PLEDGE: An IoT-oriented Proof-of-Honesty based Blockchain Consensus Protocol

Imran Makhdoom
Food Agility CRC Ltd
University of Technology Sydney
Sydney, Australia
imran.makhdoom@uts.edu.au

Farzad Tofigh
University of Technology Sydney
Sydney, Australia
Farzad.Tofigh@uts.edu.au

Ian Zhou
University of Technology Sydney
Sydney, Australia
ian.zhou@student.uts.edu.au

Mehran Abolhasan
University of Technology Sydney
Sydney, Australia
mehran.abolhasan@uts.edu.au

Justin Lipman
Food Agility CRC Ltd
University of Technology Sydney
Sydney, Australia
Justin.Lipman@uts.edu.au

***Abstract*—The existing lottery-based consensus algorithms, such as Proof-of-Work, and Proof-of-Stake, are mostly used for blockchain-based financial technology applications. Similarly, the Byzantine Fault Tolerance algorithms do provide consensus finality, yet they are either communications intensive, vulnerable to Denial-of-Service attacks, poorly scalable, or have a low faulty node tolerance level. Moreover, these algorithms are not designed for the Internet of Things systems that require near-real-time transaction confirmation, maximum fault tolerance, and appropriate transaction validation rules. Hence, we propose "Pledge," a unique Proof-of-Honesty based consensus protocol to reduce the possibility of malicious behavior during blockchain consensus. Pledge also introduces the Internet of Things centric transaction validation rules. Initial experimentation shows that Pledge is economical and secure with low communications complexity and low latency in transaction confirmation.**

***Index Terms*—Blockchain consensus, Byzantine Fault Tolerance, distributed consensus, proof of honesty, miner selection, block proposer's integrity, transaction validation rules, Internet of Things.**

I. Introduction

There are two main approaches to network consensus in blockchain-based applications: Nakamoto consensus and Byzantine Fault Tolerance (BFT). Nakamoto consensus [1] is a Proof-of-Work (PoW) based protocol that is proved to be computationally intensive. Moreover, due to the probabilistic nature of the Nakamoto consensus, temporary forks occur, which are likely to cause latency in TX confirmation, thus resulting in low TX throughput [2]. This delay in TX confirmation is not suitable for most of the real/near-real-time IoT systems requiring instant TX finality. In addition, [3] also highlighted numerous security risks in PoW-based blockchains. On the other hand, traditional BFT algorithms such as Practical Byzantine Fault Tolerance (PBFT) [4]–[6], and Delegated Byzantine Fault Tolerance (DBFT) [7], select the next block proposer in a round-robin fashion and use multiple rounds of explicit voting by a limited number of chosen validators to achieve consensus. Moreover, BFT-based protocols are also susceptible to DoS attacks due to their dependence on weak timing assumptions for liveliness [8], [9]. Consequently, weak synchrony also adversely affects the throughput of such systems [8].

Although voting-based BFT consensus protocols provide consensus finality yet a significant problem is scalability concerning the number of validator nodes [10]. The scalability problem can be attributed to multiple rounds of communications, which often involve as many as $O(n^2)$ messages per round [4]. BFT protocols also fail to operate correctly in the presence of more than $1/3$ faulty/malicious nodes. Hence, there is a strong fault-threshold assumption in BFT protocols that at least two-thirds of nodes are honest [5].

Correspondingly, to resolve the issue of nothing at stake, Clique, a Proof-of-Authority based consensus protocol [11] was developed. It seems that Clique was inspired from a statement of Warren Buffet [12], where he said that "It takes twenty years to build a reputation and five minutes to ruin it. If you think about that, you will do things differently." Hence, Clique puts the users' real-world identities (IDs)/reputation at stake. To establish the authenticity of user IDs, the public notaries being the trusted parties perform the on-chain ID verification [11]. However, it is believed that the integration of a public notary or such a government entity to a private/consortium blockchain will take some time to realize due to a lack of legislation/rules and policies on the subject. Moreover, the validation of identities by a trusted third party is against the decentralization spirit of the blockchain.

Concerning TX validation rules, currently, Bitcoin blockchain validates a TX based on its format, valid signatures, and the fact that it has not been previously spent [9], [13]. On the other hand, Ethereum validates the format, signatures, nonce, gas, and account balance of the sender's account [14]. Whereas, in Hyperledger Fabric, TXs are executed, and signed by the endorser nodes, followed by the ordering of TXs in blocks by the Ordering Service. Lastly, all the committing

peers, check TX read-write set, and endorsement policies before appending blocks to their copy of the blockchain [15]. However, there is a question about the applicability of the existing rules to the TXs in blockchain-based IoT systems that are vulnerable to cyber-attacks [9], [16]. Hence, a targeted or even a generic malware attack can infect a lot of IoT devices [17]. Therefore, TX validation rules of cryptocurrency may not be adequate for TXs initiated by IoT devices. Consequently, there is a requirement of an IoT-centric consensus protocol that must: conform to IoT-oriented TX validation rules, prevent DoS attacks (exploiting timing assumptions), provide increased fault tolerance ($> 1/3$ faulty nodes), and near instantaneous TX confirmation with low communications complexity.

Correspondingly, a lot of work has been done on blockchain-based IoT applications, such as [9], [18]–[20]. Nonetheless, these studies focus on the methodology of employing blockchain technology in eHealth, smart homes, and other areas. Also, there has been some work done on trust management or reputation systems in the blockchain environment [21]–[25]. However, these schemes have either weak assumption that users are honest in their ratings of other peers or they do not cater for the presence of malicious actors in the network. Also, few of these either rely on a third party such as an attribute provider (AP) to provide authentic credentials/attributes for the users, or they are bandwidth-intensive. Moreover, these schemes are vulnerable to various attacks against reputation systems including discrimination [26], traitors [27], and slandering attacks [25].

Hence, we present "Pledge," a Proof-of-Honesty (PoH) based consensus protocol with IoT-oriented TX validation scheme. This is an extended version of our previous work [28]. Pledge aims to reduce the participation of faulty, corrupt/malicious, and non-performing nodes in the consensus process, thereby increasing the fault tolerance to the maximum. Pledge is a PoH-based consensus protocol in which the block proposers are selected based upon the cumulative score of their honesty attributes. Whereas, the attributes are collected internally from the blockchain. Hence, we take advantage of the inherent benefits of blockchain, i.e., data immutability, ability to operate in a trustless environment, and protection against data forgery. Consequently, no trusted IDP (Identity Provider), AP, Notary Public, or a third party is required to validate the attributes. Therefore, it is nearly impossible to forge or emulate fake honesty attributes.

The rest of the paper is organized as follows: Section-II unfolds the properties of an ideal consensus protocol, Pledge methodology, and context-aware TX validation rules. In Section-III, comprehensive security and performance analysis of Pledge is presented. Finally, conclusions and future work are highlighted in Section-IV.

II. The Pledge Protocol

Before getting into the details of the Pledge protocol, it is important to first sift through the properties of an ideal consensus protocol for blockchain-based IoT systems.

A. Properties of an Ideal IoT-centric Consensus Protocol

- **Fairness.** All nodes should have an equal chance of being selected as the block proposer.
- **Investment.** The cost of the block proposer selection process should be proportional to the value gained from it.
- **Verification.** It should be relatively simple to verify that the block proposer was legitimately selected [29].
- **Honesty.** Nodes participating in the consensus process should have a high probability of being honest.
- **Termination.** All honest nodes finally decide on a block.
- **Agreement.** All honest nodes agree on the same block.
- **Validity.** The block that is being agreed upon should be from a legitimate node [30].
- **Consensus Finality.** A block once agreed upon and appended to the digital ledger, is not removed any time later [10].
- **BFT.** The consensus protocol should be able to propose a valid block even in the presence of a greater number of faulty, corrupt, or malicious nodes.
- **Unforgeability.** The block proposer selection process should be unforgeable, and no node should be able to emulate fake attributes.
- **Security.** The system should be resilient against common attacks on reputation systems, and also does not subvert the fundamental security guarantees of the blockchain.
- **Decentralization.** The consensus protocol should not be quasi-centralized by abandoning the decentralization property of the blockchain.
- **Scalability.** The consensus algorithm should scale well with the increase in the number of network nodes without increasing the communications complexity.

B. Pledge Methodology

The design of the Pledge is based on certain assumptions. **Assumptions.** There is a likelihood of Byzantine failures in the blockchain network. Correspondingly, the Byzantine nodes are not expected to follow the protocol. Moreover, an adversary may control and manipulate the behavior of the nodes resulting in deteriorated performance. The adversary may also disrupt the communications and split the network. Nonetheless, being a consortium blockchain with identity management (IDM), it is

TABLE I
ATTRIBUTES' WEIGHT CRITERIA AND RANGE OF SCORE

Attributes	Weight Criteria	Attribute Score	
		Min	Max
Blockchain protocol ver (P_V)	Ver up-to-date or not	-1 for old ver	1 for latest ver
Client ver C_V	Ver up-to-date or not	-1 for old ver	1 for latest ver
Network ID (N_{ID})	Correct or not	-10 for incorrect ID	1 for correct ID
Number of valid blocks proposed (B_F)	1 mk for each block	0	f
TX count in the previous valid blocks (B_{TXC})	1 mk for every TX	0	c
Number of TXs sent (TX_S)	1 mk for every TX	0	s
Number of TXs received (TX_R)	1 mk for every TX	0	r
TX errors (TX_E)	-10 mk for every error	-e	0
Number of pending TXs (TX_P)	-1 mk for every pending TX	-p	0
Is the node listening for peers? (L_G)	-10 mks for not listening	-10	1
Number of connected peers (P_N)	2 mks for each connection	0	n

very difficult for the malicious nodes to impersonate other honest nodes. Lastly, it is assumed that a typical private/consortium blockchain-based IoT system comprises a large number of resource constrained end-nodes (IoT devices). Besides, it has a limited number of full-nodes (potential block proposers) that can generate new blocks and maintain a copy of the blockchain. Hence, the term "node" in this paper refers to a full-node.

Pledge Protocol. When a new block is published, or the blocks proposed by the proposers of the last round are rejected, the consensus process to select the next pair of block proposers starts. As shown in Fig. 1, an honesty metric (H_{Mat}), is computed and maintained for all the registered full-nodes (potential block proposers) on the blockchain. Hence, whenever a block is successfully appended to the blockchain, an event [31] is triggered that starts the process of updating the H_{Mat} for every node based on the predefined attributes extracted/computed through the blockchain. The value of each attribute is obtained and weighted to compute the H_{Mat} for every node. Subsequently, a cumulative H_{Mat} score is calculated for each node, i.e., $H_{MAT1}CumScore$, $H_{MAT2}CumScore$, and $H_{MAT3}CumScore$ respectively for Node 1, Node 2, Node 3, and so on. Next, a priority list of K honest nodes is formed based on $Honesty_{MAT}$, which comprises the individual $H_{Mat}CumScore$ of all the nodes.

The nodes with $H_{MAT}CumScore >= H_{MAT}Threshold$ form part of the K honest nodes list. It is followed by a random selection of "Primary" and "Secondary" block proposers for the next block, from the K honest nodes. Finally, the primary proposes a new block followed by the validation of its $H_{MAT}CumScore$ and TXs in the proposed block by the rest of the K honest nodes before that block is committed. If there is any violation of the TX validation rules or the $H_{MAT}CumScore$ of the primary was not computed correctly, the proposed block is rejected, the primary is blacklisted, its owner organization is reprimanded, and a new block is introduced by the secondary proposer. The same checks are performed on the blocks proposed by the secondary proposer as well, and if a block is valid, then it is accepted and appended to the chain by all the nodes. Otherwise, secondary is also blacklisted, and new primary and secondary block proposers are selected for the current round. Finally, when the block is accepted and appended to the blockchain, the next round of $Honesty_{MAT}$ computation, and selection of a new primary and a secondary block proposer starts.

The biggest challenge in this process is selecting the attributes that optimally describe the honest behavior of the nodes and further help identify faulty, malicious, and Byzantine nodes. These attributes may differ for every blockchain technology such as Bitcoin, Ethereum, Hyperledger Fabric, etc. However, we have determined some traits common to every blockchain platform. As shown in Fig. 1, the first three attributes, including the blockchain protocol version running on the node, the client application version, and the network ID, may contribute to the faulty or impaired behavior by a node. Whereas the rest of the attributes such as the number of valid blocks proposed, the total number of TXs included in the valid blocks, TX errors, number of TXs sent and received, number of pending TXs at a particular moment, number of connected peers, and whether the node is listening for peers or not, reflect the conscientious performance of the node. If a node is honest, its performance would be exceptional as it will mine more blocks with the maximum possible number of TXs in a block. An honest node is also expected to be connected to most of its neighboring nodes and process a high number of TXs. Depending upon the type of blockchain platform, some other attributes can also be included, such as for Bitcoin or any other fintech blockchain, the total reward earned by a node, number of confirmations for the TXs, and number of blocks relayed can be considered.

Additionally, inactivity can also be an attribute such that any period of inactivity higher than time Δt will earn a negative score for each period of non-activity exceeding Δt. Similarly, for Proof-of-Stake (PoS) based blockchains, current balance can be one of the attributes. In addition to the specific aspects, certain facets resonating misbehavior of the nodes, and anomalies in their performance, can be detected by employing a layer of deep learning over the blockchain network.

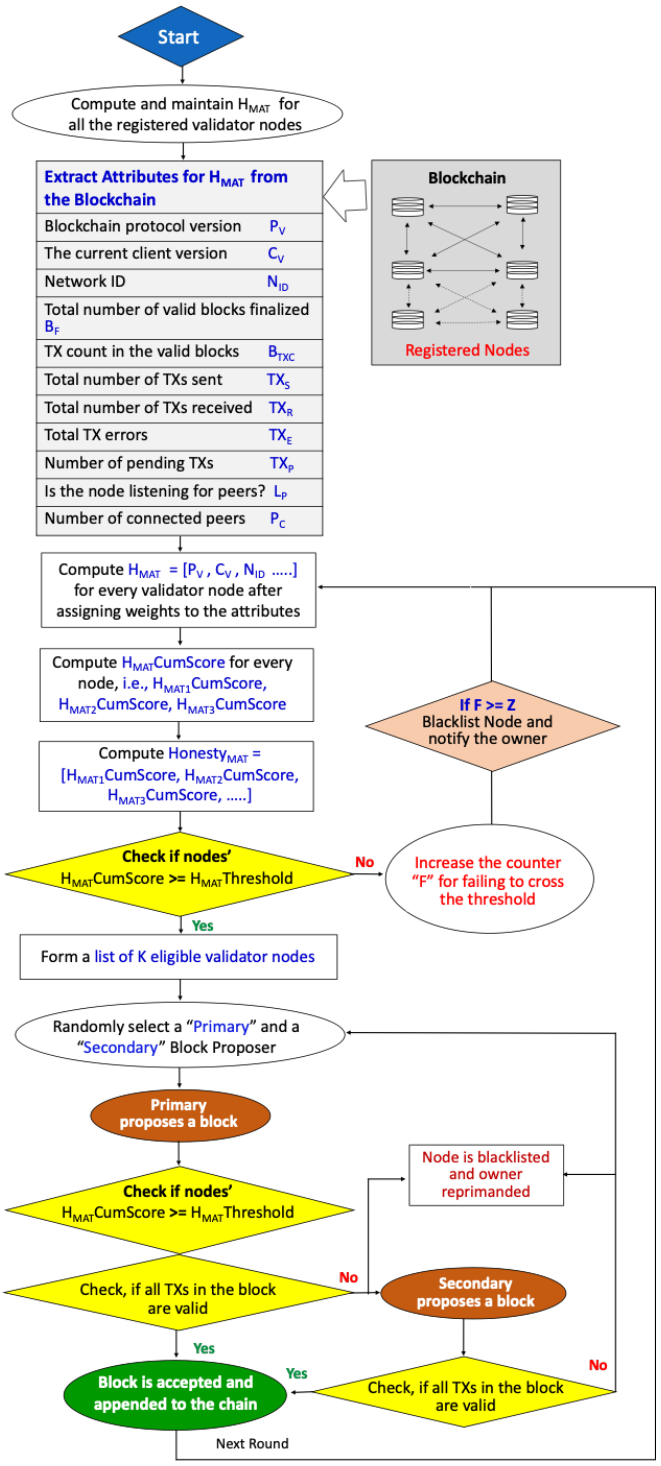

Fig. 1. Pledge methodology.

C. Computing H_{MAT}CumScore

In this section, we will illustrate the approach adopted to model $H_{MAT}CumScore$ in respect of a node based on the weighted sum of its character traits/attributes. This model is not a hard and fast rule; instead, it may vary from system to system

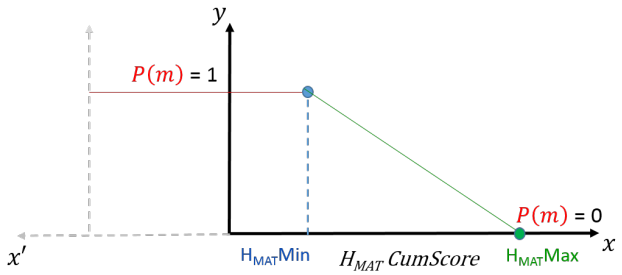

Fig. 2. Probability of being malicious.

based upon the sensitivity/criticality of the application. Table-I shows the attributes that are evaluated, the weight/scoring criteria, and the range of minimum (min) and maximum (max) values for each attribute. The min $H_{MAT}CumScore$ a node can secure is defined by (1), and the max $H_{MAT}CumScore$ score that can be achieved by a node is represented by (2).

$$H_{MAT}MinCumScore = -22 - eTX_E - pTX_P \tag{1}$$

$$H_{MAT}MaxCumScore = 4 + fB_F + cB_{TXC} + sTX_S + rTX_R + nP_N \tag{2}$$

Taking into account the $H_{MAT}MinCumScore$, and $H_{MAT}MaxCumScore$, ideally the probability of a node being malicious $P(m)$ (as shown in Fig. 2) is close to one, if the node's $H_{MAT}CumScore$ is equal to $H_{MAT}MinCumScore$. However, practically the probability of having a malicious node can be calculated as

$$P(m) = \begin{cases} 1, & \text{Malicious node} \\ ax + b & \\ 0, & \text{Honest node} \end{cases}$$

where x is a random variable that represents $H_{MAT}CumScore$. Accordingly, the slope of the line $(ax + b)$ can be defined as:

$$\frac{x2 - x1}{y2 - y1} = \frac{H_{MAT}Max - H_{MAT}Min}{0 - 1} \tag{3}$$

Correspondingly the point-slope form can be represented as (4):

$$P(m) = \frac{-1}{H_{MAT}Max - H_{MAT}Min} \times x + \frac{H_{MAT}Max}{H_{MAT}Max - H_{MAT}Min} + 1 \tag{4}$$

which can be further simplified to:

$$P(m) = \frac{-x + H_{MAT}Max}{H_{MAT}Max - H_{MAT}Min} \tag{5}$$

Now, by substituting (1) and (2) into (5), we can calculate the probability of a node being malicious, i.e,. $P(m)$, while $H_{Min} \leq x \leq H_{Max}$:

$$P(m) = \frac{-x + 4 + fB_F + cB_{TXC} + sTX_S + rTX_R + nP_N}{26 + fB_F + cB_{TXC} + sTX_S + rTX_R + nP_N + eTX_E + pTX_P} \tag{6}$$

Another important aspect is modelling the $H_{MAT}Threshold$, such that at a particular moment all the nodes that have $H_{MAT}CumScore >= H_{MAT}Threshold$, will be included in the list of K eligible block proposers. Consequently, the probability of a node being malicious will be less than 0.5, if the node's $H_{MAT}CumScore > H_{MAT}Threshold$. Hence, we define $H_{MAT}Threshold$ to be the average (avg) value of $H_{MAT}Max$ and $H_{MAT}Min$, which can be represented by (7):

$$x = \frac{-18 + fB_F + cB_{TXC} + sTX_S + rTX_R + nP_N - eTX_E - pTX_P}{2} \quad (7)$$

$H_{MAT}Threshold$ being the avg of the $H_{MAT}Max$, and $H_{MAT}Min$, is dynamic and will rise with the increase in $H_{MAT}Max$, as the honest nodes continue to perform better with the passage of time.

D. IoT-Oriented TX Validation

In our previous work [9], we identified the need for IoT-oriented TX validation rules, and also proposed a way forward. The foremost requirement for IoT systems is that the TXs should be validated based on context-aware TX validation rules. It is essential since every new TX in IoT is mostly independent of the previous TX, and a hardware malfunction, software bug, or a change in environmental conditions can induce variations in the sensor readings. The context-aware TX validation rules not only protect against malfunctioned sensors but also against malicious block proposers. Therefore, IoT TX validation rules should be carefully drafted, and they must incorporate environmental context based on the deployment scenario. This methodology can be described clearly with the help of a smart home and supply chain management system case study shown in Figure-3.

1) Smart Home: In a smart home scenario, during winters, if the temperature sensor installed in a room initiates a TX showing the temperature below threshold, e.g., 2^oC, to ignite the fireplace. This TX will only be considered valid if, during time Δt (Δt can be any value depending upon IoT application, in which co-located sensors can observe the same event and report upon it), another sensor installed in the same room also initiates a TX indicating the occurrence of the same event, i.e., falling of temperature below the defined threshold. Such confirmation will not only protect against random faults in the sensors but also ensure validation of the TXs based on multiple sensors readings. Depending on the sensitivity of the location/application, multiple cross-checks can be included as rules to verify different types of TXs initiated by IoT sensors.

2) Supply Chain Management (SCM): Let us suppose that a shipment of frozen food is being monitored for swift movement on a pre-defined route from point A to point X (as shown in Figure-3). Therefore, when the shipper initiates a TX confirming that the shipment has reached the desired customer at location X, this TX will only be considered valid if, during time Δt (Δt can be any value depending upon IoT application, in which co-located sensors can observe the same event and report upon it), some of the GPS sensors attached to the frozen food package also initiate TXs indicating the exact location of the package. The package's GPS sensors can be easily programmed to initiate a TX, once the consignment reaches location X. These cross-checks will not only protect against any TX initiated with malicious intent by the shipper but also detect a malfunctioned IoT sensor.

TABLE II
STORAGE REQUIREMENT FOR THE ATTRIBUTES

Attribute	**Storage Requirement (Bytes)**
Blockchain Protocol Ver P_V	One
Client Ver C_V	One
Network ID N_{ID}	One
Valid Blocks Finalized B_F	Four
Valid TX Count in the Valid Blocks B_{TXC}	Four
TXs Sent TX_S	Four
TXs Received TX_R	Four
TXs Errors TX_E	Four
TXs Pending TX_P	Four
Is the Node Listening L_P	One
Number of Connected Peers P_N	Two

III. SECURITY GUARANTEES AND PERFORMANCE ANALYSIS

Pledge offers numerous security guarantees with a scalable performance by satisfying most of the requirements of an optimal consensus protocol discussed in Section II-A.

A. Fairness

Every node has an equal chance of being elected as a primary or a secondary block proposer if it satisfies the $H_{MAT}Threshold$ requirement.

B. Investment

The leader selection process in Pledge is neither computationally expensive like PoW nor does it require specialized hardware, as in the case of PoET [32]. Hence, the computation, energy, and storage costs of selecting a block proposer are very economical. E.g., Depending upon the number of attributes to be evaluated for the computation of $H_{MAT}CumScore$ (eleven attributes in our case), there are eleven get operations to read the state of desired attributes from the blockchain. The storage requirement for these attributes as shown in Table-II, sums up to be at the most thirty bytes. Moreover, the computation of $H_{MAT}CumScore$ for a particular node requires one add operation.

Correspondingly, to measure the cost of computing the $Honesty_{Mat}$, and selection of the two block proposers, a simulation of Pledge protocol was run on Ethereum blockchain. The experimentation was performed using the Remix-Ethereum

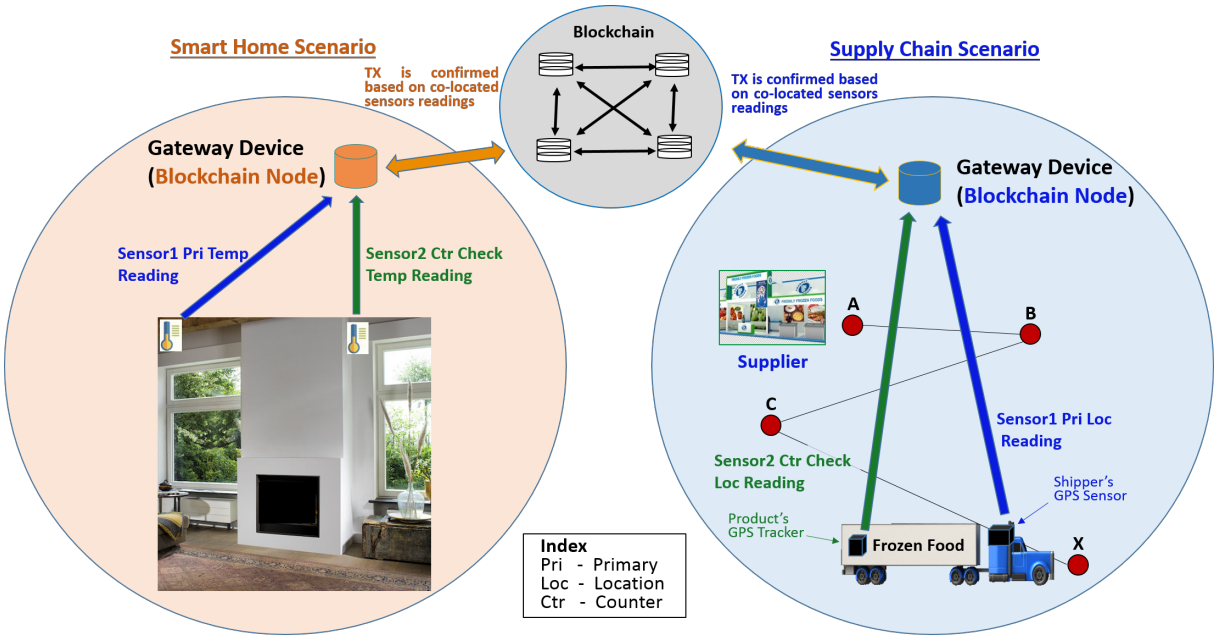

Fig. 3. IoT TX validation rules.

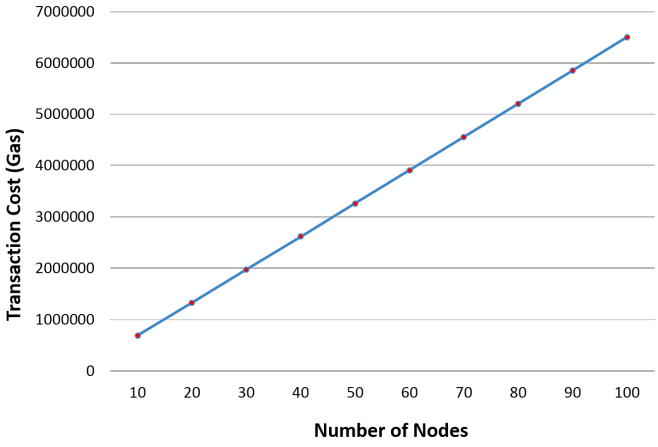

Fig. 4. Transaction cost vs Number of nodes

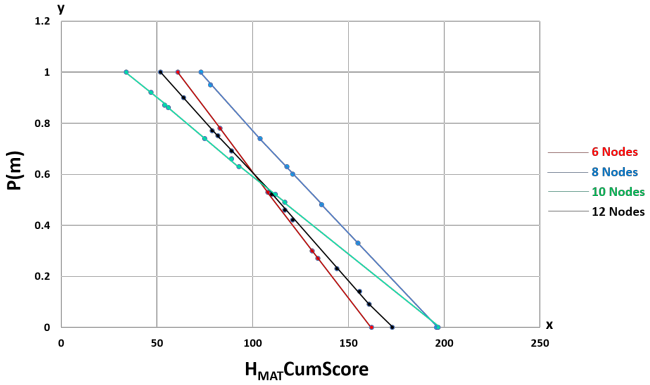

Fig. 5. Probability of a node being malicious.

IDE compiler version 0.5.19 [33], and Geth (Go Ethereum) version 1.8.27 deployed on a machine configured with an Intel Core $i5$, 6^{th} generation processor, and $8GB$ RAM. As shown in Fig. 4, avg TX cost (in terms of gas) was computed by running thirty iterations of the Pledge protocol for each set of nodes varying from ten to hundred (total 300 iterations). It can be seen that the TX cost increases linearly with the number of network nodes. The avg increase in the TX cost is $6,46,071$, with the addition of every set of ten new nodes. Nonetheless, considering the ethereum block gas limit of $8,000,000$ (8 million) gas for a block [34], and gas consumption of under 6500000 (6.5 million) for hundred nodes, it can be concluded that the proposed PoH-based block proposer selection process is relatively economical in terms of computational costs.

C. Verification

All the nodes in the network continuously try to improve their performance so that they increase their probability of being selected into the list of K honest nodes and finally get elected as block proposers. However, the verification of such a selection is straightforward. When a primary and the secondary block proposers generate a new block, the rest of the nodes in the list of K nodes run the get operation to retrieve the latest state of the attributes in respect of the block proposer, and compute the $H_{MAT}CumScore$ for verification by running just one addition operation and a logical match operation.

D. Honesty

As per (7), the probability of a node being malicious is less than 0.5 if its $H_{MAT}CumScore > H_{MAT}Threshold$.

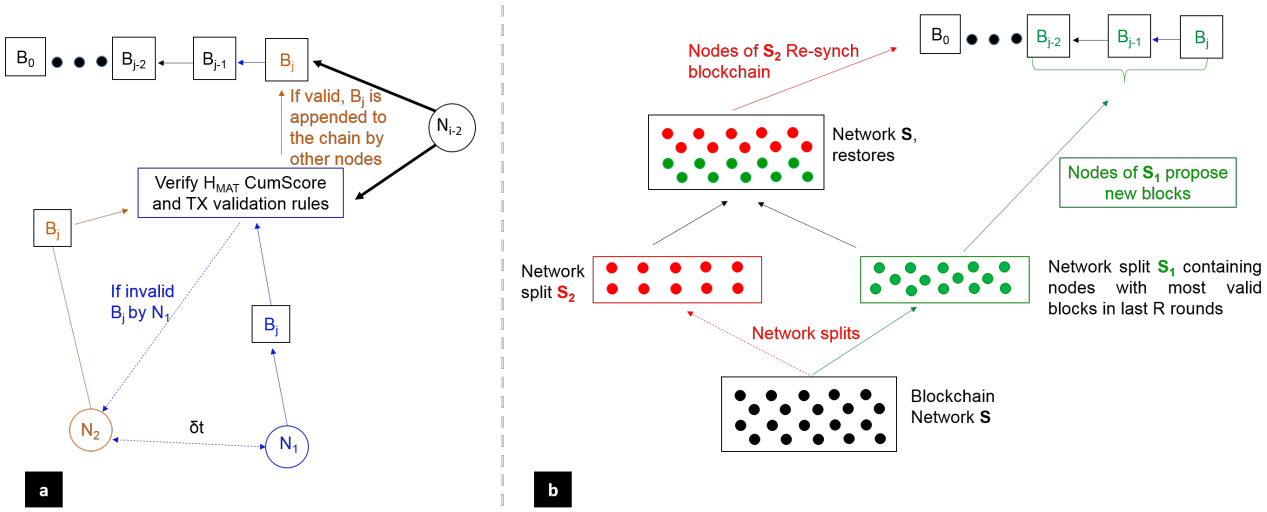

Fig. 6. Consensus termination and block agreement, a) Normal scenario. b) Split network.

Moreover, as shown in Fig. 5, the experimental results show that the probability of a node being malicious (as discussed in Section II-C) is linear with respect to its $H_{MAT}CumScore$. Also, this linearity is independent of the number of nodes in the network. Correspondingly, the lower the $H_{MAT}CumScore$ of a node is, the higher is the probability of the node being malicious. Similarly, to raise the criteria of a node being honest and to be included in the list of the K honest nodes, the threshold can be raised so that only the nodes with low probability of being malicious, e.g., 0.4, or 0.3, are eligible to be selected as the primary or secondary block proposers.

Moreover, to motivate the nodes to continuously perform honestly and achieve maximum $H_{MAT}CumScore$, the block reward/TX fee is distributed proportionally among all the K honest nodes as per their ranking in the list, i.e., the node with highest $H_{MAT}CumScore$ gets the maximum share, and the node with the lowest $H_{MAT}CumScore$ gets the smallest share. Hence, the nodes strive to achieve maximum $H_{MAT}CumScore$ to get the maximum share of the block reward.

E. Termination and Agreement

As shown in Fig. 6a, Pledge protocol assures that under normal circumstances, when the primary (N_1), and secondary (N_2) block proposers propose the block following the protocol/TX validation rules and the block is agreed upon by all the other nodes, the consensus process terminates. This property holds even if N_1 fails to propose a valid block at the first instance. Besides, considering the adversary's power to disrupt the communications and split the network, Pledge can still perform with consistency. In this context (as shown in Fig. 6b), to continue the consensus process, the network half comprising those honest nodes that have collectively proposed more blocks than the other half in the last R (eleven in this case) consensus rounds, continues to submit new blocks. Whereas, the other network half waits and synchronize its chain once the network topology is restored. It is imperative to mention that even if the network splits, still both the network halves can get the information about block proposers of the last eleven rounds (before split) from their copy of the blockchain. Moreover, in another scenario, a node may get delayed blocks due to network latency. In such a case, to avoid forks and to protect against the false invalidation of legitimate blocks, the network nodes always wait for the block that points to the block with the highest index in their local chain. Therefore, even if a node receives some blocks in random order, it will append the blocks to its local copy of the chain based on their index number in ascending order.

F. BFT

It is expected that the faulty/malicious nodes may not follow the protocol specifications and behave erratically. Pledge reduces the possibility of Byzantine behavior by a node during the consensus process by putting the node's integrity at stake. Hence, if a node proposes a block with invalid TXs, it is banished, and removed from the list of K honest nodes. Moreover, requisite clarification and corrective action is sought from the owner organization. Also, as a deterrence to others, the responsible organization is banned from participating in the consensus process for seventy-two hours, thus losing valued share of the TX fees. Depending upon the nature of the blockchain network, the organization may also be issued with a financial penalty (mechanism of issuing a financial penalty is not covered in this work). Moreover, the non-performing node's software is re-installed and it is also reconfigured to get rid of any software bug or malicious payload. Besides, the context-aware TX validation rules introduced in this paper let the nodes easily detect any malicious change in the value of a particular sensor during TX validation before committing a new block.

Moreover, as the block proposer is selected from a list of K

honest nodes, the likelihood of participation of malicious/non-performing nodes in the block proposal and consensus process, i.e., validation of proposed blocks, is reduced to a great extent. Hence, it is presumed that till the time there are at least two honest nodes in the list of K honest nodes, the consensus process is safe from most of the faults.

G. Unforgeability

The $Honesty_{Mat}$ is computed based on attributes obtained from the blockchain. Hence, due to the distributed and immutable nature of blockchain, peers/nodes cannot emulate attributes or forge any change in respective $H_{Mat}CumScore$. Similarly, Pledge is resilient to forged trust where malicious users may create fake IDs to create a spam farm to boost their trust ratings.

H. Sybil Attack

The participation of only registered nodes in the consensus process based on $H_{MAT}CumScore$ reduces the risk of a Sybil attack.

I. Targeted Attacks

The selection of K honest nodes based on their bona fide performance and further randomization to select the block proposers avoids targeted attacks by the adversaries against the next deterministic block proposer.

J. Trustless Operation

Pledge does not use any Peer-to-peer (P2P) reputation or trust model to generate the Honesty Metrics to avoid unfair rating and collusion attacks. Also, Pledge does not rely on a third party, such as an AP or a trusted IDP, to provide node attributes. Instead, the attributes for the computation of $Honesty_{Mat}$ are directly obtained from the blockchain. The idea of generating, storing, and extracting attributes from the blockchain can avoid most of the trust issues concerning the acquisition of attributes [25], [35]–[38]. Similarly, Pledge also avoids some of the significant attacks against reputation systems, including discrimination [26], traitors [27], and slandering attacks [25].

K. Whitewashing Attack

It is very likely that Pledge contains the effects of a Whitewashing attack [25], i.e., when the honesty score of a node becomes very low, he leaves the network and then joins later with a new pseudonym. Although acquiring a new pseudonym requires approval in a consortium blockchain, however, still to prevent an insider attack, Pledge provides a disincentive to the nodes for rejoining the network with a new ID. Therefore, when a new node joins the network, his honesty score is below $H_{MAT}Threshold$, due to lack of performance in the system. Thus, a malicious node stands no chance of being included in the list of K eligible nodes. Hence, the nodes with low honesty scores have no option other than to improve their performance and keep their attributes as per the required standard/threshold. However, there is a possibility that with the help of an inside attacker, a malicious node is successful in getting into the list of the K eligible nodes, and randomly gets elected for the block proposal. To counter such eventualities, whenever a primary block proposer broadcasts a new block, all the other honest nodes (in the list of K nodes) verify that whether the block proposer's $H_{MAT}CumScore$ was computed legitimately or not. In case the primary proposer is found to be malicious, the block proposed by the primary is rejected, and the secondary proposer's block is validated and accepted in the same way.

L. Protection against Non-Performing Nodes

Another vital aspect is the accountability of the nodes that violate the consensus rules or fail to surpass $H_{MAT}Threshold$. It is envisaged that the nodes that fail to get into the list of K eligible nodes for time δt equivalent to the duration of a number Z of consecutive published blocks, they are blacklisted. Where Z depends upon the the sensitivity/criticality of the system. Hence, if the system failure has serious security or safety implications, then Z can be set as the lowest as possible. E.g., if a node fails to get into the list of K eligible block proposers for eleven consecutive blocks, it will be blacklisted. Similarly, if a node's conduct is erratic and it performs below the threshold in between the episodes of making into the group of eligible block proposers, such a node's behavior is measured by analyzing node's last eleven $H_{MAT}CumScores$. If it has secured below threshold score for six or more times (this can change depending upon the criticality/sensitivity of the IoT system), it is blacklisted.

M. Replay Attacks

To protect against replay/double-spending attacks, every TX initiated by a particular node/client application has a sequence number in addition to the timestamp. Hence, a particular node cannot generate another TX with a higher timestamp but a lower or same sequence number as the previous one.

N. Decentralization

The random selection of a primary and a secondary block proposer protects the system from quasi-centralization. Otherwise, few most honest nodes may have the monopoly to mine every new block, and they may try to play foul with the system. Instead, the system gets more decentralized as the network expands. It is because the list of K eligible nodes is likely to extend with more number of nodes satisfying the threshold $H_{MAT}CumScore$. Hence, the probability of a node to be selected as a primary or a secondary block proposer decreases with the increase in the number of nodes in the list of K nodes. However, preventing quasi-centralization has an associated risk,

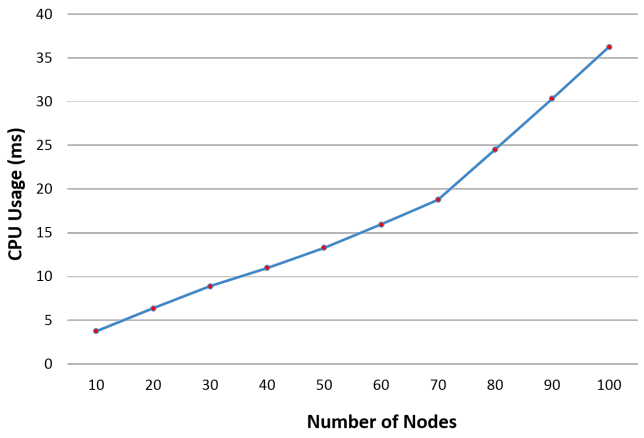

Fig. 7. CPU time to execute Pledge protocol vs Number of nodes.

i.e., the random selection of two block proposers from a list of K eligible nodes based on the threshold score entails selection of those nodes that have the probability of being malicious equal to 0.5. However, depending upon the sensitivity of the IoT application, the threshold can be raised to decrease the probability of selecting a possibly malicious node. Similarly, for systems that are not concerned about quasi-centralization, the primary and the secondary block proposers can always be selected from the top x% of the nodes with the highest $H_{MAT}CumScore$.

O. Scalability

The proposed scheme does not require energy and computationally intensive PoW for the selection of a block proposer. The computation of the $Honesty_{Mat}$ requires meagre resources. Moreover, the Pledge does not require excessive communication rounds to vote on the eligibility of the blocks or to propagate reputation scores between peers. Hence, there are no communication overheads other than routine TX and block propagation messages. Moreover, Pledge is scalable with an increase in the number of nodes as the list of K eligible nodes is dynamic. It includes all the nodes that have $H_{MAT}CumScore$ greater than or equal to the threshold.

Hence, the increase in the number of nodes does not affect the performance of the consensus process. Moreover, the logic for selecting a primary and a secondary block proposer is that in case the primary block proposer fails to propose a block in time ζt, then to avoid latency in TX confirmation by starting the process all over again, the secondary node proposes the block.

Additionally, to measure the latency in TX confirmation, we measured the avg CPU time for the execution of Pledge protocol (including computation of $Honesty_{Mat}$, and selection of Primary and Secondary block proposers) for a range of nodes varying from ten to hundred. We had total hundred iterations of the experiment for ten different sets of nodes. As shown in Fig. 7, although the avg CPU time rises with the increase in the number of nodes, yet for a hundred nodes the computation time is merely 35.47 ms. Moreover, even if the number of potential block proposers/full nodes increases to two thousand (which is very unlikely in a consortium/private blockchain) the latency in TX confirmation is expected to be 720 ms, which is still under one sec. Hence, it can be concluded that for a private/consortium blockchain settings, the latency in TX confirmation is very nominal and Pledge performs better than Bitcoin (TX confirmation is after 2-6 blocks, i.e., 10-60 mins) [40], IoTA (No specific time as it varies from 2-3 mins to even 30 mins depending upon the rate of input of new TXs in the network) [41], and Ethereum (15 sec) [40]. It is also inferred that a block proposed by an honest node, once accepted by the other honest nodes, would not be later purged from the chain. Correspondingly, a detailed comparison of the security and performance efficiency of PoH versus some renowned consensus protocols is shown in Table-III.

P. Limitations and A Way Forward

In addition to the security guarantees, we have also perceived certain limitations of the Pledge protocol, that require further research.

- $H_{MAT}Threshold$ vs. Network Bootstrapping: The current selection of $H_{MAT}Threshold$ as the avg value of $H_{MAT}Max$ and $H_{MAT}Min$ scores seems workable once the blockchain network is running for some time. However, it is observed that in the current form, the avg value may not provide the desired security once the network is being bootstrapped. Because at the start of the blockchain network, all the block proposers will have almost the same $H_{MAT}Min$ score. Hence, the block proposer's selection will rely upon random selection from the list of K eligible nodes for quite some time. Therefore, there is a requirement of working out an appropriate value of $H_{MAT}CumScore$ for validator nodes as a starting point. One option in this regard may be a random allocation of $H_{MAT}CumScore$ at the start to bootstrap the network.
- Adding a New Node to the Pledge Consensus: Currently, there is a question mark on how to onboard a new node into the honesty-based consensus protocol. It is perceived that it would take a long time for a new node to catch up with other nodes that already have a high honesty score.
- Attributes from Blockchain: At the moment, only eleven attributes (listed in Table-I) have been identified, which seems common to most of the blockchain protocols. However, in practice, there would be a requirement of extracting those attributes from the blockchain protocol that best describes the integrity and the performance of the block proposer nodes in a specific blockchain network. Therefore, it is envisaged that this aspect has to be catered for while developing a blockchain platform. The desired attributes can be directly measured from the blockchain using inbuilt functions, such as methods/functions available in web3.js library to interact with the Ethereum blockchain.

TABLE III
SECURITY AND PERFORMANCE COMPARISON OF CONSENSUS PROTOCOLS

Features	**PoW**	**PoS**	**PoET**	**PBFT**	**IoTA**	**PoH**
Area of application	Fintech	Fintech	Multiple	Multiple	Multiple	Multiple
Type of Blockchain	Permissionless	Permissionless and Permissioned	Permissionless and Permissioned	Permissioned	Permissionless	Currently Permissioned
Vulnerabilities	51% attack	51% attack, and malicious collusion of rich stakeholders	Node compromise	Fault tolerance of 1/3 faulty nodes, and DoS attack	vulnerability of Curl-P-27 hash function, and signature forging attacks [39]	Low $H_{MAT}Threshold$ at the start of the network, hence, probability of a node being selected as a primary or secondary block proposer is high with less number of nodes in the network
Address nothing at stake problem	No	No	No	No	No	Yes
Energy costs	High	Low	Low	Low	Low	Low
Computation costs	High	Low	Low	Low	Low	Low
Communication complexity	Low	Low	Low	High	Low	Low
Consensus Finality	Probabilistic	Probabilistic	Probabilistic	Instant	Probabilistic	Instant
Blockchain Forks	Yes	Yes	Yes	No	Yes	No
TX latency (Based on consensus finality)	High	Low	Low	Low	Moderate	Low
Scalable	Yes	Yes	Yes	Poor scalability concerning the number of validating nodes	Yes	Yes
The requirement of special hardware	Yes (mostly for mining)	No	Yes	No	No	No

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed "Pledge," a unique Proof-of-Honesty (PoH) based block proposer selection protocol that incorporates an IoT-centric TX validation scheme. Pledge reduces the probability of participation by non-performing, and potentially Byzantine nodes in the consensus process by restricting the block proposal responsibility to a couple of honest nodes in the network. It also prevents Sybil attack, avoids quasi-centralization, and averts various attacks against the reputation systems. Pledge is currently designed for consortium blockchains. However, with requisite modifications, it can be deployed in public blockchains as well. Based on our initial experiments and analysis, it is ascertained that Pledge not only satisfies most of the security requirements discussed in this paper but is also computationally efficient with an insignificant change in communications overhead. In the future, we aim to develop a working prototype of Pledge Protocol including a customized blockchain to further analyze the security and performance indicators of the proposed scheme.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] EconoTimes, "Blockchain project Antshares explains reasons for choosing dBFT over PoW and PoS," 2017. Last accessed 4 October 2020. [Online]. Available: http://www.econotimes.com/Blockchain-project-Antshares-explains-reasons-for-choosing-dBFT-over-PoW-and-PoS-659275

[3] V. Gramoli, "From blockchain consensus back to Byzantine consensus," *Future Generation Computer Systems*, vol. 88, pp. 173–90, 2017.

[4] M. Castro, B. Liskov *et al.*, "Practical Byzantine fault tolerance," in *OSDI*, vol. 99, 1999, pp. 173–186.

[5] M. Castro and B. Liskov, "Practical Byzantine fault tolerance and proactive recovery," *ACM Transactions on Computer Systems (TOCS)*, vol. 20, no. 4, pp. 398–461, 2002.

[6] C. Decker and R. Wattenhofer, "Information propagation in the Bitcoin network," in *Proceedings of the 13th International Conference on Peer-to-Peer Computing (P2P)*. IEEE, 2013, pp. 1–10.

[7] NEO.org, "NEO- White Paper," 2017. Last accessed 4 October 2020. [Online]. Available: https://docs.neo.org/docs/en-us/basic/whitepaper.html

[8] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The honey badger of BFT protocols," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 31–42.

[9] I. Makhdoom, M. Abolhasan, H. Abbas, and W. Ni, "Blockchain's adoption in IoT: The challenges, and a way forward," *Journal of Network and Computer Applications*, vol. 125, pp. 251 – 279, 2018.

[10] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication," in *Proceedings of the International Workshop on Open Problems in Network Security*. Springer, 2015, pp. 112–125.

[11] P. Szilgyi, "EIP-225: Clique Proof of Authority Consensus Protocol," 2017. Last accessed 4 October 2020. [Online]. Available: https://eips.ethereum.org/EIPS/eip-225

[12] J. Bridges, "Reputation lessons from Warren Buffett," 2013. Last accessed 2 October 2020. [Online]. Available: https://www.reputationdefender.com/blog/orm/reputation-lessons-warren-buffett

[13] Bitcoin-Developer-Guide, "Transactions," Developer Guide, 2018. Last accessed 3 October 2020. [Online]. Available: https://bitcoin.org/en/developer-guide#transactions

[14] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *White Paper*, vol. 3, pp. 1–36, 2014.

[15] "Transaction flow," 2019. Last accessed 5 October 2020. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/txflow.html

[16] I. Makhdoom, M. Abolhasan, and W. Ni, "Blockchain for IoT: The challenges and a way forward," in *Proceedings of the 15^{th} International Joint Conference on e-Business and Telecommunications - Volume 2: SECRYPT*, INSTICC. SciTePress, 2018, pp. 428–439.

[17] P. Ducklin, "Mirai: Internet of Things malware from Krebs DDoS attack goes open source," 2016. Last accessed 2 October 2020. [Online]. Available: https://nakedsecurity.sophos.com/2016/10/05/mirai/

[18] O. Choudhury, H. Sarker, N. Rudolph, M. Foreman, N. Fay, M. Dhuliawala, I. Sylla, N. Fairoza, and A. K. Das, "Enforcing human subject regulations using blockchain and smart contracts," *Blockchain in healthcare today*, 2018.

[19] P. Genestier, S. Zouarhi, P. Limeux, D. Excoffier, A. Prola, S. Sandon, and J.-M. Temerson, "Blockchain for consent management in the ehealth environment: A nugget for privacy and security challenges," *Journal of the International Society for Telemedicine and eHealth*, vol. 5, pp. GKR–e24, 2017.

[20] A. Dorri, S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *Proceedings of the 2^{nd} Workshop on security, privacy, and trust in the Internet of things (PERCOM), Hawaii, USA*. IEEE, 2017.

[21] R. Dennis and G. Owen, "Rep on the block: A next generation reputation system based on the blockchain," in *Proceedings of the 10^{th} International Conference for Internet Technology and Secured Transactions (ICITST)*. IEEE, 2015, pp. 131–138.

[22] Y. Wang and J. Vassileva, "Trust and reputation model in peer-to-peer networks," in *Proceedings of the 3^{rd} International Conference on Peer-to-Peer Computing (P2P2003)*. IEEE, 2003, pp. 150–157.

[23] M. Gupta, P. Judge, and M. Ammar, "A reputation system for peer-to-peer networks," in *Proceedings of the 13^{th} International Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, 2003, pp. 144–152.

[24] M. Herlihy and M. Moir, "Enhancing accountability and trust in distributed ledgers," *arXiv preprint arXiv 1606.07490*, pp. 1–15, 2016.

[25] A. Mohan and D. M. Blough, "Attributetrust a framework for evaluating trust in aggregated attributes via a reputation system," in *Proceedings of the 6^{th} Annual Conference on Privacy, Security and Trust*. IEEE, 2008, pp. 201–212.

[26] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007.

[27] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," *ACM Computing Surveys (CSUR)*, vol. 42, no. 1, pp. 1–34, 2009.

[28] I. Makhdoom., F. Tofigh., I. Zhou., M. Abolhasan., and J. Lipman, "PLEDGE: A Proof-of-Honesty based Consensus Protocol for Blockchain-based IoT Systems," in *Proceedings of the IEEE International Conference on Blockchain and Cryptocurrency*, IEEE-ComSoc. Toronto, Canada: IEEE, 2020, pp. 1–3.

[29] J. Kwon, "Tendermint: Consensus without mining," *Draft v. 0.6, fall*, vol. 1, pp. 1–11, 2014.

[30] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[31] Solidity-Documentation, "Events," 2020. Last accessed 5 October 2020. [Online]. Available: https://buildmedia.readthedocs.org/media/pdf/solidity/develop/solidity.pdf

[32] R. Kastelein, "Intel Jumps into blockchain technology storm with Sawtooth Lake distributed ledger," 2016. Last accessed 8 October 2020. [Online]. Available: http://www.the-blockchain.com/2016/04/09/

[33] "Remix-Ethereum-IDE," 2019. Last accessed 2 October 2020. [Online]. Available: https://remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-v0.5.11+commit.c082d0b4.js

[34] "Whats the maximum Ethereum block size?" 2019. Last accessed 4 October 2020. [Online]. Available: https://ethgasstation.info/blog/ethereum-block-size/

[35] J. Linn and M. Nyström, "Attribute certification: An enabling technology for delegation and role-based controls in distributed environments," in *Proceedings of the 4^{th} ACM workshop on Role-based access control*. ACM, 1999, pp. 121–130.

[36] M. K. Reiter and S. G. Stubblebine, "Authentication metric analysis and design," *ACM Transactions on Information and System Security (TISSEC)*, vol. 2, no. 2, pp. 138–158, 1999.

[37] D. W. Chadwick, "Authorisation using attributes from multiple authorities," in *Proceedings of the 15^{th} International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'06)*. IEEE, 2006, pp. 326–331.

[38] N. Klingenstein, "Attribute aggregation and federated identity," in *Proceedings of the International Symposium on Applications and the Internet Workshops*. IEEE, 2007, pp. 1–4.

[39] E. Heilman, N. Narula, G. Tanzer, J. Lovejoy, M. Colavita, M. Virza, and T. Dryja, "Cryptanalysis of curl-p and other attacks on the iota cryptocurrency," *IACR Transactions on Symmetric Cryptology*, pp. 367–391, 2020.

[40] Ethereum-StackExchange, "Block time and confirmation time," 2019. Last accessed 5 October 2020. [Online]. Available: https://ethereum.stackexchange.com/questions/56338/block-time-and-confirmation-time

[41] StackExchange, "What is the average transaction time in IOTA?" 2019. Last accessed 4 October 2020. [Online]. Available: https://iota.stackexchange.com/questions/88/what-is-the-average-transaction-time-in-iota