Cognitive Communications for Aerospace Applications Workshop (CCAAW) 2023

Securing Space Cognitive Communication with Blockchain

Authors:

Dipen Bhuva

Sathish Kumar Ph.D.

Outline

- Primary Objectives
- What is Blockchain
- Blockchain Protocols
- Related Works
- Work Flow
- Blockchain Architecture
- Primary Experiment
- Results
- Future Work
- Conclusion
- References

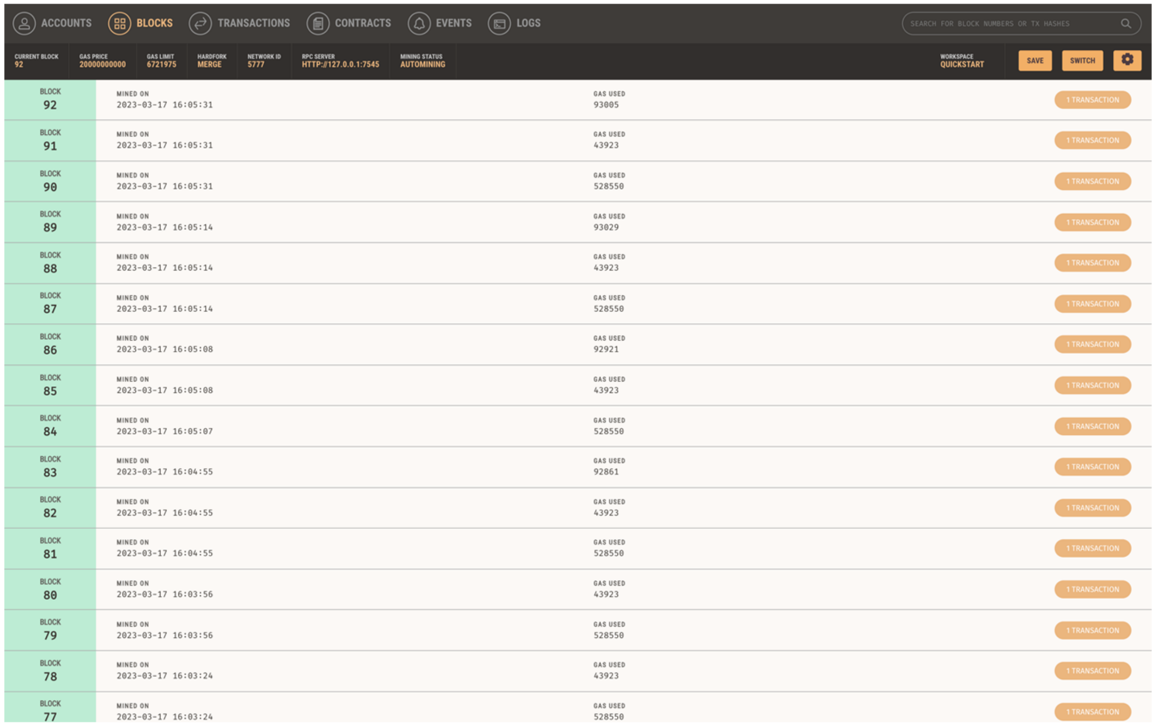
Primary Objectives

Objective of this research is to:

- Introduce blockchain-enabled Proof-of-Stake(PoS) protocol
- decentralized access control,
- decentralized authentication,
- Automatic vulnerability correction algorithm for smart contract administrators
- Enable efficient protection and provides decentralized solution for space situational awareness within space networks
- Ensuring the confidentiality and integrity of data transmitted between space nodes.

What is Blockchain

- Blockchain is a secure, transparent, decentralized ledger technology. Ethereum, a widely-used blockchain platform, allows for developing and deploying smart contracts—automated, programmable scripts that streamline processes and enforce agreements [15].



[15] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Eip-150 Revision*, 12-Apr-2017. [Online]. Available: https://www.gavwood.com/paper.pdf. [Accessed: 19-Apr-2023].

Blockchain Protocols

- **Proof of Work (PoW):** Participants solve complex mathematical puzzles to validate transactions and secure the blockchain.
- **Proof of Stake (PoS):** Participants are chosen to validate transactions based on the number of tokens they hold and are willing to "stake" as collateral.
- **Delegated Proof of Stake (DPoS):** A variant of PoS where participants elect delegates to validate transactions on their behalf.
- **Proof of Authority (PoA):** Validators are known and trusted entities authorized to validate transactions based on their identity or reputation.

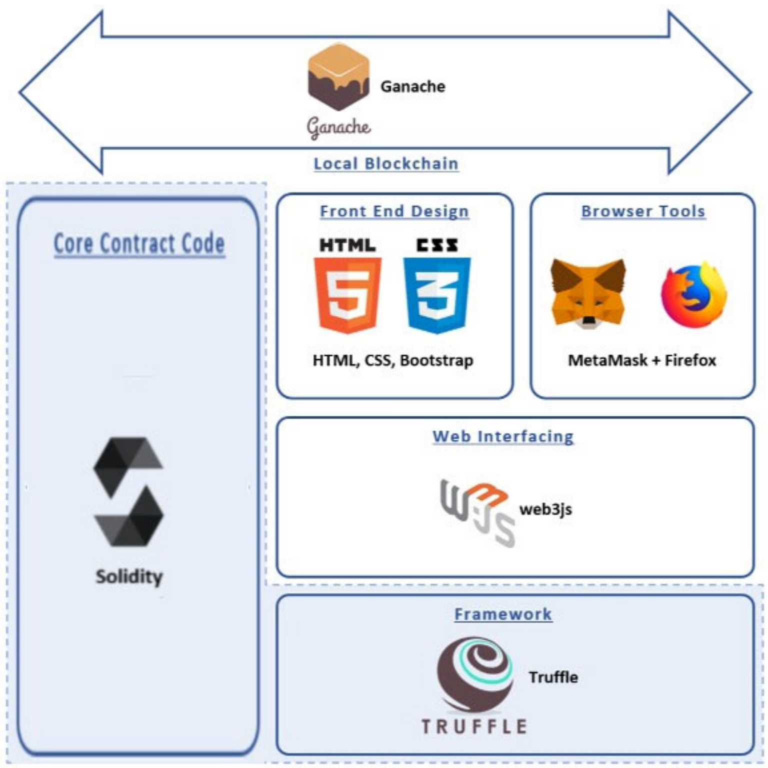
Ethereum Version 2

- Ethereum is a decentralized platform that allows developers to build decentralized applications (dApps) using smart contracts.
- Ethereum Version 2 is an Ethereum network upgrade aiming to improve scalability and security. It introduces proof of stake consensus, which is a more energy-efficient and secure alternative to the current proof of work consensus algorithm.
- Validators are chosen to add new blocks to the blockchain based on the amount of cryptocurrency they hold and are willing to lock up as collateral.

[15] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Eip-150 Revision*, 12-Apr-2017. [Online]. Available: https://www.gavwood.com/paper.pdf. [Accessed: 19-Apr-2023].

Ethereum Version 2

- Solidity is the main programming language for Ethereum smart contracts, enabling developers to build intricate applications on the platform [15]. It can be possible through various frameworks:



[15] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Eip-150 Revision*, 12-Apr-2017. [Online]. Available: https://www.gavwood.com/paper.pdf. [Accessed: 19-Apr-2023].

Smart Contracts

- Self-executing digital contract with the terms of the agreement directly written into code.
- Built on blockchain technology, typically on platforms like Ethereum, and operates automatically without the need for intermediaries.
- Secure, transparent, and tamper-proof transactions between parties.
- Only executed when specific conditions encoded in the contract are met, ensuring trust and eliminating the need for third-party enforcement.
- Smart contracts are immutable.
- High level of transparency, allowing participants to view and verify contract details.
- Various applications such as financial services, supply chain management, real estate, voting systems, and many more.

Common Vulnerabilities

Smart contracts can have vulnerabilities leading to security issues or fund loss. Common Ethereum smart contract vulnerabilities include reentrancy, integer overflow, and improper access control [16]:

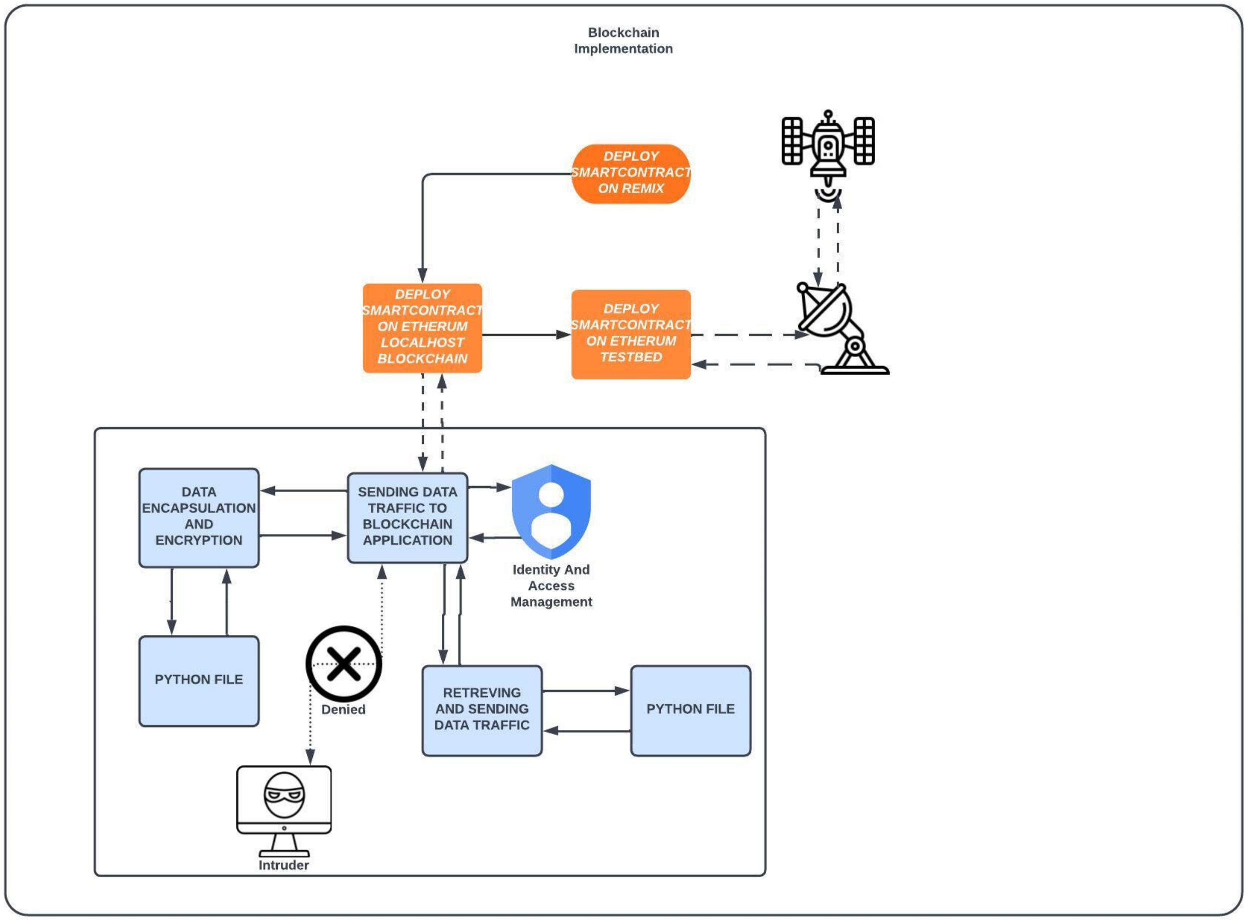
1. **Reentrancy:** Occurs when a function permits external calls to untrusted contracts before resolving, allowing attackers to repeatedly call the function and drain funds.
2. **Integer Overflow:** Happens when an operation exceeds the maximum value of its data type, causing unexpected behavior and potential security issues.
3. **Improper Access Control:** Arises when a smart contract fails to restrict access to functions or state variables, enabling unauthorized users to execute functions or alter state variables.

[16] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum Smart Contracts (SOK)," Lecture Notes in Computer Science, pp. 164–186, 2017.

Related Work

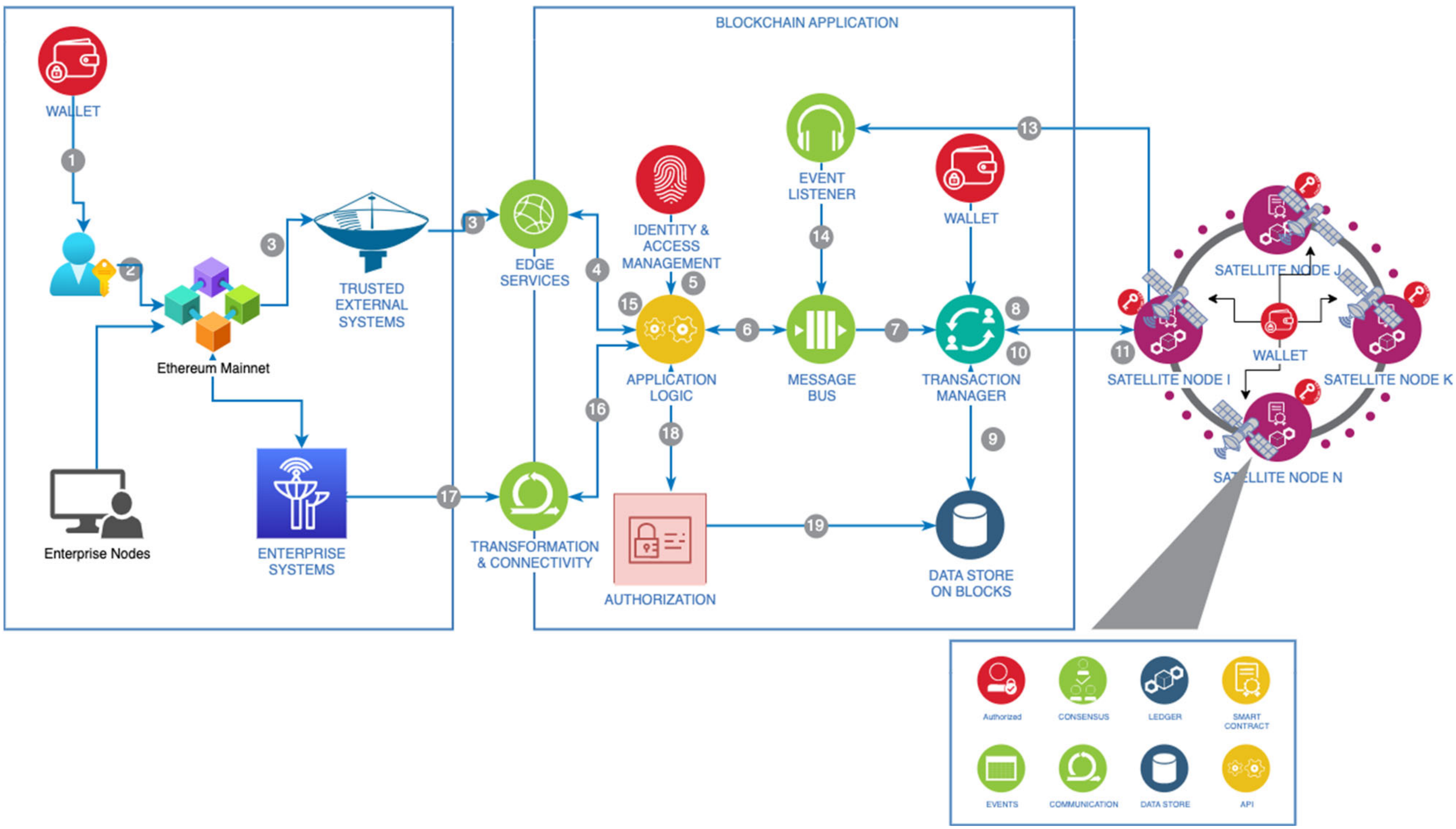
Authors	Technique Used	Advantage	Disadvantage
Ronghua Xu et. al. [2]	BlendCAC uses proof of concept protcol to verfiy identify and update each other in a trustless network environment	BlendCAC can provide a decentralized, lightweight, AC solution	The paper demonstrated Eth v1 network which is now outdated version of Eth.
Cao, S. et. al. [3]	ACID framework was introduced to resist multiple attacks while retaining all characteristics and functions of the blockchain-based SATCOM system.	Assessment in [3] that shows ACID is secure, practical, and efficient.	ACID framework should include automatic vulnerability correction tools [4] to patch security problems. The framework was tested on Eth v1 and a high-performance system.
C. Li et. al. [5]	GBS (ground base station) generates keys, nodes, parameters, and blockchains; DPC saves key parameters in tamper-proof key mechanism, allowing only GBS to identify registered sensor nodes	The proposed strategy improved satellite communications security and protection by 70% from 63% [6].	Proof-of-concept prototype tested on private bitcoin network using proof of work, requiring high computing GPU miners and degrading performance
Surdi S. et. al. [7]	The paper introduced that satellite network can employ wireless transmission and not process all nodes and it will take less time in the blockchain.	The paper proves to provide less time for transactions on blockchain or Ethereum.	Experiment not simulated, only a conceptual framework introduced; paper fails to describe blockchain network reliability and dependence on nodes within the network
Torky M. et. al. [8]	The paper uses existing SDT approach to provide security among different satellite. The approach uses asking the blockchain's last block's nonce code to establish connection between new and previous satellite.	True positive rate (TPR), true negative rate (TNR), and accuracy measures to prove its security and reliability in confirming satellite transactions	The paper fails to provide information about the experimentation Ethereum version. The paper also fails to provide an auto-patching vulnerability script for smart contracts.

Work Flow



Blockchain Implementation
DEPLOY SMARTCONTRACT ON REMIX
DEPLOY SMARTCONTRACT ON ETHERUM LOCALHOST BLOCKCHAIN
DEPLOY SMARTCONTRACT ON ETHERUM TESTBED
DATA ENCAPSULATION AND ENCRYPTION
SENDING DATA TRAFFIC TO BLOCKCHAIN APPLICATION
Identity And Access Management
PYTHON FILE
Denied
RETREVING AND SENDING DATA TRAFFIC
PYTHON FILE
Intruder

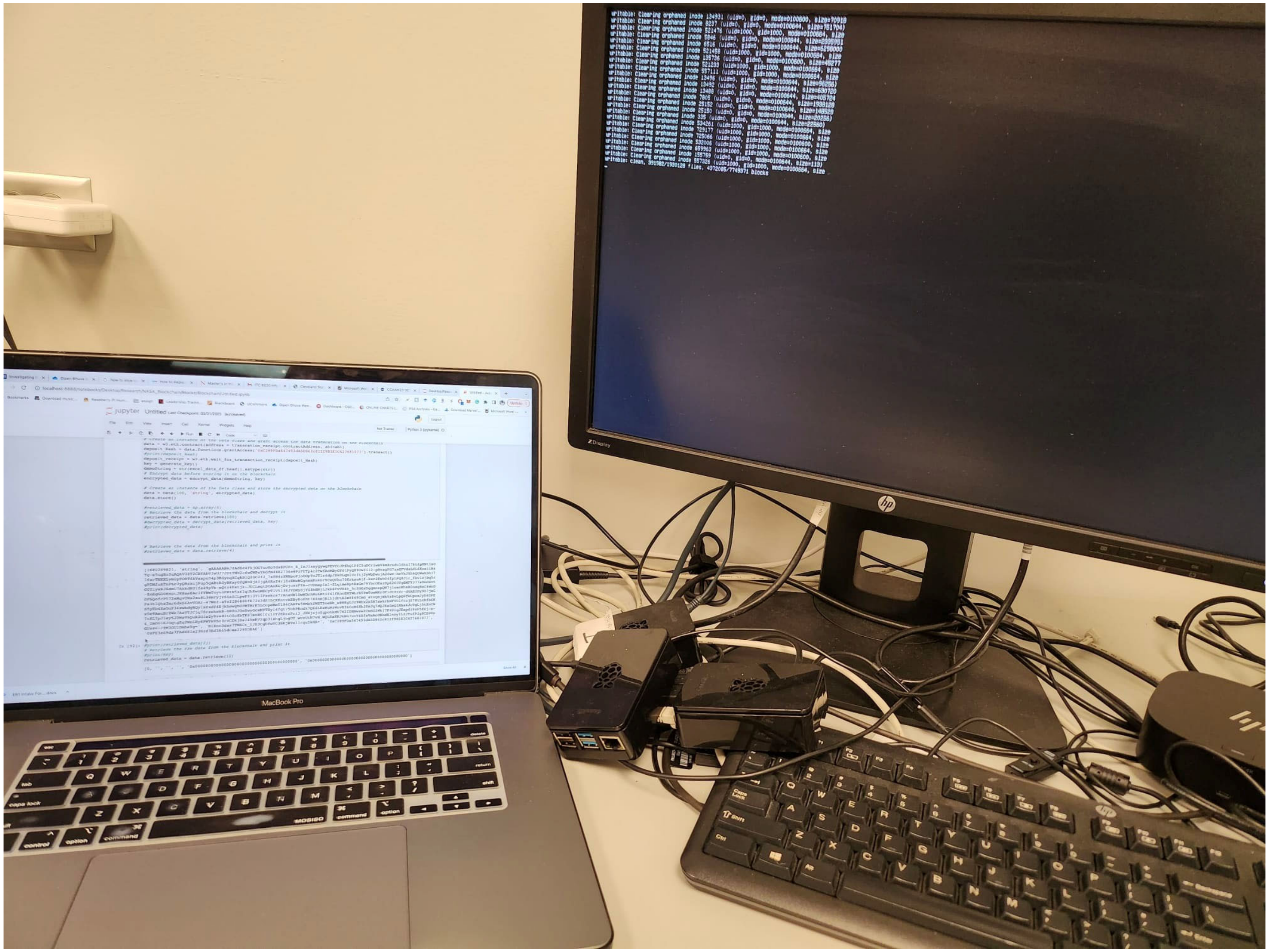
Blockchain Architecture



Primary Experiment

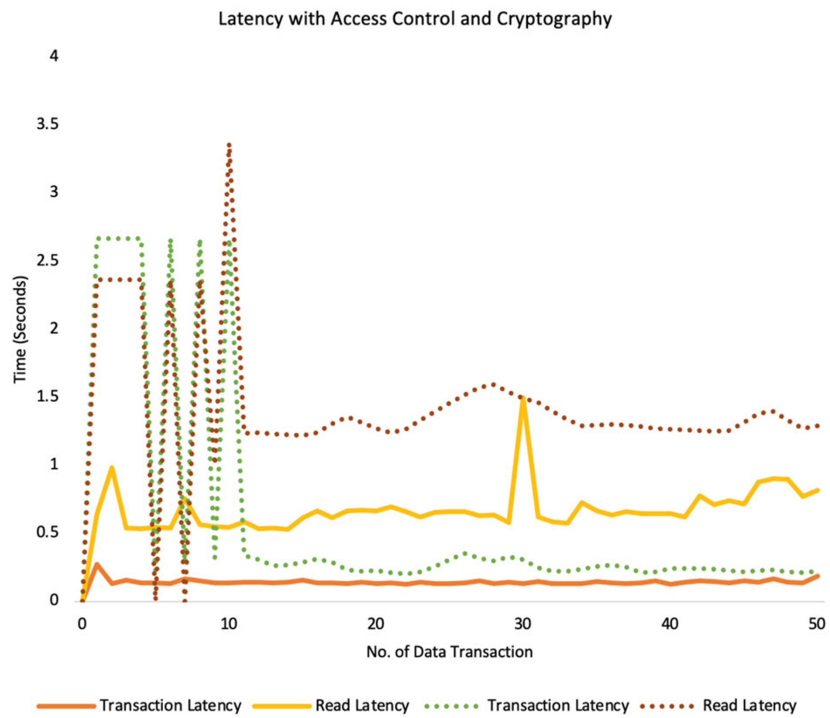
- For primary experiment we have used:
- Raspberry Pi and a 16GB RAM MacBook Pro to simulate a satellite and ground station, respectively, on a private Ethereum v2 blockchain network
- NASA-provided meteorological measurements for data transmission.
- Python Scripts for automated detection of vulnerability in smart contracts

Experimental Setup



Latency for 50 data transaction

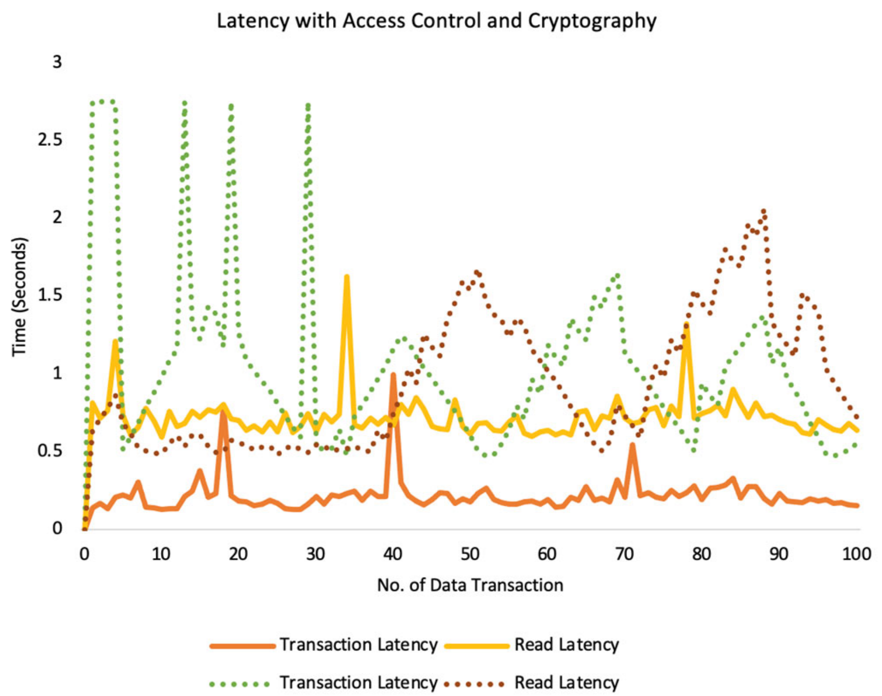
- As we can see, our approach has less latency when compared to the current to PoST.
- Our approach takes 7.16 and 33.81 seconds to send and receive, respectively, for 50 transactions.
- In contrast, it takes 29.73 and 71.81 to send and receive 50 transactions by PoST protocol with fernet encryption and decryption[8].



[8] M. Torky, T. Gaber, E. Goda, V. Snasel, and A. E. Hassanien, “A blockchain protocol for authenticating space communications between satellites constellations,” Aerospace, vol. 9, no. 9, p. 495, 2022.

Latency for 100 data transaction

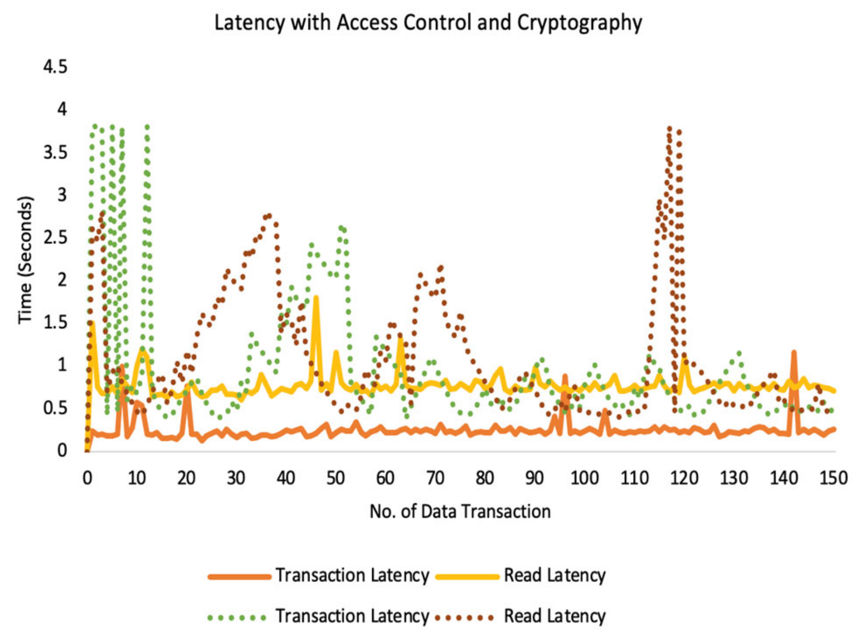
- Our approach takes 22.05 and 72.56 seconds to send and receive, respectively, for 100 transactions.
- In contrast, it takes 104.35 and 93.92 to send and receive 100 transactions by PoST protocol with fernet encryption and decryption [8].



[8] M. Torky, T. Gaber, E. Goda, V. Snasel, and A. E. Hassanien, “A blockchain protocol for authenticating space communications between satellites constellations,” Aerospace, vol. 9, no. 9, p. 495, 2022.

Latency for 150 data transaction

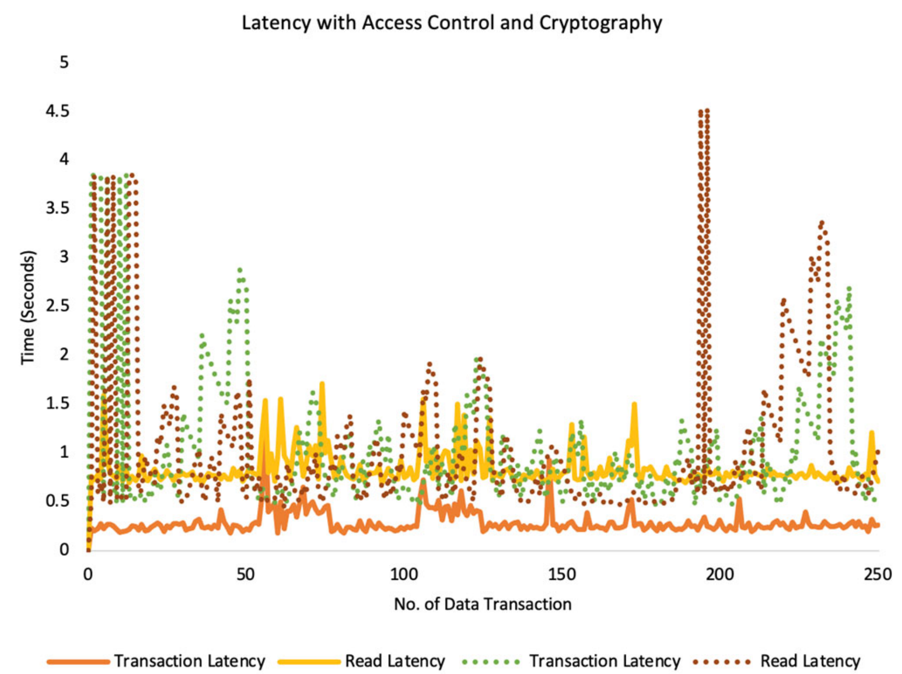
- Our approach takes 38.58 and 117.1 seconds to send and receive, respectively, for 150 transactions.
- In contrast, it takes 142.1 and 162.37 to send and receive 150 transactions by PoST protocol with fernet encryption and decryption [8].



[8] M. Torky, T. Gaber, E. Goda, V. Snasel, and A. E. Hassanien, “A blockchain protocol for authenticating space communications between satellites constellations,” Aerospace, vol. 9, no. 9, p. 495, 2022.

Latency for 250 data transaction

- Our approach takes 72.5 and 208.58 seconds to send and receive, respectively, for 250 transactions.
- In contrast, it takes 265.59 and 255.1 to send and receive 250 transactions by PoST protocol with fernet encryption and decryption [8].



[8] M. Torky, T. Gaber, E. Goda, V. Snasel, and A. E. Hassanien, “A blockchain protocol for authenticating space communications between satellites constellations,” Aerospace, vol. 9, no. 9, p. 495, 2022.

Read and Write Throughputs

- Following Table clarifies the effects of the 2049 transactional data size with respect to transaction cost (gas) on Reading Throughputs (RT) and Transaction Throughputs (TT).
- It also represents the average CPU load of raspberry pi when interacting with our blockchain architecture from a scale of 0-6.

#Satellites Trans	(RT)/RPS	(TT)/TPS	CPU Load (0-6)	Gas	Size (BYTES)
50	0.676	0.143	2.04	275,800	2049
100	0.73	0.22	2.55	275,800	2049
150	0.78	0.26	2.58	275,800	2049
250	0.85	0.29	2.60	275,800	2049

TPR, TNR, and Verification

- Following table shows confusion matrix parameters and computing True Positive Rate (TPR), True Positive Rate (TNR), and Accuracy.

#Satellites Trans	TP	TN	TPR	TNR	Accuracy
50	50	50	100%	100%	100%
100	100	100	100%	100%	100%
150	150	150	100%	100%	100%
250	150	150	100%	100%	100%

Future Work

- Incorporate a larger number of nodes, zero-knowledge proofs for enhanced privacy and optimizing the performance of the smart contract
- Simulating diverse network conditions, including nodes with limited connectivity, would help evaluate the system's resilience.
- Additionally, incorporating disconnected wallets would allow for a comprehensive assessment of the system's fault tolerance.
- Adequate number of validators in a PoS Ethereum network with respect to security, resource constraints, network latency, decentralization

Conclusion

- Introduced a Solidity smart contract for data storage and access control in a decentralized manner
- Python script for auto-patching common vulnerabilities in Solidity code.
- Secure, decentralized solution for space situational awareness, ensuring the confidentiality and integrity of data transmitted between nodes.
- Auto-patching script helps to enhance the security of the smart contract code, reducing the risk of potential vulnerabilities being exploited.
- Most efficient with regard to performance, latency, transaction throughput, read throughput, gas consumption when performing data transactions, and security.

THANK YOU

Dipen Bhuva – d.bhuva@vikes.csuohio.edu
Sathish Kumar, PhD – s.kumar@csuohio.edu

References

1) B. D. Little and C. E. Frueh, “Space situational awareness sensor tasking: Comparison of machine learning with classical optimization methods,” *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 2, pp. 262–273, 2020.

2) R. Xu, Y. Chen, E. Blasch, and G. Chen, “Exploration of blockchain-enabled decentralized capability-based access control strategy for Space Situation Awareness,” *Optical Engineering*, vol. 58, no. 04, p. 1, 2019.

3) S. Cao, S. Dang, Y. Zhang, W. Wang, and N. Cheng, “A blockchain-based access control and intrusion detection framework for Satellite Communication Systems,” *Computer Communications*, vol. 172, pp. 216–225, 2021.

4) S. Ma, F. Thung, D. Lo, C. Sun, and R. H. Deng, “VURLE: Automatic vulnerability detection and repair by learning from examples,” *Computer Security – ESORICS 2017*, pp. 229–246, 2017.

5) C. Li, X. Sun, and Z. Zhang, “Effective methods and performance analysis of a satellite network security mechanism based on Blockchain technology,” *IEEE Access*, vol. 9, pp. 113558–113565, 2021.

6) J. Yang, S. He, Y. Xu, L. Chen and J. Ren, "A trusted routing scheme using blockchain and reinforcement learning for wireless sensor networks", *Sensors*, vol. 19, no. 4, pp. 1-19, 2019.

7) S. A. Surdi, “Space situational awareness through Blockchain Technology,” *Journal of Space Safety Engineering*, vol. 7, no. 3, pp. 295–301, 2020.

8) M. Torky, T. Gaber, E. Goda, V. Snasel, and A. E. Hassanien, “A blockchain protocol for authenticating space communications between satellites constellations,” *Aerospace*, vol. 9, no. 9, p. 495, 2022.

9) N. A. NA, “A python interface for interacting with the Ethereum blockchain and ecosystem.,” *interacting with the Ethereum blockchain and ecosystem.*, 2023. [Online]. Available: https://web3py.readthedocs.io/en/latest/#. [Accessed: 14-Apr-2023].

References

10. P. Merriam, “Py-SOLC-X,” *solc*, 2016. [Online]. Available: https://solcx.readthedocs.io/en/latest/. [Accessed: 14-Apr-2023].
11. N. A. NA, “Fernet (symmetric encryption),” *Fernet (symmetric encryption) - Cryptography 41.0.0.dev1 documentation*, 2023. [Online]. Available: https://cryptography.io/en/latest/fernet/. [Accessed: 14-Apr-2023].
12. Behrenfeld, M. (2022). *NAAMES Sonde meteorological* . NASA. Retrieved April 14, 2023, from https://data.nasa.gov/dataset/NAAMES-Sonde-Meteorological-InSitu-Data-Version-1/vuj8-3hmw
13. N. A. NA, “Goerli/medalla: Ethereum 2.0 multi-client testnets,” *GitHub*, 2022. [Online]. Available: https://github.com/goerli/medalla. [Accessed: 15-Apr-2023].
14. I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, “Next Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A survey,” *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.
15. G. Wood, “Ethereum: A secure decentralised generalised transaction ledger,” *Eip-150 Revision*, 12-Apr-2017. [Online]. Available: https://www.gavwood.com/paper.pdf. [Accessed: 19-Apr-2023].
16. N. Atzei, M. Bartoletti, and T. Cimoli, “A survey of attacks on Ethereum Smart Contracts (SOK),” Lecture Notes in Computer Science, pp. 164–186, 2017.


ANY
QUESTIONS?