Evaluating FTTT Protocol via PRISM, PRISM-symm and GRIP

Sania Bhatti, Mohsin Memon, and Sheeraz Memon

***Abstract*—Probabilistic model checking is used for verifying stochastic behaviors of systems. Model checking plays imperative role for scrutinizing of novel protocols in almost every field including computer communications, networks, security, and biology prior to extensive simulations. In this paper we present an overview of few diverse case studies which are implemented using one of the leading probabilistic symbolic models checking tool, called Probabilistic Symbolic Model (PRISM). The successful applicability of PRISM in wider range of application domains motivated us to present a comparison of those applications in a precise manner. We also presented an enhanced version of Continuous-time Markov Chain (CTMC) model for evaluation of Fault Tolerant Target Tracking (FTTT) protocol implemented in wireless sensor networks. PRISM is preferred for the probabilistic modeling of FTTT protocol to aid symmetry reduction during modeling. We proceeded with probabilistic results which pinpointed the comparison of FTTT performance via PRISM, PRISM-symm and generic representatives in PRISM (GRIP) models. Modeling experiments confirmed that PRISM-symm is giving improved outcomes in comparison with PRISM and GRIP.**

***Index Terms*—CTMC model, FTTT protocol, model checking, PRISM, PRISM-symm, GRIP.**

I. Introduction

Probabilistic model checking is one of the most popular verification methods useful to accomplish accuracy evaluation which is otherwise intricate and time consuming to attain via simulations now a days. This type of model checking is used for formal verification related to the quantitative properties of stochastic behaviors of systems. Probabilistic model checking is related to the construction and scrutiny of a probabilistic model from a high-level description of a system's behavior. Model analysis is performed by specifying one or more quantitative properties which are stated in temporal logic. These properties help to measure correctness of the system along with reliability and performance [1]-[5].

Although a number of probabilistic and hybrid model checking tools are being successfully employed by researchers [6], such as PRISM [1], CaVI [7], Anquiro [8], APMC [9], Real-Time Maude [10], UPPAAL [11], AVISPA [12], HyTech [13] and Slede [14], however, this paper focuses on applications of PRISM tool only because of its excess deployment in diverse domains. In this paper, an overview of varying protocols verified via PRISM is presented. Section II discusses about PRISM model checker and the models it supports. Section III presents eleven different communication protocols which have been verified via PRISM. Section IV specifies a case study of FTTT protocol. Finally, Section V states concluding remarks.

Manuscript received August 20, 2016; revised December 6, 2016.

S. Bhatti and M. Memon are with the Department of Software Engineering, Mehran University of Engineering and Technology, Jamshoro, Sindh, Pakistan (e-mail: sania.bhatti@faculty.muet.edu.pk, mohsin.memon@faculty.muet.edu.pk).

S. Memon is with the Computer Systems Engineering Department, Mehran University of Engineering and Technology, Jamshoro, Sindh, Pakistan (e-mail: mesheeraz@hotmail.com).

II. PRISM Model Checker

PRISM is an open source probabilistic model checker and available to run on the majority of operating systems. The website of PRISM [1] developed at University of Oxford offers its download, tutorials, a huge repository of publications and case studies. It supports Discrete Time Markov Chain (DTMC), Markov Decision Process (MDP), Continuous Time Markov Chain (CTMC) and probabilistic timed automata. The system models are stated in a high level language via guarded command mutation which is based on reactive modules formalism. A set of modules representing the components of a system constitutes a PRISM model and each module comprises of states and transitions. States represent the possible configuration of the system and transitions correspond to change of states relating to time. PRISM performs model checking by expressing properties in PCTL, CSL and LTL. To specify additional quantitative measures, costs and rewards are added to the properties. Fig. 1 illustrates the two main parts of a system specification in PRISM: the model and the properties. PRISM provides results in either log form or in the form of a graph [2], [3]. PRISM has capability to work out quantitative measures corresponding to model behaviour such as expected time for transmission, expected total transmission energy. To specify these measures, it makes use of rewards, which are actually real values linked with states or transitions [15].

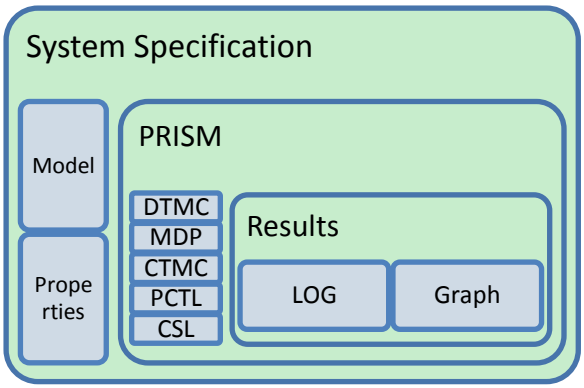

Fig. 1. PRISM overview.

The interface provided by PRISM exhibits model editor,

DOI: 10.7763/IJCTE.2017.V9.1131

an editor for properties, a simulator in support of graph plotting and model debugging. In PRISM, a storage efficient data structure is used to represent vectors and matrixes called multi-terminal binary decision diagram (MTBDD). The advantage of using MTBDD is rapid decline in storage space. PRISM supports three diverse computation engines. The first symbolic engine uses only MTBDD and is useful for regular models with maximum 10E+10 states. Second engine is based on sparse matrices, arrays and due to its predicable performance is the fastest engine. Finally the hybrid engine uses the extensions of MTBDD and is default engine in PRISM. This engine gives faster results than symbolic engine but utilizes less memory than sparse matrices and arrays [2], [3].

III. Models Checker via PRISM

Probabilistic model checking via PRISM has been effectively applied to scrutinize the performance of spacious range of systems, including gossip protocols [4], Byzantine agreement protocol [5], Fibroblast Growth Factor signaling pathway [5], Peer-to-peer protocol based on BitTorrent [5], IEEE 802.3 CSMA/CD [16], Team formation protocols [17], Power efficient algorithm [18], Robot swarm behavior [19], FTTT [20].

In this section we are presenting eleven different protocols from diverse domains which are analyzed and verified via PRISM. The key purpose of this discussion is to demonstrate that how PRISM has been widely applied in various application domains. In addition, the modeling parameters are summarized in table form.

Gossip protocols are type of communication protocols in which each node of the system periodically transmit information to large number of its peer nodes [4]. The behavior of gossip protocols is both probabilistic because in distributed network activities of individual node take place asynchronously and nondeterministic due to random selection of nodes for information exchange. The gossip protocol is modeled via MDP and best case and worst case performance of the protocol is investigated.

Symmetry reduction is a technique which is based on language-level translation of symmetric models to a compact form. A Symmetric Probabilistic Specification Language (SPSL) and an algorithm for translation of SPSL specification into corresponding generic form are presented in [5]. In addition a comparative study based on six case studies is presented. The case studies considered are Aspnes & Herlihy's randomised consensus protocol; a randomised Byzantine agreement protocol; the randomised mutual exclusion protocols of Rabin; a simplified model of the Fibroblast Growth Factor signaling pathway; a peer-to-peer protocol based on BitTorrent; and Dolev et al.'s minimum space shared memory leader election protocol. Experiments are performed using PRISM, PRISM symmetry reduction tool; PRISM-symm and Generic Representative tool; GRIP. It has been identified that symmetry reduction technique improves the performance of probabilistic model checking by allowing verification of models producing large MTBDD space.

Analysis and verification of team formation protocols performed via PRISM is presented in [17]. Both DTMC and MDP models have been developed consisting of five agents with four different network topologies: fully connected, ring, star and a network having one isolated agent. In addition stochastic two-player game (STPG) is used to verify properties of multi agent system on different network topologies in which agents split into cooperative and hostile classes. The expected performance of the organization and organization-optimal resource allocation among agents are computed.

Power Efficient Algorithm for Data Gathering (PEADG) is proposed and evaluated via PRISM in [18]. The intention of PEADG is to prolong the lifetime of data gathering wireless sensor networks. To avoid state space explosion problem, Probabilistic Automata (PAs) are created first which are mapped to DTMC module. The original approach has multiple modules and advanced approach has one module which builds a module with reachable state space only, not possible state space. Therefore, advance approach is able to deal with 226 sensor nodes instead of just 26 sensor nodes of original approach.

An interesting study to access foraging swarm robot activity using probabilistic modeling is presented in [19]. The significant characteristics of foraging robot are modeled in finite state form. A number of states such as searching, grabbing, depositing, homing, resting, coordinated and swarm connected are considered. The state space increased rapidly and even with just six robots, it reaches to 10E+21 with microscopic approach. Therefore counting abstraction approach is used which avoids low level probabilistic behavior and only global behavior is considered. In other words, with macroscopic approach, the state space explosion problem is eliminated and common behavior in a single state machine is modeled.

Reference [15] shows the performance checking of mobile wireless sensor network. A mobile wireless sensor network for hospital security is first modeled with stochastic π-calculus then translated to PRISM model. To capture system's behavior CTMC model is constructed then a number of properties are specified to formally check it. The network consists of mobile node, two alarming devices, three fixed routers and two fixed sinks. The lengths of data frames and acknowledgments used by IEEE802.11 and Ethernet protocols are also specified. The expected timings for alarm transmissions based on varying node mobility situations are also discussed.

Table I presents a comparative evaluation of varying protocols which are using PRISM model checker. Column 1 shows the name of the protocol being considered along with the PRISM model, column 2 indicates the machine specifications used during particular experiment and column 3 presents the number of nodes used in the experiment. Column 4 presents the model size in terms of states, column 5 and 6 give the model building time and model checking time respectively and column 7 demonstrates the model storage space in terms of either MTBDD node or memory used. It is noticeable that in most of the protocols, number of nodes considered during modelling are not explicitly defined which is one of the imperative parameters. It is indeed debatable to bring forward the contrast among the protocols presented in the Table I, however their results cannot be truly compared due to dissimilar machine specifications, model type and modelling parameters.

In this section, we have shown how PRISM is successfully applied for analysis and verification of gossip protocols, randomized protocols, multi agent systems, team formation protocol, foraging swarm robot and PEADG. This proves that PRISM has potential to verify probabilistic and non deterministic models relating to diverse domains.

TABLE I: COMPARISON OF VARYING PROTOCOLS USING PRISM MODEL CHECKER

1	2	3	4	5	6	7
Model	**Machine specification**	**No of nodes**	**Model size (states)**	**Build time (sec)**	**Model check time (sec)**	**Model storage (MTBDD nodes)**
Gossip protocol [4] MDP	2GHz PC with 2GB RAM	3 4	829 74034	1.73 95.0	Not given	Not given
Aspnes&Herlihy's randomised consensus protocol [5] MDP	2.40 GHz PC with 2 GB RAM running Linux	Not given	2.1e+14	15.3	>24h	116133
Randomised Byzantine agreement protocol [5] MDP	2.40 GHz PC with 2 GB RAM running Linux	Not given	1.9e+16	831.0	mem-out	1.3e+7
Randomized mutual exclusion protocols of Rabin [5] MDP	2.40 GHz PC with 2 GB RAM running Linux	Not given	4.5e+10	17.5	0.91	381184
Fibroblast Growth Factor signaling pathway [5] CTMC	2.40 GHz PC with 2 GB RAM running Linux	Not given	2.0e+9	110.2	mem-out	1.1e+6
Peer-to-peer protocol based on BitTorrent [5] CTMC	2.40 GHz PC with 2 GB RAM running Linux	Not given	3.4e+10	0.40	mem-out	54916
Minimum space shared memory leader election protocol [5] nondeterministic	2.40 GHz PC with 2 GB RAM running Linux	Not given	6.3e+66	32.5	6241	238940
IEEE 802.3 CSMA/CD [16] MDP	2.80GHz Pentium 4 PC with 1GB RAM running Linux	Not given	1.0E+11	1831	mem-out	2.1e+6
Team formation protocols [17] DTMC MDP	2.8GHz Intel Core 2 PC, 4Gb of RAM running Fedora Core 13	Not given	 35058241 8155873	 2916.2 29.7	 5-240	Not given
Power efficient algorithm for data gathering (PEADG) [18] DTMC	Intel Core Duo CPU (2.93GHz) and 2GB RAM	Original 22 Advanced 226	63 675	24.7	Not given	244 MB 63MB
Robot swarm behaviour [19]	2.4 GHz Core 2, 6 GB RAM running Mac OS	3	2.3E+10	4.35	835	>3 GB
Fault tolerant target-tracking protocol CTMC [20]	2.4 GHz dual core Processor and 4GB RAM running Windows Vista	Grid arrangement of 16 sensor nodes Random placement of 16 sensor nodes	2401992152 4.4e + 28	0.48 375.14	0.23 12.16	9747 2343661

IV. CASE STUDY: MODEL BUILDING AND SYMMETRY REDUCTION OF FTTT PROTOCOL

In this section we are demonstrating the modeling and symmetry reduction of FTTT protocol for wireless sensor network using CTMC model. The detailed working of FTTT protocol is presented in our previous work [20]. FTTT protocol works in a wireless sensor network consisting of sensor nodes, cluster heads and base stations. In first step of FTTT protocol clusters are organized to maintain data related with member sensor nodes. The second step of FTTT protocol consists of identifying redundant sensor nodes. The moving target information is send to the base station after removing redundancy in the third step of the protocol. The cluster based arrangement also help in administration of failed cluster heads as the fourth step of protocol. The final step of FTTT protocol deals with failed sensor nodes avoiding extra messages. The fault tolerance in the protocol is achieved using the deployment redundancy and avoiding the increase in cost overhead of the protocol.

The probabilistic communication involving cluster heads, base station, sensor nodes and uncertain sensor node failures leads towards probabilistic modeling of FTTT protocol. The preliminary model checking of FTTT protocol using PRISM is presented in our work [21]; considering specific number of sensor nodes at fixed locations. In initial model checking of FTTT protocol parameters and properties concerned with four key steps of FTTT protocol which are formation of cluster, reducing redundancy, target-tracking and fault tolerance are described. The analysis of FTTT protocol is focused on discovery of number of sensor nodes linking the cluster, expected count of messages, count of sensor nodes overlapping, energy consumption, tracking probability and percentage of fault tolerance. The corresponding results are graphically presented.

In this section, a performance comparison of FTTT via PRISM, PRISM-symm and generic representatives in PRISM (GRIP) models is elaborated with synchronized events. The CTMC model consists of cluster head module,

sensor node module, snetwork module and a module for target. The cluster head module communicates with snetwork module and snetwork module communicate with sensor node module. The detailed states and transitions contained in modules of FTTT, coding for development of parallel combination of sensor nodes, cluster heads and a target are presented in our work [22]. Authors in [22] also presented modules with synchronized events and value relationship of variables during modeling. PRISM transforms these modules into a multi-terminal binary decision diagram (MTBDD) representation, performs computation of set of entire reachable states and eliminates unreachable states. The construction process of PRISM-symm model is analogous to PRISM in addition to symmetry reduction step which is applied to the MTBDD representation of those modules. For GRIP, model construction procedure detailed in [23] is followed, in which generic model after translation from high-level model is passed to PRISM. In FTTT CTMC model; 16 sensor nodes at maximum and 2 cluster heads are used in two rounds of experiments.

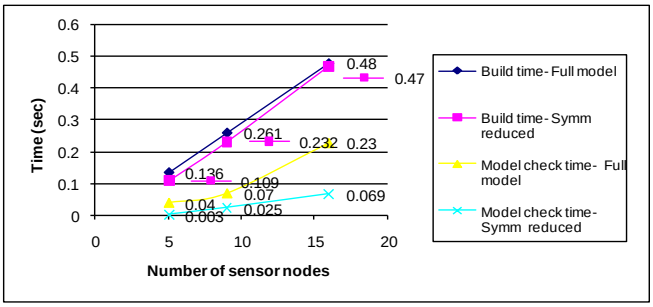

Fig. 2. Time used by PRISM (Full model) and PRISM-symm with pre defined arrangement of sensor nodes.

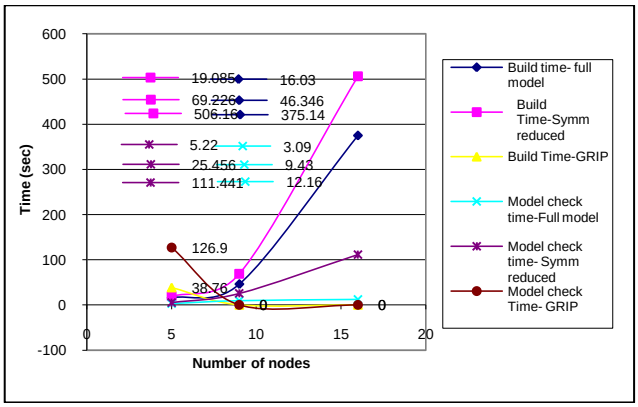

Fig. 3. Time used by PRISM (Full model), PRISM-symm and GRIP with random arrangement of sensor nodes.

In the first round of experiments, sensor nodes are placed in grid arrangement and in second round, sensor nodes are randomly placed. The model building and model checking time of full model and symmetry reduced model with respect to number of nodes and grid arrangement of sensor nodes is presented in Fig. 2. It is observed that there is small difference in the model building time of the two models, however; model checking time shows noticeable difference. PRISM-symm consumes a lesser amount of model checking time than PRISM with same number of sensor nodes.

The model building and model checking time used by GRIP, PRISM-symm and PRISM with random arrangement of sensor nodes is presented in Fig. 3. It is evident from the comparison of Fig. 2 and Fig. 3 that model build and model check time is increased in case of random placement of sensor nodes yet with equal number of sensor nodes in both PRISM and PRISM-symm. GRIP is unable to check the models with more than five sensor nodes and exhibits out of memory error which is represented by zeros in Fig. 3. Thus the state space explosion problem arises in second round due to large number of states generated in GRIP. Experiments confirmed that the results obtained via PRISM-symm are improved than GRIP and PRISM tool while comparing number of states and MTBDD nodes at the cost of more model building and model checking time.

V. Conclusion

Diverse well-known techniques are being implemented for performance measure of novel protocols in this decade. Probabilistic model checking is one of them which has been widely used for checking of qualitative and quantitative properties of stochastic systems. The emergence of probabilistic model checking in various areas and the appropriateness of PRISM in variety of applications moved us to present an overview of those case studies. In this paper we presented twelve different protocols with their implementation in PRISM. It is also found that symmetry reduction permits to perform verification on many orders of magnitude large models by occupying less memory. The FTTT case study addresses the state space explosion problem in the modeling environment with two techniques; namely PRISM-symm and GRIP; however, these techniques fail to demonstrate reduction in state space even with only sixteen sensor nodes. It concludes that for the modeling of FTTT protocol; remediation of state space explosion problem is still a key issue because of scalable behavior of wireless sensor networks.

Acknowledgment

The authors would like to thank MUET for provision of resources for this research work.

References

[1] The PRISM web site. [Online]. Available: http://www.prismmodelchecker.org

[2] M. Kwiatkowska and D. Parker, "Advances in probabilistic model checking," *Software Safety and Security - Tools for Analysis and Verification, NATO Science for Peace and Security Series – D: Information and Communication Security*, IOS Press, vol. 33, pp. 126-151, 2012.

[3] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM: Probabilistic model checking for performance and reliability analysis," *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, no. 4, pp. 40-45, 2009.

[4] M. Kwiatkowska, G. Norman, and D. Parker, "Analysis of a gossip protocol in PRISM," *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, no. 3, pp. 17–22, 2008.

[5] A. Donaldson, A. Miller, and D. Parker, "Language-level symmetry reduction for probabilistic model checking," *Proceedings of Sixth Int. Conf. on Quantitative Evaluation of Systems (QEST'09), IEEE Computer Society*, pp. 289–298, September 2009.

[6] M. Memon, S. Bhatti, and S. Memon, "Probabilistic and hybrid Model checking deployments for wireless sensor networks," *Mehran University Research Journal of Engineering and Technology*, vol. 31, no. 1, pp. 177-188, January 2012.

[7] F. A. Boulis, A. Fehnker, M. Fruth, and A. McIver, "CaVi-Simulation and model checking for wireless sensor networks," in *Proc. the Fifth International Conference on Quantitative Evaluation of Systems*, September 14-17, 2008.

[8] L. Mottola, T. Voigt, F. Österlind, J. Eriksson, L. Baresi, and C. Ghezzi, "Anquiro: Enabling efficient static verification of sensor network software," in *Proc. Workshop on Software Engineering for Sensor Network Applications*, 2010.

[9] T. Herault, R. Lassaigne, and S. Peyronnet, "APMC 3.0: Approximate verification of discrete and continuous time Markov chains," in *Proc. Third International Conference on Quantitative Evaluation of Systems*, 2006, pp. 129-130.
[10] P.C. Olveczky and S. Thorvaldsen, "Formal modeling and analysis of the OGDC wireless sensor network algorithm in real-time Maude," in *Proc. Formal Methods for Open Object-Based Distributed Systems*, 2007, pp. 122-140.
[11] G. Behrmann, A. David, K. G. Larsen, J. Hakansson, P. Petter-son, W. Yi, and M. Hendriks, "Uppaal 4.0," *IEEE Computer Society Quantitative Evaluation of Systems, ,* pp. 125-126, 2006.
[12] L. Vigano, "Automated security protocol analysis with the AVISPA tool," *Electronic Notes in Theoretical Computer Science*, vol. 155, pp. 61-86, 2006.
[13] S. Coleri, M. Ergen, and T. J. Koo, "Lifetime analysis of a sensor network with hybrid automata modelling," in *Proc. the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, ACM Press, 2007, pp. 98-104.
[14] Y. Hanna, H. Rajan, and W. Zang, "SLEDE: A domain specific framework for sensor network security protocol implementation," in *Proc. the First ACM Conference on Wireless Network Security*, Alexandria, VA, USA, March 31-April 2, 2008, pp. 109-118.
[15] R. Abo and K. Barkaoui, "A performability analysis of mobile wireless sensor networks with probabilistic model checking," *Proceedings of Wireless Advanced*, 2011, pp. 283-288, 20-22 June.
[16] M. Kwiatkowska, G. Norman, and D. Parker, "Symmetry reduction for probabilistic model checking," in *Proc. 18th International Conference on Computer Aided Verification*, pp. 234–248, August 2006.
[17] T. Chen, M. Kwiatkowska, D. Parker, and A. Simaitis, "Verifying team formation protocols with probabilistic model checking," in *Proc. 12th Int. Workshop on Computational Logic in Multi-Agent Systems (CLIMA XII 2011)*, July 2011, pp. 190–297.
[18] K. He, H. Yang, Y. Feng, Y. Liu, and Z. Qiu, "Performance analysis of data gathering protocol using PRISM," in *Proc. 17th IEEE International Conference on Engineering of Complex Computer Systems*, 2012, pp. 96-105.
[19] S. Konur, C. Dixon, and M. Fisher, "Analysing robot swarm behaviour via probabilistic model checking," *Robotics and Autonomous Systems*, vol. 60, no. 2, pp. 199-213, 2012.
[20] S. Bhatti, T. J. S. Khanzada, and S. Memon, "Clustering and fault tolerance for target tracking using wireless sensor networks," *Mehran University Research Journal of Engineering and Technology*, vol. 31, no. 4, pp. 769-776 Oct. 2012.
[21] S. Bhatti, J. Xu, and M. Memon, "Model checking of a Target Tracking Protocol for Wireless Sensor Networks," in *Proc. 10th International Conference on Computer and Information Technology*, 29 June-01 July, 2010, Bradford, UK, pp. 2867-2872.
[22] S. Bhatti, S. Memon, I. A. Jokhio, and M. Memon, "Modelling and symmetry reduction of a target tracking protocol using wireless sensor networks," *International Journal of IET Communications*, 3rd July 2012, vol. 6, no. 10, pp. 1205-1211.
[23] A. Donaldson, A. Miller, and D. Parker, "Language-level symmetry reduction for probabilistic model checking," in *Proc. Sixth International Conference on Quantitative Evaluation of Systems (QEST'09),* September 2009, pp. 289–298.

Sania Bhatti completed her bachelor of engineering in computer systems in Feb. 2004 and master of engineering in information technlogy in Feb. 2007, from Mehran, UET Jamshoro. The same year she was awarded the PhD scholarship under faculty development program. She obtained her PhD degree from the University of Leeds, United Kingdom in 2010. She is working with the Department of Software Engineering, Mehran UET, Jamshoro, Sindh, Pakistan, since Sept. 2004. Her research interests include modelling and simulation, wireless sensor networks and software engineering.

Dr. Sania has a number international conference and journal publications related to the field of wireless sensor networks, modeling and simulation.

Mohsin Memon received his bachelor of engineering in software engineering and master of engineering in information technology at Mehran University of Engineering and Technology, Pakistan, in 2006 and 2009, respectively. He achieved Ph.D. degree from the Department of Computer Science, University of Tsukuba, Japan in 2014. He is working with the Department of Computer Systems Engineering, Mehran UET, Sindh, Pakistan, Aug. 2006. His research interests include interaction technologies, life logging, and mobile computing.

Sheeraz Memon completed his bachelor of engineering in computer systems in Feb. 2004 and master of engineering in communication systems and networks in Feb. 2007, from Mehran, UET Jamshoro. The same year he was awarded the PhD, Scholarship under faculty development program, and Mr. Memon left for securing the doctoral degree from RMIT University Australia.

He is working with the Department of Computer Systems Engineering, Mehran UET, Sindh, Pakistan, since Sep, 2004. He is skilled in Digital Signal Processing (DSP), Voice/Speech Feature Extraction, Machine learning/Pattern Recognition and Digital Image Processing. During his PhD he proposed a novel Optimization algorithm called ITEM. Mr. Memon possesses extensive experience in Speech and Speaker Recognition systems, have worked on speaker recognition in adverse conditions such as speakers recorded in clinical environment and undergoing depression.