Simulation-Based Analysis of Blockchain Architectures

1st Diego Marmsoler
dept. of computer science
Technische Universität München
München, Germany
diego.marmsoler@tum.de

2nd Leo Eichhorn
dept. of computer science
Technische Universität München
München, Germany
leo.eichhorn@tum.de

***Abstract*—In Software Architectures, architectural design decisions (ADDs) strongly influence the quality of the resulting software system. Wrong decisions lead to low quality systems and are only difficult to repair later on in the development process. As of today, only little is known about the impact of certain ADDs for the development of architectures for Blockchain-based systems. Thus, it is difficult to predict the outcome of certain ADDs when developing architectures for such systems.**

In the following, we propose a simulation-based approach for Blockchain architectures in which the impact of certain ADDs can be simulated. Therefore, we implemented a simulation environment for Blockchain architectures. Then, we evaluated it by executing a series of experiments. Based on the simulation environment we also propose a first version of an empirical model for Blockchain architectures.

***Index Terms*—Blockchain Architectures, Proof of Work, Double Spend Attacks, Stale Blocks, Architectural Design Decisions**

I. Introduction

With the emergence and increasing popularity of decentralized cryptocurrencies such as Bitcoin [11] and Ethereum [2], Blockchain architectures [17] have become more and more important. Their use seems promising also for other domains, such as the medical one [1], land management [3], or even identity management [18]. The design of Blockchain architectures requires many different architectural design decisions (ADDs) to be taken, each of which with a strong impact on the quality of the resulting system [15]. Thus, wrong design decisions may have dramatic consequences on the resulting system and once the system is implemented, such decisions are only difficult to repair [7].

As of now, however, only little is known about the impact of decisions in the design of Blockchain architectures on the quality of the resulting system. Thus, it is difficult, if not to say impossible, to predict the outcome of certain ADDs on the quality of the resulting Blockchain system before implementing it.

To address this problem, we propose a *simulation-based* approach to study the effects of certain ADDs on certain quality attributes of a Blockchain architecture. Figure 1 depicts an overview of our approach: As a first step, we implemented a framework to simulate certain properties for Blockchain architectures with certain ADDs. To this end, we studied existing literature to identify classes of ADDs which might influence certain quality attributes of Blockchain architectures. Then, we implemented a simulation environment which can simulate properties for Blockchain systems built according to certain ADDs. In a second step, we used the framework to simulate the effect of certain ADDs on these properties. To this end, we configured the framework according to a set of ADDs, ran the simulation, and recorded the outcome. In a last step, we used the outcome of the simulations to propose an empirical model which relates ADDs with quality attributes. This is an iterative step: we first "guessed" a potential model based on our observations of the simulation and then we run the simulation again to test the model's predictions against the simulated output.

The major contribution of this work is twofold: 1) We provide a simulation environment which can be used to simulate the impact of 5 classes of ADDs on two types of quality attributes for Blockchain systems. 2) We provide an empirical model which can be used to estimate ADDs based on desired quality attributes for Blockchain systems.

In the following, we first provide some background for our work. Then, we describe our simulation-based approach, present results obtained from various experiments, and propose an empirical model based on the outcome of the experiments. Finally, we present related work and conclude with an outlook which leads to a discussion of future work.

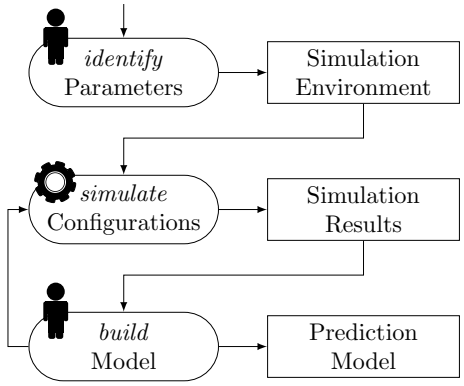

Fig. 1. Simulation-based approach showing manual (stick-figure) and automated (gear-wheel) activities (rounded rectangles) and corresponding artifacts (rectangles).

II. Background

In the following, we provide some background for our work.

A. Blockchain

A Blockchain is a public[1], distributed database used to record, identify and verify contracts, transactions or other shared data between multiple parties. The resulting data records are stored in a continuously growing list, which is locally maintained and updated by each individual member (node) of the network. Entries of the list (blocks) are cryptographically linked by including a hash of the previous block as a unique identifier in each newly added entry. More specifically, altering contents of a block changes its unique identifier, forcing a recalculation of every following block currently in the list in order to retain integrity of the ledger. Newly created blocks are broadcasted to all members to keep local Blockchain copies synchronized. By adhering to a distributed consensus protocol, the participating nodes validate potential extensions of their Blockchain copy in a peer-to-peer manner, thereby eliminating the need for an intermediary, trusted authority.

B. Proof of Work

Since our work is based on the Proof-of-Work consensus algorithm, we discuss this algorithm in more detail. Proof-of-Work (PoW) is a consensus algorithm which requires miners to solve a cryptographic puzzle, in order to mine a new block of the Blockchain. It is based on the assumption that the longest chain (in terms of number of confirmed blocks) is the "correct" one, since it requires the most computational power to be computed. Although it requires a lot of computation power (and thus electricity), it is still widely used in contemporary Blockchain architectures, such as Bitcoin [11].

C. Double Spend Attack

Double spend attacks (DSAs) are attempts to modify already confirmed entries of a Blockchain. To this end, an attacker (or a group of attackers) tries to work on an alternative extension of the Blockchain, which they hide from the rest of the network. After a block is confirmed by the network, the attackers release their alternative extension of the chain which does not contain the confirmed block [13], [14]. In order to be consistently successful, DSAs usually require the attackers to own more than 50% of the network's computing power [9], [12]–[14]. A Blockchain architecture's resistance against double spend attacks is an important property of such architectures. It is influenced by different design decisions, such as the size of the attacking network, the number of required confirmation blocks, the difficulty of extending the Blockchain, as well as the topology and latency of the underlying networks.

[1]In this paper, only permissionless Blockchains are considered. For permissioned Blockchains see [16].

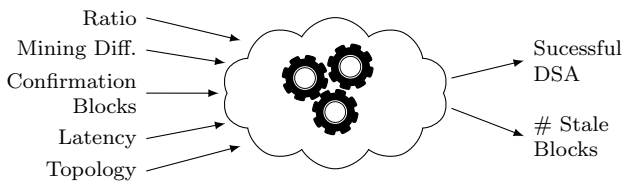

Fig. 2. Simulation-based approach to Blockchain architectures.

D. Stale Blocks

When using PoW consensus algorithms, miners usually compete to solve the next block and sometimes it might happen that different miners solve different blocks at the same time. This leads to two alternative versions of the Blockchain. After a new block is solved on top of one of the versions, this version is assumed to be the correct one and the block of the other chain is considered a stale or orphaned block (in the following denoted SB). SBs actually represent a "waste" of computing power and ideally their number should be kept low [4], [5].

III. Simulation Approach

Figure 2 depicts an overview of our approach for simulating Blockchain architectures: The framework is configured using five types of parameters to characterize an architecture according to certain design decisions. It then simulates a Blockchain system with a corresponding architecture and records the occurrence of successful DSAs as well as the amount of stale blocks. In the following, we provide some details about the simulation. The complete framework is available online and can be downloaded from [6].

A. Simulation Model

The simulated system is based on the PoW consensus algorithm (Section II-B). Its architecture consists of a network of nodes, each of which stores a private copy of the Blockchain. Nodes may receive copies of other nodes' Blockchains at every point in time, while they are also able to broadcast copies of their own Blockchain versions themselves. Moreover, a node can try to mine a new block, in which case it generates random numbers until it obtains one which is below a certain target value. In order to simulate DSAs, we distinguish between *trusted* and *untrusted* (or attacking) nodes. Trusted nodes strictly adhere to the protocol: 1) They always take the longest Blockchain as the "correct" one. Thus, they replace their own copy with a new Blockchain, whenever a longer one is obtained. 2) They always try to extend their Blockchain and new blocks are mined on top of it. Untrusted nodes, however, may deviate from the protocol in two ways: 1) They may not replace their copy of the Blockchain with a longer chain obtained from the trusted network. 2) They may also drop elements from the chain. Figure 3 shows an exemplary simulation network, consisting of two sub-networks: a trusted network consisting of eight trusted nodes and an attacking network which consists of five untrusted nodes.

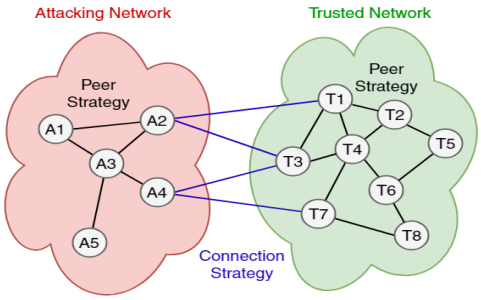

Fig. 3. Exemplary simulation network of 5 attacking and 8 trusted nodes.

B. Input Parameters

Currently, the simulation supports five types of parameters listed in table I. Parameter R is a value of the interval 0 to 1, representing the networks estimated ratio of untrusted nodes to total nodes. For example, if we expect 500 trusted nodes and 200 attacker nodes, R would be set to $2/7$. Parameter T is also a value between 0 and 1, which is used to set the difficulty of mining a new block. In the simulation, miners will generate random numbers between 1 and 0. Thus, whenever a number below T is obtained, a new block can be created. C is the number of required confirmation blocks. A simulated double spend attack is only possible if it is able to modify a block which was already confirmed by C confirmation blocks. We provide three parameters to simulate network latency: L_T, L_A, and L_C to set the average latency between trusted nodes, attacking nodes, and between the two sub-networks, respectively. Finally, network topology can be influenced with parameters D_T and D_A, representing the number of edges between trusted nodes and untrusted nodes, respectively.

C. Simulation Outcome

Currently, two aspects of a Blockchain architecture can be simulated: successful DSAs (PDS) and the amount of stale blocks (PSB). Thereby, a DSA is assumed to be successful, whenever the attacking network is able to modify an entry in the chain, which was already confirmed by a certain number of confirmation blocks. By calculating the percentage of successful attempts, a measure describing the simulated architecture's resistance against DSAs is created. The amount of stale blocks, on the other hand, is obtained by measuring the amount of blocks in both sub-networks that were added to obsolete and thereby shorter Blockchain copies. This value is again described as a percentage of all blocks mined in total.

TABLE I
POSSIBLE SIMULATION PARAMETERS

Par.	Description	p. val.
R	Ratio of untrusted nodes, compared to total nodes	$[0, 1]$
T	Mining difficulty target	$[0, 1]$
C	Number of confirmation blocks	integer
Network Latency		
L_T	Average latency in the trusted network (milliseconds)	integer
L_A	Average latency in the untrusted network (milliseconds)	integer
L_C	Average latency btw. tr. and un. network (milliseconds)	integer
Network Topology		
D_T	Ratio of existing edges in the trusted network, compared to the maximum amount of possible edges	$[0, 1]$
D_A	Ratio of existing edges in the untrusted network, compared to the maximum amount of possible edges	$[0, 1]$

IV. Evaluation

In the following, we present the outcome of some of the experiments we performed in order to evaluate the framework.

A. Impact of Ratio on DSAs

For the first type of experiment, we simulated a Blockchain architecture with $C = 6$ confirmation blocks, latencies $L_T = L_A = 10ms$, network densities $D_T = D_A = 0.8$, and different values for the ratio R of untrusted notes. Figure 4 depicts the outcome of these experiments and shows that the probability of a successful DSA grows exponentially with an increasing proportion of attacking nodes. Moreover, the data confirms the prevalent assumption that, for Blockchain architectures with a ratio of $R > 0.5$, DSAs always succeed.

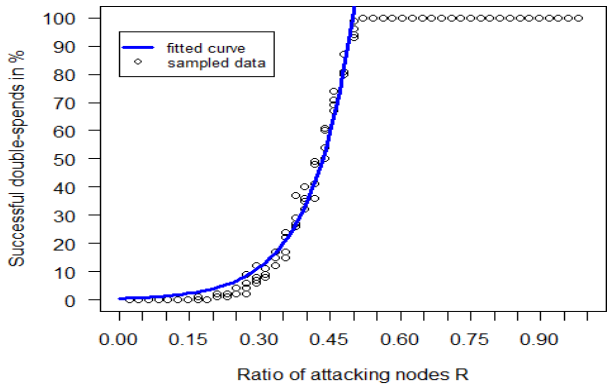

Fig. 4. Successful DSAs for different node ratios.

B. Impact of Confirmation Length on DSAs

In a second set of experiments, we simulated a Blockchain architecture with constant latencies $L_T = L_A = 10ms$, network densities $D_T = D_A = 0.8$, and different values for ratio R and number of confirmation blocks C. The results are depicted in figure 5 and show that the amount of successful DSAs drops exponentially when increasing the number of confirmation blocks. Thereby, the exponential factor seems to depend on the ratio R, i.e., a low proportion of attacking nodes leads to a high exponential factor whereas a high percentage of attacking nodes leads to a low exponential factor.

C. Impact of Difficulty on DSAs and PSBs

Another set of experiments simulates Blockchain architectures for a constant ratio $R = \frac{1}{3}$, number of confirmation blocks $C = 6$, latencies $L_T = L_A = 10ms$, network densities $D_T = D_A = 0.8$ and different mining difficulties T. Figure 6

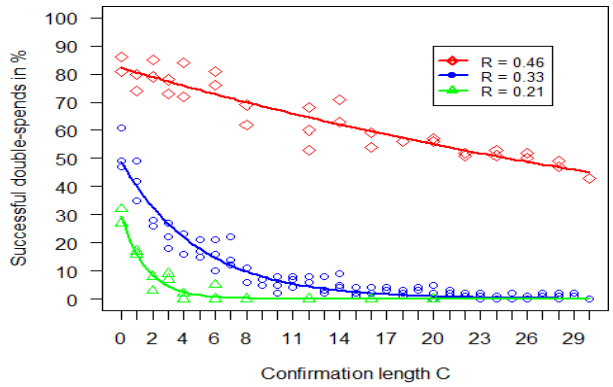

Fig. 5. Successful DSAs for different numbers of confirmation blocks.

shows a clear positive effect of mining difficulty on both, the probability of successful DSAs as well as the amount of stale blocks for corresponding Blockchain systems. However, while the probability of a DSA remains constant until it then increases exponentially, the amount of stale blocks increases from the beginning and then flattens out logarithmically.

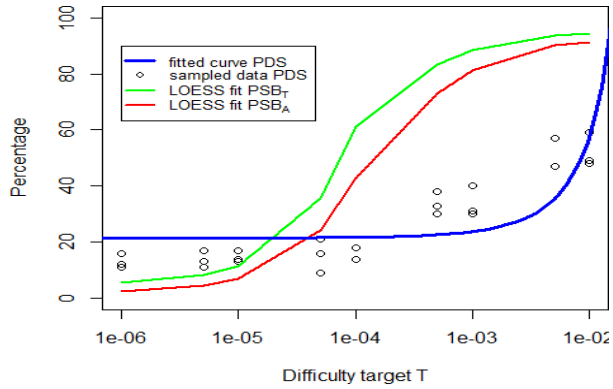

Fig. 6. Successful DSAs and produced SBs for different mining difficulties.

D. Impact of Latency on DSAs and PSBs

Our next set of experiments investigates the effect of network latency on the probability of a DSA and the amount of produced stale blocks. As mentioned above, we distinguish between three types of latencies: latency in the trusted sub-network, latency in the attacking sub-network, and latency between trusted and attacking networks. For each type we simulate a Blockchain architecture for a constant ratio $R = \frac{1}{3}$ with $C = 6$ confirmation blocks. Then, we choose a constant value for two of the latencies and investigated the impact of the remaining type on DSAs and PSBs.

1) Latency in trusted sub-network: Figure 7 shows the percentage of successful DSAs and the proportion of produced stale blocks for different latencies L_T for the trusted sub-network. It can be observed that the expected latency in the trusted sub-network indeed impacts both, the probability of a successful DSA (exponentially), as well as the amount of stale blocks (logarithmic).

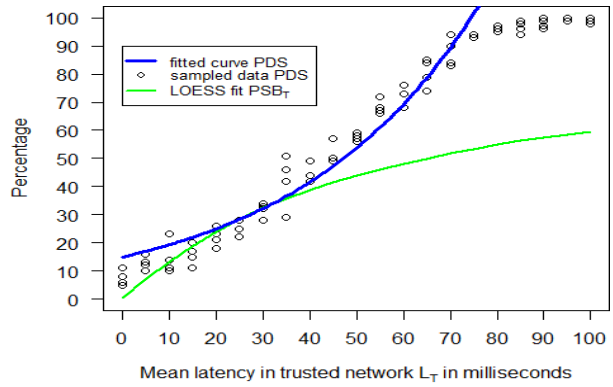

Fig. 7. Successful DSAs and produced SBs for different latencies in the trusted sub-network.

2) Latency in attacking sub-network: Figure 8 shows the percentage of successful DSAs and the proportion of produced stale blocks for different latencies L_A for the attacking sub-network. Similar to the experiments with the latency in the trusted network, we can observe a negative, logarithmic effect of latency in the attacking network to the amount of produced stale blocks. However, this time, the effect on successful DSAs is exponentially negative (compared to the positive effect observed in figure 7).

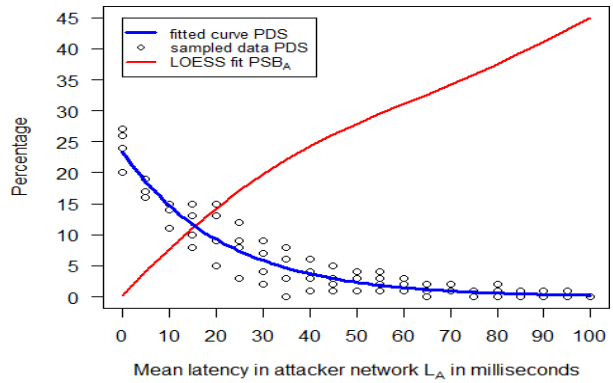

Fig. 8. Successful DSAs and produced SBs for different latencies in the attacking sub-network.

3) Latency between trusted and attacking sub-networks: Figure 9 depicts the percentage of successful DSAs for different latencies L_c in between the trusted and attacking network. Similar to the experiments with the latency in the attacking network, we can observe an exponentially negative effect on the probability of a DSA.

E. Impact of Topology on DSAs and PSBs

Finally we investigated the impact of network-topology on the probability of a successful DSA and the amount of produced stale blocks. As mentioned above, topology is measured by means of graph density (ratio of existing and maximum amount of edges in the network). Again, we distinguish between the topology of the trusted sub-network and that of the attacking sub-network.

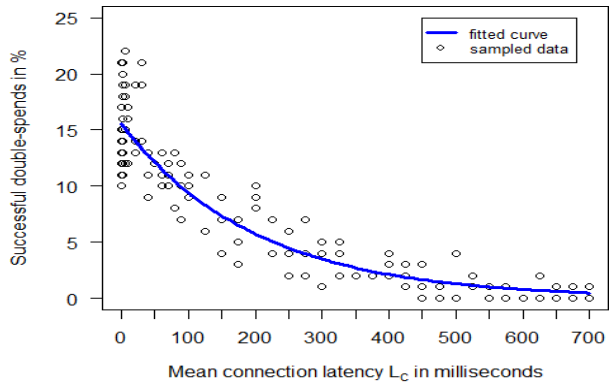

Fig. 9. Successful DSAs for different latencies between the trusted and the attacking sub-network.

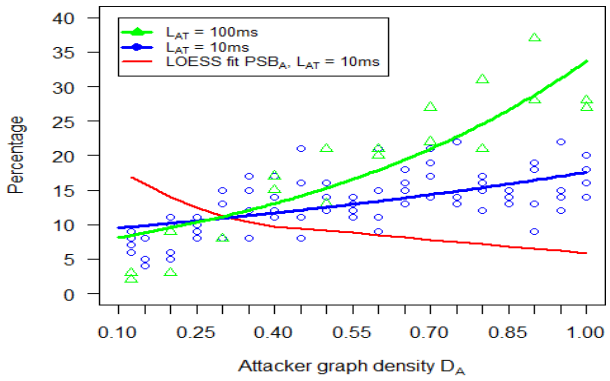

Fig. 11. Successful DSAs and produced SBs for different densities in the attacking sub-network.

1) Density in trusted sub-network: Figure 10 depicts the percentage of successful DSAs for different densities of the trusted sub-network D_T for latencies of 100 ms (green) and 10 ms (blue). For the latter, it also shows the impact of density in the trusted sub-network on the amount of produced stale-blocks (red). We can observe a negative impact on both, DSAs and PSBs, which is intensified when increasing network latency.

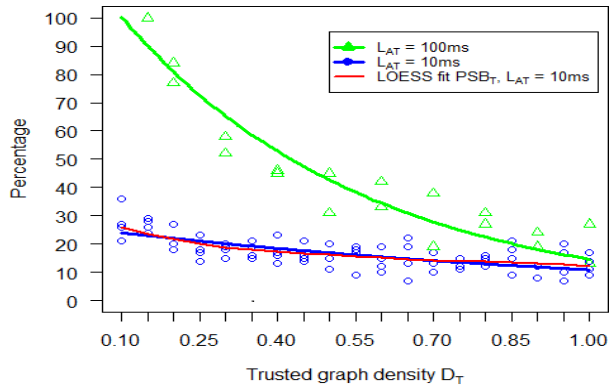

Fig. 10. Successful DSAs and produced SBs for different densities in the trusted sub-network.

2) Density in attacking sub-network: We performed similar experiments to investigate the impact of the density in the attacking sub-network on DSAs and PSBs. Its outcome is depicted in figure 11: As expected, the effect observed for the trusted density is inversed for the attacking density and we can observe a positive impact on DSAs and PSBs, which is intensified with increasing network latency.

V. Towards an Empirical Prediction Model for Blockchain Architectures

The simulation environment presented so far can be used to simulate DSAs and PSBs for different ADDs. However, it is not yet possible to estimate necessary ADDs for a Blockchain architecture with a desired resistance against DSAs. To this end, we performed several additional experiments to systematically develop an empirical model, based on the results of these experiments. In total, over 4000 datapoints were created, allowing us to perform regressions in order to fit the resulting empirical constants of the proposed model.

The model formalizes the probability of a successful DSA in terms of a function of the estimated ratio R, mining difficulty T, amount of required confirmation blocks C, latencies L_T, L_A and L_C, and network topologies D_T and D_A:

$$PDS = 100 \; exp\left(\left(a\left(R - \frac{e_1\left(e_2T + e_3\right)L_A}{d_1\left(D_A + d_2\right)} + \frac{e_1\left(T + e_3\right)L_T}{d_1\left(D_T + d_2\right)}\right) - \frac{a}{2}\right) \cdot \left(b_1C + b_2\right)\left(c_1L_C + c_2\right)\right)$$

The values of the corresponding, empirical constants are as follows:

a	10.6572
b_1	0.102625
b_2	0.409421

c_1	0.00273521
c_2	1.02279
d_1	226.506
d_2	0.868318

e_1	50000.2
e_2	0.956131
e_3	0.00001

To test our model, we compared its predictions to the outcome of our experiments. Figure 12, for example, depicts model predictions (dashed) and simulated outcome (solid) for different ratios and confirmation lengths. Although our model is not entirely accurate, the general profile of the plotted formula matches the individually fitted curves of the data.

The model can be used to estimate certain ADDs, given a desired resistance against DSAs for the resulting Blockchain architecture.

VI. Related Work

Since the introduction of Bitcoin, several attempts where made to model and predict DSAs for such systems. Similar to [13], we will call the collection of these approaches *hashrate-based* attack models. The central premise of a hashrate-based model is splitting the total computing power available to the network (hashrate H) into two parts. One of the first mathematical models has been proposed in Nakamotos original paper [11]. Since then, Nakamoto's model has been experimentally tested [12] and improved [9], [12]–[14]. While

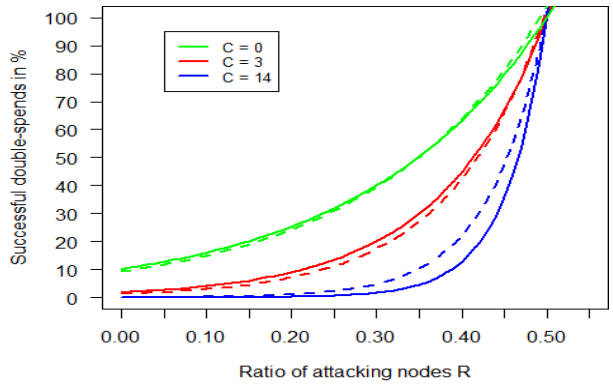

Fig. 12. Model predictions vs. simulation outcome.

these models provide first steps towards a prediction model for Blockchain architectures, the prediction is mainly based on three parameters: The probabilities of the trusted and untrusted networks mining the next block, respectively, as well as the number of required confirmation blocks. However, it should also be mentioned that Pinzón and Rocha [13] are adding onto the work of Nakamoto [11] and Rosenfeld [14] by formulating further models considering the amount of time an attacker has already spent mining on secret blocks in advance of the attack. On the contrary, our model supports predictions based on additional factors such as the mining difficulty, network latencies and network densities.

One exception to the above class of models are the works of Mwale [10] and Göbel et al. [8]. Here, a more realistic simulation of different attacks on Bitcoin Blockchains is provided, including parameters such as latency between two nodes. Compared to our work, however, the modeled network topology is simplified and does not allow the simulation of common network topologies such as stars or rings. Additionally, the creation of new blocks is modeled as a Poisson process with constant rate, meaning that no actual calculation of a PoW is performed. Finally, the model was not used to simulate and evaluate the effects of different ADDs on the resistance of Blockchain architectures against DSAs.

VII. Conclusion

This paper described a simulation-based approach for the design of Blockchain architectures. To this end, we presented a framework for the simulation of Blockchain architectures, which may simulate double spend attacks and the amount of stale blocks for various types of design decisions: 1) the estimated ratio of untrusted vs. total nodes 2) mining difficulty 3) number of confirmation blocks 4) network latency 5) network topology. Then, we described the outcome of several experiments for various design decisions: the impact of the ratio on the probability of successful DSAs, the impact of confirmation blocks on DSAs, and the impact of latency on DSAs and stale blocks. On top of these (and other experiments not presented in the paper), we provide an empirical model for the impact of ADDs on a Blockchain architecture's resistance against DSAs.

The framework can be used to simulate certain design decisions for Blockchain architectures before implementing them. The empirical model can be used to approximate design decisions for desired qualities. Thus, wrong design decisions (w.r.t. expected properties) could be avoided which reduces effort of fixing them after implementation.

As of today, however, the simulation environment (and corresponding model) supports only Blockchain architectures based on PoW. For the future we want to integrate additional consensus algorithms, such as proof of stake. Moreover, we are working on an extended version of the framework to support additional design decisions and simulate additional properties.

Acknowledgments: We would like to thank Manfred Broy and all the anonymous reviewers of *SDLT3* for their comments and helpful suggestions on earlier versions of this paper. Parts of the work on which we report in this paper was funded by the German Federal Ministry of Education and Research (BMBF) under grant no. 01Is16043A.

References

[1] Asaph Azaria, Ariel Ekblaw, Thiago Vieira, and Andrew Lippman. Medrec: Using blockchain for medical data access and permission management. In *Open and Big Data (OBD), International Conference on*, pages 25–30. IEEE, 2016.

[2] Vitalik Buterin. Ethereum, 2013.

[3] Gertrude Chavez-Dreyfuss. Sweden tests blockchain technology for land registry. https://web.archive.org/web/20161024065806/http://www.reuters.com/article/us-sweden-blockchain-idUSKCN0Z22KV, June 2016.

[4] Nicolas T. Courtois and Lear Bahack. On subversive miner strategies and block withholding attack in bitcoin digital currency. *CoRR*, abs/1402.1718, 2014.

[5] C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In *IEEE P2P 2013 Proceedings*, pages 1–10, Sept 2013.

[6] Leo Eichhorn. Simulation-based analysis of blockchain architectures. https://github.com/LeoEichhorn/Blockchain, 2018.

[7] David Garlan. Software architecture: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering*, pages 91–101. ACM, 2000.

[8] Johannes Göbel, H. Paul Keeler, Anthony E. Krzesinski, and Peter G. Taylor. Bitcoin blockchain dynamics: the selfish-mine strategy in the presence of propagation delay. *CoRR*, abs/1505.05343, 2015.

[9] Cyril Grunspan and Ricardo Pérez-Marco. Double spend races. *CoRR*, abs/1702.02867, 2017.

[10] Mabvuto Mwale. *Modelling the dynamics of the bitcoin blockchain*. PhD thesis, Stellenbosch: Stellenbosch University, 2016.

[11] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[12] A. Pinar Ozisik and Brian Neil Levine. An explanation of nakamoto's analysis of double-spend attacks. *CoRR*, abs/1701.03977, 2017.

[13] Carlos Pinzón and Camilo Rocha. Double-spend attack models with time advantage for bitcoin. *Electronic Notes in Theoretical Computer Science*, 329:79 – 103, 2016. The Latin American Computing Conference.

[14] Meni Rosenfeld. Analysis of hashrate-based double spending. *CoRR*, abs/1402.2009, 2014.

[15] Mark Staples. Software engineering research for blockchain-based systems. In *1st Symposium on Distributed Ledger Technology*, 2017.

[16] Marko Vukolić. Rethinking permissioned blockchains. In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, BCC '17, pages 3–7, New York, NY, USA, 2017. ACM.

[17] Xiwei Xu, Ingo Weber, Mark Staples, Liming Zhu, Jan Bosch, Len Bass, Cesare Pautasso, and Paul Rimba. A taxonomy of blockchain-based systems for architecture design. In *2017 IEEE International Conference on Software Architecture, ICSA 2017, Gothenburg, Sweden, April 3-7, 2017*, 2017.

[18] Bryan Yurcan. How blockchain fits into the future of digital identity. https://web.archive.org/web/20170119054131/https://www.americanbanker.com/news/how-blockchain-fits-into-the-future-of-digital-identity, April 2016.