

A Distributed Ledger with Trust-Based Data Aggregation for a Lightweight IoT Network

Tanmayee Deshprabhu, Justin P. Coon and Mihai-Alin Badiu

Department of Engineering Science, University of Oxford, OX1 3PJ, United Kingdom.

Email: justin.coon@eng.ox.ac.uk

Abstract—This paper addresses the need for a secure data management framework for lightweight Internet of Things (IoT) networks. Existing approaches rely on capable nodes or external cloud technology to carry out key functions for the lightweight IoT nodes, such as data processing, storage or routing. Instead, this paper considers a fully lightweight network or an existing network where the more capable nodes have failed. We propose a novel data management framework using a lightweight distributed ledger for IoT networks. This includes a trust-based data aggregation protocol in which data from untrustworthy nodes is not disregarded, but is instead utilised to strengthen the aggregation result. MATLAB simulations show that the proposed aggregation scheme infers data with high accuracy for both trustworthy and untrustworthy networks. Neutral networks exhibit a higher error rate but the maximum error rate decreases for larger networks.

I. INTRODUCTION

It is projected that 6G communication will rely heavily on IoT due to its flexibility in application, automation and scale [1]. However, being distributed, IoT networks can lack accountability due to the absence of a central management authority to oversee exchanges between nodes, making them vulnerable to malicious network attacks by dishonest nodes [2] or erroneous data sharing by unreliable nodes [3].

Distributed ledger (DL) technology such as blockchain (BC) [4] can be used to record all transactions in a network to maintain accountability [5]. However, in an IoT network with frequent communications or many nodes, this would result in large quantities of data being stored at each node. This poses a problem because IoT devices tend to be limited in terms of their technical capabilities [2]. There are other issues with using a DL for IoT including the high processing power and communication traffic required for DL management [6], but this paper focuses on secure storage.

Existing approaches to lightweight DL for IoT tend to limit the functions of lightweight nodes in the network, relying on the more capable nodes or a trusted third party such as a cloud platform to keep them informed. For example, Ethereum *light nodes* [7] cannot contribute to decisions, but participate in a BC network by storing only the block headers, which they receive from *full nodes* [8]. Similarly, the existing literature on lightweight DL for IoT systems relies on the assumption that there will always be intrinsically trustworthy and capable nodes available to act as the full nodes, but this is not necessarily the case in every network. A network can be fully lightweight, meaning that it consists of light nodes

only, whether that is by nature of the application or as a result of the network's full nodes failing. There is a need for an operative framework by which a fully lightweight network could function independently and securely, and this paper aims to propose a possible solution.

In our model, the nodes measure the reliability of any received data by estimating the trustworthiness of its source. Related aggregation protocols LDAT and RDAT [9] also use Bayesian trust inference in lightweight networks, but these are designed to reduce the network overhead on the channel rather than on the individual nodes. In a related distributed security framework, IOTA Tangle [10], the submitter of a transaction is required to first validate two older transactions. However, a malicious node may take advantage of this by validating transactions arbitrarily in order to participate in the network. Tangle also requires each transaction to be supported by a Proof of Work [4] result, which is too computationally complex to be completed by lightweight IoT nodes.

The contributions of this paper are as follows:

- 1) We propose a secure, scalable data storage and retrieval protocol for a distributed network that is fully lightweight.
- 2) We propose and analyze a trust-based data aggregation scheme, which aims to improve the reliability of information sharing in DL IoT systems. Unlike existing trust-based methods, the proposed method utilizes interactions from untrustworthy parties to reinforce the aggregation.
- 3) We demonstrate the performance of the trust-based aggregation scheme in the lightweight data management framework using MATLAB simulations.

II. SYSTEM DESCRIPTION

A. Network Structure

Consider a fully lightweight IoT wireless sensor network (WSN) in which a DL is maintained to keep a time-stamped and immutable record of sensor readings. There is one cluster head (CH) as chosen by the other nodes as the result of a periodic, trust-based Raft [11] consensus election. The CH has only two responsibilities: (i) packaging data into blocks (mining), a process that is checked by the remaining blocks, and (ii) broadcasting data requests from its followers.

The sensors collect some binary data from their environment, for example, motion sensors recording 1 when motion has been detected since the last update and 0 otherwise.

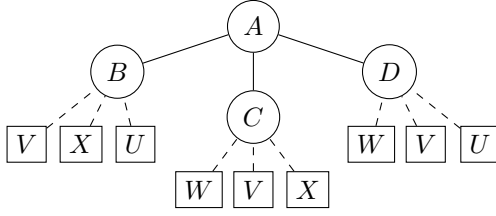


Fig. 1. Cluster of nodes after a Raft election, showing that A is chosen to take on the miner role. The solid links show the flow of data while the dashed links show data storage. This is also a small-scale illustration of how the random distribution of blocks U , V , W and X means that each node only has to store 75% of the total data, whilst each block is still stored at multiple locations.

These nodes send their readings to the CH but retain a copy temporarily. The CH ceases its sensing for the period of its leadership, and instead mines the other nodes' readings into fixed-size blocks, which are then distributed randomly. The contributing sensor nodes check for the inclusion of original reading within the mined block using a Bloom filter [12]. These nodes then update their respective trust value for the CH by recording the mining interaction as a positive or negative observation of the CH. Periodically, a Raft election is called to elect a new CH (miner) based on updated trust values.

B. Novel Lightweight Storage Scheme

Unlike typical BC implementations, the light nodes in our proposed model store whole blocks but only store a *random subset* of the many blocks from the network ledger. An example of the lightweight distributed storage is illustrated in Fig. 1, where each node uses 25% less storage space for blocks than if it were to store the full network ledger. The allocation of blocks to nodes is done on a random basis to reduce the probability of malicious collusion. Whenever a node needs to access data that it doesn't store locally, it requests this data from the network using the proposed data retrieval protocol in Section IV. Any nodes storing that particular block will send a copy to the requester, who then aggregates the many versions of the data together based on its trust values for the respective senders.

III. TRUST

A. Trust as Data Validation

In contrast with monetary ledgers such as BitCoin [4], previous data in IoT networks cannot always be used to accurately validate new contributions. This is particularly true in applications involving sensor readings because the measurand can vary independently of past data. We instead use trust inference to assess the validity of new submissions to the network ledger, given that data from a highly trustworthy source is more likely to be valid. With reference to Fig. 2, trust is defined as the probability τ that D will behave as C expects it to. This is based on a set of positive r and negative w observations that C holds about D from previous interactions.

The trust information about each node is utilised in the data aggregation scheme in Section IV, with input from both

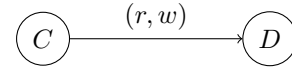


Fig. 2. Node C has r positive and w negative observations about target D .

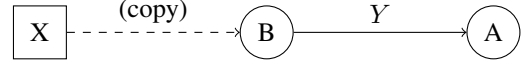


Fig. 3. Node B retrieves original data X and reports it as Y to node A .

trustworthy and untrustworthy nodes being used to strengthen the aggregation result. Secondly, in the CH elections, each node votes for the candidate for which it holds the highest trust rating to ensure that the most honest and least error-prone node is chosen. Although periodically electing a CH appears to centralise the network cluster, it is instead distributing the network decisions by allowing for the currently most trusted miner to be chosen by all of the participating nodes. This is unlike existing lightweight implementations that arbitrarily grant trust to full nodes or external cloud computing. Nodes record new observations whenever they interact with other nodes. For example, after checking a mined block or after data aggregation. The observations are then converted into a numerical representation of trust using a trust inference model.

B. Beta Reputation Model

We base our trust calculation on the Beta Reputation Model (BRM) [3]. However, the novelty of the proposed aggregation scheme in Section IV comes from exploiting the output of trust inference rather than from generating the trust inference results, thus the aggregation can be applied in conjunction with results from *any* trust inference model.

Consider two nodes, A and B in Fig. 3, in a WSN. A requests data block X from the network via the CH. B locally stores X , so it sends a copy of X to A in the form of a report Y . Ideally, the report accurately reflects the original data $Y = X$, where $X \in \{0, 1\}$. Depending on its intention, B may have lied by inverting the data ($Y \neq X$). The probability that B will tell the truth is represented by its trust value. According to the BRM in [3], τ 's probability density function can be written as

$$f(\tau | r, w) = \frac{\tau^r (1 - \tau)^w}{B(r + 1, w + 1)}, \quad \text{for all } \tau \in [0, 1] \quad (1)$$

where $B(a, b)$ represents the beta function. However, it is the nature of this model that one cannot know exactly the value of τ [3]. Instead, τ 's probability density function is used to find the expected value of trust $\hat{\tau}$.

C. Effective Trust

For the purpose of data aggregation, we extend the BRM to account for the channel error rate e , which is the probability that data is inverted during transmission from B to A due to a poor communication channel, regardless of B 's intention. For reference, $e = 0.05$ in the BitCoin BC [4].

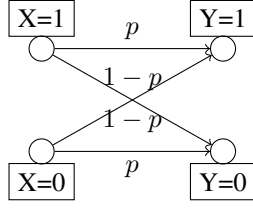


Fig. 4. Binary symmetric channel for the report Y about data X from any single source node.

As a result of e , any observations made by nodes in the network are not directly indicative of the sender's trust τ but rather of its *effective trustworthiness*, which accounts for both τ and e . We define effective trust as $p = \mathbb{P}(Y = X)$, which is given by

$$p = \tau(1 - e) + e(1 - \tau). \quad (2)$$

and, like τ , has a beta distribution meaning that we cannot know exactly the value of p but can estimate it by using observations,

$$\hat{p} = \frac{r + 1}{r + w + 2}. \quad (3)$$

The relationship between the original data X and a report about the data Y can then be represented as the binary symmetric channel (BSC) in Fig. 4. Note that $\hat{p} = 0.5$ should be used for newly discovered nodes for which there are no prior observations ($r = w = 0$) in order to ensure that their effect on the data aggregation is minimal, given that data from both effectively trustworthy ($\hat{p} > 0.5$) and untrustworthy ($\hat{p} < 0.5$) source nodes is used to reinforce the result.

IV. DATA MANAGEMENT

A. Data Retrieval

In the lightweight distributed storage proposed in Section II, nodes store a subset of the blocks from the network ledger. This aims to reduce the storage overhead on each node, but if a node requires some data that it does not locally store, then it may need to retrieve this data by requesting the associated block from other nodes in the network.

Assume in Fig. 1 that a node D requires a particular data point in block X . For simplicity, let us assume that X contains a single-bit, binary data point of interest. D does not store this block locally and so it needs to retrieve copies of it from the source nodes, which are nodes that store the desired block X . D sends a request to the CH, A , where the request contains the identifier of the desired block X . A then broadcasts a request for the block along with the identifier of the requester. B and C , acting as source nodes, then respond to D with their versions of the desired data in the form of reports Y_B and Y_C . Finally, D uses its values for \hat{p}_B and \hat{p}_C to aggregate the two reports into a *best guess* \hat{X} for the true value of X , i.e. trust-based data aggregation.

B. Trust-Based Data Aggregation Scheme

We propose here a method by which the requesting node can aggregate all of the data reports by finding the most probable value of X based on the Y and τ values of each source node. Let X be a single, binary data point of interest. Some prior information is known about the data, $s = \mathbb{P}(X = 1)$. It is possible to find the marginal probability of the report $\mathbb{P}(Y)$ by applying the law of total probability to the BSC in Fig. 4. From this, the likelihood $\mathbb{P}(Y|X)$ can be calculated using Bayes' theorem. The system in Fig. 3 can then be expanded to include multiple source nodes $n > 1$, which each store local versions of X . A makes a request for the data X and receives n versions of this data in the form of reports $Y = [Y_1, Y_2, \dots, Y_n]$. This problem can be represented as the maximum a posteriori (MAP) estimation shown in (4) where $x \in \{0, 1\}$.

$$\hat{X} = \arg \max_x \mathbb{P}(X = x|Y) \quad (4)$$

Re-writing this with the assumption that each node behaves independently,

$$\mathbb{P}(X|Y_1, Y_2, \dots, Y_n) = \frac{\mathbb{P}(X) \prod_{i=1}^n \mathbb{P}(Y_i|X)}{P(Y)}. \quad (5)$$

C. Aggregation Error Rate

The performance of the proposed data aggregation scheme can be measured by observing the error rate E , which is the probability that the aggregation result is incorrect. If E_1 and E_0 are the two cases of error, $\mathbb{P}(\hat{X} \neq 1|X = 1)$ and $\mathbb{P}(\hat{X} \neq 0|X = 0)$ respectively, then

$$E = \mathbb{P}(\hat{X} \neq X) = E_0(1 - s) + E_1 s. \quad (6)$$

Proposition 1. *Let there be a constant k such that*

$$k = \frac{n}{2} + \frac{1}{2} \frac{\ln \frac{1-s}{s}}{\ln \frac{1-\hat{p}}{\hat{p}}}, \quad (7)$$

then the conditional probabilities of error E_0 and E_1 can be calculated for any n using (8) and (9).

$$E_0 = \sum_{i=0}^{\lfloor k \rfloor} \binom{n}{i} (1-p)^{n-i} p^i, \quad (8)$$

$$E_1 = \sum_{i=\lceil k \rceil}^n \binom{n}{i} (1-p)^i p^{n-i}. \quad (9)$$

Proof. See the Appendix for full calculations. \square

As noted in the Appendix, E depends on the total number of zeros reported, $Z = \sum_{i=1}^n \mathbf{1}_{\{0\}}(Y_i)$, where Z is a binomial random variable and $\mathbf{1}_{\{0\}}(x) = 1$ for $x = 0$. Therefore, we can estimate the probability density function of Z by using a Gaussian approximation of the binomial distribution.

As this is a lightweight scheme, the presumption is that the trustworthiness of nodes will be quite high in some cases, for example, in private IoT networks where trust is used to measure the reliability of nodes. We can observe this aggregation for such networks by approximating its performance for

p near to 1. In order to do this, (8) and (9) can be expanded and approximated as

$$E_0 = \binom{n}{\lfloor k \rfloor} (1-p)^{n-\lfloor k \rfloor} + \mathcal{O}\left((1-p)^{n-\lfloor k \rfloor+1}\right), \quad (10)$$

$$E_1 = \binom{n}{\lceil k \rceil} (1-p)^{\lceil k \rceil} + \mathcal{O}\left((1-p)^{\lceil k \rceil+1}\right). \quad (11)$$

Having considered small networks, we next analyse E for large n , as would be the case in a large sensor network or public database. This allows us to assess the scalability of the proposed scheme.

Proposition 2. *Let k be defined as in Proposition 1, then for large n the conditional probabilities of error, E_0 and E_1 , are well approximated by (12) and (13) respectively:*

$$E_0 \approx \frac{1}{2} \left[\operatorname{erf} \left(\frac{n-np}{\sigma\sqrt{2}} \right) - \operatorname{erf} \left(\frac{\lfloor k \rfloor - np}{\sigma\sqrt{2}} \right) \right], \quad (12)$$

$$E_1 \approx \frac{1}{2} \left[1 - \operatorname{erf} \left(\frac{\lfloor k \rfloor - n(n-p)}{\sigma\sqrt{2}} \right) \right]. \quad (13)$$

Proof. Z can be approximated using the Gaussian distribution $Z \sim \mathcal{N}(\mu, \sigma)$, where

$$\mu = \begin{cases} n(1-p), & \text{if } X = 1 \\ np, & \text{if } X = 0 \end{cases} \quad (14)$$

$$\sigma^2 = np(1-p). \quad (15)$$

We then find the cumulative distribution function of Z for large networks by applying the Central Limit Theorem. \square

Based on the above, we can then consider scenarios with extremely large n by analysing the error functions in (12) and (13) for $n \rightarrow \infty$.

$$E_0 \approx \frac{1}{2} \left[\left(\frac{e^{-z_2^2}}{z_2\sqrt{\pi}} \right) - \left(\frac{e^{-z_1^2}}{z_1\sqrt{\pi}} \right) \right] \quad (16)$$

$$E_1 = \frac{e^{-z_3^2}}{2z_3\sqrt{\pi}} \quad (17)$$

Where $z_1 = \frac{n-np}{\sigma\sqrt{2}}$, $z_2 = \frac{\lfloor k \rfloor - np}{\sigma\sqrt{2}}$ and $z_3 = \frac{\lfloor k \rfloor - n(n-p)}{\sigma\sqrt{2}}$. Given that $\lim_{n \rightarrow \infty} (\lfloor k \rfloor, \lceil k \rceil) \simeq \frac{n}{2}$, and letting $a = \frac{\frac{1}{2}-p}{\sqrt{2p(1-p)}}$, we can approximate both types of error using

$$E_0, E_1 \simeq \frac{e^{-a^2 n}}{2a\sqrt{n\pi}}. \quad (18)$$

The above is only true for homogeneous trust, i.e. where every source node participating in the network has the same value of expected trustworthiness. Note that this does not necessarily mean that they are all equally trustworthy, but that they have the same probability of being trustworthy.

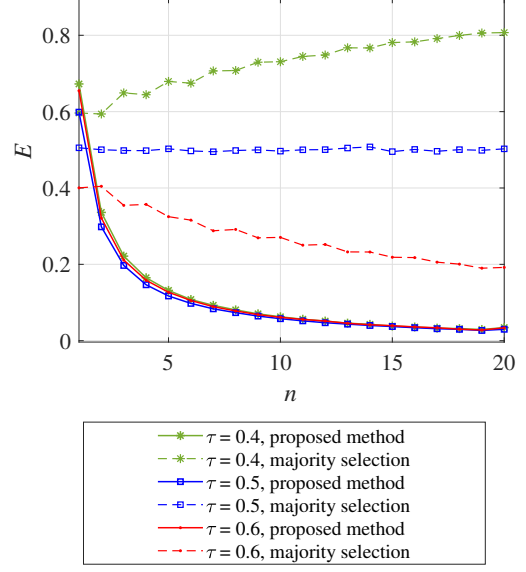


Fig. 5. Error rate E vs. source nodes n for the proposed method and the majority selection method for various values of node trustworthiness τ .

V. NUMERICAL RESULTS

A. Simulation Methodology

A MATLAB Monte Carlo simulation is used to observe the aggregation error rate E . We define the prior distribution of data $s = 0.5$ and the channel error probability $e = 0.05$.

In the first simulation, we plot E against the number of source nodes n for different p values using (8) and (9). For comparison, an alternative, commonly used method called majority selection [13] has also been simulated, which chooses \hat{X} using the statistical mode of many binary reports Y . In subsequent simulations, we randomly generate the data X and each node's corresponding report Y based on the node's respective p value. It is assumed that the receiver holds a \hat{p} value for each source node which is, for the simulation, equal to p . Using (22) and (23), we can then perform the MAP estimation and find the aggregation error rate E by comparing the MAP result with the original data.

We simulate a homogeneous network in which all nodes have the same expected trustworthiness p , but this does not mean that the nodes are all equally trustworthy. The trust probabilities of the nodes in a use-case scenario can be generated using any trust inference model [14].

B. Results

Fig. 5 shows a comparison between using the proposed trust-based aggregation method and choosing the result based on a majority vote. The proposed method demonstrates significantly lower error rate for all $n \geq 2$ showing that, even for a small number of source nodes, integrating trust inference greatly improves the aggregation reliability.

Fig. 6 shows the relationship between the aggregation error rate E and the nodes' effective trustworthiness p . The

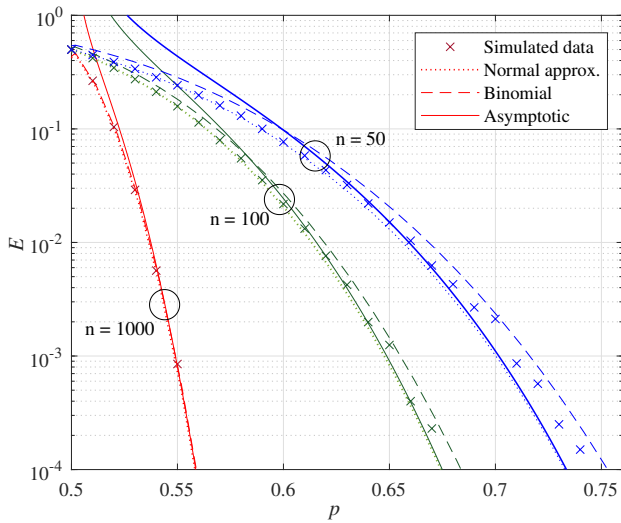


Fig. 6. Comparison between the simulation, binomial calculation, normal approximation and asymptotic analysis of E for different scales of n .

error rate decreases as the number of participating source nodes n increases. The simulation agrees with the binomial calculation of E and the normal approximation provides a close representation of the curve. The graph also shows that the asymptotic analysis is accurate to the simulation, and thus provides a reliable method by which to observe the system for large n , which represents extremely large networks.

As the aggregation utilises input from both trustworthy and untrustworthy nodes, the results are symmetrical about $p = 0.5$. However, we focus here on $0.5 < p < 1$ because this would be a more suitable range of trust for a network to operate in. E peaks at $p = 0.5$ because this represents a point where the trust information is not useful, such as in new networks. The maximum E corresponds to $\min\{s, 1-s\}$ because, when there is insufficient data about the trust and channel (in the form of p), then the aggregator chooses the most likely value of X based on the prior information.

Fig. 7 shows that the proposed scheme exhibits low error rates even for very small networks in the range $n \leq 10$, and the approximation around p close to 1 follows the simulation results closely.

VI. DISCUSSION ON IMPLEMENTATION

A. Multi-Bit Binary Data

A wide range of scientific data can be stored in single-bit binary format. For multiple-bit data, the aggregation scheme may be applied to each bit in turn. However, in order to directly apply the trust-based aggregation scheme to multi-bit data, one would need to consider the specific actions of malicious nodes. For example, a malicious node could invert all bits in the data, send old data or simply generate and submit a random number. Multi-bit binary and non-binary data formats are beyond the scope of this paper but form part of the future work.

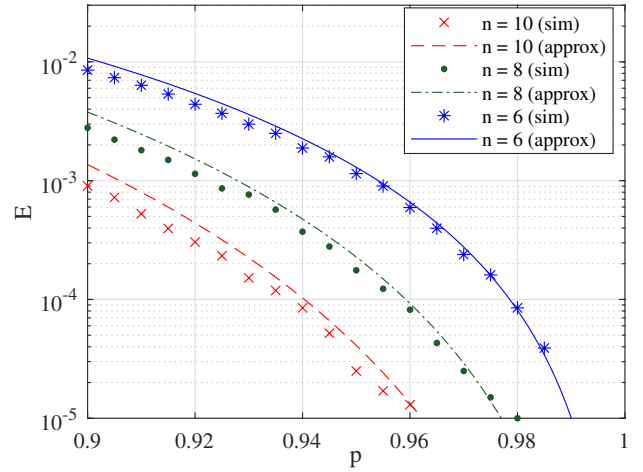


Fig. 7. Simulation and approximation of aggregation error rate in small networks for p close to 1.

B. Use Case

This paper focuses on a fully lightweight network where there are no trustworthy, highly capable nodes present to take on storage, processing and computational overheads. For example, a WSN deployed in a very remote region, or community networks in rural areas. Secondly, IoT devices have intermittent connectivity as a result of low battery life, poor communication channel or malicious attacks. These limitations mean that, even in a network that originally had access to more capable full nodes, the network may lose access to these for periods of time. Were this to happen, the network would need to function temporarily as a fully lightweight IoT network, perhaps using our proposed lightweight distributed ledger and data aggregation schemes to continue functioning until the more capable nodes are restored. As a result of the ledger, there would still be a complete record of all interactions during the network down time and, as a result of our proposed scheme, those interim interactions can be carried out within a secure, lightweight data management framework. Surprisingly, little work has been done to address this scenario and the closest protocol is the naive majority selection scheme [13] used in Section V.

C. Application to Blockchain Networks

It may be possible to apply the proposed lightweight DL scheme directly to BC networks to reduce the data stored at each node, but this would require an additional protocol for the division of the BC encryption protocols between multiple nodes. Such protocols have been discussed in literature [15]. Let the total nodes in the network be represented by N . In most BC implementations, each of the N nodes stores every block from the network ledger. The percentage reduction in storage requirement per node due to our proposed approach would be proportional to $1 - n_{min}/N$ where n_{min} is the minimum number of source nodes required to achieve some maximum allowable error rate E_{max} .

VII. CONCLUSION

In this paper, a trust-based data aggregation scheme has been proposed as part of a novel distributed ledger-based data management scheme for fully lightweight IoT networks. Simulation results for both small and large networks show that the proposed method successfully utilizes reports from untrustworthy nodes to reinforce the aggregate data and reduce the error rate. In the majority of results, the aggregation error peaks for networks around $p = 0.5$, meaning that the error rate would only be high in relatively young networks that have very few observations available about the nodes, or in networks that are extremely mixed such that the average effective trustworthiness of nodes across the network is around $\hat{p} = 0.5$. The future work includes investigating the effect of collusion-based attacks on the proposed model and extending the aggregation scheme to a non-binary data WSN.

APPENDIX

The aggregation result is defined for a single report using a MAP calculation,

$$\begin{aligned}\hat{X} &= \arg \max_x \frac{\mathbb{P}(X = x, Y)}{P(Y)} \\ &= \arg \max_x \left[\frac{\mathbb{P}(X = x) \mathbb{P}(Y|X = x)}{\mathbb{P}(Y)} \right].\end{aligned}\quad (19)$$

Expanding for n source nodes gives

$$\hat{X} = \arg \max_x \left[\frac{\mathbb{P}(X = x) \prod_{i=1}^n \mathbb{P}(Y_i|X = x)}{\mathbb{P}(Y)} \right].\quad (20)$$

The error case E_0 is defined as $\mathbb{P}(\hat{X} \neq 0 | X = 0)$, we can rewrite the MAP decision in terms of \hat{X} ,

$$\begin{aligned}E_0 &= \mathbb{P} \left[\frac{\mathbb{P}(X = 1) \prod_{i=1}^n \mathbb{P}(Y_i|X = 1)}{\mathbb{P}(Y)} \right. \\ &\quad \left. > \frac{\mathbb{P}(X = 0) \prod_{i=1}^n \mathbb{P}(Y_i|X = 0)}{\mathbb{P}(Y)} \mid X = 0 \right].\end{aligned}\quad (21)$$

The aggregation error rate E is a function of the total number of reports that equal zero Z . Substituting in $s = \mathbb{P}(X = 1)$, $p = \mathbb{P}(Y_i = X)$, and $Z = \sum_{i=1}^n \mathbf{1}_{\{0\}}(Y_i)$,

$$\begin{aligned}E_0 &= \mathbb{P} \left[(1 - \hat{p})^Z \hat{p}^{n-Z} s \right. \\ &\quad \left. > \hat{p}^Z (1 - \hat{p})^{n-Z} (1 - s) \mid X = 0 \right] \\ &= \mathbb{P} \left[(2Z - n) \ln(1 - \hat{p}) + (n - 2Z) \ln(\hat{p}) \right. \\ &\quad \left. > \ln(1 - s) - \ln(s) \mid X = 0 \right] \\ &= \mathbb{P} \left[Z < \frac{n}{2} + \frac{1}{2} \ln \frac{1-s}{\hat{p}} \mid X = 0 \right].\end{aligned}\quad (22)$$

Note the sign change based on the assumption that $0.5 < p < 1$, because $p < 0.5$ would not be a suitable operating range

for a network. Following similar steps for the remaining error case $E_1 = \mathbb{P}(\hat{X} \neq 1 | X = 1)$,

$$\begin{aligned}E_1 &= \mathbb{P} \left[\frac{\mathbb{P}(X = 0) \prod_{i=1}^n \mathbb{P}(Y_i|X = 0)}{\mathbb{P}(Y)} \right. \\ &\quad \left. > \frac{\mathbb{P}(X = 1) \prod_{i=1}^n \mathbb{P}(Y_i|X = 1)}{\mathbb{P}(Y)} \mid X = 1 \right] \\ &= \mathbb{P} \left[Z > \frac{n}{2} + \frac{1}{2} \ln \frac{1-s}{\hat{p}} \mid X = 1 \right].\end{aligned}\quad (23)$$

Note that Z is a sum of Bernoulli random variables $Y_{1..n}$, with a distribution based on p and s . As a result, Z becomes a binomial random variable $Z \sim \mathcal{B}(n, p)$ for $X = 0$ and $Z \sim \mathcal{B}(n, 1 - p)$ for $X = 1$, then we arrive at (8) and (9).

ACKNOWLEDGMENT

The authors wish to acknowledge the support of the Bristol Research & Innovation Laboratory of Toshiba Europe Ltd.

REFERENCES

- [1] L. Zhang, Y. Liang and D. Niyato, "6G Visions: Mobile ultra-broadband, super internet-of-things, and artificial intelligence," in *China Comms*, vol. 16, no. 8, pp. 1-14, 2019.
- [2] M. A. Khan, K. Salah, "IoT Security: Review, Blockchain Solutions and Open Challenges," in *Future Generation Computation Systems*, vol. 82, 2018, pp. 395-411.
- [3] A. Josang, R. Ismail, "The Beta Reputation System," in *15th Bled Electronic Commerce*, 2002.
- [4] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," whitepaper, 2009.
- [5] V. A. Red, "Practical comparison of distributed ledger technologies for IoT," SPIE, Volume 10206, 2017.
- [6] A. Reyna, C. Martin, J. Chen, E. Soler, M. Diaz, "On blockchain and its integration with IoT. Challenges and opportunities", *Future Generation Computer Systems*, vol. 88, 2018, pp. 173-190.
- [7] "Running an Ethereum Node - EthHub." [online] Available at: <https://docs.ethhub.io/using-ethereum/running-an-ethereum-node/> [Accessed 21 December 2020]
- [8] D. Gruber, W. Li, G. Karame, "Unifying Lightweight Blockchain Client Implementations," 2018.
- [9] M. Kumar, K. Dutta, "LDAT: LFTM based data aggregation and transmission protocol for wireless sensor networks," *Journal of Trust Management*, vol. 3, 2016.
- [10] S. Popov, "The Tangle," whitepaper, April 2018, version 1.4.3.
- [11] D. Ongaro, J. Ousterhout, "In Search of an Understandable Consensus Algorithm", 2014 USENIX Annual Technical Conference 2014.
- [12] K. Kanemura, K. Toyoda, T. Ohtsuki, "Design of privacy-preserving mobile Bitcoin client based on γ -deniability enabled bloom filter," 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017, pp. 1-6.
- [13] J. Rinne, "Majority selection and block-based selection diversity reception methods for 8k DVB-T in a mobile environment," 10th European Signal Processing Conference, 2000, pp. 1-4.
- [14] Z. Yan, P. Zhang, A. V. Vasilakos, "A survey on trust management of Internet of Things", *Journal of Network and Computer Applications*, vol. 42, 2014, pp. 120-134.
- [15] A. Singh et al, "Public Blockchains Scalability: An Examination of Sharding and Segregated Witness," in *Advances in Information Security*, vol. 79, 2020.