A secure and scalable IoT consensus protocol

Beverley MacKenzie
Ian Ferguson
Abdul Razaq

This paper was presented at the EAI SaSeIoT 2021 - 5th EAI International Conference on Safety and Security in Internet of Things, 25th April 2022, virtual conference.

A SECURE AND SCALABLE IOT CONSENSUS PROTOCOL

Beverley MacKenzie[1], Ian Ferguson[2] and Abdul Razaq[3]

[1]Division of Cyber Security, Abertay University, Dundee, United Kingdom

1705191@abertay.ac.uk

[2] Division of Cyber Security, Abertay University, Dundee, United Kingdom

i.ferguson@abertay.ac.uk

[3] Division of Cyber Security, Abertay University, Dundee, United Kingdom

a.razaq@abertay.ac.uk

ABSTRACT

Several consensus algorithms have been proposed as a way of resolving the Byzantine General problem with respect to blockchain consensus process. However, when these consensus algorithms are applied to a distributed, asynchronous network some suffer with security and/or scalability issues, while others suffer with liveness and/or safety issues. This is because the majority of research have not considered the importance of liveness and safety, with respect to the integrity of the consensus decision. In this paper a novel solution to this challenge is presented. A solution that protects blockchain transactions from fraudulent or erroneous mis-spends. This consensus protocol uses a combination of probabilistic randomness, an isomorphic balance authentication, error detection and synchronised time restrictions, when assessing the authenticity and validity of IoT request. Designed to operate in a distributed asynchronous network, this approach increases scalability while maintaining a high transactional throughput, even when faced with Byzantine failure.

KEYWORDS

Blockchain, IoT, Consensus, Consensus Algorithms, Byzantine failure

1. INTRODUCTION

In a few short years the Internet of Things (IoT) has become an intrinsic part of life. Creating a world where computers become ambient technologies which are always on and always available. Technologies which are capable of instinctively obeying demands while inconspicuously remaining in the background. Technologies which are already having a positive effect on the human to computer experience [1]. However, there has been inconsistency with respect to infrastructure design and the application of different protocols [2]. The IoT ecosystem is therefore filled with a range of incompatible technologies, devices and protocols which are plagued by scalability and security issues [3]. Despite this IoT connected

devices are being included in homes, cars, medical equipment, children's' toys and doorbells. In addition to this the IoT ecosystem is now having an adverse impact on the Internet. With miscreants using IoT devices to orchestrate denial-of-service attacks and distributed-denial-of-services attacks [4] [5].

There is now a considerable body of research that has recognised that IoT's heterogeneous mesh of network devices and protocols have created a unique set of risks and problems that will affect most households [6] [7]. 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 [8] [7] [9] [5]. IoT devices are presenting many security challenges to which consumers are ill equipped to protect themselves from [10].

Vulnerabilities that are due to a range of factors including: unsafe networks, infected mesh devices, poor password protection and data being transmitted in clear text [11] [12]. The potential impact of these insecurities was demonstrated by the Mirai and Persirai botnet attacks [13]. The progression of these attacks was via IoT network devices. Due to poor security, miscreants were able to use IoT devices to instigate DDoS attacks on Dyn a DNS service provider, GitHub, Twitter, Reddit, Netflix and Airbnb [13]. Even though it has been shown that the infrastructure suffers with security and scalability issues which are now compromising the internet [12] [14] new IoT devices are continually being rolled out [13], [15], [16].

Notwithstanding this, IoT technology is being embedded in many everyday items. The IoT environment has therefore been described as a poorly protected, hostile network where data may be snooped upon and exploited [7] [17] [18].

To resolve these issues it has been suggested that blockchain technology may contain the answer [19] [20] [21] [22]. Due to blockchain's security characteristics and its ability to securely transfer data across a distributed network, it has been suggested that blockchain could be capable of meeting IoT's security and safety requirements [21]. However, to achieve this Blockchain will need to resolve its security, scalability, safety and liveness issue [10] [8]. A challenge that is at the heart of blockchain breaches [20]. However, if blockchain technology is to be successfully used in the IoT environment an applicable blockchain consensus protocol will need to be identified [23]. A consensus protocol which can deal with billions of IoT request [10].

In this paper a new consensus algorithm is presented. This consensus algorithm first uses randomisation to identify its lead node – a process that reduces scalability. Next it uses error detection to achieve safety and prevent double spend / fraudulent spend. Finally, the random selection of the lead cell and partial synchronisation, between validating and authenticating nodes, enables this consensus protocol to achieve liveness.

Security is achieved via the application of cryptographic primitives to provide non-repudiation, integrity, immutability and confidentiality [10].

Under this model cells who deviate from normal behaviour, in an arbitrary way, are identified by the implementation of an error checking algorithm. The division of data into two channels provides separation of duties [24]. A robust security mechanism which can only be compromised when each node is individually attacked. Moreover, the inclusion of synchronised time, provides a mechanism which protects a cell from becoming compromised by a man-in-the middle attack [25] [26] [27] [28] [29] [30].

Finally, the model also includes cryptographic services which provide data origin authentication, along with encryption of data in transit and data at rest.

This paper is structured as follows: Section 1 contains the Introduction to the problem; in section 2 the Background is given; section 3 contains The Balance Authentication Mechanism; section 4 contains The Consensus Process; Achieving Scalability is in section 5 and the Conclusion is in 6.

2. BACKGROUND

2.1 A Blockchain Consensus Algorithm

Blockchain is the technology behind bitcoin, which was originally described as an electronic currency [31] over the past decade has gained acceptability. A technology which can store digital information in a secure and safe manner. Data held in a blockchain is protected from alterations and snooping while being transmitted across networks. At the heart of this technology is a consensus algorithm. An algorithm which is used by nodes to achieve agreement on the validity of a transaction.

It is now well established from a variety of studies that the development of a peer-to-peer blockchain consensus protocol for the IoT environment could resolve its security and scalability issues [32]. Due to blockchain's security and scalability properties it has been postulated that blockchain smart contracts (BC – smart contracts) could provide secure transportation of IoT traffic [21].

A blockchain consensus algorithm that is capable of being used in the IoT ecosystem has generated a lot of interest. Particularly a technology which could operate in a decentralised data intensive network. If such a consensus protocol could facilitate the transmission of billions of bits of information, it would also be capable of operating within an IoT data intensive environment [23]. This is because of blockchains consensus protocol's ability to provide [10]:

- Pseudo anonymity
- Confidentiality
- Authenticity
- Immutability
- Interoperability
- Scalability
- Privacy
- Non-repudiation
- Data integrity

Although before these technologies can be successfully integrated the correct blockchain consensus protocol will need to be identified.

2.2 The Consensus Problem in Context

It was Lamport et al in 1982 who introduced the network consensus protocol problem [29] In the paper 'The Byzantine General Problem' the issues which affected consensus were discussed. However, it was Fischer et al who provided details on the difficulty of achieving consensus in a distributed network if just one node failed [27]. In this paper it was shown that when an unbounded time network is faced with a single node failure safety and liveness could not be guaranteed. (Safety - All nodes agree on the authenticity of an IoT request; Liveness - All nodes responsible for consensus take part in the process [27]).

Although in 1994 it was Chandra et al who identified how it was possible to circumvent the restrictions which had been laid down in the FLP impossibility problem [25]. In Chandra et al papers four methods that could be used to circumvent these restrictions were proposed [26]:

- Randomisation
- Weak Failure Detection
- Weak Problem and Solution
- Model of Partial Synchronisation

Moreover, two of Chandra et al identified mechanisms have already been used in well-established consensus protocol:

1. Castrol et al consensus protocol uses partial synchronisation in its application of the Practical Byzantine Fault Tolerant consensus protocol. (However, due to quadratic message authentication requirement this consensus protocol contains scalability issues when used in a distributed environment.)
2. Nakamoto in 2006 blockchain Proof of Work (PoW) consensus used randomisation in its lead node identification [31]. (However, PoW's high use of resources renders it impractical for an IoT environment.)

Moreover, over the last decade there have been several consensus protocols, which have been presented as potential blockchain consensus protocols. However, none of these have provided a complete solution to the security, scalability, safety and liveness issues that affect consensus, when that consensus takes places in an asynchronous environment, that is not subjected to bounded time restrictions [33] [34] [35] [36] [37] [38] [39] [40] [31] [28].
The consensus protocol presented in this paper circumvents the FLP restrictions by implementing randomisation, error detection and partial synchronisation.

3. THE BALANCE AUTHENTICATION MECHANISM

In this section a new consensus algorithm is presented. This section contains information pertaining to how this consensus protocol achieves security and scalability.

The model is the combination of a consensus protocol and a blockchain environment. The Balance Authentication Mechanism (BAM) uses probabilistic randomness to choose the lead node. Timing synchronisation with respects to the communication of nodes in a cell [26] [25] [27]. An isomorphic balance equation and an error algorithm are used to ensure liveness and safety.

Security is achieved via the application of the cryptographic primitives of non-repudiation, integrity, immutability and confidentiality [10].

Under this model cells who deviate from normal behaviour, in an arbitrary way, are identified by the implementation of an error checking algorithm. The division of data into two channels provides separation of duties [24]. A robust security mechanism which can only be compromised when each node is individually attacked [24]. Moreover, the inclusion of a synchronised time bound between Acell and Fcell requires both nodes be compromised, if the cell is to be compromised, (a requirement that is in line with a double entry validation process [41] [42]). Both cells have to be compromised with isomorphic data, within the bounded time window. A requirement that protects against dominance attacks. This is because to gain control of the consensus process, a miscreant must gain control of all Acells and Fcells. A task that has an attack surface area of c^2 – where c indicates cells.

However, a miscreant could attempt to gain control of the lead node. A process which would require the miscreant to identify the lead node. Moreover, as the lead node identification is

based on randomness (4.1) in an environment where there are only 2 cells the probability of identification is 0.05; however, in an environment where there are 100 cells the probability of identification is 0.001.

The model also includes cryptography primitives which provide data origin authentication, along with encryption of data in transit and data at rest.

4. THE CONSENSUS PROCESS

BAM's distributed network is broken down into cells (see figure one). Each cell contains four nodes. The communication between nodes in a cell is subjected to synchronise time restrictions. The processor node is responsible for authenticating data and data origin. Data is then separated into two channels and broadcasted to the other members of the cell. The other nodes are responsible for data authentication, validation, verification and consensus. The use of randomness ensures that the byzantine failure of a cell or a node has no impact on the operation of the algorithm [31].

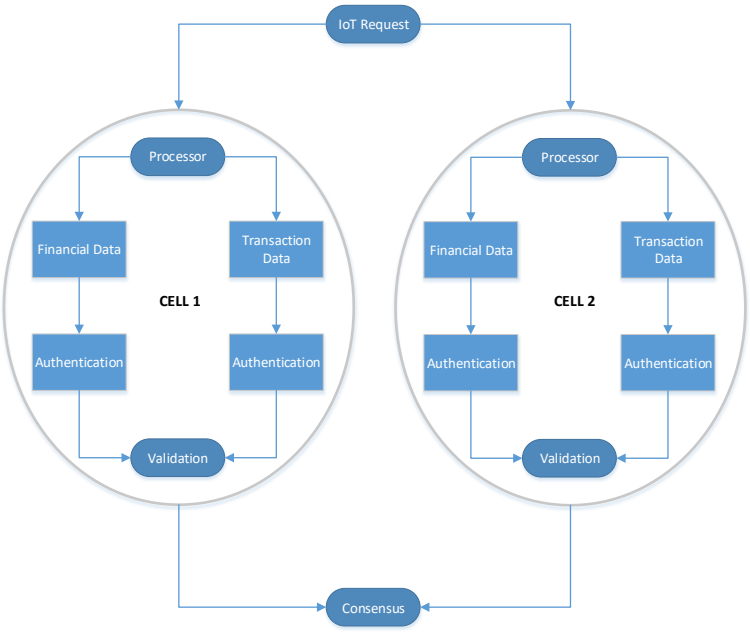


Figure 1 The Consensus Process

4.1 Choosing a Lead Cell

The Balance Authentication lead cell (BA lead cell) is responsible for authenticating and validating each IoT request. The choice of lead cell happens when a cell announces that an IoT authentication and validation has occurred. However, a second announcement of data authentication and validation must occur before the data achieves consensus. Moreover, if a subsequent cell rejects the authentication and validation of an IoT request, then the IoT request is suspended until the error checking mechanism identifies which cell is in an error state.

As in the case of proof of work, BA lead cell identification is based on its location and its transactional speed, i.e. probabilistic random factor in asynchronous network where transactional speed in a constant variable [31].

4.2 An IoT Request

IoT requests are placed in blocks. Each block in the process contains a single transaction. The block contains a request header and a request body. The header contains information pertaining to nonce, date, digital signature, transaction ID, balancing figure and code. The body of the request contains the IoT instructions. Consensus is based on the Boolean valuation of each part of the header data such that consensus is:

$$(Vcell_1(Fcell_1\&\&Acell_1)) \;\&\&\; (Vcell_2(Fcell_2\&\&Acell_2)).$$

The data contained in each IoT request is ordered and separated into two parts – financial data and transactional data. This data is transmitted along two separate and independent channels - financial channel or a transaction channel. Data types are sent along these channels for authentication. Therefore, financial authentication is:

$$a \rightarrow b$$

Transaction authentication is:

$$d \rightarrow c$$

This process complies with the principles of separation of duties and balance authentication [24] [42]. Each channel is responsible for authenticating IoT header data against ledger data, financial data and smart contract data. A Boolean checking algorithm is used in this process.

4.3 A Cell

All nodes in the distributed network are assigned to a cell. Cells contain four nodes. Nodes can either be a finance node (f), transaction node (t), process node (p) or a validation node (v). Each cell is a uniquely identifiable independent entity. All communication between cells operates within a time bounded environment via the application of either asynchronous or synchronous key exchange (a partial synchronised environment).

4.4 Double Spend Protection

To protect from double-spend or fraudulent misappropriation of funds, the consensus protocol complies with the following rules:

1. Only a correct node may propose an IoT request
2. Only a proposed IoT request may be authenticated
3. Only an authenticated IoT request can be validated
4. Only a validated IoT request can achieve consensus

Invariants that are in line with the requirements of safety and liveness [27] [29] [30] A correct node is a node who is a part of a cell. A proposed IoT request is a request which has been proposed by a processing node. An authenticated IoT request is a request which has been authenticated by both transaction and financial nodes. A validated IoT request is a request that

has been validated by a validating node. Consensus is achieved when two cells have authenticated and validated an IoT request.

4.5 The Isomorphic Algorithm

Boolean isomorphic algorithms are used to authenticate each channels data, validate the output of each channel and to error check the consensus process. Such that:

Consensus Algorithm

Input: Client request

Device requests are subjected to cryptographic checks to ensure the data is correct and complete. Only clients' request that pass these checks achieve consensus.

A devices transaction (T) request is defined as containing a: nonce (r), timestamp (t), digital signature (sig_k), code (c), balancing figure (b), action(a), finance (f) and message (m). A client's transaction is therefore defined as $T \in \{r,t,sig_k,c,m,b\}$

Sept 1 Pcell is responsible for authenticating clients' digital signatures
if $sig_k \neq\neq$ smart contract digital signature
then
exit 1, 'error, request failed'
else, data is split into two, financial data, and transactional data. Each half is placed in a block(b). Each block is broadcast via an independent channel to a financial authentication node, or a transaction authentication node.

Step 2 Acell is responsible for authenticating the data contained in $b_1 \in \{a,r,t,sig_k,c_f,m\}$ against the clients smart contract allowed action list
if request $\neq\neq$ smart contract data
then
exit 1, 'error, request failed'
else, print 'authentication data to Vcell'

Step 3 Fcell is responsible for authenticating the data contained in $b_2 \in \{f,r,t\ sig_k,c_t,m\}$
if request $\neq\neq$ smart contract data
then
exit 1, 'error, request failed'
else print 'authentication data to Vcell'

Step 4 Vcell authentication
if Acell $\neq\neq$Fcell
then

exit 1, 'error, request failed/

else print 'authentication has been validated'

---

a →b

d→c

validation (1)

For consensus to be achieved two independent cells have to authenticate the IoT request such that:

$a_1 \rightarrow b_1$

$d_1 \rightarrow c_1$

v_1

$a_2 \rightarrow b_2$

$d_2 \rightarrow c_2$

v_2

C (2)

Consensus is therefore $C_n \leftarrow (v_1 \,\&\, v_2)$ (3)

4.6 The Security Provisions

This model attempts to protect data at rest and data in transit by applying accepted cryptographic primitives. The security mechanism used in this process are in line with an IoT security framework [10] The IoT request's, TCP/IP packet header, contains security mechanisms that provide protection against hacking attacks and errors.

The consensus protocol and blockchain presented in this paper is built to be used in the internet. This protocol is therefore designed to provide security to IoT request packets that uses TCP/IP protocol to travel across the internet. Security provisions are broken down into five stages:

4.6.1 Stage one

Device requests are subjected to cryptographic checks to ensure the data is correct and complete. Only IoT requests which pass these checks are processed.

An IoT request should contain nonce (r), timestamp (t), digital signature (sig_k) two authentication codes (c_f and c_t), balancing figure (f) message (m). The use of r,t,sig_k ensures:

- The IoT request contains data origin authentication - ensuring the data was issued by an authenticated IoT device
- Data Integrity – ensures data has not been tampered with
- Non-repudiation – prevents a user from denying their action

The use of c_t,c_f,b primitives protects against:

- Double spend [43]
- Replay attack [5]
- Eclipse attack [44]

The security that is provided to an IoT request is:
$Request \in (r,t,sig_k,c_t,c_f,b,m)$

4.6.2 Stage Two

Transactional data is split into two, financial data, and transactional data. Each half is placed into a block (b). Each block is broadcast via an independent channel to a financial authentication node, or a transaction authentication node. The following cryptographic primitives are applied to each block of $b1_1 \in (r,t,sig_k,c_f,f,m)$ and $b1_2 \in (r,t,sig_k,c_t,f,m)$. Blocks are therefore provided the following protections:

- *Authentication:* This can be split into: Entity authentication - ensuring the person/system you are communicating with is the person /system you intend to be communicating with; data origin authentication - ensuring the data you received came from the correct place [45]
- *Data Integrity:* Preventing an unauthorised entity from carrying out unauthorised changes or destruction of data. The integrity of each block should be verifiable and accountable [45]
- *Non-Repudiation:* Preventing an entity from denying they took a specific action [45]
- *Access Control:* Access control relates to authorisation methods used to ensure only authorised persons have access to data [45]
- *Immutability:* Immutability provides data with a fixed and unchangeable audit trail [46] [31]

4.6.3 Stage Three

The authentication of data contained in each block is carried out by two nodes, the transaction node and the finance node. The authentication process involves both cells independently authenticating the header and body data. This security mechanism is based on the separations of duties principle [24]. A security mechanism that is used in the balance authentication of the IoT request. It protects the IoT request from carrying out unauthorised actions.

4.6.4 Stage Four

The validation of data is carried out by a fourth node, the validation node (v). The validation node ensures transactional authentication (a_t) and financial authentication (a_f) are equivalent and true, (where true means authenticated). This provides consensus validation. This security mechanism protects the IoT request from unauthorised changes. Validation (v) is achieved when: $F : \{a_t == a_f\} \rightarrow v$

4.6.5 Stage Five

The final stage requires two cells to confirm the authentication and validation of an IoT request. Moreover, because data from both cells must agree it complies with the requirements of liveness and safety.

However, if consensus is rejected the request is suspended while both cells are subjected to an error checking process [25] [26]

Only the first two nodes need to broadcast their confirmation of consensus. Consensus is therefore described as.
$C_n \leftarrow (v_1 \& v_2)$

Once consensus is achieved nodes in a cell are independently responsible for committing the block to their Merkle Tree.

4.7 Error Detection

The system uses a binary decision checking algorithm to check for errors in consensus. Because consensus is based on conditional Boolean logic it is possible to build an error checking mechanism into the system. This error checking method can identify errors which create a byzantine failure. This process also identifies stop and start errors when it prevents Fcell and Acell responding within their synchronized time window.

4.8 Merkle Tree

The Merkle Tree is a record of all IoT requests that have been authenticated or validated by a node. It uses a mathematical formular to create a hash of this data. Each blocks hash is chained to the subsequent blocks hash. It provides data integrity and data auditability.

The validator Merkle tree is the aggregate of all IoT requests. Whereas the financial node's Merkle tree is an aggregate of all financial data and the transaction node is an aggregate of all transaction data.

These Merkle trees are made up of recursive hash pairs of data. Moreover because of the design it is also possible to use the financial node's and the transactional node's Merkle tree to confirm the validity of the validation node's Merkle tree.

5 ACHIEVING SCALABILITY

The use of time complexity to measure the scalability of a consensus protocol was demonstrated in Luu et al [34]. BAM uses probabilistic randomness and an error detection mechanism which enables the process to be scalable. Unlike the quadratic message exchange that takes place in BFT style consensus process, BAM uses a single message process which gives it a time complexity of O(n).

The scalability of this consensus protocol was assessed while 1,000, 2,000, 3,000, 4,000 and 5,000 blocks were processed. It should also be noted that testing was carried out in a virtual environment which ensured time stamps and synchronisation.

The speed and scalability of this consensus process is thus:

Blockchain request	Start Time	Finish Time	Duration	Processing Time
1000	16 44:30.	16.46:11.	101	0.101
2000	16.50:19.	16.52:56.	157	0.0785
3000	16.54:02.	16.57:39.	217	0.072
4000	16:58.59.	17.04:33.	322	0.0805
5000	17.06:14.	17.11:53.	299	0.0598

Table 1 Scalability

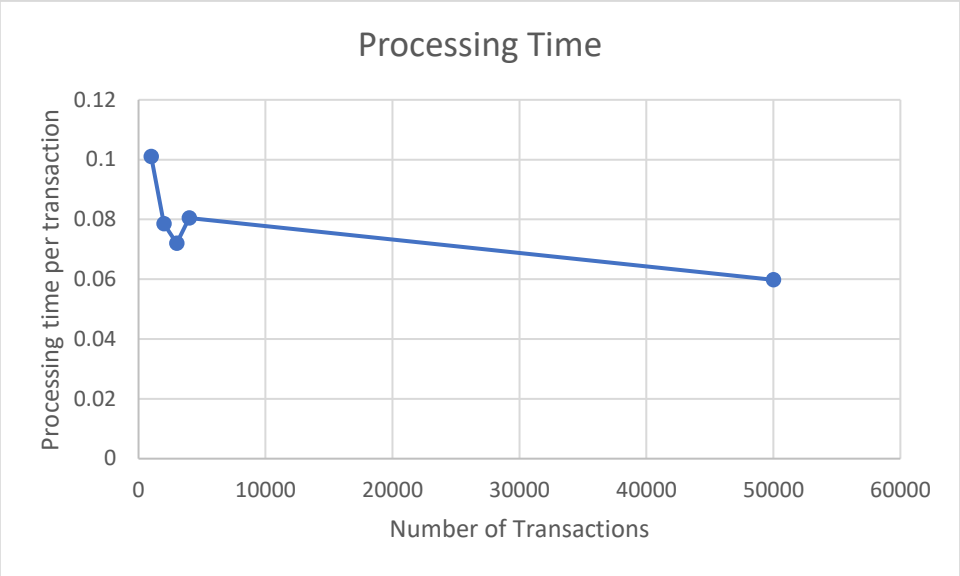


Figure 2 Time Complexity

As it can be seen in figure two and three this protocol has a constant time complexity. Therefore, BAM scalability is based on its time complexity which is constant regardless of an increase in the task.

6.0 CONCLUSION

The main aim of this research paper is to present an IoT Blockchain consensus protocol that meets the requirements of safety and liveness, whilst ensuring security and scalability. This paper first comprehensively explained the limitation of previous researchers, to take these requirements into consideration. An omission which leaves the vast majority of consensus protocols susceptible to dominance attacks, and fraudulent activity. Whilst other consensus protocols, are subjected to a time complexity issue, which leads to exponential growth.

The consensus protocol proposed in this paper offers a solution to the FLP impossibility problem, by providing integrity of the consensus decision [27]. It is a solution that does not have an adverse impact on time complexity, with a linear growth rate – i.e. it is very scalable [34]. The presented solution also used established cryptographic primitives to protect the integrity of data in transit and at rest.

Due to the use of an isomorphic HMAC validation process, the algorithm is tamper resistant – i.e. it provides both integrity protection and integrity detection. Moreover, when this equation is combined with a Merkle tree blockchain algorithm, immutability protection is also provided.

By providing integrity of data at rest, integrity of data in transit and integrity of the consensus decision, this consensus offers a method for circumventing the FLP restrictions [27]. A widely held belief that has been affecting the direction of modern day consensus discussion - with respect to achieving consensus in an asynchronous environment, when faced with unbounded time restrictions. A discussion that is having a direct effect on the security, scalability, safety and liveness of the present day Blockchain environment.

The results of this paper prove that the proposed system is affective at preventing fraudulent spending (double spend) [18] and erroneous consensus. The result of testing also demonstrated that the information security mechanisms of non-repudiation, confidentiality, integrity, authentication and authorisation provides data with protection from miscreant activity.

This consensus protocol guarantees immutability by the use of a Merkle tree, block hashing and an error checking algorithm. A process that ensures data at rest is tamper resistant.
This mathematic approach to the problem removes the need for an exhaustive threat, vulnerability and likeness analysis. Testing of this consensus protocol was in line with the assertions of cryptographic primitives that were employed. A process that confirmed this consensus protocol to be robust enough to withstand attacks on data at rest and in transit.

Testing confirmed the BAM consensus protocol linear time complexity, which means regardless of failure or an increase in requests, this consensus protocol will have a consistent linear increase in processing time. BAM is a consensus protocol that is resistant to byzantine failure, non-byzantine failure and miscreant activity. It therefore protects data at rest, data in transit and the consensus decision process.

Further work will focus on assessing the consensus protocol in a live, wild environment, to assess the robustness of its logic.

REFERENCES

[1] M. Hung, *Insight on How to Lead In a Connected World,* 2017.

[2] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum and N. Ghani, "Demystifying IoT security: an exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations," *IEEE Communications Surveys & Tutorials,* vol. 21, pp. 2702-2733, 2019.

[3] M. Dawson, "Cyber Security Architectural Needs in the Era of Internet of Things and Hyperconnected Systems," 2016.

[4] M. M. Hossain, M. Fotouhi and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the internet of things," in *2015 IEEE World Congress on Services*, 2015.

[5] Y. H. Hwang, "IoT security & privacy: threats and challenges," in *Proceedings of the 1st ACM workshop on IoT privacy, trust, and security*, 2015.

[6] 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.

[7] S. D. Gupta and S. Ghanavati, "Towards a heterogeneous IoT privacy architecture," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 2020.

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

[9] 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*, 2014.

[10] B. MacKenzie, R. I. Ferguson and X. Bellekens, "An assessment of blockchain consensus protocols for the Internet of Things," in *2018 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC)*, 2018.

[11] H. Feng and W. Fu, "Study of Recent Development about Privacy and Security of the Internet of Things," in *2010 International Conference on Web Information Systems and Mining*, 2010.

[12] M. M. Hossain, M. Fotouhi and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the internet of things," in *2015 IEEE World Congress on Services*, 2015.

[13] C. Kolias, G. Kambourakis, A. Stavrou and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer,* vol. 50, pp. 80-84, 2017.

[14] W. Zhou, Y. Jia, A. Peng, Y. Zhang and P. Liu, *IEEE Internet of Things Journal,* vol. 6, pp. 1606-1616, 2019.

[15] S. T. Ali, P. McCorry, P. H.-J. Lee and F. Hao, "Zombiecoin: Powering next-generation botnets with bitcoin," in *International Conference on Financial Cryptography and Data Security*, 2015.

[16] D. Dittrich, "So You Want to Take Over a Botnet...," in *Presented as part of the 5th USENIX Workshop on Large-Scale Exploits and Emergent Threats*, 2012.

[17] W. Shang, Y. Yu, R. Droms and L. Zhang, "Challenges in IoT networking via TCP/IP architecture," *Technical Report NDN-0038. NDN Project,* 2016.

[18] A. Biason, C. Pielli, A. Zanella and M. Zorzi, "Access control for IoT nodes with energy and fidelity constraints," *IEEE Transactions on Wireless Communications,* vol. 17, pp. 3242-3257, 2018.

[19] J. Huang, L. Kong, G. Chen, M.-Y. Wu, X. Liu and P. Zeng, “Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism,” *IEEE Transactions on Industrial Informatics,* vol. 15, pp. 3680-3689, 2019.

[20] K. Christidis and M. Devetsikiotis, “Blockchains and smart contracts for the internet of things,” *Ieee Access,* vol. 4, pp. 2292-2303, 2016.

[21] A. Dorri, S. S. Kanhere, R. Jurdak and P. Gauravaram, “Blockchain for IoT security and privacy: The case study of a smart home,” in *2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*, 2017.

[22] A. Reyna, C. Martín, J. Chen, E. Soler and M. Díaz, “On blockchain and its integration with IoT. Challenges and opportunities,” *Future generation computer systems,* vol. 88, pp. 173-190, 2018.

[23] X. Li, P. Jiang, T. Chen, X. Luo and Q. Wen, “A survey on the security of blockchain systems,” *Future Generation Computer Systems,* vol. 107, pp. 841-853, 2020.

[24] R. A. Botha and J. H. P. Eloff, “Separation of duties for access control enforcement in workflow environments,” *IBM Systems Journal,* vol. 40, pp. 666-682, 2001.

[25] T. D. Chandra, V. Hadzilacos and S. Toueg, “The weakest failure detector for solving consensus,” *Journal of the ACM (JACM),* vol. 43, pp. 685-722, 1996.

[26] T. D. Chandra and S. Toueg, “Unreliable failure detectors for reliable distributed systems,” *Journal of the ACM (JACM),* vol. 43, pp. 225-267, 1996.

[27] M. J. Fischer, N. A. Lynch and M. S. Paterson, “Impossibility of distributed consensus with one faulty process.,” 1982.

[28] M. Na and B. Liskov, “Practical Byzantine fault tolerance and proactive recovery,” *ACM Transactions on Computer Systems (TOCS),* vol. 20, pp. 398-461, 2002.

[29] M. P. R. S. L. Lamport, “The Byzantine Generals Problem,” *ACM Transactions on Programming Languages and Systems Microsoft Research,* 1982.

[30] L. Lamport and others, “Paxos made simple,” *ACM Sigact News,* vol. 32, pp. 18-25, 2001.

[31] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.

[32] A. Dorri, S. S. Kanhere and R. Jurdak, “Blockchain in internet of things: challenges and solutions,” *arXiv preprint arXiv:1608.05187,* 2016.

[33] G. Deep, R. Mohana, A. Nayyar, P. Sanjeevikumar and E. Hossain, “Authentication protocol for cloud databases using blockchain mechanism,” *Sensors,* vol. 19, p. 4444, 2019.

[34] L. Luu, V. Narayanan, K. Baweja, C. Zheng, S. Gilbert and P. Saxena, “Scp: A computationally-scalable byzantine consensus protocol for blockchains,” *See https://www. weusecoins. com/assets/pdf/library/SCP,* vol. 20, p. 2016, 2015.

[35] D. Mazieres, “The stellar consensus protocol: A federated model for internet-level consensus,” *Stellar Development Foundation,* p. 32, 2015.

[36] A. Kosba, A. Miller, E. Shi, Z. Wen and C. Papamanthou, “Hawk: The blockchain model of cryptography and privacy-preserving smart contracts,” in *2016 IEEE symposium on security and privacy (SP)*, 2016.

[37] D. Schwartz, N. Youngs, A. Britto and others, “The Ripple Protocol Consensus Algorithm.(2014),” *URL: https://ripple. com/files/rippleconsensuswhitepaper. pdf,* 2014.

[38] M. Valenta and P. Sandner, “Comparison of Ethereum, Hyperledger Fabric and Corda,” 2017.

[39] 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*, 2016.

[40] A. Poelstra and others, “Distributed consensus from proof of stake is impossible,” *Self-published Paper*, 2014.

[41] A. Sangster and G. Scataglinibelghitar, “Luca Pacioli: the father of accounting education,” *Accounting Education: an international journal*, vol. 19, pp. 423-438, 2010.

[42] L. Pacioli, R. G. Brown and K. S. Johnston, Paciolo on accounting, Facsimiles-Garl, 1963.

[43] 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*, 2012.

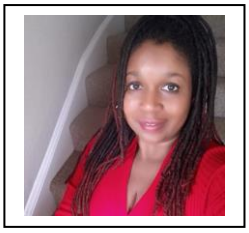
[44] E. Heilman, A. Kendler, A. Zohar and S. Goldberg, “Eclipse Attacks on Bitcoin's Peer-to-Peer Network.,” in *USENIX Security Symposium*, 2015.

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

[46] M. Jakobsson and A. Juels, Proofs of work and bread pudding protocols, Springer, 1999, pp. 258-272.

[47] D. E. Kouicem, A. Bouabdallah and H. Lakhlef, “Internet of things security: A top-down survey,” *Computer Networks*, vol. 141, pp. 199-221, 2018.

Author



Beverley A MacKenzie, is a final year Ph.D. student at Abertay University and an alumni of Royal Holloway University, Information Security Group. Her interests are cryptography, security standards and evaluation, access matrix, operating systems, blockchain security and the Internet of Things. She is a Security BSides organiser and a security evangelist. Her down time is spent with her 10 Iron-Age-Pigs.