An assessment of blockchain consensus protocols for the Internet of Things

Beverley Mackenzie
Xavier Bellekens
Robert Ian Ferguron

This is the accepted version of a paper presented at the International Conference on Internet of Things, Embedded Systems and Communications (IINTEC 2018), December 20-22, 2018, Hammamet, Tunisia which will be published in the conference proceedings by IEEE

An Assessment of Blockchain Consensus Protocols for the Internet of Things

*

Beverley Mackenzie, Xavier Bellekens and Robert Ian Ferguron
Division of Computing and Mathematics
University of Abertay, Kydd Building
Bell Street, Dundee DD1 1HG
Scotland, UK

***Abstract*—In a few short years the Internet of Things has become an intrinsic part of everyday life, with connected devices included in products created for homes, cars and even medical equipment. But its rapid growth has created several security problems, with respect to the transmission and storage of vast amounts of customers data, across an insecure heterogeneous collection of networks.**

The Internet of Things is therefore creating a unique set of risk and problems that will affect most households. From breaches in confidentiality, which could allow users to be snooped on, through to failures in integrity, which could lead to consumer data being compromised; devices are presenting many security challenges to which consumers are ill equipped to protect themselves from.

Moreover, when this is coupled with the heterogeneous nature of the industry, and the interoperable and scalability problems it becomes apparent that the Internet of Things has created an increased attack surface from which security vulnerabilities may be easily exploited.

However, it has been conjectured that blockchain may provide a solution to the Internet of Things security and scalability problems. Because of blockchain's immutability, integrity and scalability, it is possible that its architecture could be used for the storage and transfer of Internet of Things data.

Within this paper a cross section of blockchain consensus protocols have been assessed against a requirement framework, to establish each consensus protocols strengths and weaknesses with respect to their potential implementation in an Internet of Things blockchain environment.

***Index Terms*—blockchain, iot, security, privacy, consensus protocols**

I. Introduction

The Internet of Things (IoT) is a collection of physical devices, such as home appliances, medical equipment and cars, which contain software which allows them to monitor, collect and actuators personal data across networks. Devices such as smart speakers, smart cameras, smart monitors and even smart cars, which possess the capability of communicating with each other and/or us to provide information or to carry out orders - with some devices even capable of autonomous decision making. Yet, IoT data is regularly transmitted through poorly protected, hostile networks where they can be snooped upon or stolen. [1].

978-1-5386-9131-1/18/$31.00 ©2018 IEEE

In part this has been due to the fact that the IoT environment is made up of a mesh of technologies which operate within a conglomerate of protocols; and because of careless program design, various IoT security issues are being created [2]. In 'A Survey of Blockchain Security Issues and Challenges' [1], it was postulated that because of the heterogeneous structure of IoT devices the protection of data had been made a complex issue.

Thus, some of the main challenges that will be faced by IoT devices will be data integrity security and scalability; which leaves the IoT landscape, in need of a novel and unique solution to its problems. [3] [4].

This fact was confirmed by Hewlett Packard, in their Internet of Things Research Study, [1]. In this report various IoT security violations were identified. In particular it was established that 80 % of IoT devices were failing to implement secure passwords; and a further 60 % of devices were failing to encrypt users' data. It thus seems apparent that the IoT environment would benefit from a uniform framework with respect to its assessment of a device's security requirements [5]. Moreover, it has been theorised that blockchain, the architecture behind cryptocurrency, when combined with the correct consensus protocol may provide a solution to the problem. Insofar as , it is possible that it could provide IoT devices' with integrity, scalability and immutable while operating in a data intense environment.

This paper will therefore, attempt to identify IoT vulnerabilities and requirements. Next it will assess various blockchain architecture and consensus protocols against these requirements. In particularly this paper will:

1) Outline IoT security requirements;
2) Outline the security mechanisms that will be needed to facilitate these requirement;
3) Provide a framework for cataloguing a consensus protocols, compliance with these requirements;
4) And finally, drawing up an assessment of blockchain consensus protocols for the Internet of Things in accor-

[1]HP Internet of Things research study, http://www8.hp.com 2018 accessed online

dance to IoT requirements and mechanisms.

This paper is structured as follows: section I contains the Introduction to the problem; in section II the Background is given; section III contains IoT Requirements; section IV contains Services and Mechanisms; Assessments Criteria is in section V and Conclusion is in VI.

II. Background

A. IoT

IoT refers to numerous physical objects - 20 billion by 2020, according to Gartner [6] [4]. It encompasses technologies and nodes which are responsible for smart grids, power plants, smart homes, intelligent transportation and smart cities. Devices which are designed to provide ambient intelligent computing.

A burgeoning peer-to-peer network of distributed independently operable things, that are equipped to:

- Sense their environment;
- Gather data;
- Forward data;
- Receive data.

IoT networks, transmit and receive data that is sensitive to their users, across a variety of network devices. But many of these networks are unsafe and when this is coupled with the fact that up to 80 % of IoT devices fail to implement basic security [7], it becomes apparent that an interoperable security solution to IoT's requirement is necessary.

B. Blockchain

Blockchain is a major area of interest within the computing, information security and FinTec fields. Due to its potential immutability and its ability to detect integrity violations, along with its other security properties, it is possible that blockchain could be of interest to the IoT industry. [8].

A Blockchain is created by chaining together blocks of transactional data, to create a ledger. A blockchain may contain one or more ledgers, which could be relevant to one or more organisation. Digital signatures are also used by many blockchain architectures, to authenticate each transaction.

A chain is made up of several individual blocks, which are joined together. Each individual block consist of two parts, the block header and the block body. The block header includes:

1) Parent block hash, (256 bit hash value that points to the parent of the block);
2) Merkle root hash, (the hash value of all the blocks in the chain);
3) A time stamp;
4) A nonce, (a 4 byte use once unique identifier).

The Ledger is a distributed ledger, which is used to record transactions [9]; it is also the mechanism behind cryptocurrency.

In general its architecture is designed to fit one of two structures: private permissioned or public permissionless. A private ledger is owned by a centralised organisation. It is a permissioned ledger where all usage, processing, management and recording of transactions are done by authorised parties. A permissioned ledger may be representative of the transaction of a single organisation, or it may be representative of the transaction of an umbrella company, under which many organisation may operate, (e.g.: Ethereum).

The consensus protocol. The Purpose of a blockchain consensus protocols is that it provide a method for the recording and validation of transactions that take place in the distributed network. It is the authentication algorithm which is used to check the completeness and correctness of each blockchain transaction, before the transaction is committed to the chain. For a consensus protocol to be affective it must be difficult to replicate, duplicate or appropriate a transaction. It must therefore, contain mechanism for securing its self from miscreant activities. A consensus protocol is the fundamental part of blockchain technology, which provides validation and security for the data contained in a blockchain ledger.

Some consensus protocols require a numerically challenging equation to be completed before a block is committed to the chain. Blocks are then mathematically hashed to each other and it is this chaining process that gives some blockchains their immutable quality. Fundamentally, the consensus protocol is responsible for ensuring the contents of each block is valid.

Blockchain consensus protocols may be placed in to one of two groups, quorum or deterministic. Quorum algorithms (e.g.: proof of work) are based on a resources intensive analytical behaviour, with regards to choosing the transaction analyst and agreeing on a block's validity. Whereas, a deterministic algorithm uses pseudo-randomness to identifying an analyst and to agreeing on various block issues (e.g.: proof of luck).

One of the main problems associated with many consensus protocols, which are used by cryptocurrency and thus blockchain, is the requirement that all analyst participate in their consensus [10]. Due to this fact, many blockchain environment are cumbersome and resource intensive.

In an attempt to rectify this, resource issue, many more consensus protocols have been designed and proposed by individuals and various organisations. Yet many of these have not been subject to peer review and others, while asserting the qualities of their consensus protocols, fail to provide detailed algorithmic information.

The industry's attempt to rectify the resources issues, however, has of late seen many more incantation of quorum based protocols being proposed. These new breed of consensus algorithms are, often coupled with a secure chip architecture (e.g.: proof of luck and Intel SGX), which is intended to provide an extra layer of security while reducing the resources scalability problem.

Digital Signature are used by many blockchains as a means of authenticating transactions and validating a users transaction before it is placed in a block, (digital signature are based on the Diffie-Hellman key exchange algorithm. It is used in protocols such as SSL, TLS and IPsec VPN). The algorithm uses a public and private key pair, which can be used to allows a shared secret to be passed between two parties or to validate

a transaction. Digital signature's authentication and validation stages are thus:

1) Bob a blockchain user, is issued with a private and public key - digital signature;
2) Bob keeps his private keys secret, but he issues his public key to other users and blockchain analysts;
3) Bob instigates a transaction;
4) Bob then takes a hash of the transaction and encrypt the hash with his private key;
5) Bob appends the encrypted hash to the transaction - i.e. the hash is his signature;
6) Alice, a blockchain analysts, receives Bob's transaction, with the encrypted hash;
7) Alice then uses Bob's public key to decrypt the hash;
8) Alice compares data - if the decrypted transaction data and the actual transaction data are the same, authenticate and validation has been achieved.

C. Security Issues

Depending on the chosen consensus protocol, a blockchain may suffer from several security issues [1], some of which are:

- 51% attack. Consensus protocols that uses a quorum method for the agreement of transactions are susceptible to this type of attack. The 51% attack occurs when a group of miscreant analyst take control of more than 50% of a network. Thereafter, they would be able to control all network transactions. [11];
- Double fork. The double fork attack is effective when applied to blockchain environments in which the total value of a chain is used as an indicator of the validity of chain. The double fork attack occurs when miscreants create two chains which are equidistant from the genesis, with an equal number of transaction. A decision on the dominant chain is based on the value of each chain. [1];
- Double spend. The double spend attacks is effective because it exploits the spending and authorisation lag time that exists in most blockchain environment. A transaction is created that moves funds to a merchant address. Once the transaction is entered into the current block, possession of the goods is taken. But before the block is authorised a second transaction, relating to the same monies is created. The double spend attack can be due to either erroneous or fraudulent behaviour. All blockchains that use a distributed single ledger structure are susceptible to this attack. [1];
- Scaling problem. The scaling problem can manifest itself in one of two ways. Firstly, it could be caused by the high level of resources utilisation that occurs with some consensus protocols; or secondly, it may refer to network latency, which is due to the Maximum Transmission Unit (MTU) of IoT device. Which in turn could lead to packet storms. Bitcoin is susceptible to a resources scalability problem, because of its uses of Proof of Work (PoW). [1];
- Sybil attack. This occurs when identities are forged, and a networks reputation is undermined. The sybil attack is only affective if a blockchain environment fails to use integrity and authorisation security mechanisms. [12] ;
- Eclipse attack. For this attack to work your surrounding peers must be both malicious and in cahoots with each other. They then work together to prevent you from being well connected to the network. You are thereafter, ill-equipped to verify transactions. The revision attack is only affective if a blockchain environment fails to use immutability. [13].
- Revision Attack. This is when data that has already been authenticated and added to the chain, is amended at a later date. The revision attack is only effective if a blockchain environment fails to use immutability.

It is therefore important to ensure that the blockchain, consensus protocol mitigate or removes a potential attack. In part this may be achieved by the inclusion of the correct security mechanism.

D.

In order to provide transaction data flow from the IoT device to the consensus protocol, it will be necessary to use a blockchain architecture that facilitates the seamless and scalable movement of data. [14]. Such a blockchain would therefore need to possess the following key properties:

- Decentralization;
- Transaction Speed;
- Security;
- Scalability;
- Anonymity (pseudo-anonymity);
- Auditability;
- Efficiency;
- Immutability.

Although, it should be noted that some of these properties will be dependent on the blockchain consensus protocol.

This assessment looked at the following blockchain architectures so as to assess whether they met the requirements of an IoT environment: Ethereum - because it is a well developed blockchain environments that uses smart contracts; Corda - because its a prominent financial blockchain environment which was designed to uses smart contracts; Hyperledger ledger - which is an umbrella organisation that includes several blockchain smart contract designs, some of which are modular.

The aforementioned architectures were assessed against the IoT blockchain requirements with respect to their potential as architectural solutions to the IoT blockchain architectural requirements.

Ethereum is a permissionless ledger, which initially used a proof of work consensus protocol with respect to the ordering and validity of a transaction; although, it is presently transiting to casper, a proof of stake consensus protocol. It is a non-modular smart contract platform, with a transaction block speed of 12 seconds.

Moreover, it has been shown to contain security and efficiency issues at the solidity, Ethereum Virtual Machine

(EVM) and blockchain levels [15] which are compromising its immutability, efficiency, auditability and security.

Corda was designed by the financial industry, for the financial industry. It is a permissioned private smart contract platform, which combines smart contracts with smart legal contracts. It uses a consensus of state validity, consensus of state uniqueness and a notary with respect to transaction authentication and validation. However, it is proprietary with restricted data on its blockchain architecture and consensus protocol being available. It also has not been subjected to peer review, with respect to its efficiency and security [16].

Hyperledger [2] is a collective name used by a collection of blockchain solutions offered up by the Linux Foundation. Under the umbrella of Hyperledger five blockchain solutions have been created.

- Burrow
- Fabric
- Iroha
- Sawtooth
- Indy

After reviewing the aforementioned, four were considered to be at a sufficient enough development stage to be of interest. These being: Sawtooth, Iroha, Burrow and Fabric.

Sawtooth Sawtooth [17] is a blockchain ledger that has been published by Intel under the umbrella of the Linux Foundation. The blockchain can be used over a permission or permissionless environment. It has been designed to be used in industries who require a blockchain mechanism to facilitate their transaction data storage and transmission. In regards to its consensus protocol it uses a lottery protocol and a Proof of Elapsed Time (PoET), which uses the provision contained within the TEE (trusted execution environment) [18] of the Intel SGX [19] processor to instigate and manage its consensus protocol.

Whereas, the ledger's ability to handle transaction data appears to meet some of the requirements of an IoT environment, its architectural design has been discounted on the grounds that it was not possible to find algorithmic details of the consensus protocol, PoET. However, as this transaction ledger is still in its development phase, further consideration maybe given to it at a later time.

Iroha Iroha [20] uses C++ to provide ledger capabilities for mobile and web development projects. Within the Iroha environment Hyperledger blockchain can store two types of data objects and functions. It is therefore an object orientated environment, and as such it could meet the needs of IoT devices. The server validation system was able to perform the following checks on each activity: Data throughput test; a version test; a computational test; and a data consistency test.

Once again, this was still at an early stage and little to no information was given on its consensus algorithm. It is therefore a project that may require further consideration, at a later stage in its development cycle.

Burrow At the time of carrying out this assessment, the Burrows blockchain environment was still at its incubation stage. As a consequent of this it was difficult to obtain technical details.

However, a review of available data indicated that Burrow would be based on the smart contract model, with many of the technical qualities associated with Ethereum.

Moreover, with little to no information on its technical schematics, it was not possible to carry out an in-depth technical review of this blockchain model.

Fabric Fabric is a modular pluggable architecture, which can be changed in accordance to a user's requirements. It also provides its user with access to a certification authority and it facilitates public / private encryption and digital signature. But, PBFT and Kafka are the consensus protocols that are being used by Fabric, and as it will be demonstrated in section V, both of these fall short of IoT's requisites. However, it should also be noted that due to the pluggable nature of Fabric, these consensus protocols could either be substituted out, or a mitigating technical architecture could be put in place.

III. IoT Requirements

During the product development life cycle, often the security development life cycle is omitted, and as it is often shown this is invariably to the detriment of the product. Yet this situation is avoidable. By identifying the security requirements of a product, alongside its technical requirements, it may be possible that the correct mechanisms could be put in place, which would have mitigated many present day IoT security vulnerabilities. Moreover, the application of this approach is demonstrated by the Microsoft Development Security, Life Cycle; an approach which incorporate security and privacy consideration at every stage of the development life cycle. [3]

In this vain, it is the intent of this assessment to identify IoT's security and service requirements and their implementation methods.

With respect to IoT services requirements, several frameworks and models were reviewed, in an attempt to identify an applicable structure for an IoT consensus protocol structure. In particular the:

- CIA triad
- Parkerian Hexad
- ISO 7498-2

were referenced because of their importance with respect to network security and application security.

However, after reviewing Parkerian Hexad it was establish that only two of its six elements were applicable to an IoT consensus protocol, (see table one for the IoT consensus protocol requirements), i.e. integrity and authenticity.

In the case of the CIA triad, only integrity was found to be relevant.

ISO 7498-2 was, however with respect to this review, found to be sufficient with respect to its range of security requirements.

[2]Linux foundation, Hyperledger http://hyperledger.org

[3]https://www.microsoft.com/en-us/securityengineering/sdl/

IV. Services and Mechanisms

The term services [21], as used by this report, was first defined by ISO 7498 to mean the security goals of an application. This report has, however, expanded this definition to include the overall goals of an application, that is: security and functionality.

The term security mechanisms [21], as used by this report, was defined by ISO 7498 to mean the way in which a service goal would be achieved.

The term requirements, as used in this paper, will refer to the combination of an IoT's service and mechanism requirements.

The review classification will therefore be based on an IoT's services and mechanisms requirements.

A. Security Services and Mechanisms

As outlined by ISO 7498 there are five categories of security services, but only four security services are relevant to this research, these being:

Authentication, this can be split into: Entity authentication - ensuring the person you are communicating with is the person you intend to be communicating with; data origin authentication - ensuring the data you receive is complete and correct.

Authentication, in all its guises, is important to IoT, insofar as, it is a service which provides assurance with respect to the person from whom you receive data and the data itself.

Data Integrity, preventing an unauthorised entity from carrying out unauthorised changes or destruction of data is achieved by the implementation of an data integrity mechanism.

The integrity of each blockchain transaction should be verifiable and accountable; Therefore, data integrity is a desirable goal.

Non-Repudiation, preventing an entity from denying they took a specific action is achieved by the instigation of a non-repudiation mechanism.

In line with the transactional nature of blockchain non-repudiation should be considered a fundamental security goal.

As stated above a security mechanism is a means by which a security service may be implemented. ISO 7498 identified eight mechanisms but only six are relevant to the overall IoT architecture and only two are relevant to the consensus protocol. These being:

Encipherment, is a way of hiding information. it uses mechanisms such as steganography and encryption. Encipherment may also be subdivided into reversible and irreversible.

Due to the open nature of a ledger, it would not be possible to encipher IoT data, at the application layer (i.e. within the ledger). Although, at the physical and network layer encipherment of transaction data would be necessary to protect it against: a man in the middle attack, snooping and eavesdropping.

A blockchain consensus protocol would therefore, not be required to provide this security service.

Services	**Mechanisms**
Entity Authentication	Encipherment, Digital Signature
Data Origin Authentication	Encipherment, Digital Signature
Data Integrity	Encipherment
Non-repudiation	Digital Signature
Immutability	Auditability, Merkel Chain, Hash Mac
Scalability	Resource Management

TABLE I
IoT Services Requirements and Mechanisms Framework

Digital Signature, there are two parts to digital signature: signing and verifying. A digital signature can provide: non-repudiation; entity authentication and integrity which are IoT requirements.

Access Control, relates to methods used to ensure only authorised persons have access to data. However, due to the open nature of a blockchain ledger, such a control would not be needed at the application layer. Although, at the network layer of a permissioned ledger, it is possible that some industries will require an access control matrix.

Threfore, blockchain access control is outside the scope of this paper.

Notarisation, is implemented by a third party, usually a certification authority, who provides guarantees with respect to data origin and integrity. it would therefore not be relevant to the consensus protocol, but it would be relevant to the blockchain architecture.

It is also worth noting that security mechanisms may take the form of either a specific security mechanism or a pervasive security mechanism. A pervasive mechanism is a mechanism that is not mutually inclusive to a single security service. Whereas, a specific security mechanism is one that is relevant to a specific security service.

Consensus protocols will be assessed with respect to both their pervasive and specific mechanisms.

B. Immutability

It should be noted that, a security service that was not mentioned in ISO 7498 is immutability, a service that is important to IoT. Immutability relates to the requirement that it should be impossible / difficult to change a blockchain; from the genesis block to the present block; the contents of a blockchain, once verified, should be constant and fixed. As this is a fundamental blockchain mechanism it should be considered in an assessment.

In conclusion for an IoT consensus protocol to be considered secure it must include mechanism which provide: authentication; integrity; non-repudiation and immutability.

C. Technical Requirements

The Internet of Things has been described as a paradigm that is gaining control within the world [22]. Yet, its requirements have not, as yet been standardized; which has led to vulnerabilities and operability issues. In 'The Internet of Things: A Survey' [22], it was highlighted that security issues relating to integrity, anonymity and adaptability / scalability

were blockchain's main areas of concern. These concerns have been echoed in several other papers [2] [23].

The Open Web Application Security Project [24] has also confirmed the top vulnerabilities that are facing IoT as being: Insecure Interface; Insufficient Authentication/Authorisation; Insecure Network Services; Lack of Transport Encryption; Privacy Concerns; Insecure Software/Firmware; and, Poor Physical Security, which is in line with earlier summation.

In The Computer for the 21st Century [25], it was also proposed that ubiquitous computing should have three main services, which must first be met, if IoT is to become an invisible ambient intelligence [26], these being: cost, power consumption and networking capability, i.e. scalability.

As of 2014 Bitcoin electricity consumption was on par with Ireland's electricity usage [27] and it has been predicted that bitcoin will will be consuming approximately 0.5 % of the worlds energy by the end of 2018, (as cited in 2018 [28]). Scalability would therefore be considered a fundamental IoT requirement.

IoT's requirements were further explored by Zhi-Kai Zhang et al [2]; in this paper the security concerns affecting IoT devices and networks, which were defined as:

- Identification and location method problems;
- Authentication and authorisation methods need to be established;
- Storage and memory restrictions data privacy;
- Malware vulnerability; privacy;
- Software vulnerability;
- and IoT malware.

Threfore, it has also been established that scalability should also be considered a requirement.

V. Assessment Criteria

A. Framework

Entity Authentication objective: Mechanism should be put in place which ensures the person you are talking to is the person you intend to be communicating with. **Purpose:** Protecting data from unauthorised access and unauthorised changes.

Non-repudiation objective: Data verification mechanism should be implemented **Purpose:** Preventing an entity from denying they took a specific action.

Data Integrity objective A System for verifying each blockchain transaction should be in place **Purpose:** To prevent an authorised entity from carrying out authorised action.

Immutability objective Once data is committed to a chain it should impossible or computationally difficult to change said data. **Purpose** Provides a fixed completeness of data.

Scalability objective Use of these security instruments should have limited and/or no impact on service. **Purpose** To ensure implementation, these security requirements should not have a significant impact on service provision.

B. Proof of Work PoW

In 1993 Cynthia Dwork and Moni Naor (as cited in [29]) came up with the first proof of work concept. However, it wasn't until 1999 when Markus Jakobsson and Ari Juels [29] coined the term Proof of Work [30] in their paper, 'Proof of work and Bread pudding' protocols. PoW is a protocol which requires a resource intensive level of work from a miner, with respect to authentication, verification and commitment of a transaction. Used by Bitcoin, it is a decentralised, permission-less network with the blockchain being replicated on multiple nodes, throughout the network.

It uses Diffie Hellmen public / private key pairs for both the authentication of a user, and the verification of their transactions. Users sign all of their transactions with their private key and minors uses public key to authenticate relevant transaction. It is a quorum-based consensus protocol, in that all minors have to agree on the validity of each transaction, before a transaction can be added to a chain.

The header of each new block contains:

- The previous block hash, which when combined with the Merkle chain gives the chain Integrity;
- A difficulty requirement. The blocks header is hashed by the use of a hash algorithm, usually sha-256, the hash output string must also meet a formatting requirement that it contains a specific number of leading zeros, (at the time of writing this paper the requirement was 18). This feature gives PoW its immutability quality, although it is also responsible for its resource usages scalability issues, as minors hash and rehash each block, in an attempt to meet this requirement [8].
- A nonce, which is a pseudo-random number which is used to protect against a reply attack and/or changes to the data. It therefore, provides PoW with data integrity.
- The previous blocks hash.

PoW consensus protocol is also theoretically susceptible to: 51 percent attack, scaling problem; Sybil attack; eclipse attack and the double fork attack.

C. Proof of Stake

PoS [31] was created as an alternative to PoW. Very much like PoW it is a distributed network consensus protocol which is based on quorum agreement. However, with PoS the values of a miner's vote is directly proportionate to the number of coins a miner owns; (i.e. If Bob owns 100 coins and Alice owns 10 coin, then the value of Alice's vote, in the quorum, would be ten times less than that of Bob's).

The formula is made more complex by the inclusion of coin age. The age of a coin is defined as the amount of time a currency has been held, multiplied by the value of the coin; (e.g. if bob received 10 coins from Alice and he holds them for 90 days, it would be said that Bob had accumulated coin age of 900 days). Coin age increase the value of a miner's quorum vote. Coin age is also used to:

- Establish coin ownership, (miners are given voting incentives if they own their coins for a long time);
- Decide a dominate fork (i.e.: a dominate fork is the one with the greatest coin value with respect to miners' votes);
- Decide which miner will be responsible for transaction processing (i.e. the miner with the greatest coin age, is

always chosen as the miner responsible for transaction processing).

Like PoW, PoS uses a private / public encryption algorithm, with respect to the authentication and validation of a user's transaction. However, due to the structure of PoS quorum algorithm, PoS blockchains do not provide immutability. PoS is also susceptible revision attack, since the energy and cost requirement for revising a chain is not prohibitive and the double spend attack [32].

Finally, coin ownership is not indicative of an individual's vested interest in the efficient functionality of the consensus protocol [33]; its usage could therefore lead to nothing at stake (NoS), whereby minors who are only interested in accumulating coins, may make decisions which could lead to erroneous action with respect to the ledger.

D. Proof of Luck Consensus Protocol

Proof of Luck Consensus Protocol [8] consists of three parts: proof of ownership; proof of time; and proof of owner. Proof of ownership uses the Intel SGX, (a hardware solution), to provide each analyst with a unique ID, known as EPID. This protects analyst from Sybil attacks and thereby provides analysts with data integrity. By using the Intel SGX Trusted Execution Environment,(TEE) it is possible for proof of time to enforce, a bitcoin like, proof of work requirement, (i.e. that a sufficient amount of time has passed before a new block may be processed). This protects against the double spend attack.

The combination of the protocols and hardware and software allows proof of luck to provide integrity. It also means the consensus protocol has low latency with respect to transaction validation which makes it scalable.

Intel SGX also includes a random number generator, which allows the consensus protocol to carry out random miner selection. [31] [34] However, it was not possible to ascertain whether a digital signature was to be used with respect to clients' and their transaction; therefore it was not possible to establish whether entity authentication and data origin authentication could be achieved.

It also failed to provide information on how it would achieve non-repudiation and immutability.

Due to limited data it was only possible to establish that PoL appears to be susceptible to the revision attack. It should also be noted that the Intel SGX does come with its own security concerns. [35] [19] [34] In particular:

- DRAM bus tapping attack;
- Memory mapping attack;
- Software attacks on peripherals.

E. Kafka

Kafka was developed by LinkedIn [36] as a way of providing transaction scalability. It is a commitment log which is replicated across a distributed network. It is capable of processing between 5,000 and 50,000 queries per second; while suffering little latency. Hyperledger Fabric is also using it as a consensus protocol. Kafka messages (transactions) are distributed by producer nodes, to its subscribed consumer nodes (channels). Consumer nodes are consumer groups that are responsible for running a single consumer message process. Consumer processes are assigned a partition. Messages are held by Kafka's commitment log, for a pre-determine time. During this time period message deletion or amendments are not permitted. Kafka therefore has a limited level of immutable. Kafka uses a cache replication system, to preserve the integrity of its data. It partitions are replicated among replication servers known as brokers. Brokers therefore, guarantee limited data integrity; i.e. during the data retention period. Moreover, Kafka was designed as a fault tolerant scalable ordering system, distributed messaging system for log processing. Its replication system provides system redundancy, which enables it to provide fault tolerance. But it does not provide: entity authentication; data origin authentication; non-repudiation and immutability. Kafka's processes are, however, scalable.

Services	**Consensus**
Entity Authentication	PoW, PoS
Data Origin Authentication	PoL, PoW, PoS
Data Integrity	Pos, PoW, PoL, Kafka
Non-repudiation	PoW, PoS
Immutability	PoW
Scalability	PoS, PoL, kafka, PBFT

TABLE II
IOT SERVICES SECURIRTY REQUIREMENTS AND CONSENSUS PROTOCOLS

F. PBFT

Practical Byzantine Fault Tolerance protocol [30] uses a combination of primary backup and quorum replication techniques to order requests. Initially designed as a means of dealing with byzantine computer memory failure, in the last five years, it has been adapted to be used as a consensus protocol. A PBFT network is managed by a leader node, who is responsible for decision making, based on quorum voting. Within a window of vulnerability, (which relates to the time between replication backups). PBFT is cable of functioning even if a third of the nodes are acting malicious. Node leaders are responsible for receiving requests from clients. Leader node multicast (broadcast) these request to their peer nodes. Once a request is actioned and a third of peer nodes have confirmed the same result, the result is then accepted as the result of the operation. Leader nodes then communicate the result to the client. PBFT has been designed to be used within a permissioned network. Moreover, it does not use any of the traditional security protocols to protect clients and their data and Its security mechanisms, appear to be undocumented. It was therefore not possible to assess its consensus protocol against the IoT's framework.

VI. CONCLUSION

Over the last decade there has been a large-scale role out of various IoT devices. But many of these devices have weak passwords, insecure networks, a lack of operability and scaling

problems. Yet, the IoT industry continues to rapidly grow, while failing to deal with its fundamental security issues. However, if a reoccurrence of previous technological security failures is to be prevented, IoT will require robust security frameworks and standards. Moreover, it has been conjectured that blockchain may provide a solution.

In this paper it was established that IoT security requirements for: entity authentication; data original authentication; data integrity; non-repudiation; immutability and scalability, could in part, be provided by blockchains immutability, confidentiality, integrity and authorisation mechanisms. However, at present no blockchain and consensus protocol could simultaneously meet both the security and scalability requirements.

Finally, the failure to resolve this issue continues to expose the IoT environment to potential miscreant activities. Further research will therefore need to be carried out into design a secure and scalable IoT consensus protocol.

REFERENCES

[1] I.-C. Lin and T.-C. Liao, "A survey of blockchain security issues and challenges.," *IJ Network Security*, vol. 19, no. 5, pp. 653–659, 2017.

[2] Z.-K. Zhang, M. C. Y. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, and S. Shieh, "Iot security: ongoing challenges and research opportunities," in *Service-Oriented Computing and Applications (SOCA), 2014 IEEE 7th International Conference on*, pp. 230–234, IEEE, 2014.

[3] W. Shang, Y. Yu, R. Droms, and L. Zhang, "Challenges in iot networking via tcp/ip architecture," *Technical Report NDN-0038. NDN Project*, 2016.

[4] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," *Computer networks*, vol. 76, pp. 146–164, 2015.

[5] W. Wang, D. T. Hoang, Z. Xiong, D. Niyato, P. Wang, P. Hu, and Y. Wen, "A survey on consensus mechanisms and mining management in blockchain networks," *arXiv preprint arXiv:1805.02707*, 2018.

[6] M. Hung, "Insight on how to lead in a connected world." online, 11 2017. https://www.gartner.com/.

[7] HP, "Internet of things research study." http://www8.hp.com, 1 2018. Accessed online.

[8] M. Milutinovic, W. He, H. Wu, and M. Kanwal, "Proof of luck: An efficient blockchain consensus protocol," in *Proceedings of the 1st Workshop on System Software for Trusted Execution*, p. 2, ACM, 2016.

[9] M. Pilkington, "11 blockchain technology: principles and applications," *Research handbook on digital transformations*, p. 225, 2016.

[10] A. Kiayias and G. Panagiotakos, "Speed-security tradeoffs in blockchain protocols.," *IACR Cryptology ePrint Archive*, vol. 2015, p. 1019, 2015.

[11] D. K. Tosh, S. Shetty, X. Liang, C. A. Kamhoua, K. A. Kwiat, and L. Njilla, "Security implications of blockchain cloud with analysis of block withholding attack," in *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 458–467, IEEE Press, 2017.

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

[13] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network.," in *USENIX Security Symposium*, pp. 129–144, 2015.

[14] M. A. Khan and K. Salah, "Iot security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[15] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts (sok)," in *Principles of Security and Trust*, pp. 164–186, Springer, 2017.

[16] M. Valenta and P. Sandner, "Comparison of ethereum, hyperledger fabric and corda," tech. rep., FSBC Working Paper, 2017.

[17] Hyperledger, "Hyperledger sawtooth." Available https://www.hyperledger.org, 7 2018. Accessed online.

[18] B. Schneier, *Applied cryptography: protocols, algorithms, and source code in C*. john wiley & sons, 2007.

[19] V. Costan and S. Devadas, "Intel sgx explained.," *IACR Cryptology ePrint Archive*, vol. 2016, no. 086, pp. 1–118, 2016.

[20] C. Cachin, "Architecture of the hyperledger blockchain fabric," in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, vol. 310, 2016.

[21] A. W. Dent and C. J. Mitchell, *User's Guide To Cryptography And Standards (Artech House Computer Security)*. Artech House, Inc., 2004.

[22] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[23] Z. Zheng, S. Xie, H.-N. Dai, and H. Wang, "Blockchain challenges and opportunities: A survey," *Work Pap.–2016*, 2016.

[24] C. Smith, "Top 10 iot vulnerabilities." online, 11 2015. http:// https://www.owasp.org/.

[25] M. Weiser, "The computer for the 21st century: specialized elements of hardware and software, connected by wires, radio waves and infrared, will be so ubiquitous that no one will notice their presence," in *Readings in Human–Computer Interaction*, pp. 933–940, Elsevier, 1995.

[26] M. Weiser, "The computer for the 21 st century," *Scientific american*, vol. 265, no. 3, pp. 94–105, 1991.

[27] K. J. O'Dwyer and D. Malone, "Bitcoin mining and its energy footprint," 2014.

[28] A. de Vries, "bitcoin energy consumption," *online*, 2018.

[29] M. Jakobsson and A. Juels, "Proofs of work and bread pudding protocols," in *Secure Information Networks*, pp. 258–272, Springer, 1999.

[30] 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.

[31] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," *self-published paper, August*, vol. 19, 2012.

[32] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in *Proceedings of the 2012 ACM conference on Computer and communications security*, pp. 906–917, ACM, 2012.

[33] A. Poelstra *et al.*, "Distributed consensus from proof of stake is impossible," *Self-published Paper*, 2014.

[34] M. Hoekstra, "Intel® sgx for dummies (intel® sgx design objectives)," *retrieved Sep*, vol. 3, p. 4, 2014.

[35] V. Costan, I. Lebedev, S. Devadas, *et al.*, "Secure processors part ii: Intel sgx security analysis and mit sanctum architecture," *Foundations and Trends® in Electronic Design Automation*, vol. 11, no. 3, pp. 249–361, 2017.

[36] J. Kreps, N. Narkhede, J. Rao, *et al.*, "Kafka: A distributed messaging system for log processing," in *Proceedings of the NetDB*, pp. 1–7, 2011.