

Demo Abstract: Attack and Vulnerability Simulation Framework for Bitcoin-like Blockchain Technologies

Fabian Schüssler
Technical University of Munich
fabian.schuessler@in.tum.de

Pezhman Nasirifard
Technical University of Munich
p.nasirifard@tum.de

Hans-Arno Jacobsen
Technical University of Munich

ABSTRACT

Despite the very high volatility of the cryptocurrency markets, the interest in the development and adaptation of existing cryptocurrencies such as Bitcoin as well as new distributed ledger technologies is increasing. Therefore, understanding the security and vulnerability issues of such blockchain systems plays a critical role. In this work, we propose a configurable distributed simulation framework for analyzing Bitcoin-like blockchain systems which are based on Proof-of-Work protocols. The simulator facilitates investigating security properties of blockchain systems by enabling users to configure several characteristics of the blockchain network and executing different attack scenarios, such as double-spending attacks and flood attacks and observing the effects of the attacks on the blockchain network.

CCS CONCEPTS

• **Computer systems organization** → **Peer-to-peer architectures**;

KEYWORDS

Blockchain, Bitcoin, Simulation

1 INTRODUCTION

Understanding the security and vulnerability issues of different blockchain systems, such as Bitcoin, plays a significant role as the popularity of Bitcoin-like systems increases. In this work, we propose a simulation framework for simulating different types of attacks on Bitcoin-like blockchains. The proposed framework is a configurable blockchain simulator capable of conducting distributed large-scale network simulations and simulating different types of attacks, such as double-spending attacks and flood attacks, for deriving empirical insights on the behavior of the system under attack. A double-spending attack on Bitcoin represents a situation where the malicious miners try to outpace the longest valid chain. A transaction spam attack, also known as flood attack, represents a scenario where one or multiple nodes disrupt the normal operation of the network by issuing a large number of spam transactions. The spam transactions can increase the transaction costs and the transaction confirmation time.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Middleware '18, December 10–14, 2018, Rennes, France

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-6109-5/18/12...\$15.00

<https://doi.org/10.1145/3284014.3284017>

2 RELATED WORK

We built this work upon *VIBES*, a generic blockchain simulator for simulating the issuance of transactions, the block mining and the interaction of thousands of miners [5]. Our work extends *VIBES* with the latest implementation of Bitcoin and enables Bitcoin attack simulations. In addition to *VIBES*, other works attempted to simulate distributed ledger technologies [1, 2, 4]. However, other works focus solely on a single aspect of the Bitcoin network, where [1] simulates the block mining and effects of mining rewards and transaction fees, and [2, 4] focus on simulating the Bitcoin network and propagation of messages. In contrast to other works, we propose a simulator which addresses every aspect of a Bitcoin network, as well as offering a rich user interface for configuring the simulation environment and investigating the simulation results.

3 TECHNICAL APPROACH

Figure 1 demonstrates the architecture of our system which is based on the *VIBES* simulator. We make use of and improve several concepts introduced by *VIBES*, such as the fast-forward computing and the Coordinator Node. We also make use of *VIBES* primary blockchain configuration panel, which enables the user with providing several main simulation metrics such as the block size limit, block time, and transaction sizes.

We provide a web-based user interface where the user configures the different network and attack parameters. For demonstrating a double-spending attack, the user can configure the following simulation metrics: the attacker's hash-rate as a percentage of the network's total hash-rate (q), the attack duration as the number of mined blocks (d) and the number of desired transaction confirmations (n). Once the user provides all the simulation inputs, the simulation starts, and the system assigns every simulated miner node an *honesty attribute* based on the attacker's hash-rate which specifies if the node is *honest* or *malicious*. After the nodes started with mining on the genesis block, the network separates into an honest public blockchain and a private malicious blockchain. The attackers include their transactions for double-spending attempts into the public blockchain, and the malicious nodes start mining on the competing blockchain. The attack succeeds if the malicious blockchain is longer than the honest blockchain after the fraudulent transactions reached the required number of confirmations. For the attack to fail, the honest network's blockchain needs to stay longer until the end of the attack simulation. After the outcome of the attack is decided, both networks merge back together. The system visualizes the blockchain tree and the branch selection and shows the results of the simulation such as the attack success probability, and the maximal safe transaction value which is the maximal transaction value a merchant should accept; otherwise, it would be profitable for an attacker to attempt a double-spending attack.

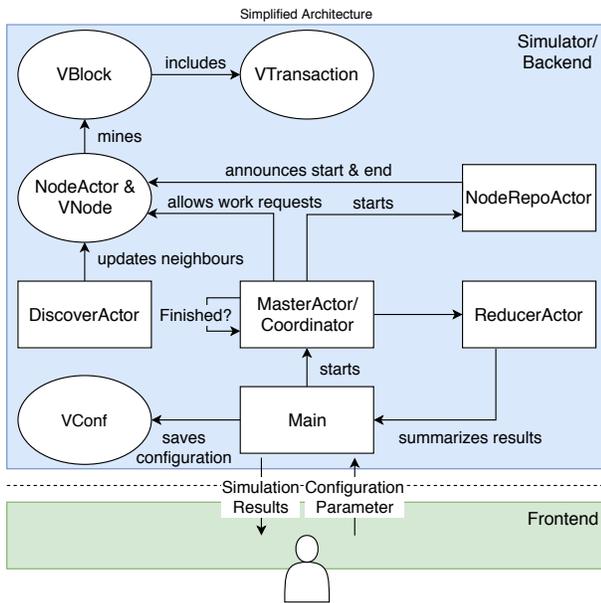


Figure 1: The architecture of the simulator.

The system tests the theoretical and empirical outcomes of the double-spending attack against the calculated success probability r . It has to be taken into account that the simulation needs an attack duration to decide if the attack failed, but the Equation 1 does not factor in the attack duration, but we consider this effect negligible with $d = 50$. Table 1 compares the results of 101 simulations with the calculated double-spending attack success probability.

$$r = \begin{cases} 1 - \sum_{m=0}^n \frac{(m+n-1)!}{m!(n-1)!} (p^n q^m - p^m q^n), & \text{if } q < p \\ 1, & \text{if } q \geq p \end{cases} \quad (1)$$

Figure 2: Double-spending success probability where p is the honest network’s percentage of the total hash-rate and $q = 1 - p$. [3]

Table 1: Outcomes of a double-spending attack and their simulated and expected probabilities with a sample size of 101 simulations ($n = 6, q = 30, d = 50$).

Outcome	Simulated Probability	Expected Probability
Attack successful	13.86%	15.645%
Attack failed	86.13%	84.355%
TOTAL	100%	100%

For simulating a flood attack, the user requires to specify one parameter, the target transaction fee, which is the minimum fee that the attacker desires to gain from every transaction issuer in the network. In our implementation, the Coordinator sends requests to the nodes to create the transactions which are marked with a

transaction spam attribute. In the case of a very short block generation time, transactions might not get propagated fast enough to the miner, who includes the transactions in the next block. To ensure that every transaction issuer in every block requires the targeted transaction fee, the attacker has a buffer of spam transactions in the transaction pool, which leads to maintaining the target fee in the event of very short block generation times despite the existence of transactions with lower fees in the transaction pool. In our simulations, a buffer of double the size of maximal transactions per block ensures the attacker’s goal. After the simulation, the system provides a report containing several metrics such as the number of confirmed flood attack transactions, the attacker’s spent transaction fees in Bitcoin and the number of confirmed transactions below the target fee.

We developed a web-based user panel based on *React*¹ and the simulation framework is implemented based on *Scala*² and *Akka*³. We open-sourced the code of simulator, which is provided publicly⁴.

4 CONCLUSIONS

In this demo, we present an approach for simulating a few attacks on Bitcoin-like blockchain networks. Furthermore, the users can configure several simulation metrics, and the simulations outputs are displayed in detailed reports for the user to explore the security properties. As part of future work, we plan to extend the simulation framework with characteristics of mining pools, the effects of decreasing mining rewards, the mining energy consumption, and investigating user privacy measures.

ACKNOWLEDGMENTS

Alexander von Humboldt Foundation supported this project.

REFERENCES

- [1] M. Carlsten, H. Kalodner, M. Weinberg, and A. Narayanan. 2016. On the Instability of Bitcoin Without the Block Reward. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 154–167.
- [2] T. Neudecker, P. Andelfinger, and H. Hartenstein. 2015. A simulation model for analysis of attacks on the Bitcoin peer-to-peer network. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 1327–1332.
- [3] M. Rosenfeld. 2014. Analysis of Hashrate-Based Double Spending. *CoRR* (2014).
- [4] M. B. Sarrias. 2015. *Bitcoin Network Simulator Data Exploitation*. Master’s thesis. Open University of Catalonia.
- [5] L. Stoykov, K. Zhang, and H. A. Jacobsen. 2017. VIBES: Fast Blockchain Simulations for Large-scale Peer-to-peer Networks: Demo. In *Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference: Posters and Demos*. ACM, 19–20.

¹<https://reactjs.org/>

²<https://www.scala-lang.org/>

³<https://akka.io/>

⁴<https://github.com/i13-msrg/vibes>