Optimised Test and Security Solution for P2P-based M2M Applications

B. Shala, P. Wacht, U. Trick, A. Lehmann
Research Group for Telecommunication Networks, Frankfurt University of Applied Sciences, Frankfurt am Main, Germany,
{shala, wacht, trick, lehmann}@e-technik.org

Abstract

Peer-to-Peer (P2P)-based application provisioning in Machine-to-Machine (M2M) communication systems offers the possibility to integrate the end user for realisation of M2M applications considering the individual requirements without the use of centralised instances. Despite many advantages, this flexible methodology of M2M application provisioning does not consider the opportunity of testing the new created and provisioned applications. Moreover, it does not pay attention to security concerns. This publication presents a novel concept for automated testing of P2P connected M2M networks. Different challenges and requirements for testing are defined and a novel testing architecture for functional testing is introduced. Also, this publication evaluates different security problems inside the P2P-based M2M application (P2P4M2M) framework and introduces the importance of trust for ensuring security. To overcome the security drawbacks of P2P4M2M, this publication presents a novel concept for securing services and applications using the integration of a trust management system.

1 Introduction

Machine-to-Machine (M2M) communications is applied in many different application fields, such as energy management, ambient assisted living, building surveillance, smart home, traffic management and electro mobility. These application fields aim to increase the quality of life and efficiency. The European Telecommunications Standards Institute (ETSI) defined M2M applications as “applications that run the service logic and use Service Capabilities accessible via open interfaces” [1]. Previous papers [2, 3] have defined requirements and concepts to realise service and application provisioning in M2M. The authors in [2] present the P2P4M2M framework that realises service and application provisioning using P2P networking in M2M application field. A service, as well as an application, can be realised by peers using technical or non-technical principles. An application consists of one or more underlying services that are combined (i.e. aggregated or composed). These peers are represented by technical devices who are networked using P2P mechanism. The use of the M2M community concept described in [2] forms a social network of peers and helps to avoid legal restrictions, to adjust different interests among the peers and to ensure optimisation and forming P2P networks. The networking enables the participating peers to provide a service or an application that can be consumed by others [2]. According to [3], the information exchange between the peers for the service utilisation and the signalling to generate the application is enabled by using various M2M communication protocols (e.g., CoAP, HTTP, SIP) based on SUBSCRIBE/ NOTIFY principle. The Service Management Framework (SMF) described in [4] is the main component for service and application provisioning in M2M based on the P2P4M2M framework [2]. Reference [4] introduces a Service Management Framework (SMF) installed in the local households, consisting of Service Delivery Platform (SDP) and Service Creation Environment (SCE), and uses the concept of P2P networked energy-community. The SCE brings the functionality to design and configure services graphically compliant to the personal needs of the users [4]. Thus, the SMF used in the P2P4M2M framework gives every peer in the M2M community the possibility to create and configure M2M applications using the SCE which is integrated in the peer. Fig. 1 shows the structure of the P2P connected peers within an M2M community.

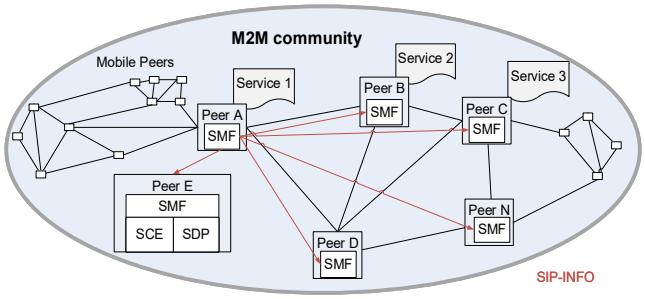

Figure 1: P2P connected peers within a M2M community [2]

The concept presented in [2] and [3] does not consider approaches for testing P2P-based M2M services and applications. Therefore, a novel testing framework is required to enable testing of heteregoeneous and decentralised services and applications. Another drawback of this concept is that it does not provide an evaluation of different security issues in P2P4M2M and no strategies to handle security risks. The increasing number of attacks in M2M networks creates the necessity to develop technologies for preventing attacks and system failures [5].

One aim of this paper is to illustrate the challenges and requirements of testing M2M services and applications based on the P2P4M2M framework (chapter II) and to define a novel concept for functional automated testing (chapter III). The testing methodology in this concept is based on model-based testing because of its advantages described in [6]. Another aim of this paper is to define security issues based on the different layers of the P2P4M2M framework (chapter IV) and to optimise and ensure security by introducing a trust management system (chapter V).

2 Challenges and Requirements for Testing P2P4M2M

The process of creating M2M applications based on [3] makes functional testing very complex and can be described as follows: The application creator developes and designs an M2M application using his SCE. The M2M application consists of several services which are part of an M2M community. Each service involved in an application is described by its Service Interface Description. The Service Interface Description includes service ID, service functions, input, output and further configuration parameters of a service. Services can be provided by different peers participating in a P2P network without the use of a central authority. The creation of an application will generate an SCXML (State Chart XML) description which precisely describes the potential functionality the application should deliver in a formal manner. In principle, such an SCXML description includes the involved services, the connection of services, as well as conditions and definitions of input/ output parameters.

A special testing framework is required for testing the M2M application. First of all, the P2P4M2M framework is a distributed system. Based on [7], the size and complexity of distributed systems is growing and the system should be able to run over a wide variety of different platforms and should access different kinds of interfaces. Considering the distributed characteristics of the P2P4M2M framework and the need for exchanging relevant information for testing, it is important to ensure collaboration between the services, applications and elements of the testing framework. The decentralisation of the peers and their volatility (nodes leaving and entering suddenly) in the P2P-based M2M application community has to be considered in the testing framework. Especially in the P2P4M2M concept, the application creator could be a user who has no technical background and who is not able to prepare the testing. Also, based on [8], the application creator should not be the test creator. If the application creator has already interpreted a specification incorrectly during the development, he will also misinterpret it for the test. For this reason and for the advantages of test automation [8] the testing needs to be automated using a mechanism which utilises the information about the functionality provided by the application and the services. Another problem is the procedure for defining test cases. Testing distributed services and applications in M2M networks requires different methods for deriving and generating test suites and for running the test. Reference [9] presents several problems for testing distributed systems including the test data generation and the specific execution behaviour. The testing framework must have the ability to derive test cases from the information gathered by the application and the distributed services. Based on the characteristics of the P2P4M2M framework presented in [3] the execution of the test cases on the participating services and the composed application should also be considered. Furthermore, for dealing with several security issues inside the P2P4M2M framework the testing framework needs to enable the integration of a trust management system. The importance of this system and the concept of trust will be introduced in chapter IV. Due to these different challenges, the general requirements for the testing framework can be summarised as follows:

- **Collaboration** – It is necessary to have collaboration between the application creator, the elements of the testing framework and the peers, which are part of the application, and are providing or consuming services.
- **Deployment** – The testing framework needs to have the ability to deal with a high number of peers and also the volatility of nodes in P2P network should be considered by the framework.
- **Test Automation** – Based on the complexity of the P2P4M2M framework the whole testing process needs to be automated considering the distributed architecture of the system.
- **Test Derivation** – Test suites need to be derived and generated from the gathered information about the composed M2M application and the participating services.
- **Test Execution** – The generated test cases need to be executed on different services in a timely manner. Also the test cases for the whole application should be executed after its creation.
- **Verification** – The testing process should deliver results about the functionality of the considered System under Test (SUT), which could be a service or an application.
- **Tool support** – The framework should provide tools to generate, execute and manage tests.
- **P2P and M2M capability** – The framework should consider the included P2P mechanism and its characteristics within the application framework. M2M communication protocols should also be supported.
- **Trust Management System support** – The framework should provide the possibility to integrate a trust management system in its architecture.

3 Proposed Testing Framework

The challenges of testing (see chapter II) and the complexity of the P2P4M2M framework lead to the necessity to define a suitable testing framework for functional testing of

M2M services and applications. Functional testing is the process of verifying the functions in a system to assure that they meet the specified requirement. Reference [10] defines that in functional testing "a tester does not need to know the internals of the SUT as the focus is to evaluate the functional correctness of a given system, independently of its internal implementation".

Three black-box testing scenarios can be derived based on the application creation process described in [3]. The first scenario deals with the testing of a service after it enters the M2M community. This happens in order to ensure the availability of the correctly working services in the community and should be done after predefined time intervals. The second and third testing scenarios will happen after the application creator has built an application using several services participating in the M2M community. The services, which are part of the composed application, need to be tested according to their special configurations within the application (second scenario). In the third scenario the composed and created application needs to be tested based on its configuration and according to the special conditions of each participating service in the composition.

Based on related work [11-16], the requirements for the testing framework for P2P-based M2M applications and the need for a load balanced and effective testing mechanism, a test architecture with a combination of a global tester called Test Master and distributed testers called Test Agents is presented in this work. An additional test component for load balancing is required due to the increasing number of participating peers and provided services in the M2M community and the inability of the Test Master to scale up with the increasing number of distributed services. Therefore, a Test Generation Environment (TGE) is included in the testing framework which derives and generates test cases and also interacts with the Test Master, the services and the application creator. Fig. 3 shows the conceptual test architecture consisting of a Test Master, Test Agents and TGE.

The TGE gets an SCXML description of a composed application and the service interface descriptions of each participating service and generates test cases for this application respectively each of the participating services. Afterwards, the TGE will send the relevant test instructions to the Test Master who is the coordinator of the overall testing framework. The Test Master will send test instructions to the Test Agents, which will afterwards execute the test cases on the SUT. SUT are all services which are part of the community and the composed application. The detailed functionality of the Test Master can be described as follows: controlling and managing test processes, receiving test instructions from the TGE, receiving and evaluating test results, providing the application creator and the service providers with information about the test results, sending test instructions to the test agents, interacting with all test elements, and maintaining list of test agent. The functionality of the TGE is the following: receiving Service Interface Descriptions and other relevant Service Descriptions from the services, receiving SCXML descriptions from the Application Creator, deriving and generating test cases, sending test instructions to the Test Master. The functionality of Test Agents can be described as follows: receiving instructions from the Test Master, executing test instructions on SUT, sending test results to the Test Master and exchanging test related information with the Test Master.

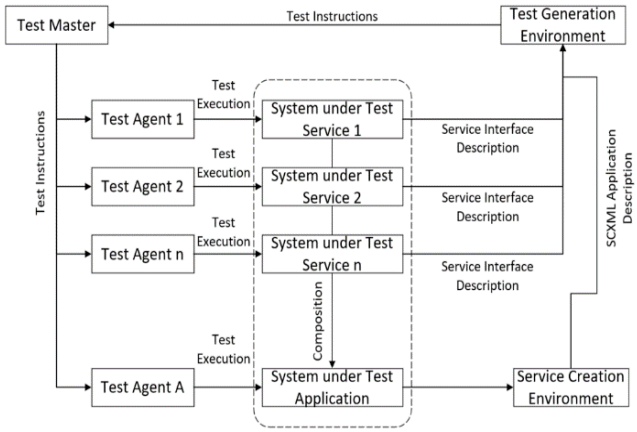

Figure 2: Conceptual Test Architecture of P2P-based M2M environments

4 Security Issues and Concept of Trust

Fig. 3 illustrates the functional architecture of the P2P4M2M framework which consists of several components. Considering the security aspect, the following general categorisation can be determined based on [3]: a) M2M network – includes M2M application, M2M service, M2M communication protocol, b) P2P network – includes P2P communication and P2P overlay, c) IP network. It has to be emphazised that security for IP networks is out of scope in this research due to the fact that there already is a huge amount of publications dealing with IP networks and network security (e.g. [17] and [18]) which describe different vulnerabilities and several security solutions for IP networks.

For M2M communications, [19] defines several potential security issues by dividing them into three categories: Physical attacks, logical attacks and data attacks. Also, [19] lists the different attacks for each category: Physical attacks include side channel attacks, software modification and malwares, destruction or theft of the M2M device. Logical attacks include impersonation, denial of service and relay attacks. Data attacks include privacy attacks, data modification and false information injection as well as selective forwarding/ interception. Furthermore, reference [5] provides an overview of the current state of security in sensor and ad-hoc networking for M2M communications.

Exemplary for the application field of smart homes, [20] provides a landscape of threats assumed for smart home assets.

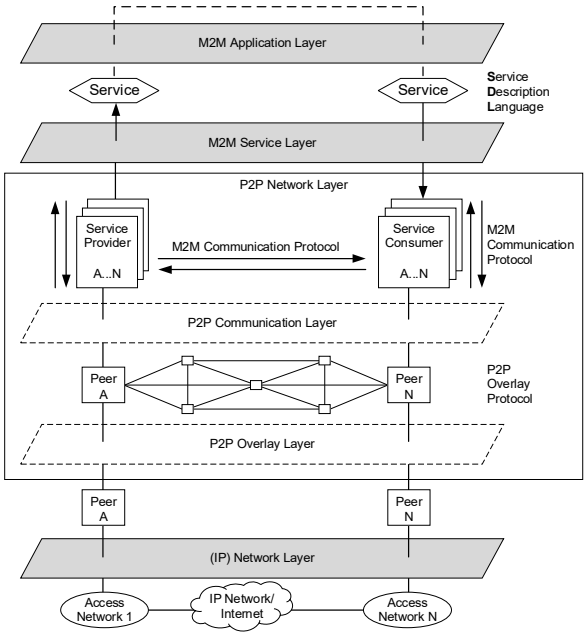

Figure 3: Functional Architecture of the P2P4M2M framework [3]

P2P communication between the different services and applications plays a crucial role in P2P4M2M framework. P2P overlays are virtual topologies that are built on top of physical networks. During the past years different P2P overlay protocols have been developed. The P2P overlay protocols used in P2P communications define different rules for communication in the overlay network, such as routing the messages over the overlay, bootstrapping into the overlay, mapping the nodes in the network and maintaining the nodes in the overlay. The authors in [2] mention the different advantages in using P2P communication instead of client/ server architectures. Security threats in P2P networks can be classified based on [21] in: eavesdropping, communication jamming, injection and modification of data, unauthorised access, repudiation, man-in-the-middle attack and sybil attack. Furthermore, based on several publications [22-29] most of the different P2P protocols are not secure against several security attacks and do not provide an efficient protection mechanism. Furthermore several publications state that bootstrapping, Denial of Service (DoS) and identity attacks have the worst impact on security in the P2P overlay. Ensuring anonymity among the P2P nodes is also not solving the issue for most of the P2P protocols.

Two main categories of problems related to security can be identified for P2P-based M2M applications: a) attacks from outside of the M2M community e.g. peers who want to harm the system by bootstrapping into the community. b) Attacks from the inside of the M2M community e.g. peers with bad intentions trying to break down by falsifying information in the community, network, or P2P layer. In order to successfully deal with these attacks, a security concept for preventing the entrance of malicious peers inside the community should be developed. The concept should also include a solution for preventing malicious behaviour of existing peers in the community. Based on [2] and Fig. 3, peers are using the P2P overlay for finding each other and for storing relevant information. Furthermore, they communicate using M2M communication protocols and are able to use and provide services. The different security issues described in [19] are executed by malicious peers and therefore the focus for ensuring security inside the P2P4M2M framework should be on peers.

Attacks on the P2P layer can have a significant impact for the correct functionality of the whole system in P2P4M2M. Based on [30] it is difficult to implement security protections in P2P systems compared to centrally administered systems and security strategies need to be decentralised. Additionally, it is difficult to validate without centralised control peer identity and trustworthiness between peers [31]. Reference [31] also states that a P2P system relies on a set of distributed peers working fairly and properly together and defines the level of trust as “the level of confidence of one peer toward another peer with which it is communicating”. As stated above on the basis of trust, many attacks can be mitigated by removing trustless peers from the system. This way, the existing peers are able to continue working reliably and providing trustworthy services without getting harmed by attackers. According to [32], trust can be defined as “an accumulated value from the history and the expecting value for the future. Trust is quantitatively/ qualitatively calculated and measured which is used to evaluate values of physical components, value chains among multiple stakeholder and human behaviours including decision making. Trust is broader concept that can cover security and privacy“. Moreover, trust can be applied to peers providing a service and peers using a service. Furthermore, trust can be applied for provided services and applications. In this research paper the focus is to apply trust to peers providing a service and the provided services. For evaluating trust, the following three main steps need to be accomplished: data collection, data analysis and trust decision. For ensuring the collection of the right data, trust metrics need to be defined. [32] defines trust metric as “a measure to evaluate a level of trust by which a human or an object can be judged or decided from trustworthiness”. Based on defined requirements the collected data has to be analysed and evaluated by the trust decision process. This process sets up the level of trust for the tested entities.

5 Requirements and Principles of Trust Management System

The huge amount of data collected in the P2P4M2M should be processed and analysed in a trustworthy way. Based on the trust metric parameters and the results of the trust evaluation, the peers are categorised as either trustworthy or

untrustworthy. For the Trust Management System (TMS) presented in this research, the following requirements were initially defined. To avoid centralised management and controlling, trust computing and evaluation have to be realised without any central authority, thus this process has to be **autonomous and decentralised**. For ensuring trust from the beginning of a working service, the **initial trust level of it** has to be considered and evaluated. This enables the possibility for the peers to figure out quickly trustworthiness among other peers and services. The TMS needs to ensure **flexibility** since one of the challenges in the P2P4M2M framework is the heterogeneity of peers and services. An important requirement is also the **volatility of peers and services.** In the M2M community, peers are able to suddenly enter or leave the network and this leads to rapid changes in existing trust relationships between peers. As the number of peers and services follows an increasing trend, the TMS has to ensure **scalability and stability.** Peers are able to provide more than one service and the trust evaluation must not be based only on one service but has to consider the **variety of different services** provided by the peer. Furthermore, the TMS needs to consider **context-dependency** and to ensure that a peer can trust e.g. service 1 but mistrust service 2 of another peer**.** The trust computing and evaluating will generate a significant amount of trust data among the peers and the TMS has to provide a mechanism for securing trust data storage and to ensure with that the **trustworthiness of trust data.**

As mentioned in the previous section, the concept of trust in this research is interpreted as a value for measuring the reliability and correctness of different working services provided by different peers and used in several composed applications. As any peer can provide many services within the P2P4M2M framework we consider that the total trust level of a peer consists of the trust levels of the services it provides. For that reason, we focus on trust evaluation based on services. The testing framework described in chapter III is considered for the integration of a TMS. Taking into account the heterogeneity and complexity of the services and applications, a decentralised approach for the architecture of the TMS is considered in this research. For trust evaluation we consider the trust level of a newly provided service. For a newly provided service, trust has to be computed and evaluated by integrating this process into the test framework with the Test Master and Test Agents defined in [2]. Fig. 4 shows the integration of the TMS inside the testing framework and the workflow of computing trust for an entering service. After service 1 enters the M2M community, its functionality will be tested using the testing framework described in chapter III. Moreover, using this test architecture presented it is possible to derive the initial trust level and to check whether or not the service entered the community is trustworthy. Based on the functionality of the service, the TGE will generate suitable test cases for security tests and the test agents will execute these cases on the service. During the test execution, the service will exchange messages with the test system or its Test Agent that are responsible for it. After the test is executed, the Test Master will receive the test report and will evaluate the trust level of the entered service 1. Then, the Test Master will send the trust information via the test agent to the peer who is providing the service. The trust information will be stored inside the P2P network and the SMF which is included in the peer and will be updated accordingly to the behaviour of the peer. These information will be used for ensuring a reliable and trustworthy environment.

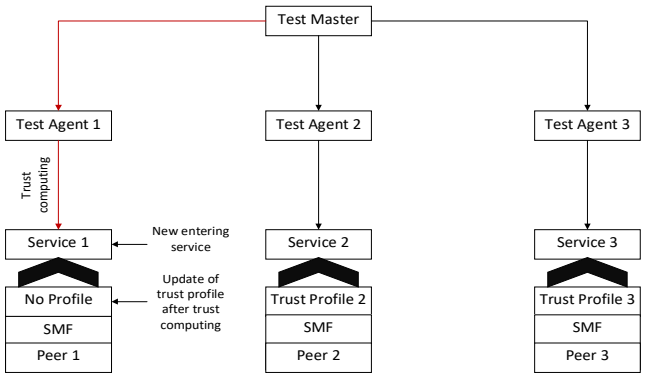

Figure 4: Computing trust of an entering service using the testing framework

6 Conclusion

This publication presents a novel concept for automated functional testing of P2P-based services and applications in M2M. The presented concept aims to deal with the complexity of testing applications which are composed by several heterogenic services with different service functionalities and configuration parameters. Furthermore, the presented test architecture ensures load balancing and efficient automated testing of M2M services and applications. The missing role of a test creator in the P2P4M2M framework is solved using the integration of the TGE.

Despite the fact that service and application provisioning in M2M renders many advantages, it also forms an attractive platform for many attackers and malicious peers. This publication presents the so far missing security risks and requirements for the P2P4M2M framework. Furthermore, the concept of trust is introduced and major requirements for an effective and stable trust management concept are presented. Moreover, considering the initial trust level, a trust management system is presented which enables the trustworthy service and application provisioning in M2M and decreases the risks of security attacks.

7 Acknowledgments

The research project P2P4M2M providing the basis for this publication was partially funded by the Federal Ministry of Education and Research (BMBF) of the Federal Republic of Germany under grant number 03FH022IX5. The authors of this publication are in charge of its content.

8 References

[1] ETSI TR 102 725, V1.1.1, 2013-06: Technical Report, "Machine-to-Machine communications (M2M); Definitions", ETSI TISPAN

[2] M. Steinheimer, U. Trick, W. Fuhrmann and B. Ghita, "P2P-based community concept for M2M Applications", Proc. of Second International Conference on Future Generation Communication Technologies (FGCT 2013), London, UK, December 2013

[3] M. Steinheimer, U. Trick, B. Ghita and W. Fuhrmann, "Autonomous decentralised M2M Application Service Provision," unpublished

[4] M. Steinheimer, U. Trick, P. Ruhrig, R. Tönjes, M. Fischer and D. Hölker, „SIP-basierte P2P-Vernetzung in einer Energie-Community", ITG-Fachbericht 242: Mobilkommunikation, pp. 64, Mai 2013

[5] European Union Agency for Network and Information Security (ENISA), "Ad-hoc & sensor networking for M2M Communications – Threat Landscape and Good Practice Guide", 2017

[6] P. Wacht, "Framework for Automated Functional Tests within Value-Added Service Environments", PhD Thesis, School of Computing and Mathematics, University of Plymouth, UK, December 2015

[7] A. Saifan and J. Dingel, "Model-based testing of distributed systems," Technichal report, vol. 548, 2008

[8] M. Winter, T. Roßner, C. Brandes and H. Götz, "Basiswissen modellbasierter Test", dpunkt Verlag Heidelberg, Germany, ISBN: 978-3-86490-297-0, 2016

[9] S. Ghosh and A. Mathur, "Issues in testing distributed component-based systems," In Proceedings of the First International Conference on Softaware Engineering Workshop on Testing Distributed Component-Based Systems, Los Angeles, CA, May 1999

[10] M. Pezzè and M. Young, "Software testen und analysieren" (translated title: "Testing and analysing software"), Oldenbourg, Munich, Germany, ISBN: 3-486-58521-6. 2009

[11] E. Almeida, G. Sunye, Y. Traon and P. Valduriez, "A framework for testing peer-to-peer systems," Dans 19th International Symposium on Software Reliability Engineering (ISSRE 2008), Redmond, Seattle, USA, IEEE Computer Society, 2008

[12] A. Ulrich and H. Konig, "Architectures for testing distributed systems", Dans Proceedings of the IFIP TC6 12th International Workshop on Testing Communicating Systems, pp. 93–108, Deventer, The Netherlands. Kluwer, B.V., 1999

[13] P. Rosenkranz, M. Wählisch, E. Baccelli and L. Ortmann, "A Distributed Test System Architecture for Open-source IoT Software," IoT-Sys´15 Proceedings of the 2015 Workshop on IoT challenges in Mobile and Industrial Systems, pp. 43-48, ACM, New York. 2015

[14] C. Torens and L. Ebrecht, "RemoteTest: A Framework for Testing Distributed Systems," 2010 Fifth International Conference on Software Engineering Advances, pp. 441-446, Nice, France, 2010

[15] A. Khoumsi, "Testing distributed real time systems using a distributed test architecture", Sixth IEEE Symposium on Computers and Communications, 2001

[16] P. Wacht, U. Trick, W. Fuhrmann and B. Ghita, "Efficient Test Case Derivation from Statecharts-Based Models", Proceedings of the Eleventh International Network Conference, Frankfurt, Germany, 2016

[17] M. Kappes,"Netzwerk- und Datensicherheit", Springer, Wiesbaden, Germany, ISBN: 978-3-8348-0636-9. 2013

[18] J. Vacca, "Computer and Information Security Handbook", Elsevier, Burlingtion, USA, ISBN: 978-0-12-394397-2. 2013

[19] A. Barki, A. Bouabdallah, S. Gharout and J. Traore, "M2M Security: Challenges and Solutions," IEEE Communications Surveys & Tutorials, Volume: 18, Issue: 2, 2016

[20] European Union Agency for Network and Information Security (ENISA), "Threat Landscape and Good Practice Guide for Smart Home and Converged Media", 2014

[21] ITU-T, Framework for secure peer-to-peer communications, X.1161, 2008

[22] Z. Trifa and M. Khemakhem, "Taxonomy of Structured P2P Overlay Networks Security Attacks", International Journal of Computer, Electrical, Automation, Control and Information Engineering, 2012

[23] E. Sit and R. Morris, "Security Considerations for Peer-to-Peer Distributed Hash Tables", International Workshop on Peer-to-Peer Systems, Springer, 2002.

[24] T. Reidemeister, K. Böhm, P. Ward and E. Buchmann, "Malicious Behaviour in Content-Addressable Peer-to-Peer Networks", 3rd Annual Communication Networks and Services Research Conference, 2015

[25] M. Srivatsa and L. Liu, "Vulnerabilities and security threats in structured overlay networks: A quantitative analysis", ACSAC`04, pp.252-261, IEEE, Los Alamitos, 2004

[26] A. Malatras, "State of the art survey on P2P overlay networks in pervasive computing environments", Journal of Network and Computer Applications, Elsevier, vol. 15, pp. 1-23, 2015

[27] J. Arnedo-Moreno and J. Herrera-Joancomarti, "A survey on security in JXTA applications", Journal of Systems and Software, Elsevier, vol. 82, nr. 9, pp.1513-1525, 2009

[28] G. Dosanjh, B. Lodmell, A. Van Der Star and S. Wang, "Gnutella Peer-to-Peer Security", 2007, available from:

[29] I. Clarke, O. Sandberg, B. Wiley and T. Hong, "Freenet: A Distributed Anonymous Information Storage and Retrieval System", Proc. ICSI Workshop, Berkeley, CA, June 2000

[30] C. Selvaraj and S. Anand, "A survey on Security Issues of Reputation Management Systems for Peer-to-Peer Networks", Computer Science Review, Elsevier, p. 145-160, 2012

[31] J. Buford, H. Yu, E. K. Lua, " P2P Networking and Applications", Elsevier, Burlington, USA, ISBN: 978-0-12-374214-8. 2009

[32] ITU-T, Technical Report, Trust Provisioning for future ICT infrastructures and services, 2016