Hardware-Based Approach in Blockchain to Detect Node Compromise in Distributed Ledger IoT Technology

Yigit Alparslan (ya332@drexel.edu), Drexel University

June 5, 2019

1 Introduction

IoT (Internet of Things) has been focused a lot recently due to increased complexity of systems[1][2][3][4]. Distributing resources at a network in an IoT platform can help us achieve solving important problems[5][6]. Recently, block-chain technology has seen popularity due to its distributed nature over the network and scalability [7]. Blockchain is a distributed ledger, which simply means that a ledger is spread across the network among all peers in the network, and each peer holds a copy of the complete ledger. Bitcoin was originally created by Satoshi Nakamato to be the world's first peer-to-peer cash system designed to eliminate modern day payment methods that rely almost "exclusively on financial institutions" to process these electronic payments. Blockchain based payment systems show us a glimpse of how we can begin eliminating these centralized points, but any current system shows the lack of settlement and mainstream consumer adoption. Besides the cryptocurrency implementation of blockchain, we see it in other fields such as energy, public utilities. This paper explores one of such implementations as an IoT technology.

Blockchain consists of several nodes. Each node can have a sensor. Sensors are not directly connected to a local board that we are running our decision making process on. Such separation helps reduce creating single point of failure.

One problem that is encountered in such implementation is node compromise, where sensor and node data don't match. A redundant sensor, that play the role of a vault, placed on the chain can give us insight on how to verify data coming from sensors. However, there are three cases where node compromise can occur:

1. Chain is attacked. Redundant sensor will notify the chain. Compromise will be detected.

2. Redundant sensor data inside the vault happens to be hacked. Local board will sense that the vault data is different. Compromise will be detected.

3. If vault data and local board data are same, but there happens to

be hack, then time stamps will be different on each node transaction, based on the delay of verified time stamp on the node that has the vault data and the node that has the local board data, compromise will be detected.

2 Algorithm

1. A node starts a transaction by first creating and then digitally signing it with its private key (created via cryptography) . A transaction can represent various actions in a blockchain. Most commonly this is a data structure that represents transfer of value between users on the blockchain network. Transaction data structure usually consists of some logic of transfer of value, relevant rules, source and destination addresses, and other validation information.

2. A transaction is propagated (flooded) by using a flooding protocol, called Gossip protocol, to peers that validate the transaction based on preset criteria. Usually, more than one node are required to verify the transaction.

3. Once the transaction is validated, it is included in a block, which is then propagated onto the network. At this point, the transaction is considered confirmed.

4. The newly-created block now becomes part of the ledger, and the next block links itself cryptographically back to this block. This link is a hash pointer. At this stage, the transaction gets its second confirmation and the block gets its first confirmation.

5. Transactions are then reconfirmed every time a new block is created. Usually, six confirmations in the a network are required to consider the transaction final.

2.1 System Layout

The system consists of two major components: A web platform locally hosted by the user (User Interface), and a Backend Software (Simulation Script to drive the logic).

1. UI accepts urls to mine, verify, resolve node conflicts.

2. Software calculates the mine hashs, and does the proof of work to verify/resolve conflicts.

2.2 Code

In order to keep this paper concise, the full code for the project can be found on github "github.com/ya332/blockchain". Collaborator can be added on requests made to ya332@drexel.edu. Github code includes the simulation script, web app, and blockchain implementation.

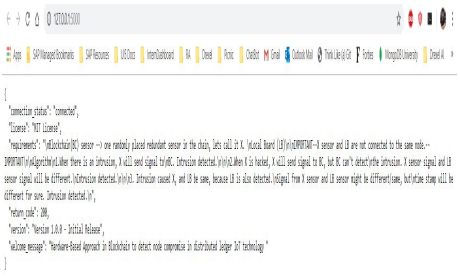

Figure 1: Snippet of the simulation code from blockchain simulation

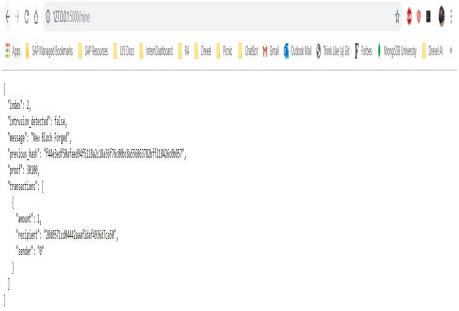

Figure 2: Snippet of the simulation code from blockchain simulation

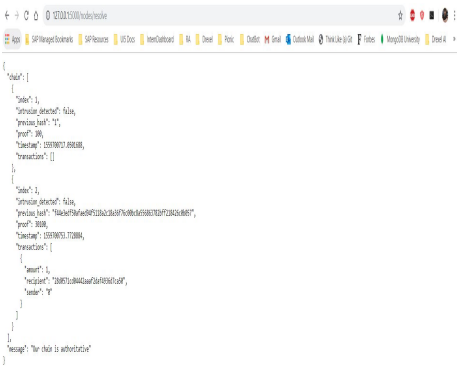

Figure 3: Snippet of the simulation code from blockchain simulation

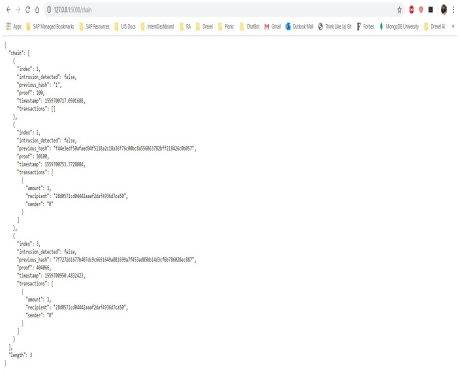

Figure 4: Snippet of the simulation code from blockchain simulation

3 Conclusion and Future Work

Blockchain is a promising field that touches several industries. Public utility, energy sectors being a few of them. Decentralized chain with publicly available nodes gives insight at every step of the transaction, and make the business process more transparent and failproof since it is really hard to modify the ledger and any change is permanent[8][9]. One disadvantage of such implementation is a node compromise[10], and its defense heavily depends on the location of the hack[11]. We offered three cases to detect such node compromise, and our web-based platform can detect intrusions based on the vault data on all of the cases. Our platform works with HTTP requests, and as of now, can't support an iterative simulation to test several cases. As a future work, we would like to automate the input selection, and run in a loop that makes several API calls per iteration to simulate the entire chain.

4 Acknowledgements

We would like to thank Bradley Zhou who supported us throughout the entire research term.

5 References

1. Liu Z., Yao C., Yu H., Wu T. Deep reinforcement learning with its application for lung cancer detection in medical Internet of Things Future Generation Computer Systems, Volume 97, 2019

2. Rebouças Filho P.P., Gomes S.L., e Nascimento N.M.M., Medeiros C.M.S., Outay F., de Albuquerque V.H.C. Energy production predication via Internet of Thing based machine learning system, Future Generation Computer Systems, Volume 97, 2019

3. M.T. Hammi, B. Hammi, P. Bellot, A. Serhrouchni, M. Tahar Hammi Bubbles of Trust: a decentralized Blockchain-based authentication system for IoT Comput. Secur., 78 (2018), pp. 126-142

4. M. Ge, J.B. Hong, S.E. Yusuf, D.S. Kim Proactive defense mechanisms for the software-defined Internet of Things with non-patchable vulnerabilities Futur. Gener. Comput. Syst., 78 (2018), pp. 568-582

5. C. Wang, Q. Wang, K. Ren, W. Lou, "Ensuring data storage security in cloud computing", Proc. of IWQoS'09, 2009.

6. M.A. Shah, M. Baker, J.C. Mogul, R. Swaminathan, "Auditing to Keep Online Storage Services Honest", 11th Workshop on Hot Topics in Operating Systems (HotOS-XI) Usenix, 2007.

7. C. Qu, M. Tao, J. Zhang, X. Hong, R. Yuan Blockchain based credibility verification method for IoT entities IEEE Transaction on Information Forensics and Security, 2018 (2018), pp. 1-11

8. H. Lulu Liang, Kai Zheng, Zilong Wei, Yanmei Wang, Sihan Wu, Xin Huang, "Model Checking of IoT System in Microgrid", Information Technology in Medicine and Education (ITME) 2016 8th International Conference on, pp. 601-605, 2016

9. Charalampos Papamanthou, Roberto Tamassia, and Nikos Triandopoulos. 2008. Authenticated hash tables. In Proceedings of the 15th ACM conference on Computer and communications security (CCS '08). ACM, New York, NY, USA, 437-448. DOI: https://doi.org/10.1145/1455770.1455826

10. Deqing Zou , Wenrong Zhang , Weizhong Qiang , Guofu Xiang , Laurence Tianruo Yang , Hai Jin , Kan Hu, Design and implementation of a trusted monitoring framework for cloud platforms, Future Generation Computer Systems, v.29 n.8, p.2092-2102, October, 2013

11. Sheeja S. Manakattu , S. D. Madhu Kumar, An improved biased random sampling algorithm for load balancing in cloud based systems, Proceedings of the International Conference on Advances in Computing, Communications and Informatics, August 03-05, 2012, Chennai, India