Ensuring Trustworthiness for P2P-based M2M Applications

Besfort Shala, Patrick Wacht, 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, UK

Abstract—**P2P-based M2M application frameworks have several advantages, such as increased flexibility, efficiency and a lack of single point of failure as compared to centralised approaches. However, there are several security drawbacks which need to be addressed in order to provide the user a secure environment for the provision and usage of M2M applications. This publication presents different security issues inside P2P-based M2M application frameworks and evaluates P2P protocols based on security. In order to avoid different security attacks, the concept of trust and its importance are emphasized. Furthermore, a trust management system with special trust metric parameters is presented which considers the architecture of P2P-based M2M applications. Finally, blockchain principles are integrated for optimising the overall security in the system by improving data storage between peers, avoiding volatility of peers and ensuring correct working M2M applications.**

Keywords— M2M; P2P; Service and Application; Security; Trust

I. Introduction

Many different application fields, such as energy management, ambient assisted living, building surveillance, smart home, traffic management and electro mobility can be established by applying Machine-to-Machine (M2M) communications. The integration of intelligent and complex devices are realised using the provision of M2M services and applications. There are several centralised and decentralised approaches for application provision. Decentralised approaches appear more frequently as a result of their many advantageous in terms of resource constraints and fault tolerance. Previous publications [1, 2] have defined requirements and concepts which enable service and application provision in M2M without the use of a centralised authority.

A framework that realises service and application provisioning using P2P networking in M2M application field is defined in [2]. Peers have the ability to realise services, as well as applications. An application consists of one or more underlying services that are combined. Peers are represented by technical devices or humans who are networked using P2P mechanism without the use of central authorities. For avoiding legal restrictions, adjusting different interests among the peers, a social network of peers called M2M community is introduced in [2]. According to [4], 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, MQTT) based on SUBSCRIBE/ NOTIFY principle. The Service Management Framework (SMF) described in [5] is the main component for service and application provisioning in M2M based on [2] and gives every peer the possibility to create and configure M2M applications.

The concept presented in [2] lacks the evaluation of different security issues in P2P-based M2M environments and do not provide 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 [9]. This publication aims to define different security issues in P2P-based M2M application frameworks by evaluating security risks in P2P and M2M networks. In order to deal with the distributed nature of services and applications in M2M, the testing architecture presented in [3] is used for the security optimisation process of M2M services and applications. For ensuring security within the framework, the concept of trust and special trust metrics based on different evaluations are introduced. Furthermore, the security between the peers is enhanced through the integration of blockchain technology inside the presented security framework.

II. Security issues for P2P-based M2M applications

Fig. 1 shows the structure of P2P connected peers within an M2M community and the composition of three services (service 1, service 3 and service 5) to an application. Based on the decentralised M2M application creation process described in [3], the following general categorisation can be made for security considerations: 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 pointed out that there is a huge amount of publications dealing with IP networks and network security (e.g. [6] and [7]) which describe different vulnerabilities and several security solutions. However, the security for the IP Network layer in this research is out of scope. Several security issues in M2M communications are classified into three categories based on [8]: Physical attacks

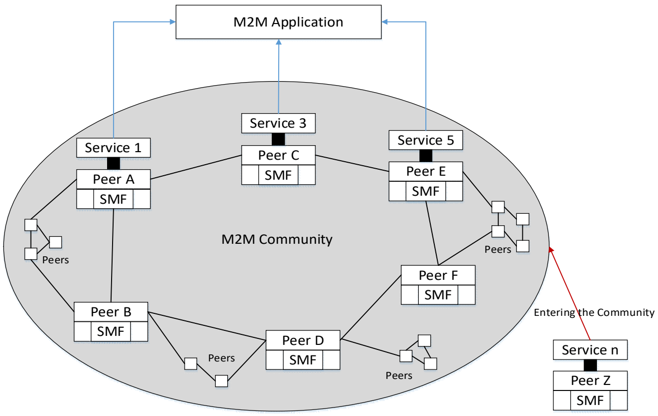

Fig. 1. M2M community with P2P-connected peers

include side channel attacks, software modification and malwares, destruction or theft of the M2M device. Logical attacks include impersonation, denial of service, relay attacks. Data attacks include privacy attacks, data modification and false information injection, selective forwarding/ interception. Furthermore, an overview of the current state of security in sensor and ad-hoc networking for M2M communications is provided in [9]. Exemplary for the application field of smart homes, a landscape of threats assumed for smart home assets is provided in [10]. The author in [8] states that M2M communications have to deal with all security issues of other network-based communications and [11] provides some security mechanisms including detecting the node compromise attack, lode location identifiers, two-way authentication and dual system.

In contrast to server/ client systems, all peers have the equivalent authority and responsibility in P2P architectures. 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. Security threats in P2P networks can be classified based on [12] in: eavesdropping, communication jamming, injection and modification of data, unauthorised access, repudiation, man-in-the-middle attack and sybil attack. Furthermore, based on self-assessment, previously researches and publications [13-23] the most relevant security attacks and their impact on different P2P protocols were derived and shown in Table I. The evaluation shows that most of the different P2P protocols are not secure against several security attacks and do not provide an efficient protection mechanism. Furthermore Table I shows, 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. The evaluation made in Table I concludes that the P2P protocol Freenet [22] mitigates best the different security attacks in comparison with other evaluated protocols and should be considered for further investigations.

TABLE I. P2P OVERLAY SECURITY COMPARISON

Security issues	P2P Protocols								
	Chord	CAN	Tapestry	Pastry	Gnutella 0.4	Freenet	JXTA	Fast Track	Gnutella 0.6
Incorrect lookup routing	-	o	o	o	N	o	o	o	o
Incorrect routing updates	o	o	+	+	N	o	o	o	o
Incorrect forwarding	-	o	o	o	-	o	o	o	-
Sybil attacks	o	-	o	o	-	o	o	o	-
Eclipse attacks	o	-	o	o	-	o	o	o	-
Query hit attacks	N	N	N	N	-	N	o	o	o
Man in the middle attacks	-	o	o	o	o	o	o	o	o
Denial of Service attacks	-	-	o	-	-	o	o	o	o
Multiple joins and leaves	o	-	o	o	N	N	-	o	+
Invalid splits	N	-	N	N	N	N	N	N	N
Assigning node IDs attack	o	-	o	o	N	N	o	N	N
Bootstrap attacks	-	-	-	o	-	o	o	o	o
Virus injection	N	N	N	N	-	N	N	N	N
IP harvesting	N	N	N	N	-	o	N	N	N
Privacy	N	N	N	N	o	+	o	-	-
Anonymity	-	-	-	-	o	+	-	-	-

The following notations are used to assess the level of security: + high; - low; o medium; N not available.

Summarising the security issues for M2M and P2P networks described above and considering the distributed nature of the M2M service and application provision, two main categories of problems related to security can be identified: 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 trying with a bad behaviour 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. Peers and the services they use or provide play the most significant role for application provision. Based on [2], 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 [8] are executed by malicious peers and this is why the focus for ensuring security inside the M2M community should be on peers. In order to deal with the complexity and unique characteristics of the decentralised M2M application process described in [4] and its security issues, the concept of trust has been introduced in the following chapter.

III. Trust in P2P-based M2M Communities

Attacks on the P2P layer can have a significant impact for the correct functionality of the whole system. Based on [24] 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 [25]. As stated in [25], 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 [26], 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 publication, the focus is on evaluating the trust level of peers which provide a service and the services themselves. 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. Trust metric is defined in [26] as "a measure to evaluate a level of trust by which a human or an object can be judged or decided from trustworthiness". Furthermore, the concept of trust model is defined in [26] as "a method to specify, build, evaluate and ensure trust relationships among entities". Based on defined requirements the collected data has to be analysed and evaluated by the trust decision process – process for setting up the level of trust for the tested element.

There are several publications dealing with trust management systems in the M2M, Internet of Things (IoT) and P2P domain. A summary of the most relevant publications is presented as follows. The authors in [27] propose a distributed trust management system in combination with reinforcement learning for using in mobile M2M communications by utilising the history of node´s interactions to build trust among other nodes. This approach performs good results in terms of execution time and energy consumption. For evaluating trust between nodes [28] introduces a trust management model which uses information generated from direct communication with the node and allows nodes to be completely autonomous in the decision-making about the behaviour of other nodes in the IoT domain. A trust management scheme is presented in [29] where trust is evaluated based on both direct user satisfaction experiences of past interaction experiences and recommendations from others considering social relationships. A fuzzy-based approach for ensuring trustworthiness in IoT is proposed in [30] who defines network related trust metrics and also considers the energy consumptions for trust evaluation. The drawback of the approaches described above is that they do not consider the initial trust level of a peer and relay at the beginning on predefined trusted existing peers in the system. A different approach is provided by [31] who proposes a centralised trust management system with different trust management servers covering different geographical locations for trust computation. The problem of this solution is the single point of failure of the centralised system and the low level of usability in large scale systems. In [32] a reputation-based trust supporting framework, named PeerTrust, is introduced which includes a coherent adaptive trust model for quantifying and comparing the trustworthiness of peers based on a transaction-based feedback system. The disadvantage of this approach is that it does not consider the special characteristics of M2M systems and that it focuses only in P2P social communities, such as online markets. The main drawback of the trust management systems described in [27-32] is that they do not consider the different services a peer can provide and the trust level of the composed M2M application. Moreover, they do not consider the trustworthiness of collected trust data of each peer.

IV. Integration of a Trust Management System

The aim of this research is to present a trust management system (TMS) which secures the trust evaluation process considering the distributed nature of the peers providing or consuming services and applications as described in the introduction. The huge amount of data collected between the peers 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 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 faster trustworthiness among other peers and services. The trust management system needs to ensure **flexibility** by considering the heterogeneity of peers and services. An important requirement is also the **volatility of peers and services.** In a P2P-based community, where a huge amount of peers are connected with each other without central authority, 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 trust management 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 trust management system 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 trust management system has to provide a mechanism for securing trust data storage and to ensure with that the **trustworthiness of trust data.**

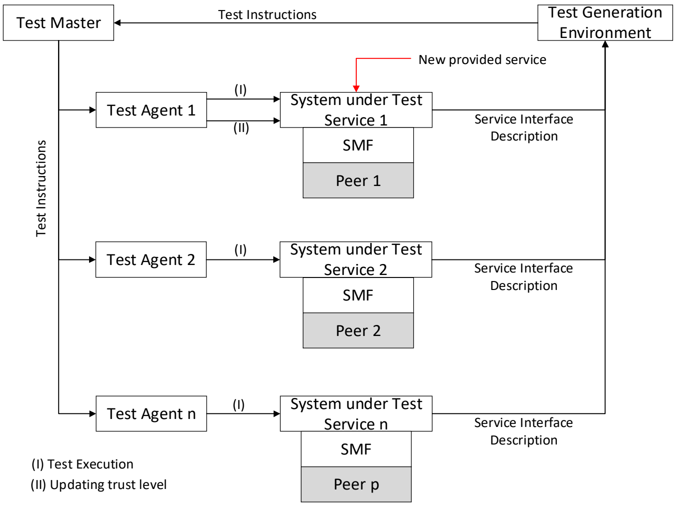
Test Master
Test Instructions
Test Generation Environment
New provided service
Test Agent 1
(I)
(II)
System under Test Service 1
SMF
Peer 1
Service Interface Description
Test Instructions
Test Agent 2
(I)
System under Test Service 2
SMF
Peer 2
Service Interface Description
Test Agent n
(I)
System under Test Service n
SMF
Peer p
Service Interface Description
(I) Test Execution
(II) Updating trust level

Fig. 2. Trust evaluation for new provided service using the Test Architecture

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 we consider that the total trust level of a peer consists of the trust levels the services it provides. For that reason, we focus on trust evaluation based on services. The architecture testing framework described in [3] 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, two cases are defined in this publication. The first one is evaluating trust of a newly provided service. The second case deals with the trust evaluation of existing services.

The author in [3] presents a concept for automated testing of decentralised services and applications in M2M. Also in [3] a testing framework with a special testing architecture for functional testing is introduced. The testing architecture is based on a global tester, called Test Master, and distributed testers, called Test Agents. A Test Generation Environment (TGE) is also embedded in order to generate test cases based on functional description of the System under Test (SUT). SUT can be all services which are part of the community and the composed application. This publication propose to use the test architecture presented in [3] for computing and evaluating trust levels for new provided services. The generated test cases for the service will be executed by the test agents for every service and the outgoing test report will be sent to the Test Master. From this report the Test Master is able based on defined criteria to derive the initial trust level and to verify whether or not the service entered the community is trustworthy. The evaluated trust level information will be sent to the SMF of each peer and used for trust related issues after the service is part of the community. Fig. 2 shows the integration of the TMS inside the testing framework which is used for testing P2P-based services and applications and the above presented workflow of computing trust for an entering service.

Trust computation and evaluation for existing services and applications is made with a completely different approach. For this case, an autonomous decentralised TMS is introduced and trust is evaluated based on the behaviour of each service using trust agents which will be automatically assigned to new entering peers. The procedure for evaluating trust for existing services can be described with an example where two services have interactions with each other and the trust data collection/ evaluation is made by trust agents. They will measure different trust metric parameters in order to compute trust. The services will communicate with each other using SIP SUBSCRIBE and NOTIFY messages as introduced in [2]. In Fig. 3 Service 1 is trying to subscribe service 2 by sending a SUBSCRIBE message. During their life time the activities of two services are monitored based on the defined trust metric parameters. After receiving the SUBSCRIBE-message, service 2 will ask trust agent 2 about the trust level of service 1. Trust agents are connected P2P and evaluated data are stored in the P2P data stores. Trust agent 2 will inform service 2 about the trust level of service 1 and based on that value service 2 will decide to accept or not to accept a session with service 1.

There are different trust metric parameters defined in [27-32] which are not completely suitable or enough for our TMS because they are used in different scenarios which do not consider the special application composition nature of the P2P-based M2M applications. Furthermore, they are not applicable for evaluating trust of an entering service. This publication defines different trust metric parameters for each of the two cases described above. The author in [33] identifies three perspectives of metrics, such as network performance metrics, knowledge quality metrics and accuracy of detection metrics. In order to compute the trust level of M2M services and applications, we identified a fourth metric perspective, namely, the service availability metric. For a newly provided service, we defined the corresponding metric parameters: Trust based on the functionality of the service - the service description corresponds to the service functionality. DoS attacks - the reaction of the service against a huge number of service requests. This attack gives the opportunity to figure out the robustness of the service and its willingness to accept service requests. Based on its performance against DoS attacks, part of the initial trust level can be derived. For computing trust based on the availability of the services, the following metric parameters for existing services are defined: Number of attendance of a service in various applications. Time a service stays online and time a service stays offline [34]. Number of online/ offline actions. Number of execution times. Number of subscribe-messages and accepted-/ not accepted-messages.

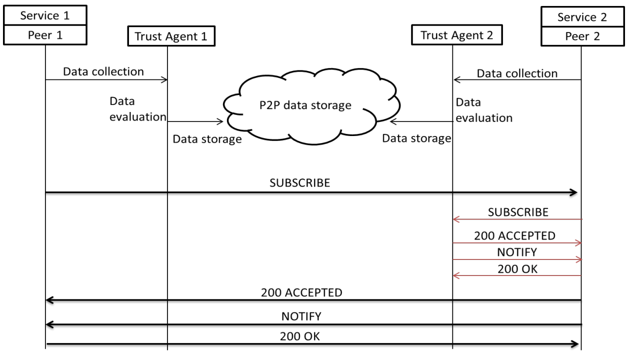

Fig. 3. Example of trust computation for existing services

V. Security Optimisation using Blockchain

A trusted third party represents a single point of failure and is also vulnerable to hacking and manipulating. To overcome this, Bitcoin [35] uses a cryptographic proof instead of trust for executing transactions between peers. Originally introduced for the crypto-currency Bitcoin, blockchain is a P2P distributed database which records all transactions, agreements, contracts and/ or other digital assets between peers participating in that community. In Bitcoin, blockchain is used for storing all transactions in the network. According to [36], trust is established due to the fact that everyone in Blockchain has a direct access to a shared "single source of truth". All transactions, which are public, comprise specific information, such as date, time, and number of participants. Every peer in the network has a copy of the blockchain and the transactions are validated by the so-called miners using cryptographic principles. These enable nodes to automatically recognize the current state of the ledger and every transaction in it [36]. As stated in [36], a corrupted transaction will be immediately refused by the nodes since they do not reach a consensus for validating that transaction. The author in [37] states that "once a new block is formed, any changes to a previous block would result in different hashcode and would thus be immediately visible to all participants in the blockchain". Based on [38], in order to attack the blockchain network, it is required to compromise more than 50% of the system which is very hard to achieve for malicious peers. These advantages make the use of blockchain for securing P2P-based M2M applications very attractive. Blockchain principles can be used besides bitcoin in various application fields and in combination with different technologies. In order to overcome the different security risks for P2P-based M2M applications and for enabling trustworthiness of the presented trust management system, this publication propose to reuse the principles of blockchain and to integrate them for different aspects of P2P-based M2M application frameworks.

Smart contracts are often used in conjunction with the blockchain technology to enable reliable relationships or application agreements between peers. They are first introduced in [39] and defined as transaction protocols that implement and execute terms of a contract in form of a script. Details of the contract can be stored in a blockchain address and after triggering that address with a transaction, the smart contract will be executed automatically in every node part of the contract according to the previously arranged agreement [40]. Smart contracts can be implemented in Ethereum [41] which is a blockchain based platform used for building decentralized applications. This publication proposes to use smart contracts for setting up rules for entering or leaving the M2M community within the context of P2P-based M2M application. Using smart contracts the volatility of peers and services is controlled and increases the high availability of services and applications. Furthermore, smart contracts can be implemented for ensuring the correct functionality of an M2M application which is composed by different services. Thus, they disable a possible misbehaviour of the composed M2M application by respecting the predefined rules using smart contracts.

Blockchain can also be used for building a secure decentralized storage network. One example is Storj [42], a P2P cloud storage network which uses blockchain features like a transaction ledger, public/ private key encryption and cryptographic hash functions for security. Storj is built on a Distributed Hash Tables (DHT) and enables peers to negotiate contracts, transfer data, verify the integrity and availability of remote data, and pay other nodes [42]. Every user has the possibility to rent storage from a storage provider called farmer. The encrypted file which needs to be stored in the storage is split into many shards and transmitted to the network. The file owner generates challenges for all the shards of the file and verifies periodically the farmers by requesting a special proof of work related to the selected challenge. In P2P-based M2M applications the blockchain features can be utilised for storing trust related information generated for each service by the trust agents. Trust agents will evaluate the trust levels of the services and will send this information to the secure blockchain network. Other trust agents are then able to retrieve this information for providing the requesting peers with trust information about others. This enables avoiding the usage of trust third parties for verifying peers trustworthiness. No one has the possibility to change past transactions which contains the trust information about a service. Additionally, blockchain features can be incorporated to overcome different security attacks described in TABLE I by combining them with P2P protocols, such as Chord and Pastry, for secure and reliable data storage.

VI. Conclusion

P2P-based M2M applications are exposed to a large number of various security attacks such as identity, bootstrapping or Denial of Service attack. In order to benefit from the many advantages of distributed application provision, it is crucial to provide a secure environment for all peers participating in the M2M community. This publication presents an overview of different security risks for P2P-based M2M applications and evaluates the level of security for P2P protocols against different attacks. Furthermore, the concept of trust in the context of an environment without centralised entity is introduced and a novel trust management system for ensuring the trustworthiness between the peers is presented. This system considers the initial trust level of services and enables trustworthy application provision in M2M by decreasing the risks of security attacks. Additionally, the security level of the overall system is optimised by the integration of blockchain.

The next step will be to describe and formalise a specific trust model for the proposed trust management system and to assess several trust evaluation models. Furthermore, future works include the integration of a security policy in the M2M community and identifying relevant trust related security attacks. The P2P network for sharing trust information among the trust agents also needs to be developed. The incorporation of the blockchain technology in different aspects will be evaluated in more detail.

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] K. J. Lin, N. Reijers, Y. C. Wang, C. S. Shih, J. Y. Hsu, "Building Smart M2M Applications Using the WuKong Profile Framework", Proceedings of the 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, pp. 1175-1180, IEEE, 2013

[2] M. Steinheimer, U. Trick, W. Fuhrmann and B. Ghita, "P2P-based community concept for M2M Applications", FGCT 2013, London, UK, December 2013

[3] 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), in press, 2017

[4] M. Steinheimer, U. Trick, B. Ghita and W. Fuhrmann, "Decentralised System Architecture for autonomous and cooperative M2M Application Service Provision", International Conference on Smart Grid and Smart Cities (ICSGSC), in press, 2017

[5] 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

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

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

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

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

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

[11] X. Nie and X. Zhai, "M2M Security Threat and Security Mechanism Research", 3rd International Conference on Computer Science and Network Technology, 2013

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

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

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

[15] E. Keong Lua, J. Crowcroft, M. Pias, R. Sharma and S. Lim, "A Survey and Comparison of Peer-to-Peer Overlay Network Schemes," IEEE Communications Surveys & Tutorials. vol. 7,no. 2, pp. 72.93, 2005

[16] 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

[17] 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

[18] G. Ciaccio, "Recipient Anonymity in a Structured Overlay", AICT-ICIW`06, IEEE, 2006

[19] 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

[20] 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

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

https://courses.ece.ubc.ca/cpen442/previous_years/2007_1_spring/modules/term_project/reports/2007/gnutella_security.pdf [23 June 2016]

[22] 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

[23] J. Liang, R. Kumar, K. Ross, "The FastTrack overlay: A measurement study", Computer Netzworks, 50(6):842-858, 2006.

[24] 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

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

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

[27] F. Boustanifar and Z. Movahedi, "A Trust-Based Offloading for Mobile M2M Communications", Ubiquitous Intelligence & Computing, IEEE, 2016

[28] C. V. L. Mendoza and J. H. Kleinschmidt, "Mitigating On-Off attacks in the Internet of Things using a distributed trust management scheme", International Journal of Distributed Sensor Networks, 11 (11), 2015

[29] R. Chen, J. Guo and F. Bao, "Trust Management for SOA-Based IoT and Its Application to Service Composition", IEEE Transactions on Services Computing, Volume: 9, Issue: 3, 2016

[30] D. Chen, G. Chang, D. Sun, J. Li, J. Jia and X. Wang, "TRMIoT: a trust management model based on fuzzy reputation for internet of things", Computer Science and Information Systems, vol. 8, no. 4, 2011

[31] Y. B. Saied, A. Olivereau, D. Zeghlache, M. Laurent, "Trust management system design for the Internet of Things: A context-aware and multi-service approach", Computers & Security, 39, 351-365, 2013

[32] L. Xiong and L. Liu, "PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communieties", IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 7, 2004

[33] Z. Movahedi, Z. Hosseini, F. Bayan and G. Pujolle, "Trust-Distortion Resistant Trust Management Frameworks on Mobile Ad Hoc Networks: A Survey", IEEE Communications Surveys & Tutorials, vol.18, no.2, 2016

[34] ITU-T, "Security requirements and mechanisms of peer-to-peer-based telecommunication networks", X.1163, 2015

S. Nakamota, "Bitcoin: A Peer-to-Peer Electronic Cash System", 2008, available from: https://bitcoin.org/bitcoin.pdf [12 January 2017]

[35] S. Hashemi, F. Faghri and P. Rausch, "World of Empowered IoT Users", International Conference on Internet-of-Things Desing and Implementation, Berlin, Germany, 2016

[36] M. Samaniego and R. Deters, "Using Blockchain to push Software-Defined IoT Components onto Edge Hosts", BDAW´16, Blagoevgrad, Bulgaria, 2016

[37] K. Biswas and V. Muthukkumarasamy, "Securing Smart Cities Using Blockchain Technology", International Conference on High Performance Computing and Communications, Sydney, Australia, 2016

[38] N. Szabo, "Formalizing and Securing Relationship on Public Networks", 1997, available from:

http://journals.uic.edu/ojs/index.php/fm/article/view/548/469#1 [17 April 2017]

[39] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things", IEEE Access, 2016

[40] "A Next-Generation Smart Contract and Decentralized Application Platform", White Paper, available from:

https://github.com/ethereum/wiki/wiki/White-Paper [9 March 2017]

[41] S. Wilkinson, T. Boshevski, J. Brandoff, J. Prestwich, G. Hall, P. Gerbes, P. Hutchins and C. Pollard, "Storj – A Peer-to-Peer Cloud Storage Network", 2016, available from: https://storj.io/storj.pdf