The Applicability of Blockchain in the Internet of Things

Yash Gupta
Department of Computer Science and Engineering
Indian Institute of Technology Delhi
New Delhi, India

Rajeev Shorey, Devadatta Kulkarni and Jeffrey Tew
Tata Consultancy Services (TCS) Innovation Labs
Tata Consultancy Services
India/USA

***Abstract*—In this paper, we address the applicability of Blockchain technology to ensure security of data transmitted and received by the nodes in an Internet of Things (IoT) network. We propose a Blockchain consensus model that is suitable for resource constrained devices. We also propose a model for implementing IoT security on top of the Blockchain model. We simulate our proposed model to understand its feasibility.**

***Index Terms*—blockchain, IoT security, distributed consensus**

I. Introduction

Securing data on Internet of Things (IoT) is a challenging issue faced by the industry. Global spending on IoT security is expected to grow at an annual rate of 25%, reaching up to 900 million USD in 2020 [1]. Further, Gartner projects that more than 25% of all identified cyber attacks in enterprise environments will involve IoT [1]. Securing IoT networks is a huge challenge due to its distributed, heterogeneous and resource-constrained nature ([4], [5]).

Blockchain is a data structure that enables creation of a tamper-proof distributed ledger in a peer-to-peer setting [2]. Primarily popularized by the peer-to-peer electronic cash system, Bitcoin [3], such a distributed ledger coupled with cryptographic primitives such as hash functions, symmetric encryption, asymmetric encryption and Merkle trees can be used to provide the key security properties of Confidentiality, Integrity and Availability to data in a distributed system [2].

Due to the distributed nature of IoT networks, blockchain based distributed ledger technology could be explored to provide a secure, tamper-proof IoT network without any single point of failure. The work in this paper is a preliminary attempt to understand the applicability of blockchain in an IoT system.

II. Problem

Our objective is to propose a mechanism to provide confidentiality, integrity and availability to data transmitted and received by nodes in an IoT network using blockchain technology. The key contribution of our work is a blockchain based security model which is feasible on resource constrained IoT nodes.

III. Proposed solution

We propose a security model for IoT backed by a blockchain based distributed ledger. We borrow the concept of a token tracked by the blockchain from Bitcoin [3], but instead of holding any monetary value, tokens in our system decide distribution of voting power across nodes and rate-limit certain transactions to prevent Denial-of-Service (DoS) attacks.

We introduce a blockchain protocol layer and a blockchain application layer to the base IoT architecture layers, as shown in Fig. 1. We discuss this in detail in the next section.

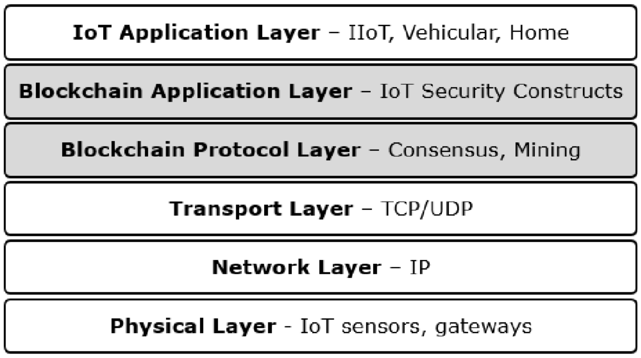
IoT Application Layer – IIoT, Vehicular, Home
Blockchain Application Layer – IoT Security Constructs
Blockchain Protocol Layer – Consensus, Mining
Transport Layer – TCP/UDP
Network Layer – IP
Physical Layer - IoT sensors, gateways

Fig. 1. The Proposed Layered Architecture

A. Blockchain Protocol Layer

The blockchain protocol layer encompasses the consensus algorithm for nodes in the network. We define certain categories of messages in this layer to help achieve a common view of the blockchain among all participating nodes.

- *hello(net_id, from, to), hello_ack(peer_list)* - Allows the nodes to discover peers.
- *get_block_inventory(locator)* - Request for the inventory of blocks available with a node's peers, with ”locator” summarizing its own block inventory.
- *get_tx_inventory()* - Request for transactions in the mempool (i.e., not yet mined into a block)
- *block(header, tx_list), block_header(hash, timestamp, miner, merkle_root)*
- *transaction(input_list, output_list)* - Transactions have a list of inputs which it is spending, and a list of outputs which it creates. Certain categories of transactions may be allowed without any input to create or ”mine” new tokens.

In order to ensure that all IoT nodes have a uniform view of the blockchain, we define the rules followed by the blockchain protocol layer so as to achieve consensus. A summary of these rules is as follows.

- On receiving a transaction, each node stores it in its memory pool and broadcasts it further.
- Once every clock tick , a node tries to mine a new block based on its mining_token balance b, difficulty d, timestamp t, new block's merkle root m and mining condition [6]:

$$hash(prev\_blk|m|t) \leq elapsed\_time \times b/d$$

- Upon generating a new block, the node updates its view of the blockchain and broadcasts the block to its peers.
- The difficulty value for the next block is updated such that average expected time for finding the next block (taken over some fixed number of last blocks) is equal to a set value.
- Upon receiving a new block, the node verifies that the block satisfies all protocol rules, and that all transactions in the block follow the protocol and application rules.
- After verifying a received block, the node updates its view of the blockchain, clears included transactions from the mempool, and broadcasts the block further.
- Eventually, all nodes have a uniform view of the blockchain ([3]).

B. Blockchain Application Layer

The blockchain application layer defines the IoT security specific transactions and their semantics for the higher layers. The core function provided by our proposed application model is authentication and authorization for devices on IoT networks.

We define a node in the IoT network by the tuple $(id, K_{pu}, K_{pr}, \Pi_{nonce,firmware,K_{pu}})$, where (K_{pu}, K_{pr}) is the public-private key pair for the node, id is a shorter version of the public key; $\Pi_{nonce,firmware,K_{pu}}$ is a proof-of-firmware generated by the node using hardware root of trust (such as Physically Unclonable Functions (PUFs)), to prevent Sybil attacks on the network.

$\Pi_{nonce,firmware,K_{pu}}$ is a function of the private key, the firmware contents and a nonce value derived from the latest block in the blockchain. This is to provide dynamic authentication and avoid replay attacks on the network.

We define the following transactions for the blockchain application layer:

- join_net($id, K_{pu}, \Pi_{nonce,firmware,K_{pu}}$), leave_net(...)
  A node can join the blockchain with a *join_net* transaction. *join_net* is allowed to have a sister output of *pay_token* which pays a set number of token to the newly joined nodes id. These newly issued tokens are used by the node for all further actions which require the node to be joined to the network.
  leave_net indicates that the node has left the network, and must spend all the tokens issued by *join_net*.
- *begin_session(id$_A$,id$_B$,), end_session(...)*
  These transactions are used by a member node to initiate and end authenticated and authorized communication session with another node in the network. *begin_session* must spend at least 1 token issued by the *join_net* of the session initiating node. This spent token is then either released by a corresponding *end_session* tx or after a set timeout in number of blocks.
- *add_to_group(id, group), remove_from_group(...)*
  These transactions are used to add a node to a group for access control management.
- *add_rule(group, resource, action), remove_rule()*
  These transactions are used to add/remove domain specific access control rules for the IoT node sessions.
- *pay_token(id, amount, type)*
  This transaction is used to pay tokens of a given "type" to a node id. New tokens can only be generated along with a join_net() transaction.
- *pay_mining_token(id, amount, type)*
  Mining tokens are a special token type which grant mining privilege to the nodes which hold them. Mining tokens reduce the difficulty of mining blocks for a node. Only the first block of the blockchain can generate new mining tokens, which are then transferred to different nodes in subsequent blocks for decentralization of trust.

IV. Simulation Results

We build a simulation (Fig. 2) of the proposed model using OMNET++ [7] to evaluate the feasibility and performance of our proposed model. We define the following metrics to measure the feasibility of our proposed model:

A. Metrics

- F_{tx} – Transactions added to the blockchain per second
- F_{blk} – Blocks added to the blockchain per second
- $M_{mempool}$ – Memory space utilized by the transaction memory pool (i.e., transactions that are not yet included in a block).

B. Assumptions

We make certain simplifying architectural assumptions in the simulation, which we list below for each abstraction layer:

- IoT Application Layer - Nodes initiate and end sessions with inter-transaction time sampled from an exponential distribution with a given mean.
- Blockchain Protocol Layer - Inner nodes of the network are involved in generating new blocks, while leaf nodes only verify the blocks received by them.
- Transport Layer - We assume that the underlying transport is reliable with packet queuing.
- Network Layer - We assume a tree topology for the network.
- Physical Layer - We assume wireless links have 54 Mbps bandwidth and 1.0 ms delay, whereas Ethernet links have 1 Gbps bandwidth and 0.1 ms delay.

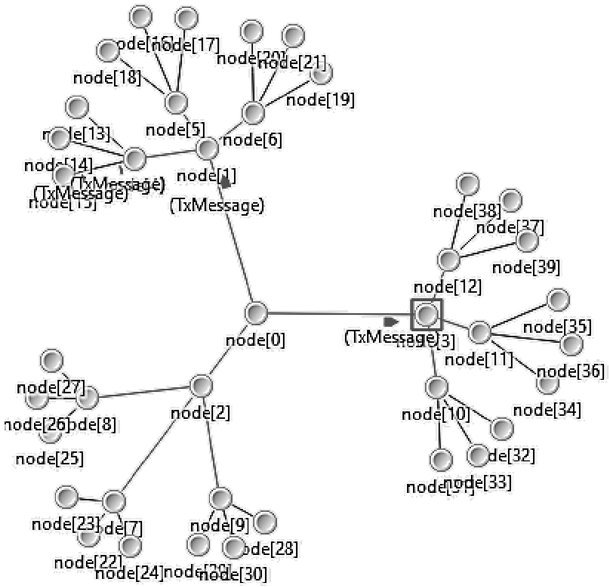

Fig. 2. Topology of the 40-node network we simulate

C. Observations

We simulate the IoT nodes connected in a tree topology with a branching factor 3 for all our simulation runs. We show block and transaction arrival rate for different number of nodes using 1 Gbps Ethernet links in Fig. 3, Fig. 4 and Fig. 5.

We observe from our experiments that the transaction mempool shows high variability, but is bounded by a set maximum size. This ensures that we use a constant amount of working memory.

We observe that block arrival rate is unstable for a long time (400 seconds) on a 4 node wired network due to high variance. The network is unable to sustain high transaction throughput, as new transactions keep on arriving but no blocks may be found for long periods. The block arrival rate stabilizes much faster (in first few seconds) in 13 node and 40 node networks. Consequently, this results in a stable transaction arrival rate.

We observe that block size is much more stable in case of 40 node network as compared to smaller networks, as the bigger network allows for larger fan-in of transactions to the network. This results in the transaction memory pool to be consistently filled with about 1000 transactions, which can be placed in to the next block. This also results in a higher throughput of 800 tx/sec achieved in case of 40 node network, as compared to 500 tx/sec in case of a 13 node network.

We also simulate a 13 node network with WiFi links (Fig. 6). In this case, the block arrival time is not affected much by the increased latency and reduced bandwidth, as block difficulty adjustment adapts dynamically to the network conditions. Transaction rate is also similar to the 13 node network with wired links. Therefore, substituting wired links with lossy WiFi links does not affect the network operation.

We are able to conclude from the simulation studies that the blockchain based model is stable and able to process transactions at acceptable rates when the network has a significant number of nodes (e.g., 13 or more). In addition, we observe that the performance of the model does not degrade significantly with wireless links that are characterized by higher latency and lower bandwidth.

V. Conclusion

Blockchain is a relatively unexplored area in the IoT security space, and we show that it is a viable solution to the IoT security problem. The key properties of tamper-resistance and decentralized trust allow us to build a secure authentication and authorization service which does not have a single point of failure. It is to be noted that the work in this paper is a preliminary attempt to understand the implementation challenges of blockchain in an IoT network. At this point in time, we do not have detailed results on the scalability or the performance of blockchain in an IoT network.

The key contribution of this work is in application of blockchain to provide an authentication and authorization service in IoT networks to:

- Secure the network from remote and local adversaries
- Provide visibility in the form of blockchain history of active nodes and sessions
- Detect and prevent outlier behavior from certain nodes

Further, we validate the stability and performance of our proposed model using simulations. We also show that our model's stability or performance does not degrade significantly on a lossy wireless network.

VI. Future Work

IoT networks have vast attack surfaces and vulnerabilities. The work in this paper could be extended to cover additional attack surfaces and support specific IoT deployments such as (i) firmware update delivery, (ii) firmware tampering detection, and (iii) spurious component detection.

The proposed model could be modified to work over Bluetooth Low Energy (BLE) links, unreliable transport layer protocols such as UDP, and optimized to meet performance requirements for a real-time system.

References

[1] Gartner. ”Report on IoT security spending”, Gartner Newsroom. 2016.

[2] ScorexFoundation. ”A treatise on Blockchain concepts + Scorex 2.0 tutorial.” https://github.com/ScorexFoundation/ScorexTutorial, 2017.

[3] S. Nakamoto. ”Bitcoin: A peer-to-peer electronic cash system.” https://bitcoin.org/bitcoin.pdf, 2008.

[4] Open Web Application Security Project. ”Internet of Things Project - Attack surfaces.” https://www.owasp.org/, 2017

[5] Abera, Tigist, et al. ”Things, trouble, trust: on building trust in IoT systems.” 53rd Annual Design Automation Conference. ACM, 2016.

[6] V. Buterin. ”On Stake” https://blog.ethereum.org/, 2014

[7] A. Varga and R. Hornig. ”An overview of the OMNeT++ simulation environment.” Simulation tools and techniques for communications, networks and systems & workshops, Simutools 08, 2008.

[8] Dorri, Ali, et al. ”Blockchain for IoT security and privacy: The case study of a smart home.” IEEE International Conference on Pervasive Computing and Communications Workshops, 2017.

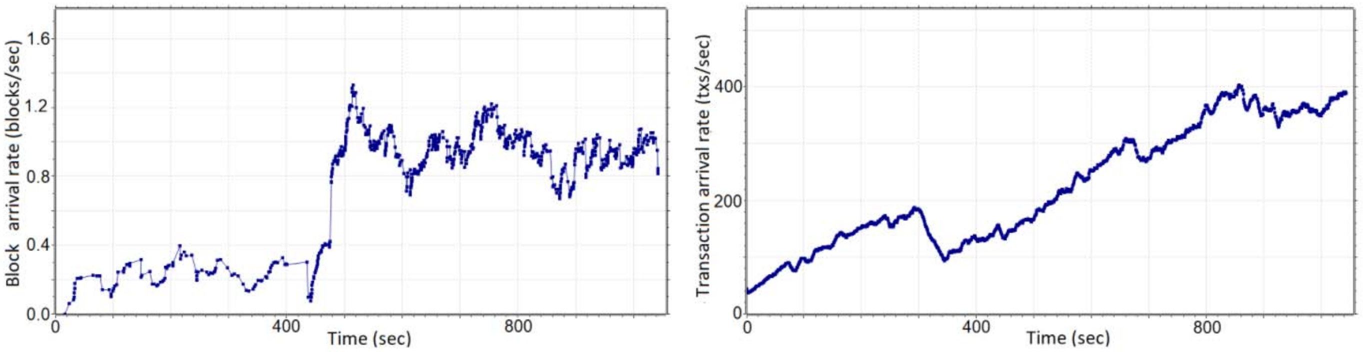

Fig. 3. Block arrival rate and transaction arrival rate v/s time for simulation of a 4 node network over Gigabit Ethernet

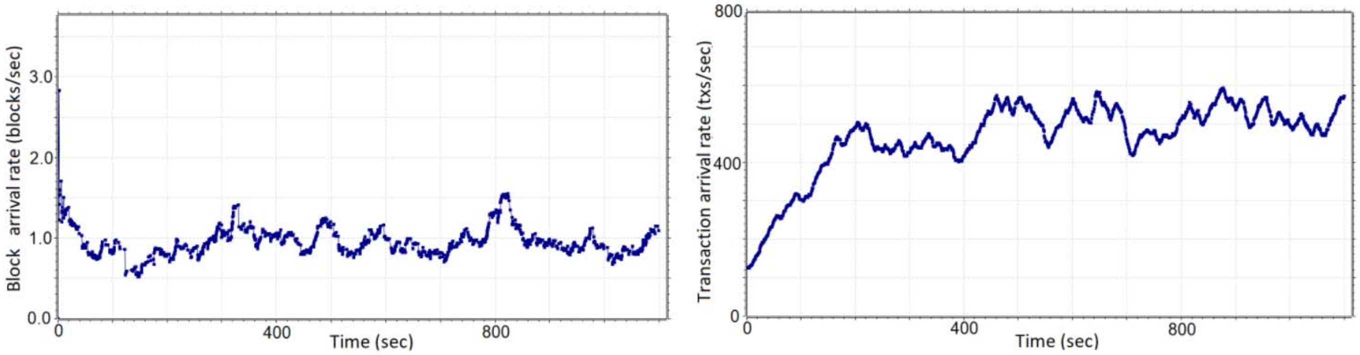

Fig. 4. Block arrival rate and transaction arrival rate v/s time for simulation of a 13 node network over Gigabit Ethernet

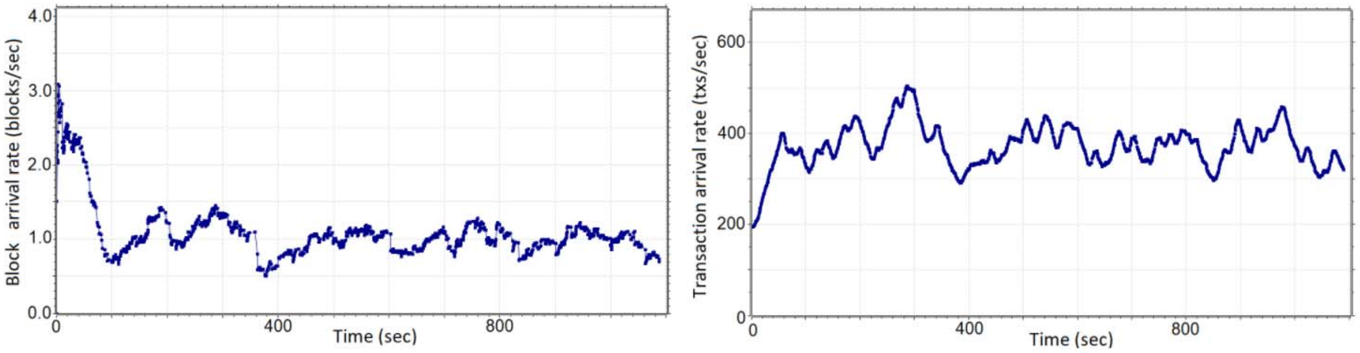

Fig. 5. Block arrival rate and transaction arrival rate v/s time for simulation of a 40 node network over Gigabit Ethernet

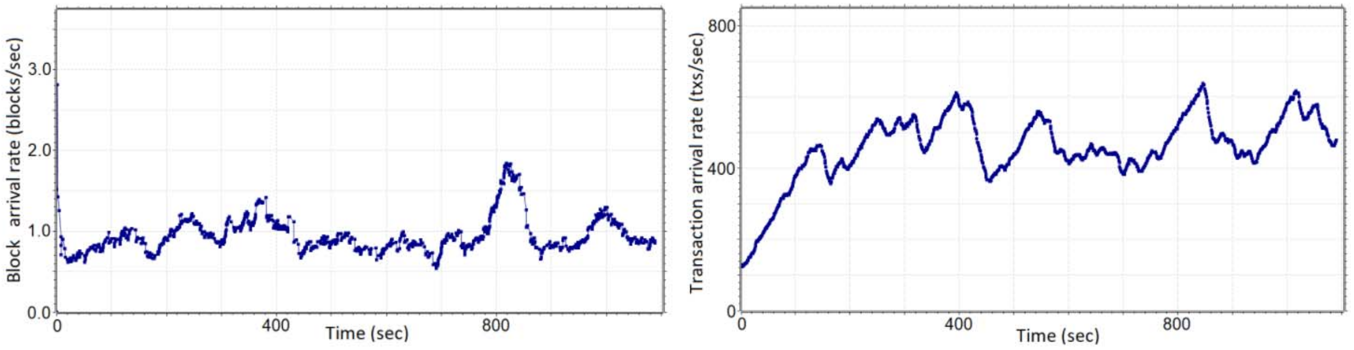

Fig. 6. Block arrival rate and transaction arrival rate v/s time for simulation of a 13 node network over WiFi