used it to communicate with the PGW.
- *Expected Result:* The UE must not be able to modify the assigned IP address, as this may lead to users being able to steal other users' services theft of service and/or unauthorized access. The UE should only be able to communicate via its assigned IP address.
- *Observed Result:* In the given test case we statically assigned an IP in the valid range to the UE and use it for communicating with the MME as shown in Figure 6 and 7. We saw that the IP address is assigned as a virtual IP in Windows and pinging via the static IP to the core network was unsuccessful. Here 192.168.3.2 was our assigned IP while 192.168.3.20 was the static IP allotted by us. We observed that communication with the core network was only possible when using the network

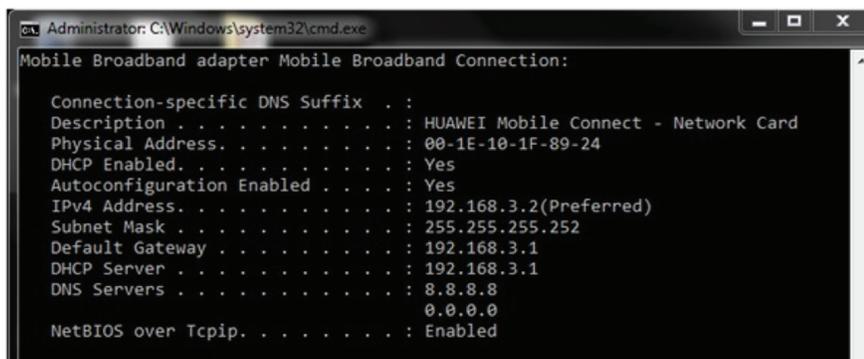
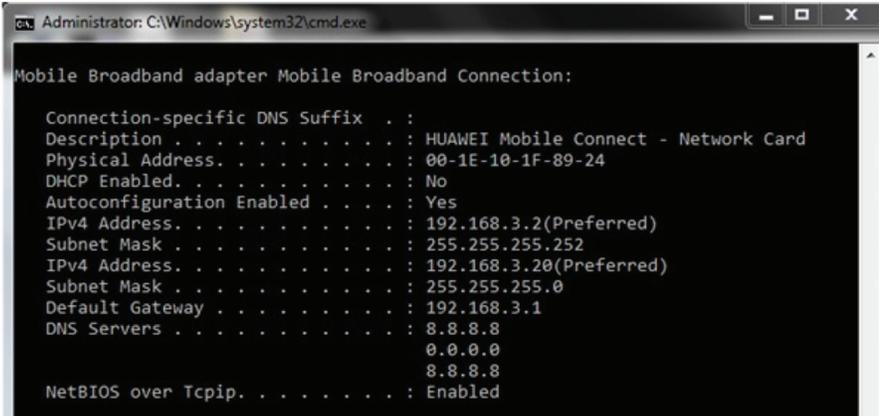


Figure 6 The default IP configuration.



```

Administrator: C:\Windows\system32\cmd.exe
Mobile Broadband adapter Mobile Broadband Connection:

Connection-specific DNS Suffix . :
Description . . . . . : HUAWEI Mobile Connect - Network Card
Physical Address. . . . . : 00-1E-10-1F-89-24
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
IPv4 Address. . . . . : 192.168.3.2(Preferred)
Subnet Mask . . . . . : 255.255.255.252
IPv4 Address. . . . . : 192.168.3.20(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.3.1
DNS Servers . . . . . : 8.8.8.8
                        0.0.0.0
                        8.8.8.8
NetBIOS over Tcpip. . . . . : Enabled

```

Figure 7 The manual IP assignment test case.

assigned IP i.e. 192.168.3.2, thus proving that the implementation is secure.

4.2 Test Case II – Information Disclosure Vulnerability

- *Scenario:* Scanning the IP addresses using any network sniffer we can sniff for the communication between the UE and the PGW. As a simple test on information disclosure vulnerability we check whether network information, in form of IP address, is sent to the UE.
- *Expected Result:* In a secure mobile communication system there should be no leakage of information on network topology. The communications between the UE and PGW should not contain any such protected information about the core network.
- *Observed Result:* We captured the packets in the UE interface using wireshark as shown in Figure 8. Based on the observation of the captured pcaps file; data from the protocol hierarchy, contents summary and endpoints we found information about one of the network IP interfaces i.e. 192.168.3.3 apart from our tunnel IP range. As we already know that our IP address is 192.168.3.2 from the previous test case, the detected IP address is an interface to the core network. This core network IP should not have been visible to the UE and leads to Information Disclosure vulnerability.

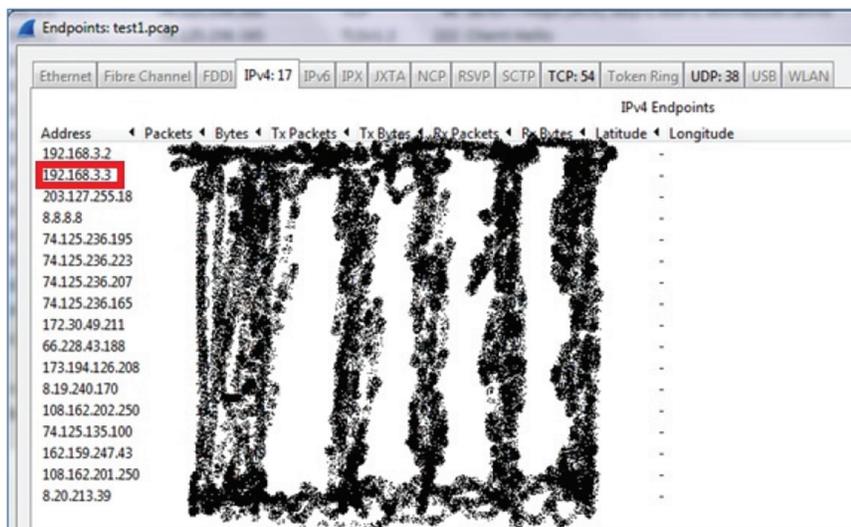


Figure 8 The core network IP disclosure.

4.3 Test Case III – Integrity Check

- Scenario:* To ensure integrity, the receiver verifies that the received NAS message is exactly the message that the transmitter sent. This is done using an integrity value which is derived using the KNASint Key. MME initiates the NAS security procedure by sending the message which includes encryption and integrity protection algorithms. Key selection identifier (KSI-ASME) is also included in the message. UE responds back to the MME with a message which is ciphered and integrity protected.
- Expected Result:* The SAE/LTE architecture mandates that the NAS signaling between the UE and MME be integrity protected. Apart from the first communication of the UE, the IMSI is not sent over the network only the Globally Unique Temporary Identifier (GUTI) is used for communicating, and setting up of the encryption and integrity keys between the UE and the MME.
- Observed Result:* In our testbed on scanning the MME and ENB logs we were able to monitor the NAS signaling data gaining information about the transmitted attach request for setting up the initial connection request.

The NAS message as given in Figure 9 is integrity protected to ensure the origin of the sender but not ciphered to protect the communication. It uses the GUTI to set up the attach process with the MME which can be seen in the NAS messages.

The Figure 10 shows NAS signaling Attach accept setup once the integrity and ciphering has been setup. It shows a secure communication being setup with integrity protection and ciphering to protect the signaling information exchange. The NAS message shows the signals sent when setting up user context and exchanging APN info and IP info with the new user.

```

4408 12:17:39.913 [NAS] UL 01 0057 EMM: Attach request
4409
4410     Protocol discriminator = 0x7 (EPS Mobility Management)
4411     Security header = 0x1 (Integrity protected)
4412     Auth code = 0xb71dfb3b
4413     Sequence number = 0x03
4414     Protocol discriminator = 0x7 (EPS Mobility Management)
4415     Security header = 0x0 (Plain NAS message, not security protected)
4416     Message type = 0x41 (Attach request)
4417     EPS attach type = 2 (combined EPS/IMSI attach)
4418     NAS key set identifier:
4419         TSC = 0
4420         NAS key set identifier = 0
4421     Old GUTI or IMSI:
4422         MCC = 001
4423         MNC = 01
4424         MME Group ID = 32769
4425         MME Code = 1
4426         M-TMSI = 0x00000002
4427     UE network capability:
4428         Length = 4
4429         Data = e0 e0 00 00
4430     ESM message container:
4431         Protocol discriminator = 0x2 (EPS Session Management)
4432         EPS bearer identity = 0
4433         Procedure transaction identity = 1
4434         Message type = 0xd0 (PDN connectivity request)
4435         Request type = 1
4436         PDN type = 1 (IPv4)
4437         Protocol configuration options:
4438             ext = 1
4439             configuration protocol = 0
4440             Protocol ID = 0x8021 (IPCP)
4441             Data = 01 00 00 10 81 06 00 00 00 00 83 06 00 00 00 00
4442             Protocol ID = 0x000a (IP address allocation via NAS signalling)
4443             Data =
4444         Last visited registered TAI:
4445             MCC = 001
4446             MNC = 01
4447             TAC = 0x0001
4448         MS network capability:
4449             Length = 3
4450             Data = e5 c0 04
4451         TMSI status = 0
4452         Mobile station classmark 2:
4453             Length = 3
4454             Data = 4f 18 a6
4455         Remaining bytes:
4456             f1

```

Figure 9 Unencrypted attach request using GUTI.

```

332 14:47:17.174 [NAS] DL 0064 EMM: Attach accept
333
334 Protocol discriminator = 0x7 (EPS Mobility Management)
335 Security header = 0x2 (Integrity protected and ciphered)
336 Auth code = 0x2164dc69
337 Sequence number = 0x01
338 Protocol discriminator = 0x7 (EPS Mobility Management)
339 Security header = 0x0 (Plain NAS message, not security protected)
340 Message type = 0x42 (Attach accept)
341 EPS attach result = 2 (combined EPS/IMSI attach)
342 T3412 value:
343     value = 0
344     unit = 7 (deactivated)
345 TAI list:
346     Length = 6
347     Data = 00 00 f1 10 00 01
348 ESM message container:
349     Protocol discriminator = 0x2 (EPS Session Management)
350     EPS bearer identity = 5
351     Procedure transaction identity = 1
352     Message type = 0xc1 (Activate default EPS bearer context request)
353     EPS Qos:
354         Length = 1
355         Data = 09
356     Access point name = "test123"
357     PDN address:
358         PDN type = 1 (IPv4)
359         IPv4 = 192.168.3.2
360 Protocol configuration options:
361     ext = 1
362     configuration protocol = 0
363     Protocol ID = 0x8021 (IPCP)
364     Data = 03 00 00 0a 81 06 08 08 08 08
365 GUTI:
366     MCC = 001
367     MNC = 01
368     MME Group ID = 32769
369     MME Code = 1
370     M-TMSI = 0x00000002
371 Location area identification:
372     MCC = 001
373     MNC = 01
374     LAC = 0x0001
375 MS identity:
376     Length = 5
377     Data = 04 00 00 00 02

```

Figure 10 Integrity protected and ciphered attach accept containing the APN and assigned IP.

5 Conclusion and Future Work

5.1 Conclusion

This paper demonstrates creation of an end-to-end testbed for GSM, GPRS and SAE/LTE using open source projects. We implemented the various network elements in a virtual platform to make it scalable and flexible to deploy. In this paper we demonstrated the usability of the testbed based on a few test cases. Some of the sample categories of test cases that have been performed include: theft of service, information disclosure vulnerability and integrity check. Finally, the observed results were compared with the expected results to ascertain any defects in the system. The utilization of this testbed is not

only limited to perform security tests, but can also be used for various other studies as well.

5.2 Future Work

Moving forward we plan on taking our whole implementation to a more open architecture implementation using Network Function Virtualization (NFV). As a first step, the virtualization aspects of Linux platform will be implemented and managed using OpenStack. OpenStack is an open source implementation that is used to control large pools of processing, networking and storage resources [34]. The whole system can be managed and resources can be assigned by an administrative dashboard and web interface. This will facilitate us to easily build and manage mobile network elements on a flexible virtual platform.

Further, we plan to implement project Clearwater. This project is specially designed to be scalable over the cloud network to provide IMS solutions to the users. And it does so by using the concept of NFV as it has been built from the ground up to run in a virtualized environment and take full advantage of the flexibility of the cloud [35].

References

- [1] Burgess, D. A, Samra H. The Open BTS Project an opensource GSM base station, Sept 2008.
- [2] OpenBTS: <http://openbts.org/>
- [3] Heimerl, K., Brewer, E. The village base station. “In Proceedings of the 4th ACM Workshop on Networked Systems for Developing Regions Systems.” San Francisco (CA), 2010, pp. 5–6.
- [4] Anand A., Johnson, D. L., Belding, E. M. “Village Cell: cost effective cellular connectivity in rural areas. In Proceedings of the International Conference on Information and Communication Technologies and Development”. Atlanta (GA), 2012, pp. 180–189.
- [5] Mathias Kretschmer, Peter Hasse, Christian Niephaus, Thorsten Horstmann, and Karl Jonas. Connecting Mobile Phones via Carrier-Grade Meshed Wireless Back-Haul Networks. E-Infrastructures and E-Services on Developing Countries. Africomm 2010, 2010.
- [6] Amarisoft. Amari LTE 100, Software LTE base station on PC. Available at: <http://www.amarisoft.com/>
- [7] OSLD Project. Open Source Long-Term Evolution (LTE) deployment. Available at: <https://sites.google.com/site/osldproject/>

- [8] 3GPP TS 23.002 – Network Architecture.
- [9] Gottfried Punz, *Evolution of 3G Networks: The Concept, Architecture and Realization of Mobile Networks Beyond UMTS*, Springer Wien-New York, 2010.
- [10] Heikkei Kaarannen, *UMTS Networks: Architecture, Mobility and Services*, John Wiley and Sons, 2005.
- [11] 3GPP TS 23.016: “Subscriber data management; Stage 2”.
- [12] 3GPP TS 23.060: “General Packet Radio Service (GPRS); Service description; Stage 2”.
- [13] Virtualization – <http://www.ibm.com/developerworks/cloud/library/cl-hypervisorcompare/>
- [14] Hypervisor – <http://www.tricerat.com/resources/topics-library/hypervisor-virtualization-software>
- [15] The term Hypervisor – Gerald J. Popek and Robert P. Goldberg (1974). “Formal Requirements for Virtualizable Third Generation Architectures”. *Communications of the ACM* 17
- [16] OpenBTS: <http://wush.net/trac/rangepublic>
- [17] OpenBSC Network from scratch: [http://openbsc.osmocom.org/trac/wiki/network from scratch](http://openbsc.osmocom.org/trac/wiki/network%20from%20scratch)
- [18] 3GPP-IMS: <http://www.3gpp.org/technologies/keywords-acronyms/109-ims>
- [19] TS GSM 08.52 BSC-BTS Interface Principles.
- [20] TS GSM 08.54 BSC-BTS Layer 1 Specification.
- [21] TS GSM 08.56 BSC-BTS Layer 2 Specification.
- [22] TS GSM 08.58 BSC-BTS Layer 3 Specification.
- [23] TS GSM 12.21 BSC-BTS Operation/Maintenance Signalling.
- [24] Osmo-SGSN_OpenBSC: <http://openbsc.osmocom.org/trac/wiki/osmosgsn>
- [25] OpenGGSN Readme – <http://cgit.osmocom.org/openggsn/tree/README>
- [26] LTE – MME Document by Amarisoft.
- [27] LTE – ENB Document by Amarisoft.
- [28] Documentation-OpenIMS.org: http://www.fokus.fraunhofer.de/en/fokus_testbeds/open_ims_playground/components/osims/index.html
- [29] Mohammad Ilyas, Syed A. Ahson, *IP Multimedia Subsystem (IMS) Handbook*, CRC Press, 2009.
- [30] Dragos Vingarzan, Peter Weik and Thomas Magedanz, *Design and Implementation of an Open IMS Core*, Springer Berlin Heidelberg, 2005.
- [31] OpenBSC-GPRS Implementation: <http://openbsc.osmocom.org/trac/wiki/OpenBSCGPRS>

[32] USRP N210 – <https://www.ettus.com/product/details/UN210-KIT>

[33] Fedora Project Downloads-<http://fedoraproject.org/en/get-fedora-all>

[34] OpenStack Open Source Cloud Computing Software: <https://www.openstack.org/software>

[35] Project Clearwater: <http://www.projectclearwater.org/about-clearwater/>

Biographies



K. J. George received B.Tech in Computer Science and Engineering from Kurukshetra Institute of Technology and Management, India in 2011. He has 2 years of experience in research and development of mobile communication networks. At present he works as Member Technical Staff in NEC India Standardization (NIS) Team at NEC Mobile Network Excellence Center (NMEC), NEC India Pvt Ltd, Chennai. Prior to joining NECI he was associated with IIIT, Bangalore as Research Associate. In his current role, he is working on security aspects of telecom networks and testbed development of next generation mobile networks. His research interest includes Penetration Testing, Network Security and Telecom Security.



S. Arumugam received Ph.D in Electrical Engineering from Indian Institute of Technology Kanpur, India in 2008 and M.Tech degree from

Pondicherry University, India, in 2000. He has 14 years of experience in Academic teaching and Research. Presently he works as Manager for Research at NEC Mobile Network Excellence Center (NMEC), NEC India Pvt Ltd, Chennai. Prior joining NECI he was associated with ABB Global Services and Industries Limited, Bangalore as Associate Scientist. He has published more than 25 papers in various International Journals and Conferences and also participated in many National and International Conferences. In his current role, he is representing NEC for Global ICT Standards forum of India (GISFI). His research interest includes Next Generation Wireless Networks.



P. Thiruvassagam received Master of Design in Communication Systems from Indian Institute of Information Technology Design and Manufacturing-Kancheepuram, Chennai, India in 2014. He has 2 years of experience in academic teaching. At present he works as Research Engineer in NEC India Standardization (NIS) Team at NEC Mobile Network Excellence Center (NMEC), NEC India Pvt Ltd, Chennai. His research interest includes Information Security, Wireless Telecom Security and Next Generation Wireless Networks.



A. R. Prasad, Dr. & ir., Delft University of Technology, The Netherlands, is Chief Advanced Technologist, Executive Specialist, at NEC Corporation,

Japan, where he leads the mobile communications security activity. Anand is the chairman of 3GPP SA3 (mobile communications security standardization group), a member of the governing body of Global ICT Standardisation Forum for India (GISFI), founder chairman of the Security & Privacy working group and a governing council member of Telecom Standards Development Society, India. He was chairman of the Green ICT working group of GISFI. Before joining NEC, Anand led the network security team in DoCoMo Euro-Labs, Munich, Germany, as a manager. He started his career at Uniden Corporation, Tokyo, Japan, as a researcher developing embedded solutions, such as medium access control (MAC) and automatic repeat request (ARQ) schemes for wireless local area network (WLAN) product, and as project leader of the software modem team. Subsequently, he was a systems architect (as distinguished member of technical staff) for IEEE 802.11 based WLANs (WaveLAN and ORiNOCO) in Lucent Technologies, Nieuwegein, The Netherlands, during which period he was also a voting member of IEEE 802.11. After Lucent, Anand joined Genista Corporation, Tokyo, Japan, as a technical director with focus on perceptual QoS. Anand has provided business and technical consultancy to start-ups, started an offshore development center based on his concept of cost effective outsourcing models and is involved in business development.

Anand has applied for over 50 patents, has published 6 books and authored over 50 peer reviewed papers in international journals and conferences. His latest book is on “Security in Next Generation Mobile Networks: SAE/LTE and WiMAX”, published by River Publishers, August 2011. He is a series editor for standardization book series and editor-in-chief of the Journal of ICT Standardisation published by River Publishers, an Associate Editor of IEEK (Institute of Electronics Engineers of Korea) Transactions on Smart Processing & Computing (SPC), advisor to Journal of Cyber Security and Mobility, and chair/committee member of several international activities.

He is a recipient of the 2014 ITU-AJ “Encouragement Award: ICT Accomplishment Field” and the 2012 (ISC)² Asia Pacific Information Security Leadership Achievements (ISLA) Award as a Senior Information Security Professional. Anand is Certified Information Systems Security Professional (CISSP), Fellow IETE and Senior Member IEEE and a NEC Certified Professional (NCP).