Trust-based Composition of M2M Application Services

Besfort Shala, Ulrich Trick, Armin Lehmann
Research Group for Telecommunication Networks
Frankfurt University of Applied Sciences
Frankfurt/M., Germany
shala@e-technik.org

Besfort Shala, Bogdan Ghita, Stavros Shiaeles
Centre for Security, Communications and Network Research
University of Plymouth
Plymouth, United Kingdom
besfort.shala@plymouth.ac.uk

Abstract— The end-user integration for M2M application service creation and the decentralisation of M2M application service platforms are creating new possibilities for different application fields in the M2M domain. However, besides several advantages, these improvements inherit several security-related issues such as intentionally or accidentally misconfigured M2M application services harming other end-users in the M2M community. This research focuses on evaluating the trustworthiness of new joining decentralised M2M application services provided by end-users. Therefore, this publication presents a novel concept for trust evaluation by combining several model-based testing techniques. Moreover, it defines an approach for trust-based M2M application service selection and composition for end-user consumption. Finally, the overall framework for functional verification and trust evaluation is optimised by full decentralisation of all involved entities.

Keywords— M2M; P2P; Model-based Testing; Service and Application; Trust; Security

I. Introduction

The number of Machine-to-Machine Communication (M2M) devices is exponentially increasing and providing great resources for creating sophisticated M2M application services. The integration of end-users within the M2M application service provision process enables the creation of individual smart environments by building and sharing end-user-based M2M application services with others. The authors in [1] introduce a framework for M2M application service provision where every end-user has the possibility to provide or to consume M2M application services without the use of centralised service providers and without expert knowledge. Therefore, the author in [1] presents a Service Management Framework (SMF) which consists of a local Service Creation Environment (SCE) and a Service Delivery Platform (SDP). Moreover, the SMF includes all available devices and services present in the personal environment of the end-user and integrates also remote services which are provided by other end-users. The SCE provides the possibility for end-users to combine graphically devices or services (local or remote) with each other and to create complex composed M2M application services, which can be made available to other end-users. The M2M application services are described by machine-readable Service and Service Interface Descriptions. To avoid centralised entities, the author in [1] propose to use a Peer-to-Peer (P2P) network for communication and information storage between the peers. To enable social networking between all participating peers (service providers and service consumers) the author in [1] introduce a M2M community. The advantageous of this community are that user can easily access the P2P network and also create sub communities which are addressing different application fields and interests.

Avoiding central instances and transferring all responsibilities for service provision to end-users can increase the risk of failures and malicious behaviours. Less technical knowledge or not manageable behaviour of end-users could lead to serious problems in the M2M community such as wrong, malicious or not working application services. These issues could happen intentionally or accidentally from end-user side. The application service creation process done by the end-user results with a new single or composed application service in the community. In comparison with existing application services, there is no prior knowledge of the new application service in the community nor are there some recommendations, observations or historical data about the past which could give a short overview about the behaviour of the new application service. Moreover, the new application service provides no transaction list and also no rating score in contrast to existing application services in the community which are continuously evaluated regarding their trust level. This knowledge gap about the application service could lead to enormous problems such as security attacks performed by the end-user using the new provided application service. The community and the participants are not able to decide either to trust the application service and to start an interaction or to ignore the application service and/or banning it out of the community

The aim of this paper is to present an optimised framework for functional verification and trust evaluation of M2M application services. This framework enables the identification of the trust level for new provided M2M application services. Therefore, a review of test- and trust-related publications is made and requirements for the framework are defined. Furthermore, this research paper evaluates several approaches for initial trust evaluation and introduces a novel concept for assigning trust levels for new M2M application services by combining model-based testing techniques. Finally, this research defines a mechanism to enable trust-based service selection and composition by end-users.

978-1-5386-4646-5/18/$31.00 ©2018 IEEE

II. Existing Approaches for Testing and Trust in M2M/ IoT and P2P Networks

In order to define requirements for an overall framework, several existing approaches for functional verification and trust evaluation are examined. For functional verification, most relevant testing approaches in M2M/Internet of Things (IoT) are selected [4-6]. The review of trust management systems consists of centralised and decentralised approaches used for ensuring trust among the nodes within the M2M/IoT domain [7-12]. As the focus of this research is a decentralised framework avoiding disadvantages of centralised entities, the related work will also include P2P approaches [13-14]. None of the presented works contribute to an overall framework which could test M2M application services after their deployment and also evaluate their behaviour in order to compute their trust level. Regarding the decentralised capability, only the trust projects presented in [7, 8, 14] support a fully decentralised architecture avoiding centralised authorities and problems of single point of failure. The evaluated test approaches are using semi-centralised or centralised architectures. Another important requirement is the availability of the platform to compose services with each other based on their trust level. Only the approach introduced in [7] considers the evaluated trust levels of the services for service composition. Regarding the test case generation, the reviewed trust approaches do not deal with generation of test cases which could be used for trust evaluation of the behaviour among the nodes. They focus on trust evaluation of existing nodes where the evaluation is made based on mathematical computation of the observation and recommendation scores got from the participating nodes. All of the reviewed projects for testing [4, 5, 6] support functional testing of services but do not consider services which are provided by end-users without the use of centralised authorities. Regarding the integration of the end-user in the test process, the authors in [4] provide a user-friendly web front-end where end-users are able to configure and launch test campaigns by selecting test cases or including specific test data to the database. Regarding the formal description of M2M application services, the work in [5] provides an interesting approach with a so-called Service Test Description. However, this description has missing information regarding security related questions which could be used for trust evaluation. The approach in [6] includes in the service description also semantic models in order to generate test cases considering functional and non-functional properties. Nonetheless, they do not describe in detail the description definition procedure and the test case generation. Most of the evaluated trust management approaches in M2M/IoT do not provide any possibility for evaluating trust of new services. Only the authors in [9] and in [12] propose methods for initial trust assessment of new services/ devices. However, the initiate average rating method proposed in [9] is not a suitable idea because it does not consider the characteristics or the behaviour of a new node. The missing information about how challenges are derived for the challenge-response process used for initial trust evaluation in [12] is a drawback. The fact that a centralised controller is introduced for performing challenges on the device represents another drawback in [12]. Most of the trust approaches also do not provide or consider any solution for a secure data storage system of trust related data. The authors in [7] try to solve the storage management problem by considering only nodes with good trust values and with high impact on the community. However, the framework should consider all trust values because bad trust levels of nodes are also very important in order to mitigate bad behaviour in the community as well as trust values from nodes with low impact on the community. Regarding suitable trust metrics which are used for trust evaluation, the work in [13] and [14] provide interesting trust parameter which could be reused also for the framework presented in this research although they are not enough and should be supported by additional trust parameters. The other publications regarding testing do not consider the above mentioned trust-based requirements in their work. TABLE I summarises the analysed approaches and an extract of the requirements resulting from the strengths and weaknesses of the projects. These requirements are classified into three categories: general, test-based and trust-based requirements. General requirements consider the general functionality of the framework. Test-based requirements consist of requirements which are necessary for functional testing of decentralised M2M application services created and provided by end-users. Trust-based requirements are requirements used for building an appropriate trust model or trust management system which could evaluate the behaviour of the services or end-users and share the trust values among the nodes.

TABLE I. Evaluation of Test and Trust Platforms

Requirements		Testing Approaches			Trust Management Approaches							
		[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]
General	Framework flexibility	-	-	-	-	-	-	-	-	-	-	-
	Decentralisation capability	o	-	-	+	+	o	o	-	-	o	+
	Trust-based service composition	-	-	-	+	-	-	-	-	-	-	-
	Test case generation	o	o	o	/	/	/	/	/	/	/	/
Test-based	Functional verification	+	+	+	/	/	/	/	/	/	/	/
	Decentralised and end-user-based M2M application services	-	-	-	/	/	/	/	/	/	/	/
	End-user test integration	+	-	-	/	/	/	/	/	/	/	/
	Formal description of M2M application services	-	o	o	/	/	/	/	/	/	/	/
Trust-based	Trust evaluation	/	/	/	+	+	+	+	+	+	+	+
	Initial trust level	/	/	/	-	-	o	-	-	o	-	-
	Existing trust level	/	/	/	+	+	+	+	+	-	+	+
	Secure trust data storage	/	/	/	o	-	-	-	-	-	-	-
	Suitable trust metrics	/	/	/	-	-	-	-	-	-	o	o

The following notations are used to assess the satisfaction for the requirements: + satisfied; - not satisfied; o partially satisfied; / not available.

III. Trust Evaluation For new M2M Application Services

Most relevant trust approaches in M2M/IoT and P2P do not consider trust evaluation for new M2M application services. However, it is required to provide trust information about a new M2M application service to other peers of the M2M community [1] in order to enable them to decide to either start an interaction based on the trust level with the service or to ignore it. The literature provides a low number of publications dealing with trust for new services in other domains or with different focus. This subsection presents some approaches for trust evaluation of new services in other application fields.

The authors in [15] propose a so-called "trust bootstrapping" process where services without any trust level are going to be rated. The steps for evaluating the initial trust level of a service starts with the service provider which publishes his service together with other information such as the provider ID, service ID, service functional properties, a list of trust metrics, minimum and maximum values of the metrics by using a Graphical User Interface (GUI). The service requester uses also the GUI in order to find services based on his own preferences and selected metrics. The trust behaviour of a new service is evaluated with monitoring, certification or feedback techniques and stored in the registry of the system [15]. However, this approach does not mention in detail how the monitoring process is going to be and if this is a central or decentralised process. Moreover, the author does not provide more information about the storage of the trust metrics in the registry of the system without defining that system. One benefit of the work presented in [15] lies in some of the trust metric parameters such as execution time and response time of a service which play an important role for evaluating the availability and the participation willingness of a service in the community. One major drawback is that the providers have to specify the trust metrics themselves and are therefore able to falsify these and the ranges. This could have a bad impact on the trust evaluation process by providing not reliable trust information for the service requester.

Another approach is presented in [16] which proposes trust bootstrapping for web services. According to possible characteristics of new web services, they employ three generic mechanisms presented in the following: the inheritance mechanisms where the new web service gets the trust score based on the trust score of its service provider, the referral mechanisms where the new web service gets the trust score based on the referrals from other communities and the guarantee mechanisms where the web service gets a temporary trust score under guarantee conditions. The approach presented in [16] does not provide any mechanisms for storing the generated trust data in a trustworthy way using the three introduced trust assigning mechanisms. Moreover, these mechanisms used in three different cases provide some limitations. In the first case using the inheritance mechanism, the service provider A, for instance, has six existing services which are trustful. This does not mean that a new service from service provider A is also or will also be trustful. The trust level of a service should not be derived from the trust level of existing services. Therefore, the inheritance mechanism is not sufficient for providing trust bootstrapping of new services. The referral and guarantee mechanism are also not efficient and secure because considering only the behaviour of the service in past communities without considering its initial behaviour is not enough for enabling a reliable trust level of services.

The authors in [17] introduce a bootstrapping technique where new web services are getting initial estimated reputation values based on their Quality of Service attributes and their similarities with services that have long feedback records constructed from collected user feedback ratings. However, the authors in [17] do not explain in detail the architecture of the system, whether it is centralised or decentralised, nor provide any information on how the service testing is done by the system. Moreover, the focus of that work is reputation and not trust, which have similar meaning but are not the same. Additionally, same as the work in [16], the authors in [17] also use the values of other services provided by the service provider for assigning the initial value for the new service. As mentioned in [16], this provides not an efficient and reliable way to determine the initial trust or reputation value of new coming services.

Computing the initial trust level of defence agents which are used to deal with modern distributed and collaborative network attacks is the focus in [18]. The authors in [18] emphasize that considering all defence agents as trustworthy from the beginning of the lifecycle is not very realistic and a trust bootstrapping process for assigning trust levels for defence agents is required. The presented trust model in [18] divides the defence agents into the following categories: management agent, evaluation agent and new coming agent. The new coming agent will go through the trust bootstrapping process in order to get an initial trust level. The management agent and the evaluation agent are considered to be trustworthy. The trust evaluation process starts with the classification of defence agent`s trust type which is related to the behaviour pattern of the agent and is identified by analysing its feedback. This process is done by the evaluation agent. The trust type can be assigned based on the behaviour and the response of the defence agent for receiving a service request [18]. The next step in the trust evaluation process is identifying constraints by contrasting the benefits and costs for performing defence task between two entities. This step also includes the benefit that the trust brings to the entities by calculating the gained trust utility. Afterwards, the next step is assigning the initial trust level of the new defence agent using the methods of assigning corresponding values and computation of weighted averages for the defence agents. Limitations of this work are that the authors do not consider the trust level of other defence agents such as the evaluation agent and also do not provide more information regarding the data storage and the architecture of the trust model. Moreover, there are missing information on how the evaluation agents assign the initial trust level for new defence agent and how the assigning of corresponding values and weighted averages is done.

The several approaches in this section do not provide a full independent and decentralised trust management system for assigning or evaluating the initial trust level for new M2M application services and service providers. Their focus is not on the M2M domain or on decentralised M2M application services provided by the end-user. Moreover, they do not provide the possibility to test the functionality of new entering application services and do not consider any possibility to store the gained values in a trustworthy way. Additionally, most of the approaches rely on recommendations or on trust values from similar application services which provide no reliable source for trust evaluation. Another limitation is that they do not consider any technique or solution for generating test cases for evaluating the initial behaviour of new services.

IV. Framework for Functional Verification and Trust Evaluation of M2M Application Services

In order to test the functionality of new provided M2M application services, the authors in [2] propose an approach by introducing a test architecture consisting of a Test Master, Test Agents, and a Test Generation Environment. The Test Master coordinates the overall testing framework by sending and exchanging information with the TGE and the Test Agents. Furthermore, the Test Master gets test instructions from the TGE and forwards them to the Test Agents for test execution on the System under Test (SUT) which in this research are M2M application services. The obtained results of the test execution from the Test Agents are then evaluated by the Test Master. The TGE collects information about the M2M application services and derives based on that information suitable test cases which are then sent as test instructions to the Test Master. For evaluating the initial trust level of new provided M2M application services the author in [3] propose to integrate the trust evaluation process within the functional testing process by using the test architecture and the outcomes of the test execution for evaluating the trust level. However, the approach presented in [2] and [3] contains centralised elements such as the Test Master, which represents a drawback regarding single point of failure or centralised management about the test and trust reports. This research paper proposes an optimisation of the overall framework by distributing the role of the Test Master among other peers/end-users part of the M2M community, which will autonomously do the test execution and the evaluation of the obtained test results. First of all, the service provider designs a M2M application service logic using the GUI which is part of the SCE as described in [1]. Therefore the end-user graphically (see Fig. 1) creates a state machine that represents the behaviour of the system. The SCE generates from this logic a formal Service Description and deploys the M2M application service to other users by providing the Service Interface Description of the M2M application service. The Service Description containing also the Service Interface Description is sent to and stored in the P2P network.

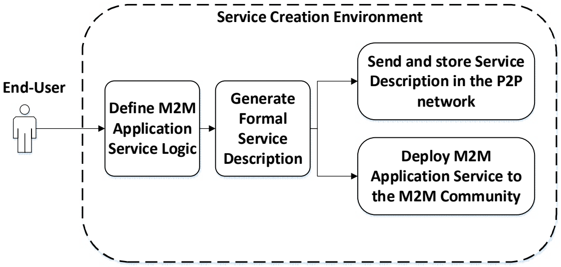
Service Creation Environment
End-User
Define M2M Application Service Logic
Generate Formal Service Description
Send and store Service Description in the P2P network
Deploy M2M Application Service to the M2M Community

Fig. 1: Service Creation Environment

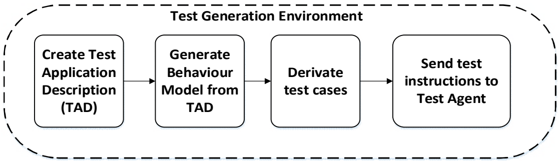
Test Generation Environment
Create Test Application Description (TAD)
Generate Behaviour Model from TAD
Derivate test cases
Send test instructions to Test Agent

Fig. 2: Test Generation Environment

After the deployment of a new M2M application service all peers/end-users will receive a notification about the new joining M2M application service and are also able to pick up the so-called Application Description Package stored in the P2P network, which contains the Service Description and the Service Interface Description of the new M2M application service. These information are used from the TGE (see Fig. 2) to create a Test Application Description (TAD) [2]. The TAD is used to generate a behaviour model from which test cases for functional verification and trust evaluation of new M2M application services are derived. In the proposed optimised approach, the test execution (see Fig. 3) can be done independently by one ore many peers/end-users acting as Test Agents. The obtained tests results are evaluated and an initial trust level [3] for the new M2M application service is assigned and stored among all other peers in the P2P network. To ensure secure and trustworthy data storage, the authors in [22] propose to store all the trust data in a blockchain. In order to verify and evaluate new M2M application services all participating end-users in the test process are honoured for their contribution by the community with credits, which can be used by them for consuming available M2M application services.

The benefit of this approach is that a new M2M application service could be evaluated by many end-users independently and the different test results obtained by the end-users can be combined to calculate an overall verdict about the new M2M application service. The calculation of the verdict considers also the trust level of the different end-users performing the tests. The total trust level of an end-user consists of the trust levels the M2M application services it provides. End-users with better trust levels are more weighted in the calculation process than end-users with low trust levels. Thus, this approach enables a distributed and efficient way to verify new joining M2M application services.

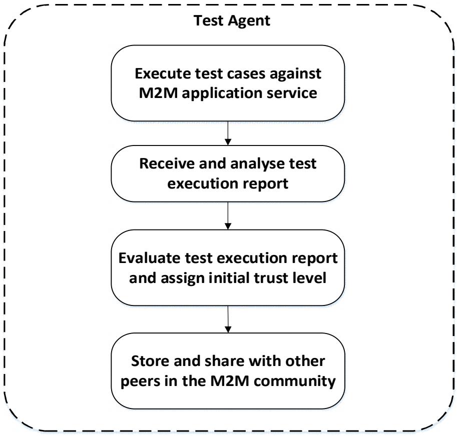
Test Agent
Execute test cases against M2M application service
Receive and analyse test execution report
Evaluate test execution report and assign initial trust level
Store and share with other peers in the M2M community

Fig. 3: Test Agent Activities

V. Combining Model-based Testing Techniques for Trust Evaluation of new M2M Application Services

There are several methodologies in testing but the focus of this research is on model-based testing. Several advantages described in [21], such as less errors in early stages of service development, automatic test case generation or the specification of the system behaviour make model-based testing a powerful technique. After a new M2M application service is provided by an end-user, it should be tested. Basically, functional testing is done against the new M2M application service. This test should ensure the verification of its functionality and considers the SUT as a black-box by analysing only the input and output values. Testing the functionality of a M2M application service provides a first impression about its behaviour. The functional testing process concludes with the result if the M2M application service is behaving like it is mentioned in its system model. For example, after an end-user provides a new M2M application service to the community the test system will analyse the system model of the M2M application service and will derive test cases that are going to be executed against the M2M application service. If the M2M application service behaves like it was described, the test will pass and the M2M application service can be considered as initially trustworthy. To sum up, the evaluation of this behaviour could build a first initial trust level of the M2M application service.

Security testing is also an important aspect for testing system requirements of a system related to predefined security properties. This kind of testing can be classified in security functional testing which validates the correct implementation of security features in the system and security vulnerability testing which tries to identify unintended system vulnerabilities [19]. The focus of this research is on decentralised M2M application services [1] which are created by end-users with basic or no technical background. Therefore, it can be supposed that decentralised M2M application services do not contain security features which have to be verified using security functional testing. The second category of security testing is security vulnerability testing which could be interesting for this research because unintended vulnerabilities can be happen in end-user provided M2M application services. However, the authors in [19] state that security vulnerability testing "requires more specific testing techniques" by defining and evaluating several attacks manually. This is also not completely in line with the aim of this research which is to provide an automated and end-user friendly framework considering the decentralised and distributed architecture of the end-user and the M2M application services. The authors in [20] describe several activities which are part of security testing, such as risk assessment and risk-based security testing, functional testing of security features, performance testing, robustness testing, and penetration testing. Most of these activities focus on testing the security attacks or their impact on the system under test whereas performance testing verifies that the system under test "can tolerate required constant load of service requests […] and will have adequate response time for valid requests even while under load-based attacks" [20]. Moreover, performance testing aims to find the performance drawbacks of the system and provides the possibility to identify the stress level "that will result in denial of service" [20]. This leads to the conclusion that good performance results of new M2M application services are related to trustworthy behaviours and provide the possibility to identify the willingness level of a service to participate in interactions with others.

In order to assess the trust level for new provided distributed M2M application services, this research proposes to combine the results of functional and performance testing. As mentioned above, the functional testing step is accomplished using model-based functional testing where, based on the system model, adequate test cases are generated and used for test execution. For performance testing a model-based approach can also be used by building so-called performance models from system component and their interactions. After the end-user creates the new M2M application service, the tester will verify the correct functionality of it. Moreover, the tester will do performance testing in order to confirm the participation willingness of the M2M application service. Then these results are combined and calculated to finally obtain the final verdict. For example, the M2M application service will successfully pass the functionality test and will also respond positively to a predefined amount of requests using performance testing. This gives a first trust overview about the initial behaviour of the M2M application service and can be used for further trust evaluation process of existing M2M application services.

To sum up, this section proposes to reuse model-based functional testing and model-based performance testing by combining them for trust evaluation processes for new provided M2M application services. These techniques are used to verify the initial behaviour of the M2M application service under specific conditions.

VI. Trust-based Selection of M2M Application Services

During the M2M application service composition process described in [1] the end-user configures and selects different single application services based on his own interest in order to create a composed application service in form of a service chain (Fig. 4). However, the community consists of several same or different M2M application services. Multiple end-users can offer different instances of the same M2M application service. The random selection [1] of instances providing one of the M2M application services part of the created service chain is not secure and could lead to selecting M2M application services provided by unsecure or trustless peers. This could result to an unstable and not efficient composed M2M application service. Therefore, this research proposes to consider the trust level of M2M application services and end-users for the application service selection and composition process. The peer who is the first in the service chain, in this case the end-user configuring the application

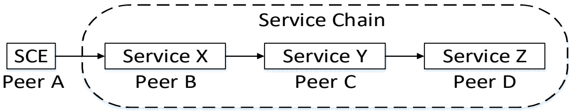

Fig. 4: Service Chain of composed M2M Application Service

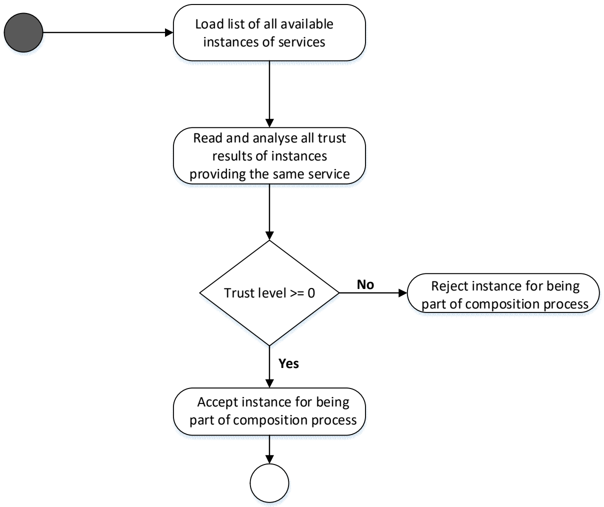
Load list of all available instances of services
Read and analyse all trust results of instances providing the same service
Trust level >= 0
No
Reject instance for being part of composition process
Yes
Accept instance for being part of composition process

Fig. 5: Algorithm for Selecting trustworthy M2M Application Services

(using the SCE), loads a list of all available instances of services from the P2P network. Then, the peer identifies all instances providing the first service in the service chain. Afterwards, the trust results of all instances providing this same service are analysed and based on the trust level ("-1" bad, "0" average, "+1" good) the instances are selected or withdrawn for being part of the composition process. The trust level for new M2M application services is computed using the approach presented in section V by combining model-based functional and performance testing. The steps described in Fig. 5 are continuously performed by every next peer part of the service chain in order to build a trustfully composed M2M application service at the end.

VII. Conclusion

Nowadays, the decentralisation of the market enables the deployment of a huge amount of new M2M application services, where end-users are also involved in the application creation process by modelling the application or providing the resources. The wide range of M2M application services renders difficulties to consumers to select the right services. Moreover, there is no prior knowledge nor are there data about the behaviour of the new service. This often leads to unsatisfied consumers who select not well-functioning or trustless services. Therefore, this research paper proposes an optimised approach for functional verification and trust evaluation of new M2M application services. Moreover, it introduces a novel trust evaluation mechanism by combining different model-based techniques and proposes an algorithm which integrates the trust evaluation results in a trust-based selection and composition process for M2M application services.

Acknowledgements

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.

References

[1] M. Steinheimer, U. Trick, W. Fuhrmann and B. Ghita, "Autonomous decentralised M2M Application Service Provision", ITA 2017, IEEE Wrexham, UK, 2017

[2] B. Shala, P. Wacht, U. Trick, A. Lehmann, B. Ghita and S. Shiaeles, "Framework for Automated Functional Testing of P2P-based M2M applications," 9th International Conference on Ubiquitous and Future Networks (ICUFN 2017), IEEE, Milan, Italy, 2017

[3] B. Shala, P. Wacht, U. Trick, A. Lehmann, B. Ghita and S. Shiaeles, "Trust Integration for Security Optimisation in P2P-based M2M Applications," TrustCom 17, IEEE, Sydney, Australia, 2017

[4] A. Ahmad, F. Bouquet, E. Fourneret, F. Le Gall, B. Legeard, "Model-Based Testing as a Service for IoT Platforms", ISoLA 2016, IEEE, 2016

[5] P. Wacht, U. Trick, „A Novel Test Creation Framework for Value-Added Services", SoftCOM 2016, IEEE, Split, Croatia, 2016

[6] E. Reetz, D. Kuemper, , R. Tönjes, A. Lehmann, "Test Driven Life Cycle Management for Internet of Things based Services: a Semantic Approach", VALID 2012: The Fourth International Conference on Advances in System Testing and Validation Lifecycle, 2012

[7] I. Chen, J. Guo, F. Bao, „Trust Management for SOA-Based IoT and Ist Application to Service Composition", IEEE Transactions on Services Computing, vol.9, no.3, 2016

[8] C. V. L. Mendoza, J. H. Kleinschmidt, „Mitigating On-Off attacks in the Internet of Things using a distributed trust management scheme", International Journal of Distributed Sensor Networks, Hindawi Publishing Corporation, 2015

[9] S. Asiri, A. Miri, "An IoT Trust and Reputation Model Based on Recommender Systems", PST 2016, IEEE, 2017

[10] H. Benkerrou, S. Heddad, M. Omar, "Credit and Honesty-based Trust Assessment for Hierarchical Collaborative IoT Systems", SETIT 2016, IEEE, 2016

[11] Y. B. Saied, A. Oliverau, D. Zeghlache, M. Laurent, „Trust management system design for the Internet of Things: A context-aware and multi-service approach", Elsevier Journal, Computer & Security, 2013

[12] T. Nguyen, D. Hoang, D. Nguyen, A. Seneviratne , „Initial Trust Establishment for Personal Space IoT Systems", INFOCOM WKSHPS: MobiSec 2017, IEEE, 2017

[13] Y. Ma, D. Wang, „A Novel Trust Model for P2P Networks", ICNC-FSKD 2016, IEEE, 2016

[14] S. Nakahira, S. Nakamura, T. Enokido, M. Takizawa, "Trustworthiness in Peer-to-Peer Systems", 18th International Conference on Network-Based Information Systems, IEEE, 2015

[15] Z. Aljazzaf, M. Capretz, M. Perry, „Trust Bootstrapping Services and Service Providers", Ninth Annual International Conference on Privacy, Security and Trust, IEEE, 2011

[16] H. Nguyen, J. Yan, W. Zhao, "Bootstrapping Trust and Reputation for Web Services", International Conference on Commerce and Enterprise Computing, IEEE, 2012

[17] O. Tibermacine, C. Tibermacine, F. Cherif, "Regression-Based Bootstrapping of Web Service Reputation Measurement", International Conference on Web Services, IEEE, 2015

[18] Y. Yang, X. Chunhe, L. Shiying, L. Zhong, "Trust Type Based Trust Bootstrapping Model of Computer Network Collaborative Defence", IEEE, 2015

[19] M. Felderer, P. Zech, R. Breu, M. Büchler, A. Pretschner, „Model-based security testing: a taxonomy and systematic classification", Software Testing, Verification and Reliability, published online in Wiley Online Library, DOI: 10.1002/stvr. 1580, 2015

[20] ETSI TR 101 583, V1.1.1 (2015), "Methods for Testing and Specification (MTS); Security Testing; Basic Terminology"

[21] M. Winter, T. Roßner, C. Brandes, H. Götz, „Basiswissen Modellbasierter Test", dpunkt Verlag Heidelberg, Germany, ISBN: 978-3-86490-297-0

[22] B. Shala, P. Wacht, U. Trick, A. Lehmann, B. Ghita, S. Shiaeles, "Ensuring Trustworthiness for P2P-based M2M Applications", ITA 2017, IEEE, Wrexham, UK, 2017