Energy Level Spoofing Attacks and Countermeasures in Blockchain-enabled IoT

Ali Hussain Khan, Humza Ikram, Chuadhry Mujeeb Ahmed, Naveed Ul Hassan, and Zartash Afzal Uzmi

***Abstract*—The Internet of Things (IoT) ecosystem is witnessing widespread deployments for emerging applications in diverse domains such as remote sensing, smart homes, and industry 4.0. There is also a growing need to secure such deployments against malicious IoT devices to sustain normal network operations. Since the IoT deployments encompass geographically distributed nodes, blockchain technology, which inherently offers distributed trust in such scenarios, is gaining popularity in providing a secure and trusted IoT deployment. In this paper, we present a use case in which an IoT deployment is retrofitted with a blockchain. The use of blockchain prevents malicious nodes from falsifying information about their energy levels. We first present attack scenarios where IoT nodes can spoof energy while joining or being a part of the network. We then build a defense strategy and evaluate its performance under various attack scenarios. Our results indicate that the IoT deployment is robust under the proposed defense strategy which can detect if a node is spoofing its energy levels over 75% of the time.**

I. Introduction

With rapid development in communication and computation technologies, the world is becoming increasingly connected. Internet of Things (IoT) is a paradigm where a massive number of low-power and resource constrained devices are deployed for sensing, data collection, data sharing and other applications. This is a technology where devices communicate with each other without human intervention. The current scale of IoT devices is huge. According to [1], the number of IoT devices in 2020 was expected to be more than 20 billion increasing at an exponential rate. IoT devices have distributed nature and rely on cloud infrastructure due to the limited resources on endpoints. Cloud is essentially a centralized entity that has a conflict with the distributed nature of IoT device deployment. Cloud-based architectures introduce congestion, delay and security issues [2]. Sensory data dominates the IoT ecosystem and it tends to be extremely sensitive and critical, therefore, security and privacy are important. Centralized security architectures lack scalability which is a big concern in the context of IoT [3].

Recently, blockchain has emerged as a solution to the privacy, security and scalability concerns in many application domains. Blockchain is a distributed ledger technology that utilizes cryptography, hash algorithms and consensus to form an immutable chain of blocks. Data is encrypted using cryptography, and the blocks are linked via hashes. A new block is added to the chain after it has achieved consensus in the system. Legacy consensus mechanisms are Proof-of-Work (PoW), Proof-of-Stake (PoS), delegated Proof-of-Stake (dPoS) and Practical Byzantine Fault Tolerance (PBFT). Due to its distributed nature, it has been utilized in applications other than cryptocurrency which was the intended use case of blockchain. It has been recently utilized in smart grid [4], next generation cellular networks [5], vehicular networks [6], etc.

Due to its distributed nature and decentralization, blockchain is a very good candidate for being used in IoT. Due to its inherent features, it also provides privacy and security of data. Blockchain itself is very resource consuming. Deploying blockchain on IoT will have issues regarding the delay, bandwidth requirements, and limitation of computation resources. Hierarchical architecture solves this problem by utilizing network infrastructure for aiding in blockchain deployment. Recently, [7] have considered a blockchain-enabled internet of underwater things model, where they have considered the notion of cluster head (CH) and cluster members (CMs). The inter-cluster communication is supported by CH. However, a constraint in this work is that they are utilizing cloud servers for blockchain which introduces the distance and latency constraint. Similarly, in [8], the authors present an IoT framework where they are performing lightweight encryption of IoT nodes. IoT nodes are distributed into a cluster and each cluster had CHs and CMs and CHs act as miners as well as blockchain nodes.

The nature of IoT devices is heterogeneous and energy is an integral requirement. Different types of IoT devices have expected longevity requirements from a couple of days to several years. Therefore, the role of energy in IoT devices is extremely important. Due to the massive scale of IoT devices, there is a huge cost associated with the maintenance of device batteries [9]. Energy harvesting has been deemed as an attractive solution to improve the longevity of IoT devices. Therefore, energy harvesting solutions should also be implemented to increase the battery life of IoT devices.

In this work, we consider a clustered dynamic IoT network to facilitate the IoT network with routing and consensus management. There are multiple clusters that are connected with Base Stations (BSes). Each cluster has one CH and multiple CMs. The CH is a device that has to maintain a high level of energy in its cluster and is responsible for routing the data from the CMs to the BS with which it is connected. The BSes are interconnected and they achieve consensus on the

A. H. Khan, H. Ikram, N. U. Hassan and Z. A. Uzmi are with the Department of Electrical Engineering, Lahore University of Management Sciences (LUMS), 54792 - Lahore, Pakistan. (Emails: 18060048@lums.edu.pk, 20100031@lums.edu.pk, naveed.hassan@lums.edu.pk, zartash@lums.edu.pk).

C. M. Ahmed is with the department of Computer And Information Sciences, University of Strathclyde, Glasgow. (Email: mujeeb.ahmed@strath.ac.uk).

block and the block is added to the blockchain which is also held at the BS.

We consider an attack where a joining node spoofs its energy level to become the cluster head. A malicious node can spoof and become cluster head and then withhold data or run out of the battery to deny blockchain services to the system. Doing so will also bypass the incentive structure of the blockchain and claim incentives disproportionate to its energy value. It could spoof its energy while joining the network to have a chance of becoming the cluster head right away. It could also join the network and spoof the energy value while being a part of the network.

We present a detection algorithm where the spoofing node spoofs its capabilities while running the system. We can detect spoofing by keeping track of all the communication to and from this node by reports from all the other cluster members. We also keep track of the energy harvested by different nodes in the system. We also present deterministic and random energy harvesting mechanisms. We use all this information in the simulation and detect the spoofed energy values of a malicious node.

Rest of the paper is organized as follows: In Section II, we present the system model and blockchain integration in the IoT scenario, in Section III, we present the threat model and attack scenarios that we will be considering, in Section IV, we present the detection algorithm for the energy levels spoofing, in Section V, we give the simulation setup and results of the detection algorithm and in Section VI, we conclude the paper.

II. System Model and Blockchain Integration

In this section, we present the integration of a blockchain in an IoT deployment:

A. System Model

We consider an IoT network which has device to device (D2D) connections among each other. The network comprises of a trusted authority (TA), large number of IoT devices and BSes or sink nodes. We assume that secure and timely data sharing is the most important requirement in this network for coordinated/cooperative decision making. For ensuring IoT data security, we are assuming the use of blockchain and BSes will take part in consensus mechanism as well as act as blockchain nodes. As we want timely consensus, we assume that a non-PoW type consensus mechanism such as delegated Proof-of-Stake (dPoS) or a variant of PBFT is used in the network. We assume energy-constrained IoT devices which form clusters based on their deployed positions. To summarize, we have the following nodes in the network:

TA: The TA is responsible for registering the IoT devices and BSes in the system. IoT devices and BSes submit their registration information and get their public/private key pairs and the digital certificates.

BSes: BSes are the backbone network providers in the system. They are the consensus as well as blockchain storage nodes owing to the high on-board computational as well as storage capabilities.

IoT Devices: IoT devices are the users of the blockchain service. IoT devices are deployed in clusters based on their position in the system. Each cluster has one CH and the rest are CMs. CMs are relatively low energy devices, which utilize the CHs, which are relatively high energy devices to relay their data to BSes for consensus and adding to the blockchain. CMs also report if they shared data with CHs in a particular cycle to audit the energy level of the CH. IoT devices are provided with apparatus for energy harvesting which will be the only source of their energy, which means that a device will die if it does not harvest the intended energy to stay alive.

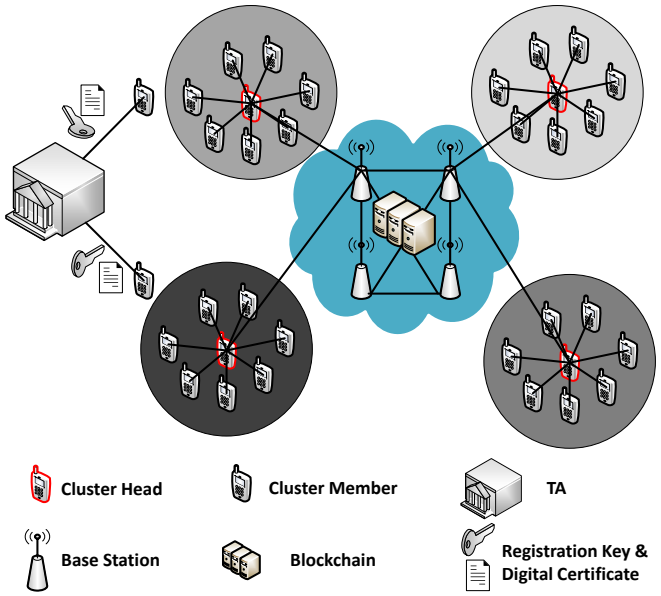

Fig. 1. System Model of Blockchain-enabled IoT System

B. Blockchain Integration

Here, we describe a particular application scenario of blockchain-enabled IoT network. It goes as follows:

Step 1: First, all the joining nodes get authenticated by the TA. The blockchain is a consortium blockchain, where the BSes act as blockchain nodes, which perform consensus as well as storage of blockchain based on their enhanced communication, computation and storage capabilities. The IoT nodes, along with their identity information, also share their battery level, which is a very important metric for allocating CHs and CMs.

Step 2: The TA gets all the identity information from the IoT nodes and estimates their current battery level using physical methods. Then it compares the reported as well as the measured battery level. If there is small difference between the two, the battery level is deemed correct and the node is authenticated. If the battery level is deemed incorrect, the node is not authenticated and not permitted to join the blockchain network. All the authenticated nodes will get their public-private key pairs from the TA.

Step 3: All the IoT nodes form clusters. Multiple clusters are connected to one BS. And multiple BSes form the basis of the consensus in the blockchain network. Based on the reported battery levels, one node is selected as CH for one

cycle while the others are selected as CMs. The probability of being selected as a CH depends on the battery level of the nodes. At the beginning of each cycle each IoT node shares channel state information (CSI), distance with the CH as well as the energy harvested during the last cycle. During each cycle, CMs also share their sensing data with the CH of that cycle. The CH transmits this data to its respective BS for consensus. There is a time limit within which this data is to be transmitted from CMs to CH and from CH to BS. But due to varying CSI, the transmitter node cannot transmit with fixed value of transmit power. Therefore, it transmits at varying powers and thus it can spoof its communication energy.

Step 4: BSes form consensus on the data shared with them. This consensus is based on dPoS or PBFT since these mechanisms are communication intensive and less time taking. This verified data is available for all the blockchain users to audit. All the IoT nodes also share their updated battery levels after each cycle for consensus and it is uploaded to the blockchain.

Analogy with Leader/Follower Systems: This application scenario bears a lot of resemblance with leader/follower systems that are used widely in the communication systems as well as blockchains. Leader-based dPoS and PBFT systems are examples of such blockchain systems. These schemes are analogous to the cluster based distribution in IoT networks or other networks in general. In leader/follower based communication systems, there is a criteria of leader selection. In cluster-based networks, there is also a criteria of CH selection. In leader/follower based blockchain systems, leaders have different incentive structure than the followers. In cluster-based systems, we suggest a different incentive structure for CH than those of the CMs. The motivations of employing a leader/follower system in a communication network is similar to the motivation of employing CHs in cluster-based systems. Leaders conduct whatever relevant activity of interest in the communication network which in the case of dPoS and PBFT is leading the consensus mechanism and CHs are responsible for conducting activity within cluster e.g., data aggregation and data relay. Since our application scenario has similar distribution of roles and is analogous to leader/follower based systems, there is a natural inclination of using such consensus algorithms in an architecture like this.

III. Threat Model and Attack Scenarios

In this section, we describe the general threat model in blockchain-enabled IoT application scenario context and present the energy level spoofing attack scenarios in particular

A. Threat Model

In a blockchain-enabled IoT scenario, we have base stations as well as IoT nodes. Both of them could turn malicious.

Malicious BSes: Malicious BSes could add incorrect data to the blocks. They can also collude with other BSes to get that data verified and added to the blockchain. Especially, they could modify data on energy levels to be added to the blockchain. In this way, they can control which nodes become CHs for the upcoming cycles.

Malicious IoT Nodes: Malicious IoT nodes could report spoofed values of their battery levels. They could also collude with other IoT nodes in the network to report spoofed interactions that support enhanced battery levels. They could also collude with other BSes to support their battery-level reporting.

B. Energy Level Spoofing Scenarios

There have been attacks presented in recent literature [10] where an adversary takes advantage of the inherent vulnerabilities of energy constrained devices and spoofs the current energy level of that device. We observe that a malicious adversary can make use of such an attack to spoof the energy capabilities of itself, to launch a DoS attack on the blockchain network. In this work, we look at attacks where adversaries report upgraded energy levels while joining or being a part of the network. We call these attacks Energy Upgrade while joining (EnU-Join) and Energy Upgrade while running (EnU-Run) the blockchain network. Next, we look at these attack scenarios in some detail.

1) EnU-Join: In this attack scenario, an IoT node can report an upgraded battery level while joining the network. Along with that, it also overestimates its capability of energy harvesting. The significance of doing this is that if it is authenticated as a high energy node in the network, it will be utilized as a CH. Becoming a CH in the context of IoT nodes will have multiple advantages. Firstly, it can act maliciously by not relaying the data from CMs to the BSes for consensus This type of attack is studied in the blockchain literature called censorship attack [11]. Here, censorship attacks are defined as a majority coalition building a chain which rejects transactions or messages that an ordinary validator, miner, or client would accept. Even if it does not withhold data maliciously, it can claim incentive which is disproportionate to its actual battery level [12] and thus denying the system its fair incentives. This type of spoofing can be detected using accurate energy state estimation methods and spoofing nodes will be denied access to the blockchain.

2) EnU-Run: In this attack scenario, an IoT node can report upgraded battery level while being a part of the network. Another dimension of this attack is that it can underestimate the amount of data being received from all IoT nodes. This way, it can also spoof the amount of energy required for the next cycle and increase its probability of becoming CH. Here also, the motivation is that the node will get to become CH and gain an advantage in terms of incentive. It can be detected since all the communication to and from that node is tracked by the blockchain network. If spoofing is detected, the node will be kicked out of the network.

IV. Detection Algorithm

In this section, we present a mechanism for the detection of energy level spoofing in an IoT-based application scenario. We consider a queue-based recursive energy model where we have two different types of energies i.e., communication energy consumed $E_c(t)$ and harvested energy arrived $E_h(t)$.

The queue-based energy model is shown in Fig. 2. At any point in time, the updated energy at the next time instant can be determined by utilizing the energy harvested, energy consumed and the current energy level. The queue-based recursive energy can be represented by equation 1.

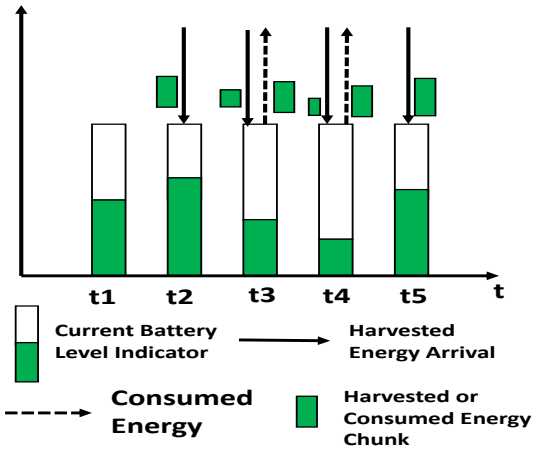

Fig. 2. Queue Based Energy Update Model

$$E(t+1) = [\min\{E_{\max}, E(t) + E_h(t) - E_c(t)\}]^+, \quad (1)$$

where $[x]^+ = \max\{0, x\}$. After every cycle, all the CMs share the amount of data they shared with the CH and their respective distances from the CH. This information is shared with the CH of the next cycle as this information is intended to be included in the blockchain which is further transmitted to the BS for consensus. Since the IoT nodes are energy constrained, in order to avoid their shut down, energy harvesting models are considered as discussed in the previous section. We will use two different models, details of which will be given later on. Based on this, an energy dissipation model is used which calculates the energy after being dissipated for CH as well as CMs. This model is given as follows [13]:

$$E_{Tx}(k,d) = \begin{cases} E_{ele} \times k + E_{fs} \times k \times d^2 & d \leq d_0 \\ E_{ele} \times k + E_{mp} \times k \times d^4 & d \geq d_0 \end{cases} \quad (2)$$

$$E_{Rx}(k) = E_{ele} \times k. \quad (3)$$

Here, E_{ele} is the electronics energy and specifically, it is the energy for transmitting or receiving one bit of data. E_{fs} and E_{mp} are the amplifier energy parameters specifically free space path loss and multipath fading energy parameters respectively and k is the number of data bits transmitted or received. Apart from these energy values, there is some energy required for aggregation at the cluster head which is denoted by E_{DA} and the cluster head has to relay the aggregated information to the BS which is located at a longer distance, so that energy dissipation also follows equation 2.

Similarly, as discussed before, nodes are equipped with energy harvesting apparatus which is able to harvest energy from different sources. We also assume that we have predetermined spatial and temporal patterns of energy harvesting. These patterns follow a certain type of distribution. Knowing that distribution, we put a δ bound on the energy harvested.

Based on the model explained in equation 2 as well as the energy harvesting model and knowing the initial attack free conditions, each node can estimate the energy consumption at each node based on parameters as discussed in eq. 2. For each CM node, energy consumption for transmitting the data to CH will be calculated based on the data and the distance to relay the data to the CH. For the CH, energy required to receive the data as well as transmit the data to the BS will be calculated based on the amount of data and the distance between CH and BS. Similarly energy harvested is estimated from the available distribution. The calculated values of energy are compared with the reported energy values. If the difference between these values is less than ϵ, then the values of reported energies are considered correct, otherwise spoofing is detected. The spoofing nodes will be removed from the system. Next we describe the energy spoofing cases based on $E_c(t)$ and $E_h(t)$ which are also presented in Table I.

Case I (Deterministic $E_c(t)$ and Deterministic $E_h(t)$): In this case, the communication energy and harvested energy are deterministic. $E_c(t)$ can be made deterministic by deploying CM nodes within Line of Sight (LoS) from the CH node. Since there will be no multipath fading, the energy dissipation can be quantified. Similarly, $E_h(t)$ can be also be made deterministic. This can be done if we have reliable energy sources attached to the IoT devices. For example, if the devices are connected to the grid, it can deterministically harvest certain amount of energy at a given time. Spoofing is not possible in this case since there is no source of uncertainty and with the analytical model, the spoofing will be detected.

Case II (Random $E_c(t)$ and Deterministic $E_h(t)$): In this case, $E_h(t)$ is deterministic but $E_c(t)$ is non-deterministic. This is the case when there are non-LoS links between the CH and the CM nodes, there is fading and variable channel states so the CSI is not the same. Here, spoofing is possible since there is uncertainty from the communication side of the energy model. The channel conditions are not always the same. Therefore, the idea of channel inversion power control (CIPC) is utilized [14]. CIPC is a technique which is used to ensure fairness between channels with low and high gains. The transmitter ensures a constant data rate by increasing or decreasing its transmit power to compensate for the fading. For example, if the channel is Rayleigh fading, using CPIC, some inverse of Rayleigh distribution will be utilized at the transmitter to ensure constant stream of data at the receiver.

Case III (Deterministic $E_c(t)$ and Random $E_h(t)$): In this case $E_c(t)$ is deterministic but $E_h(t)$ is random. $E_h(t)$ can be random when there is an uncertain source of energy to which the devices are connected for energy harvesting. For example, a solar panels based energy harvesting system is largely non-deterministic. The energy harvested depends on the temperature, intensity of light which ultimately depends on how exposed the panels are to the sunlight. So, the malicious nodes can spoof energy harvested at a given time. This type of spoofing can possibly be detected by using a similar solution proposed in Case II but based on $E_h(t)$. That is, we can assume bounds on harvested energy based on the deployment of energy harvesting devices available, anomalous deviation from which can be considered as spoofing.

Case IV (Random $E_c(t)$ and Random $E_h(t)$): In this case both $E_c(t)$ and $E_h(t)$ are random due to their respective sources of uncertainty. This is the most interesting and practical scenario since most real life situations are expected to have non-LoS deployment of nodes and uncertain sources of energy harvesting. Here, there are two avenues of spoofing and malicious adversary could utilize both of these to spoof the energy levels at a given time. This type of spoofing can be detected by either a solution combining both of the Cases II and III as well as by a weighted solution of both cases.

Case	$E_h(t)$	$E_c(t)$	Spoofing and detection
I	Deterministic	Deterministic	Spoofing not possible as both energies can be accurately determined.
II	Deterministic	Random	Nodes can misreport $E_c(t)$. Can be detected based on reported CSI.
III	Random	Deterministic	Nodes can misreport $E_h(t)$. Can be detected based on the predetermined energy harvesting patterns.
IV	Random	Random	Spoofing based on both $E_c(t)$ and $E_h(t)$. Can be detected by combining Case II and Case III.

TABLE I
$E_h(t)$ AND $E_c(t)$ SPOOFING CASES

V. Simulation Setup and Results

In this section, we describe the simulation setup for the spoofing detection and derive results based on that setup.

A. Simulation Setup

The simulation setup of our work is very similar to [13]. We consider that IoT devices are divided into very small clusters comprising of 20 devices. The maximum distance d between CMs and CH is considered to be 150m. We assume that our clusters are uniformly, randomly distributed between 1 to 150m for each simulation. The size of message transmitted between CMs and CH k is 20 bytes. The value of E_{ele}(electronics Tx/Rx) depends on the communication load and its value is considered to be 50nJ/b. The values of E_{fs} and E_{mp} depend on the amplifier characteristics and is supposed to be 10 pJ/b/m^2 and 0.0013 pJ/b/m^4, respectively. The energy of data aggregation is considered to be 5nJ. The BS is located at a distance of 500m away from the CH on average. At this range, the inter-cluster distances are considered negligible and we assume every node is 500m away from the BS. Using these values, we can calculate the evolution of energy values of the CMs as well as CH. We also consider a deterministic energy harvesting model where 0.05J of energy is harvested after every 5 rounds of consensus and adding the data to the blockchain. We randomly allocate the mean, μ_h, of the energy supplied to the node. We tailor a distribution for the energy harvested, $E_h(t)$ centered around μ_h. We assume a solar panel as the source of energy harvested so we will model the harvested energy's probability density function using the data provided by [15]. We assume that μ_h does not vary in a short time frame and that any node can estimate the μ_h of any other node accurately. In cases I and case II, we simply use $E_h(t) = \mu_h$. In cases III and IV (random $E_h(t)$), we are using the distribution derived from solar panel data mentioned above. We also consider that the maximum energy levels of all IoT devices in our system model is 0.5J.

Based on the distance and current energy levels, we calculate the average energy consumed for relaying the message to CH for every CM and for the CM to relay the aggregated message to the BS. In case III, we simply set $E_c(t) = \mu_c$ where μ_c is the average energy required to receive, aggregate and transmit a message. In cases II and IV, we assume that $E_c(t)$ has a clipped inverse exponential power distribution. This distribution has been used before as shown in [16].

In our simulations, a malicious node will report energy values by over-reporting their overall energy gained. In case II, this implies that they over-reported their energy harvested and in case III, this implies that they under-reported their energy consumed. It may be a mixture in case IV. A malicious node will pick a mean value that is a certain percentage more or less than μ_c and μ_h and report an energy change centered around this new mean.

B. Results

Based on the simulation setup described in the previous subsection, we generate results for the probability of detection of malicious IoT nodes.

1) Case II: We estimate the measure of energy they have spoofed as the percentage difference between our μ_c and their reported $E_c(t)$. We also set three different thresholds at 0.3, 0.5 and 0.7 as fractions of μ_c. If a node reports a communication power consumption beyond these values, it is considered malicious. Fig. 3 is a stacked bar graph where the complete bar (blue and orange shaded area) represents the true positives and the lower part (shaded blue) represents the false positives. It shows that at even when the nodes are under-reporting their energy consumed as 30% of μ_c, there is a 63% chance of detection. When they under-report their energy consumed as 70% of μ_c, the probability of detection rises to 68%. Increasing the threshold results in greater values to be spoofed, however, it benefits as benign nodes are less likely to be falsely flagged.

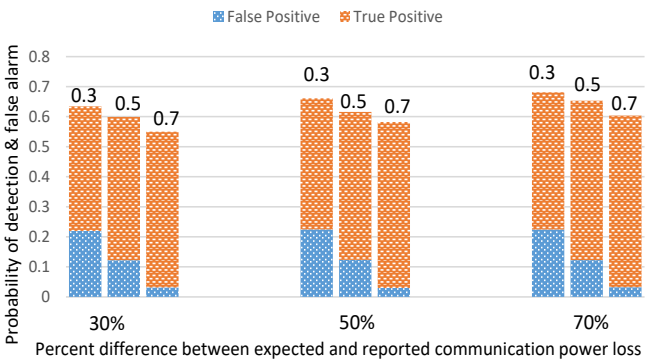

Fig. 3. The probability of detection of malicious nodes and false alarm for various fractional thresholds and differing amounts of energies spoofed in a Case II scenario.

2) Case III: We threshold the amount of energy they are spoofing as a percentage of μ_h. Fig. 4 shows that for various configurations of thresholds, the probability for detection falls slightly but remains between 76% and 65%. The probability of a false flag scenario is found to be 0 in our simulations as the distribution of the data is very well defined and there are very small benign deviations over short time intervals.

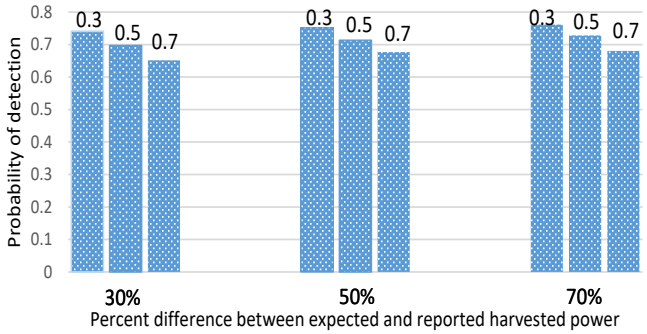

Fig. 4. The probability of detection of malicious nodes for various fractional thresholds and differing amounts of energies spoofed in a Case III scenario.

3) Case IV: We may not know either of $E_h(t)$ or $E_c(t)$ deterministically for a node. However, we can assume an average of the results for when $E_h(t) = \mu_h$ or $E_c(t) = \mu_c$ which corresponds to case II and case III respectively. We calculate the perceived fractional difference for case II and case III independently and take an average and threshold this average as in earlier cases. Fig. 5 shows that detecting malicious nodes is still viable in this scenario. In this scenario, a node is spoofing *both* of μ_c and μ_h as a fraction. The detection probability remains close to 80% in this scenario. This is attributed to the relatively low deviation in $E_h(t)$. As the deviation in $E_h(t)$ is smaller than $E_c(t)$, any deviation between reported and estimated energy may be attributed to $E_c(t)$ which increases the likelihood for detection significantly as μ_c is usually much smaller than μ_h. Fig. 5 also shows that in this scenario, the probability of falsely flagging a benign node drops when the threshold is increased.

Fig. 5. The probability of detection of malicious nodes and false alarm for various fractional thresholds and differing amounts of energies spoofed in a Case IV scenario.

VI. Conclusion

In this paper, we considered the problem of energy constraint in IoT nodes and asserted that a cluster-based system is an appropriate solution for that problem, where a CH with high energy deals with relaying data from CMs to BSes. However, this leads to energy spoofing attacks. In this work, we presented an effective blockchain-based solution to the problem based on pre-known distributions of harvested as well as communication energy. We noticed that when the nodes spoof as low as 30% of their energy, they are detected with high accuracy of close to 75%. Higher spoofing levels will lead to even better accuracy. Since the spatial and temporal patterns of harvested energy are known and the communication energy depends on the CSI between the sender and the receiver, it is hard to spoof the energy values. Therefore, this work has the promise to be included in energy-based blockchain-enabled IoT scenarios.

References

[1] S. Sun, R. Du, S. Chen, and W. Li, "Blockchain-based IoT access control system: Towards security, lightweight, and cross-domain," *IEEE Access*, vol. 9, pp. 36 868–36 878, 2021.

[2] M. Rehman, N. Javaid, M. Awais, M. Imran, and N. Naseer, "Cloud based secure service providing for IoTs using blockchain," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–7.

[3] P. Angin, M. B. Mert, O. Mete, A. Ramazanli, K. Sarica, and B. Gungoren, "A blockchain-based decentralized security architecture for IoT," in *International Conference on Internet of Things*. Springer, 2018, pp. 3–18.

[4] N. Ul Hassan, C. Yuen, and D. Niyato, "Blockchain technologies for smart energy systems: Fundamentals, challenges, and solutions," *IEEE Industrial Electronics Magazine*, vol. 13, no. 4, pp. 106–118, 2019.

[5] A. H. Khan, N. Ul Hassan, C. Yuen, J. Zhao, D. Niyato, Y. Zhang, and H. V. Poor, "Blockchain and 6G: The future of secure and ubiquitous communication," *IEEE Wireless Communications*, vol. 29, no. 1, pp. 194–201, 2022.

[6] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, 2019.

[7] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasurbramanian, "A lightweight blockchain based framework for underwater IoT," *Electronics*, vol. 8, no. 12, p. 1552, 2019.

[8] S. Khan, W.-K. Lee, and S. O. Hwang, "AEchain: A lightweight blockchain for IoT applications," *IEEE Consumer Electronics Magazine*, vol. 11, no. 2, pp. 64–76, 2022.

[9] H. Jayakumar, A. Raha, Y. Kim, S. Sutar, W. S. Lee, and V. Raghunathan, "Energy-efficient system design for IoT devices," in *2016 21st Asia and South Pacific design automation conference (ASP-DAC)*. IEEE, 2016, pp. 298–301.

[10] J. Wu, Y. Nan, V. Kumar, D. J. Tian, A. Bianchi, M. Payer, and D. Xu, "{BLESA}: spoofing attacks against reconnections in bluetooth low energy," in *14th USENIX Workshop on Offensive Technologies (WOOT 20)*, 2020.

[11] V. Buterin, "Automated censorship attack rejection," 2017. [Online]. Available: https://silo.tips/download/automated-censorship-attack-rejection-buterin-aug-2017

[12] Z. Liu, N. C. Luong, W. Wang, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "A survey on blockchain: A game theoretical perspective," *IEEE Access*, vol. 7, pp. 47 615–47 643, 2019.

[13] T. M. Behera, S. K. Mohapatra, U. C. Samal, M. S. Khan, M. Daneshmand, and A. H. Gandomi, "Residual energy-based cluster-head selection in wsns for IoT application," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5132–5139, 2019.

[14] P. I. Tebe, K. Ntiamoah-Sarpong, W. Tian, J. Li, Y. Huang, and G. Wen, "Using 5G network slicing and non-orthogonal multiple access to transmit medical data in a mobile hospital system," *IEEE Access*, vol. 8, pp. 189 163–189 178, 2020.

[15] A. Branco, L. Mottola, M. H. Alizai, and J. H. Siddiqui, "Intermittent asynchronous peripheral operations," in *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, 2019, pp. 55–67.

[16] S. Weber, J. G. Andrews, and N. Jindal, "The effect of fading, channel inversion, and threshold scheduling on ad hoc networks," *IEEE Transactions on Information Theory*, vol. 53, no. 11, pp. 4127–4149, 2007.