International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

SLIDING WINDOW BLOCKCHAIN ARCHITECTURE FOR INTERNET OF THINGS

Gadde Ajay Kumar (MCA Scholar), B V Raju College, Vishnupur, Bhimavaram, West Godavari District, Andhra Pradesh, India, 534202.

K. R. Rajeswari, B V Raju College, Vishnupur, Bhimavaram, West Godavari District, Andhra Pradesh, India, 534202.

Abstract- Internet of Things (IoT) refers to the concept of enabling Internet connectivity and associated services to nontraditional computers formed by integrating essential computing and communication capability to physical things for everyday usage. Security and privacy are two of the major challenges in IoT. The essential security requirements of IoT cannot be ensured by the existing security frameworks due to the constraints in CPU, memory, and energy resources of the IoT devices. Also, the centralized security architectures are not suitable for IoT because they are subjected to single point of attacks. Defending against targeted attacks on centralized resources is expensive. Therefore, the security architecture for IoT needs to be decentralized and designed to meet the limitations in resources. Blockchain is a decentralized security framework suitable for a variety of applications. However, blockchain in its original form is not suitable for IoT, due to its high computational complexity and low scalability. In this paper, we propose a sliding window blockchain (SWBC) architecture that modifies the traditional blockchain architecture to suit IoT applications. The proposed sliding window blockchain uses previous (n 1) blocks to form the next block hash with limited difficulty in Proof-of-Work. The performance of SWBC is analyzed on a real-time data stream generated from a smart home testbed. The results show that the proposed blockchain architecture increases security and minimizes memory overhead while consuming fewer resources.

Index Terms—Blockchain, Internet of Things, smart home, security, sliding window

1. INTRODUCTION

Blockchain is a distributed ledger used to record transactions between two or more parties. Unlike relational database systems, blockchain is a data structure where new entries get appended at the end of the ledger, and there exist no administrator permissions within a blockchain which allow modification of the data. Also, the addition of a new block to the chain needs to be verified by all other parties through a consensus algorithm. Since there exists a distributed control over the blockchain, it is difficult for attackers to modify the data compared to a relational database system. Relational databases are primarily designed for centralized data storage and blockchain are specifically designed for decentralized data storage. There exist two types of blockchains: (i) permissioned and (ii) permissionless. A permissioned blockchain is a private blockchain which requires pre-verification of the participants within the network who are assumed to know each other whereas, a permissionless

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
IJARST
ISSN: 2457-0362

blockchain is a public blockchain [1]. Traditional blockchain approach is not suitable for IoT with real-time data streams due to their computationally complex Proof-of-Work (PoW) [2]. As the computational time increases, blockchain security becomes infeasible to be used for IoT.

The two major challenges involved in applying blockchain to IoT environments include: (i) computational complexity and (ii) scalability. The computational complexity depends on

difficulty level and Merkle tree size. Merkle tree is a tree in which every leaf node is labeled with the hash of a transaction data and every non-leaf node is labeled with the cryptographic

hash of the labels of its child nodes. Merkle tree grows with the number of transactions made and, thereby, increasing the time consumed for Proof-of-Work, which is less favorable for

an IoT network. Scalability refers to the limits on the number of transactions a blockchain can process within a specific time period. Bitcoin is a popular example of a blockchain. Bitcoin blockchain is a payment system that does not rely on a central authority to secure and control its money supply. Each block in a Bitcoin blockchain has limited block size. In Bitcoin, the block size is limited to 1 MB and a block is mined every ten minutes. Interestingly, the existing literature [3] suggests blockchain as one of the data security and privacy algorithms that can be implemented for IoT applications due to its distributed architecture. In this paper, we propose a new blockchain architecture for IoT environments, especially in the context of smart home applications. A smart home monitors, analyzes, and reports the state of the home. Smart homes use devices connected to IoT to automate and monitor in-home systems [4]. Smart home can be considered as the smallest unit of a smart city. The security standardization of a smart home supports a smart city and vice versa. In a smart home, the real-time data streams are generated

by sensors which help us to monitor the current status of the home, analyze energy consumption, and investigate any accidents inside a smart home. The volume of data generated by a smart home depends on the number of sensors deployed and the frequency of data acquisition. Therefore, proper sampling of sensor data is required to produce meaningful information which can be later stored in the blockchain. The volume of data stored in a blockchain decides the packet overhead, memory overhead, and computational overhead. In this context,

our proposed sliding window blockchain architecture tries to improve the security and reduce the memory overhead of IoT in a smart home environment.

ARCHITECTURE

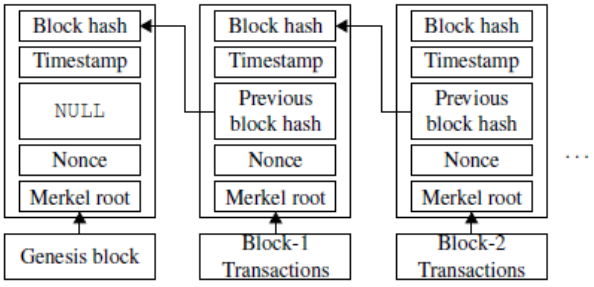

Figure 1: Blockchain architecture.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
IJARST
ISSN: 2457-0362

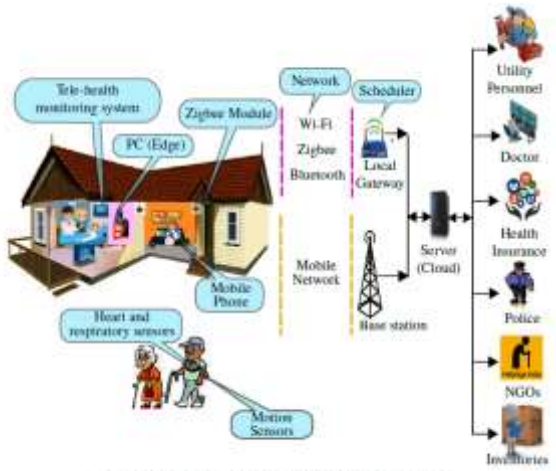

Figure 2: A typical smart home system for assisted living.

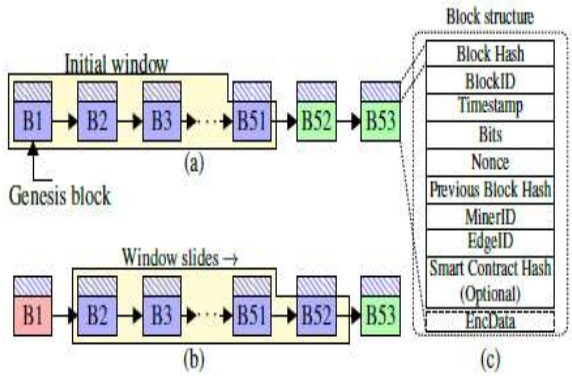

Figure 3: Sliding window blockchain.

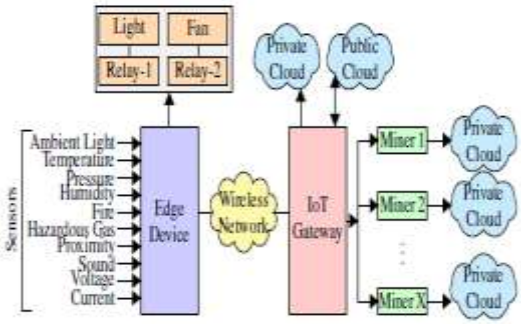

Figure 4: Smart home testbed used for studying sliding window blockchain.

2.SYSTEM STUDY

2.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

ECONOMICAL FEASIBILITY

TECHNICAL FEASIBILITY

SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
IJARST
ISSN: 2457-0362

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3. SYSTEM TEST

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

3.1 TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
IJARST
ISSN: 2457-0362

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or –

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
IJARST
ISSN: 2457-0362

one step up – software applications at the company level – interact without error.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

4.CONCLUSION

IoT devices face constraints on resources such as computational capability, energy sources, and memory. Therefore, the standard security algorithms are not feasible for IoT. We proposed a sliding window blockchain that meets the requirements of a resource constrained IoT network by reducing the memory overhead and limiting the computational overhead. The memory overhead is reduced by storing only a limited part of the blockchain, as defined by the sliding window size

in the IoT device and maintaining the whole blockchain in the private cloud. Computational overhead is limited by using the difficulty level between 1 and 5 and by eliminating the Merkle

tree. The security is increased by generating the block hash using the properties of n blocks in the sliding window. A false miner cannot mine a block unless he gets the previous (n□1) blocks and the window size information. From the experimental results, we observed the following: (i) The computational time of PoW for each level of difficulty increases exponentially. (ii) The total block addition time increases with the increase in the number of miners in the group. (iii) As the window size increases, the hash computation time increases linearly. (iv) A random selection of difficulty for each block in a blockchain reduces the total block addition time. Future work can be carried out to analyze the impact of a variable size sliding window. New consensus algorithms can be developed to suit the IoT environment. Furthermore, energy consumption of the blockchain can also be analyzed to draw more insights on energy resources required for an IoT device.

5.REFERENCES

[1] S. Kulkarni, "The beauty of the blockchain," Open Source for You, vol. 06, pp. 22–24, June 2018.

[2] T. M. F. Carames and P. F. Lamas, "A review on the use of blockchain for the Internet of Things," IEEE Access, vol. 6, pp. 32 979–33 001, May 2018.

[3] A. Dorri, S. S. Kanhere, and R. Jurdak, "Blockchain in Internet of Things: challenges and solutions," arXiv preprint arXiv:1608.05187, August 2016.

[4] IoT Agenda, "Smart home or building," April 2018. [Online]. Available: https://internetofthingsagenda.techtarget.com/definition/ smart-home-or-building

[5] L. Jiang, D. Y. Liu, and B. Yang, "Smart home research," in Proceedings of 2004 International Conference on Machine Learning and Cybernetics, vol. 2, August 2004, pp. 659–663.

[6] theinstitute.ieee.org, "Towards a definition of the Internet of Things (IoT)," May 2015. [Online]. Available: https://iot.ieee.org/images/files/pdf/IEEE IoT Towards Definition Internet of Things Revision1 27MAY15.pdf

[7] J. Wan, X. Gu, L. Chen, and J. Wang, "Internet of Things for ambient assisted

living: Challenges and future opportunities," in International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), October 2017, pp. 354–357.

[8] D. Abbasinezhad-Mood, A. Ostad-Sharif, and M. Nikooghadam, "Novel anonymous key establishment protocol for isolated smart meters," IEEE Transactions on Industrial Electronics, vol. 67, no. 4, pp. 2844–2851, April 2020.

[9] S. K. Das, D. J. Cook, A. Battacharya, E. O. Heierman, and T. Y. Lin, "The role of prediction algorithms in the MavHome smart home architecture," IEEE Wireless Communications, vol. 9, no. 6, pp. 77–84, December 2002.

[10] C. Qu, M. Tao, J. Zhang, X. Hong, and R. Yuan, "Blockchain based credibility verification method for IoT entities," Security and Communication Networks, vol. 2018, pp. 1–11, June 2018.