IMPROVING THE DIGITAL SECURITY OF SMART ENERGY SYSTEMS WITH SMART CONTRACTS

Pekka Koskela, Jarno Salonen and Juha Pärssinen

VTT Technical Research Centre of Finland,
P.O. Box 1000, FI-02044 VTT, Finland

ABSTRACT

Smart grids are evolving towards intelligent electricity grid where the operation of systems is distributed and automatised. Technical solutions to achieve these future needs are proposed using blockchain with smart contracts in many studies, where smart contracts enhance automation. Fundamentally smart contracts will increase security because of their distributed nature and since it inherits the security of blockchain. However, smart contracts are software components, which have special features like the unstoppable nature of applications and may use special languages like Solidity. Our aim in this paper is to get a holistic review in the smart contract life cycle, what potential new vulnerabilities and threats will they introduce and how can they be prevented, and what smart contract specific issues programmers should focus on. We also propose a future direction to achieve more secure smart contracts in smart energy systems.

KEYWORDS

Blockchain, Smart contracts, Smart energy systems.

1. INTRODUCTION

As the energy industry develops towards smart and micro grids, where a large amount of energy producers and consumers interact with each other using heterogeneous devices. These new activities require new and economically viable solutions for trustworthy communication between the devices and trading of energy and storage capacity. A solution of the trading has been proposed to utilize blockchains and smart contracts [1],[2],[3],[4].

Smart contracts can help solve multidimensional problems related to achieving and securing the integrity and reliability of distributed, complex energy events and information exchange, as well as systems optimization. Blockchain-based smart contracts help eliminate the need for third parties to build mutual trust between different parties and enabling automation facilitating the deployment and commercialization of decentralized energy transactions and exchanges, both in terms of energy flows and financial transactions [5],[6].

Recently there have been many broad reviews concerning the use of blockchain in future smart grids analysing the applicability of blockchain technology [7] and identifying possible applications of blockchain in smart grid [8]. There have also been numerous studies covering the analysis of the security threats and vulnerabilities of smart contracts [9],[10],[11], some of which have been concerned with machine learning detection [12]. The vulnerabilities are strongly

David C. Wyld et al. (Eds): DMML, SEAS, ADCO, NLPI, SP, BDBS, CMCA, CSITEC - 2022
pp. 129-138, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120711

dependent on the construction and implementation of specific systems and thus depends on the case. For instance the public and private blockchains differ from each other in terms of permission to participate, access to the network and consensus methods. In the public blockchain everybody is permitted while in the private blockhain permission is granted only to specific user groups. Public blockchains such as Ethereum use proof of work (PoW) that is a probabilistic consensus algorithm, in the Ethereum virtual machine environment while private blockchains such as Hyper Ledger Fabric use Raft, which is a deterministic consensus algorithm, in the docker container environment. Many existing papers study Ethereum-based approaches, whereas in this paper we try to form a more holistic overview based on the smart contract life cycle. Our focus is more on what secure improvements and new threats the smart contracts will bring and estimate the risk level of threats and how to prevent them.

This article is organized as follows. In section 2 we describe smart contracts, their features and benefits. Section 3 provides an overview of past studies promoting smart contracts in smart energy systems. Section 4 presents what new threats are posed by smart contracts during their life cycle, including their potential damage and risk and generic means to prevent the aforementioned threats. In section 5 we discuss the future directions on how to achieve more secure smart contracts and finally, we conclude the article in section 6.

2. SMART CONTRACT

A smart contract is a program component attached to a blockchain. As such, the program component is not smart and may not even be a contract, but only code that performs some pre-programmed tasks. The smart contract related software code is connected to the blockchain, which guarantees the integrity, i.e., that the program code has not been altered after the connection. This provides code that the various parties can rely on when the code is executed. The basic features of the smart contract are the following:

- It is stored in a blockchain
- It can be checked by everyone and the operating logic and results are visible to everyone
- It can no longer be changed after it is in the blockchain
- Its operation “cannot” be blocked and the results “cannot” be modified
- It performs the functions assigned to it, programmed automatically and always in the same way, i.e. certain input values always produce the same output result.

In the cybersecurity environment, the smart contracts are connected to the blockchain and the blockchain is connected to cloud services as depicted in the figure below (fig1).

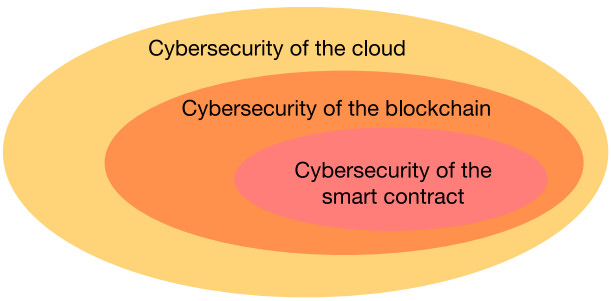

Figure 1. The cybersecurity environment of smart contract.

Because the smart contract is connected to a blockchain, its security is substantially affected by the security of the blockchain and how the blockchain implementation has been done. The following questions are important when determining the security and risk level of the smart contract and the blockchain ”platform”: What are the transaction consensus mechanisms? How decentralized is the blockchain, in other words, what is the number of “block approvers”? Is it a private or public blockchain? One of the basic principles of the blockchain and also the smart contract is that they operate in a decentralized manner, thus avoiding a single point vulnerability, which would exist in a centralized system. Therefore, an essential part of blockchains and thus also smart contracts for information security is how effectively the decentralization has been implemented. The benefits of using a smart contract linked to a blockchain include the following: Increased security based on decentralization and the inherited security of the blockchain, ability of building mutual trust between unknown parties, automation of contract-based functions like sales, outlets, approval chains, material, product and quality monitoring, and eliminating the need to use trusted third parties, see fig 2.

The security of blockchain is based on cryptography where every new block contains a cryptographic hash code generated from the block itself and the previous block. Thus if some malicious third party wanted to create a fake block, they would need to change both the hash code of the previous and the current block, which in practice means that all hash codes of the blockchain would need to be changed. This is difficult to do at a limited time before a new block is accepted and when the blockchain is long enough as it will be in a normal case.

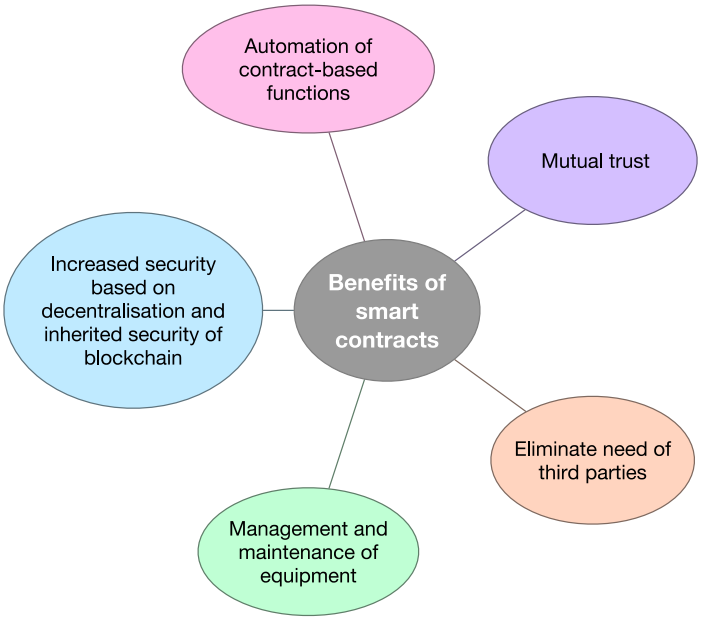

Figure 2. The benefits of smart contracts.

Mutual trust between participants is commonly based on the knowledge that information hasn’t been modified after it has been saved in the blockchain (integrity) and secure consensus mechanisms guaranteeing that the saved information is mutually accepted and any cheat attempt is detected and excluded.

Basically the smart contract is small program, like any other conventional program and can be programmed to consist of any functions like sales, web shops, approval chains, material, product and quality monitoring, which automatise the process. Because the smart contract is connected to

a blockchain, it cannot be modified after implementation and it will always return exactly the same output to a certain input. There will be no additional need of monitoring the results or acceptance by a third party.

We can use the blockchain to form a digital identity for persons, devices and software. When all actors of a system have a digital identity it will be easier to manage resources and devices as well as make maintenance operations in the security point of view. With smart contracts maintenance activities like software updates and resource provisioning can be done automatically.

3. USE OF SMART CONTRACTS

The exploitation of smart contracts and smart grid technologies in the energy sector has been extensively studied and several test environments have been made in connection with the subject [13],[14].

In the energy networks of the future, smart grids and micro-networks, one challenge is to create an economically viable and efficient trading place between the parties connected to the grid. Challenges include among others:

- How to build trust between different actors?
- How are privacy and security issues handled?
- How to make the system work as automatically as possible?
- How to make the system work as efficiently and economically as possible?
- How to optimize energy production, using and storing?

As a solution to the challenges, an online store based on blockchains and smart contracts has been proposed, where the sale and purchase of energy can be done automatically between the parties connected to the grid [1],[2],[3],[4]. We could also create a similar system which includes only the electricity producers and consumers and where sales are automated through smart contracts without a separate sales agency.

In the energy networks of the future, one challenge will be how to charge electric cars economically. A solution to the problem has been proposed to use smart contracts in order to reduce power variation and charging costs [15],[16]. In addition to selling and buying energy, the marketplace could sell storage capacity such as electric car batteries and household energy storage capacity [4],[17].

Similarly, automated mechanisms based on smart contracts can be created for the management and maintenance of equipment, which alert on equipment failures and allocate the cost of maintenance measures according to a service contract taking into account, for example, warranty periods, the cause of the fault and the time taken for maintenance [17].

Solutions based on smart contract security can increase the security and resilience and prevent cyberattacks on distributed network devices, network peripherals, and related infrastructure [6].

When secure digital identity and remote attestation support is formed based on blockchain technology [18] many operations and functions of a system can be automatised and it also allows the development of new services. These services can among others be warranty follow-up, covering following up the operation time, operation conditions and interruptions of a specific device and providing automatised maintenance services and compensations according to the smart contract. Furthermore when devices can be identified we can include automatic provisioning to the system, software updates, calibration, certificate verification and data source

identification. Based on blockchain technology the online store can be established for sale and purchase of energy, energy storage and devices, that uses smart contracts for automatised functionalities. Fig 3 presents the different possible functions which may be automatised with smart contracts.

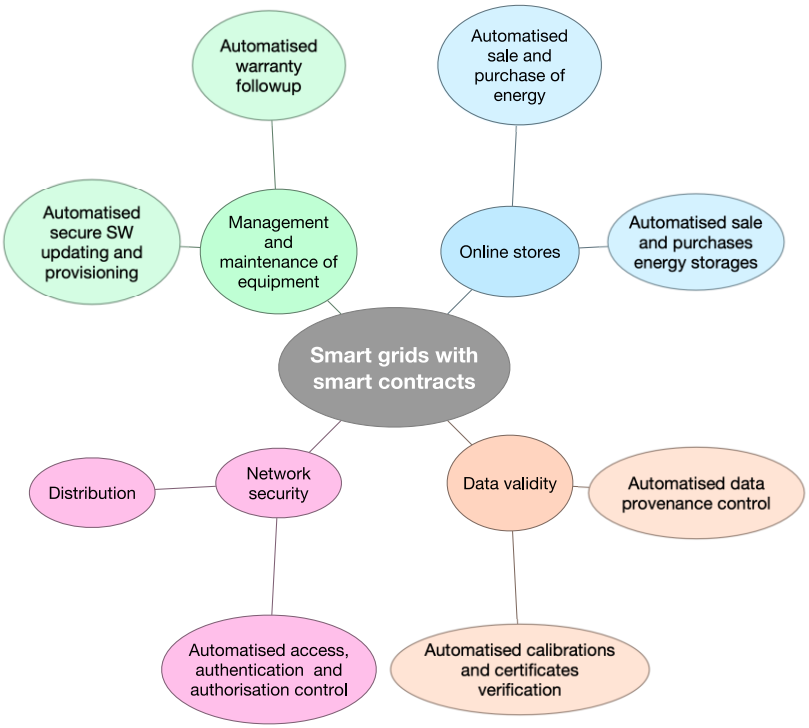

Figure 3. The exploitation of smart contract in the smart energy sector.

4. THREATS TO SMART CONTRACTS

Smart contracts are connected to blockchains that act as a cloud service on virtual machines. As a result, smart contracts face the same threats as cloud services and blockchains. In addition, the smart contracts themselves introduce their own threats to the system. This study does not address the threats to cloud services and blockchains, but only what additional threats are posed by smart contracts.

In fig 4 we present the life cycle of smart contracts which consist of creation, deployment, execution and termination. During the creation phase participants negotiate and make mutual agreement on the content of the smart contract. When the content is clear the smart contract can be designed, programmed, tested and verified. Because the programming work may be demanding due to strange programming languages and syntax, accurate testing is critical in terms of security and operation of smart contract. After verification the contract will attached and stored in the blockchain. During the execution of a smart contract there will be injected input values to the contract, which produce output values. If the contract is unbroken the same input values must provide always same output values. Finally there is a need to design and implement how to terminate the smart contract in a controlled manner, because the code will never disappear in a blockchain.

Figure 4. The life cycle of a smart contract, the creation-deployment-execution-termination.

During the life cycle of a smart contract, the creation-deployment-execution- termination process is subject to various threats. The realization of the smart contract threats is primarily influenced by the chosen blockchain platform such as the decision between private or public blockchain and its implementation, i.e., how well the code is tested. In terms of the extent of the damage, it largely affects what the smart contract does, e.g., how many users does the contract have, and how long the attack proceeds before the system can be repaired. Preventing smart contract related threats may include smart contract, blockchain, and data integrity. We will discuss more about the accuracy of different threats, their damages and risk and how to prevent them in the next chapters.

4.1. Creation

Threat: In the creation phase a programming error in a smart contract can be either intentional, accidental or due to a security bug in the programming environment [19],[20].

Damage and risk: The error can cause the smart contract to malfunction, enable a security breach, or cause operational disturbance of the blockchain. The error risk increases, when the source code is not available or it lacks a well-known language like Solidity which have familiar-like syntax, but in the semantics level there can be significant differences. Obviously the risk level of programming errors will depend on the competences of programmers and testing processes. In general we assumed the risk of errors to be between low and middle, where low risk represents using well known languages and middle risk not so well known ones. Some examples of attacks have been discussed in [21].

Prevention actions: Smart contracts should be carefully tested before commissioning and the first implementations should be done with a smaller test group. The idea is to use techniques and tips which are available [7],[8] and use programming languages/methods which support good human readable source code when they are available [9],[12]. The correctness of the contract can be tested with many tools [11],[22],[23] and methods [26]. Furthermore there are many tools for analysing the vulnerability of smart contracts [16],[22].

4.2. Deployment

Threats: During deployment the smart contract cannot be approved, which is due to attack, e.g., distributed denial-of-service (DDoS), against the blockchain. As a result deployment of the smart contract is not possible. Another possible threat is as a result of the attack against the blockchain, when the attacker modifies an existing smart contract or installs his own smart contract. As a result of the attack, a false or erroneous smart contract will be used. An example of these attacks is decentralised autonomous organisation (DAO)-attack, where an erroneous smart contract is used to steal crypto currency ETH [22]. A possible method to change the smart contract in the blockchain that is based on work of proof or same kind of consensus is, if attackers, i.e. poll, have enough computing power to solve the nonce fairly quickly, create two or more consecutive blocks and then propose and accept their own malicious smart contract into the blockchain.

Damage and risk: The first attack will prevent the start of activities related to the smart contract, like trading whereas in the second attack the smart contract doesn't work properly and will do incorrect things. The risk of attacks like DDoS depends among others on how distributed the implementation or how protected the network is. In general the public blockchain can be assumed as more distributed and therefore more protected than the private one. However, private blockchains have a better control over access and network operations and in that sense the private chain can be considered to be safer against DDoS attacks than the public one.

Generally we assume the risk of DAO attack to be similar to the risk of a programming error and it will depend on a case basis on what kind of a blockchain platform is used and what are the specific vulnerabilities of that platform. However, we generally assume private blockchains to be safer against DAO kind of attacks, because the operations in blockchain can be more restricted and authenticated, which is not the case in public blockchains. This is because public blockchains are managed by the community whereas private blockchains management is limited and more controlled by a pre-selected group and therefore the recovery from an attack will be faster and easier when compared to the public chain.

Because private blockchains can be more controlled and better monitored we assume that it may react to both the DDoS and DAO type of attacks faster than public blockchains, which means that the potential damage will be smaller.

Prevention actions: One should use sufficiently distributed and fast blockchain platforms to prevent DDoS type attacks . Private chains can use fast consensus algorithms and mechanisms and are therefore often faster than public ones. In order to prevent DAO type attacks, monitoring the operation and integrity of smart contracts should be implemented and the blockchain itself should include control programs [23]. Private blockchains have better access and more strict operational control, which is missing from public blockchains.

4.3. Execution

Threats: During the execution, slowing down the operation of a blockchain by an attack may result in a slowdown or total blocking of smart contracts. Another attack affecting the result of the smart contract may be initiated either by changing the initial input or output values which results to an incorrect result.

Damage: The first attack will prevent or slowdown activities related to the smart contract for instance in trading. The actual damages will depend on the tasks of the contract and how efficient the attack is. We assume the risk of this attack being low, but the risk will increase by poor design and/or implementation of the blockchain. In case of an attack manipulating either the input or output values of the smart contract, results into the smart contract not working properly and therefore producing wrong results. In this case the smart contract will be missing sufficient integrity of the input or output values. In general we argue that the case where integrity needs to be improved afterwards, the operation will be much easier in the private than the public chain, because the public blockchain forms a rather loose community which does not have any easy means to make any required updates at once.

Prevention actions: The efficiency of the DDoS attack will mainly depend on the distribution level of blockchain implementation and in general the public blockchain can be assumed more distributed and therefore more protected than the private chain. However, the efficiency of DDoS attacks will affect also the privacy of the operational pace of blockchain. In this case the private blockchain consensus algorithms and therefore the operations can be made faster than in the public chain. We argue that private blockchains which often are faster, have better control with

regards to the access and operation control than public chains. Therefore it can be assumed that private blockchains will be safer against DDoS attacks than public blockchains.

To prevent the integrity and data modification attacks, there is a need to monitor the contract activity, to have input and output data encryption and integrity control programs. Flow control and interaction of other smart contracts can be monitored among others using graph-based analysis and path-searching [24]. We argue that since private blockchains may form a more controlled environment, they can be considered to be safer and the responses to attacks are faster than in public chains and therefore the potential damage might have smaller influence.

4.4. Termination

Threat: Due to the attack against the blockchain or the programming error, the smart contract cannot be terminated at the desired time.

Damage: Influencing the expiry date of an intellectual property contract results into the contract activities being terminated too early or too late. We assume that in general there is a low risk of the termination threat, which depends on the termination process and its implementation.

Prevention actions: Influencing the expiry date of an intellectual property contract is planned in such a way that the termination process of the contract takes into account any possible attacks. We argue that in general detecting the attack and fixing any issues is faster in private chains than public ones, because the private chains have a more controlled environment.

5. FUTURE DIRECTIONS TO MORE SECURE SMART CONTRACTS

Based on previous chapter most of the attack sources are not platform issues but result from smart contract programming errors. One possible means to prevent any errors is to develop smart contracts using simple programming languages. which are often considered non-touring complete [26]. Another means to prevent errors is using integrated development environments, where the operations that are not vital or even harmful in the security point of view -like recursion - are removed or prevented. In case of the smart grid domain, we should conduct an extensive study on what kind platform(s) [26] and language(s) [26], [27] will be used and what tasks the smart contracts are intended to do. If smart contracts need operations like interconnection with other smart contracts or the use of random number generators, time stamps, which must be update identically to all distributed apps, we must study how this can be done in a secure way so that we can keep the apps consistent during an attack. Similarly when smart contract is connected with AI deep learning, we need to take care that all smart contracts have the same data input all the time to maintain the consistency of the contracts.

6. CONCLUSION

As energy networks will become increasingly complex towards smart and micro grids, new solutions will be needed for their management in an efficient, economical and secure manner. One proposed solution has been to utilize blockchains and smart contracts. The exploitation of both technologies has been extensively studied in the energy sector and several test environments have been made in connection with this subject. The possibilities of blockchains and smart contracts are wide in the energy networks of the future, and several solution options have been presented for their utilization. Since the security depends on the implementation, we might obtain a more accurate picture of the security of the different solutions by making small scale test platforms of the most interesting use cases both in the laboratory and the real environment. In

parallel to this testing work we should also study what tasks smart contracts need to do and develop programming tools, languages and environments, which enable secure programming methods, that minimise the possibility of harmful operations. When taking into account the critical nature of energy services, where good cybersecurity is an uttermost important property, our recommendation is to start blockchain implementations with with private blockchains, because they are more manageable and secured environment than public chains.

REFERENCES

[1] C. Zhang, J. Wu, C. Long, and M. Cheng, "Review of existing peer-to-peer energy trading projects," *Energy Procedia*, vol. 105, pp. 2563–2568, 2017.

[2] Z. Guan, X. Lu, W. Yang, L. Wu, N. Wang, and Z. Zhang, "Achieving efficient and privacy-preserving energy trading based on blockchain and abe in smart grid," *Journal of Parallel and Distributed Computing*, vol. 147, pp. 34–45, 2021.

[3] I. Perekalskiy, S. Kokin, and D. Kupcov, "Setup of a local p2p electric energy market based on a smart contract blockchain technology," *in 2020 21st International Scientific Conference on Electric Power Engineering (EPE)*. IEEE, 2020, pp. 1–4.

[4] S. Kushch and F. P. Castrillo, "A review of the applications of the blockchain technology in smart devices and dis-tributed renewable energy grids," ADCAIJ: *Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 6, no. 3, p. 75, 2017.

[5] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, and A. Peacock, "Blockchain technology in the energy sector: A systematic review of challenges and opportunities," *Renewable and Sustainable Energy Reviews*, vol. 100, pp. 143–174, 2019.

[6] M. Mylrea and S. N. G. Gourisetti, "Blockchain for smart grid resilience: Exchanging distributed energy at speed, scale and security," in *2017 Resilience Week (RWS). IEEE*, 2017, pp. 18–23.

[7] C. Yapa, C. de Alwis, M. Liyanage, and J. Ekanayake, "Survey on blockchain for future smart grids: Technical aspects, applications, integration challenges and future research," *Energy Reports*, vol. 7, pp. 6530–6564, 2021.

[8] H. Arezoo, M. H. Seyed, S.-k. Miadreza, and A. Hasan, "Blockchain technology in the future smart grids: A comprehensive review and frameworks[j]," *International Journal of Electrical Power and Energy Systems*, vol. 129, 2021.

[9] N. Ashizawa, N. Yanai, J. P. Cruz, and S. Okamura, "Eth2vec: Learning contract-wide code representations for vulnerability detection on Ethereum smart contracts," in *Proceedings of the 3rd ACM International Symposium on Blockchain and Secure Critical Infrastructure*, 2021, pp. 47–59.

[10] O. Lutz, H. Chen, H. Fereidooni, C. Sendner, A. Dmitrienko, A. R. Sadeghi, and F. Koushanfar, "Escort: Ethereum smart contracts vulnerability detection using deep neural network and transfer learning," *arXiv preprint arXiv:2103.12607*, 2021.

[11] C. Liu, X. Zhang, K. K. Chai, J. Loo, and Y. Chen, "A survey on blockchain enabled smart grids: Advances, applications and challenges," *IET Smart Cities*, vol. 3, no. 2, pp. 56–78, 2021.

[12] F. Mi, Z. Wang, C. Zhao, J. Guo, F. Ahmed, and L. Khan, "Vscl: Automating vulnerability detection in smart contracts with deep learning," in *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 2021, pp. 1–9.

[13] M. B. Mollah, J. Zhao, D. Niyato, K.-Y. Lam, X. Zhang, A. M. Ghias, L. H. Koh, and L. Yang, "Blockchain for future smart grid: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 18–43, 2020.

[14] A. Bhardwaj, S. Shah, A. Shankar, M. Alazab, M. Kumar, and T. Gadekallu, "Penetration testing framework for smart contract blockchain," *Peer-to-Peer Networking and Applications*, vol. 14, p. 2635–2650, Sep. 2021.

[15] A. Bahga and V. K. Madisetti, "Blockchain platform for industrial internet of things," *Journal of Software Engineering and Applications*, vol. 9, no. 10, pp. 533–546, 2016.

[16] C. Liu, K. K. Chai, X. Zhang, E. T. Lau, and Y. Chen, "Adaptive blockchain-based electric vehicle participation scheme in smart grid platform," *IEEE Access*, vol. 6, pp. 25 657–25 665, 2018.

[17] T. Alladi, V. Chamola, J. J. Rodrigues, and S. A. Kozlov, "Blockchain insmart grids: A review on different use cases," *Sensors*, vol. 19, no. 22, p.4862, 2019.

[18] U. Javaid, M. N. Aman, and B. Sikdar, "Defining trust in iot environments via distributed remote attestation using blockchain," in *Proceedings of the Twenty-First International Symposium on*

Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing, 2020, pp. 321–326.
[19] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 254–269.
[20] "list of known bugs," accessed: 2021-9-14. [Online]. Available: https://docs.soliditylang.org/en/v0.8.7/bugs.html.
[21] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum smart contracts (sok)," in *Principles of Security and Trust*. POST 2017. Lecture Notes in Computer Science, M. Maffei and M. Ryan, Eds., vol. 10204. Springer, Berlin, Heidelberg, 2017, pp. 164–186.
[22] A. L´opez Vivar, A. T. Castedo, A. L. Sandoval Orozco, and L. J. Garc´ıa Villalba, "An analysis of smart contracts security threats alongside existing solutions," *Entropy*, vol. 22, no. 2, p. 203, 2020.
[23] Y. Huang, Y. Bian, R. Li, J. L. Zhao, and P. Shi, "Smart contract security: A software lifecycle perspective," *IEEE Access*, vol. 7, pp. 150 184–150 202, 2019.
[24] Z. Zheng, S. Xie, H.-N. Dai, W. Chen, X. Chen, J. Weng, and M. Imran, "An overview on smart contracts: Challenges, advances and platforms," *Future Generation Computer Systems*, vol. 105, pp. 475–491, 2020.
[25] M. Jansen, F. Hdhili, R. Gouiaa, and Z. Qasem, "Do smart contract languages need to be turing complete?" in *International Congress on Blockchain and Applications. Springer*, 2019, pp. 19–26.
[26] ´A. J. Varela-Vaca and A. M. R. Quintero, "Smart contract languages: A multivocal mapping study," *ACM Computing Surveys (CSUR)*, vol. 54, no. 1, pp. 1–38, 2021.
[27] D. Harz and W. Knottenbelt, "Towards safer smart contracts: A survey of languages and verification methods (2018)," 2019.

AUTHORS

Pekka Koskela received the D.Sc. degree in 2018. Over 20 years he has been worked in several research projects both researcher and project leader. Currently he is studying among others the exploitation of quantum, homomorphic and digital ledger technologies.

Jarno Salonen is working as a Senior Scientist in the applied cybersecurity team at VTT He has a professional background of over 20 years in making the digital world a better place for ordinary users especially in the areas of cybersecurity, privacy, resilience and development of electronic services.

Juha Pärssinen obtained his M.Sc in 1998 and Lic.Sc. in 2003 from Helsinki University of Technology, Department of Computer Science and Engineering. He is working as a Senior Scientist in the applied cybersecurity team at VTT. During the years he has participated in multiple European joint research projects and ETSI standardization efforts. His research interest includes system architecture and workflow analysis, industrial control systems security and digital forensics.