# Review of “Effective Blockchain-based Data Storage Scheme in Zero Trust” by Jin Wang [?]

Gagan Chowdary Chatu*,
CS5378 Advanced Computer Security
Texas State University, San Marcos, Texas, USA
Email: *akl103@txstate.edu

Internet of Things (IoT) is a large-scale, heterogeneous, and dynamic distributed network for devices. Blockchain is distributed ledger technology popularly used for crypto-currency and security. blockchain is a chain of blocks that are immutable. Zero trust is a security model where no entity is trusted. All entities must be authenticated before granting access to the system. Since blockchain is distributed in nature there are no need for third parties to trust. This makes blockchain an appropriate choice for the zero trust model. The efficiency of blockchain is far from application requirements of IoT devices. This paper talks about an effective way of storing information generated by IoT devices using zero trust security model and blockchain technology.

Zero trust IoT (ZIoT), is a security framework that combines IoT with blockchain [1]. Since zero trust follows never trust always verify policy there is a high verification overhead. Because of the extra overhead, throughput is extremely low. To improve throughput, a concept called Sharding is used. Sharding is a technique used to divide a single entity into shards and each shard can be processed in parallel. In blockchain we divide chain into different sub chains. Since we can use parallel processing through put is improved [2]. The verification method used in traditional blockchain is Merkle proof. It uses a data structure called Merkle tree. The proof size of Merkle tree is huge and it cannot be parallel processed. When we apply for zero trust architecture, frequency of verification is high. With high proof size, there would be network delays. This paper suggests the use of Insertable Vector Commitments (IVC) instead of Merkle tree. Three main contributions of this paper are as follows:

- Since traditional blockchain is inherently open and anyone can join, we need to secure it using user authentication.
- Use of sharding is proposed to increase throughput.
- Merkle proof is heavy. So, to decrease proof size, use of Insertable Vector Commitment (IVC) is proposed.

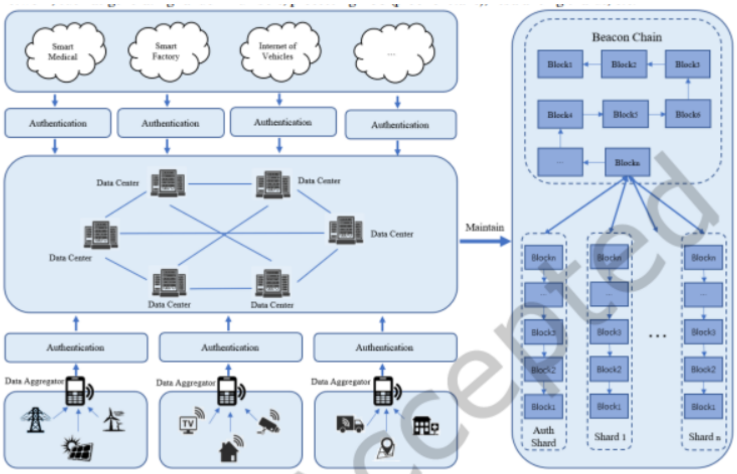

Fig. 1: Framework of S-BDS [?]

Commitment is one of the most basic cryptographic protocols. It can be seen as a sealed letter. Once a message is committed, it can be available to the public but only to those who knows the hiddenness of the message can open it. Vector commitment is a special type where it commits a vector of length n and can open commitment at any position. Merkle tree is also considered as a vector commitment. The root of the tree is the commitment value, and the message is at the lead nodes. The only downside of Merkle tree is that proof of verification is heavy and cannot be parallelly processed.

As visualized in Figure 1, S-BDS in ZIoT is unlike traditional blockchain, Users need to be authenticated to access the data of storage nodes. To improve efficiency, we divide the network into several sub networks. This allows a transaction to be divided and processed individually. This allows us to use parallel processing to increase the speed of the transaction. The authentication chain will be responsible for user login information. The Beacon chain is responsible for operations like generating random numbers, POS (proof of stake) rebalancing shards etc.

The Authentication chain generates a public key and private key for registered users and devices. This chain records public key and private key is stored at user. It uses a 2-step process of asymmetric encryption on addresses

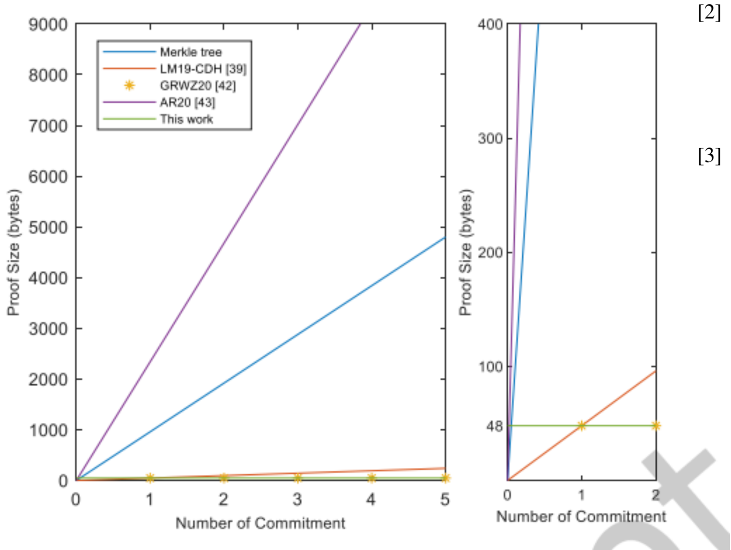

Fig. 2: Comparison of proof size in blockchain **[?]**

and randomly generated numbers between the user and the system to authenticate user. Post authentication, node can upload data to the blockchain. Data uploaded is encrypted using the user's public key and can only be decrypted by the user's private key. Blocks are generated by consensus algorithm. Where it generates key value pairs and calls the insert algorithm to generate commitment and stores it in the block header and transactions are stored in the block body. Open algorithms are used to generate proof for data integrity and correctness. In this step data, storage nodes convince the users that data need to be validated. So that user decrypts and checks whether the data they uploaded is the one that is stored in the blockchain. After verification is completed, the data may be stored in different shards. Aggregate algorithms such as SCA.Agg and MCA.Agg are used to aggregate proof [3]. They reduce the communication overhead. From Figure 2, we can determine that the performance of IVC against other data structures is much better than others.

In conclusion, Use of IVC reduces proof size of verification and improves communication delay. Sharding helps in parallel processing and data aggregation techniques improve efficiency of the ZIoT system. Further research is required on making sure that there are no limits on number of users. There is a need of a system that protects from DDoS attacks.

## References

[1] S. Zhao, S. Li, F. Li, W. Zhang, and M. Iqbal, "Blockchain-enabled user authentication in zero trust internet of things," in *Security and Privacy in New Computing Environments*, D. Wang, W. Meng, and J. Han, Eds. Cham: Springer International Publishing, 2021, pp. 265–274.

[2] M. Zamani, M. Movahedi, and M. Raykova, "Rapidchain: Scaling blockchain via full sharding," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 931–948. [Online]. Available: https://doi.org/10.1145/3243734.3243853

[3] S. Gorbunov, L. Reyzin, H. Wee, and Z. Zhang, "Pointproofs: Aggregating proofs for multiple vector commitments," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 2007–2023. [Online]. Available: https://doi.org/10.1145/3372297.3417244