Probabilistic Blockchains: A Blockchain Paradigm for Collaborative Decision-Making

Tara Salman
Dept. Of Computer Science & Engineering
Washington University in St. Louis
St. Louis, USA
tara.salman@wustl.edu

Raj Jain
Dept. of Computer Science & Engineering
Washington University in St. Louis
St. Louis, USA
jain@cse.wustl.edu

Lav Gupta
Dept. of Computer Science & Engineering
Washington University in St. Louis
St. Louis, USA
lavgupta@wustl.edu

***Abstract*— A blockchain provides a secured paradigm to achieve consensus using a distributed and peer-to-peer network in which no trusted central party is required. As a result, it has the potential to resolve many challenges that are faced with current centralized controllers in globally distributed applications. To date, the blockchain technology has been used for recording transactions and tracking objects in which multiple participants reach a consensus on whether a transaction is valid or not. This paper introduces the novel paradigm of probabilistic blockchains, an extension of the current blockchains that allows building efficient and distributed risk assessment and decision-making applications in which multiple untrusting parties collaborate but may not completely agree on the outcome. The paradigm is particularly useful for risk assessment, where a group of decision-makers needs to decide or analyze an event based on imperfect information. The proposed approach can be used in applications like intrusion detections, stock market predictions, insurance, and recommendation systems. The paper presents and analyzes the application of probabilistic blockchains for intrusion detection systems for computer networks. The results show the feasibility and efficiency of the proposed paradigm in making such decisions.**

Keywords— blockchain technology; Probabilistic blockchains;, Risk assessment; Intrusion detection systems; IDS.

I. INTRODUCTION

There are several massively collaborative applications where the participating entities do not necessarily trust each other and may be competitive. These are called *multi-trust domain* applications. An example of a multi-trust domain is the current banking systems. The banks do not necessarily trust each other and, therefore, need the services of a centralized trustworthy organization, called SWIFT (Society for Worldwide Interbank Financial Telecommunication), to transact. This centralized solution introduces delays and additional costs.

An alternative solution, in a multi-trust domain system, is to distribute the decision-making process. Blockchains meet this objective efficiently and securely. They allow the parties to collaborate and achieve consensus without mediation by a centralized trusted authority. The distributed architecture and the continuous updates make the blockchains solutions provably secure against attackers, who may otherwise succeed in taking control of decision making in centralized solutions. The technology started with Bitcoin, the first digital payment system used mainly for financial applications [1]. Now it is used in many other applications including logistics, assess tracking, management, and events recording [2].

Currently, the blockchain technology is not usable for decision making in multi-trust domains where each participant has a different view of the transaction or event. The extensions to the blockchains that we propose in this paper can be significantly useful in making consensus decisions in risk assessments for multi-trust domain applications. With our proposed approach, an individual decision made by an agent, or a participant can be deterministic, as for a yes or a no, or it can be probabilistic, i.e., assessing that an event would happen with a certain probability. Our proposal would allow blockchain to connect these participants and process their decisions to achieve a group decision, or a "consensus." The consensus can also be deterministic or probabilistic where all members would agree to a decision or some would agree, and others would disagree.

Note that we will be using the term *"decision"* to represent an agent's individual decision while the term *"consensus"* will be used to represent a group decision made by more than one party.

As defined in Wikipedia, *"consensus decision-making" is a group decision-making process in which group members develop, and agree to support a decision in the best interest of the whole. Consensus may be defined professionally as an acceptable resolution, one that can be supported, even if not the "favorite" of each individual* [3]. To make blockchains suitable for such applications, the paradigm needs to be extended such that it can process differing decisions and achieve a probabilistic "consensus" about events.

This extension to the blockchain paradigm leads to what we refer to as *"probabilistic blockchains."* Probabilistic blockchains assess events, to which it is applied, and return a probabilistic consensus evaluation of their occurrence. The proposed paradigm would be suitable for many applications, including the ones that currently use traditional blockchains.

An example application that could benefit from the probabilistic blockchains is Intrusion Detection Systems (IDS). Different algorithms or different IDS products or different agents will give a differing opinion (decision) about whether a particular packet or sequence of packets represents an attack. In this case, a group consensus is essential even when the agents have differing "decisions" for the particular packet inspected.

A. Paper's Contributions and Organization

Risk assessment and decision making in multi-trust domain applications pose considerable research challenges. Consensus

decisions need to be made continuously about resource distribution, security, and several other application aspects. Decision makers need to have a global view of the system and sometimes require access to even private information. This is currently done by semi-distributed decision-making platforms such as Adaptive Decision Making Broker (ADMB) [4]. These platforms are geographically distributed and managed. However, critical decisions are made at a centralized global controller (also known as a broker) that is trusted by all system entities. The centralized controller, however, introduces a single point of failure, which is vulnerable to many security risks.

The main contribution of this paper is to introduce the probabilistic blockchain paradigm to resolve problems with centralization in multi-trust domain systems. We begin with a brief background of the blockchain technology in Section II. The extension of current blockchains to probabilistic blockchains is introduced in Section III. In Section IV, the security and benefits of the proposed concept are analyzed. We discuss the results of the experimental evaluation of probabilistic blockchain-based IDS in Section V. Other decision-making applications that may benefit from blockchain-based solutions are discussed in Section VI. Finally, Section VII provides the conclusions of the paper.

II. Blockchain Background And Related Works

This section presents a brief overview of blockchain technology. It also reviews some of the work related to blockchain-based decision making.

A. Blockchain Technology Overview

A blockchain consists of two main components: a database and a network of nodes, as illustrated in Fig. 1. A blockchain's database is a distributed, shared, tamper-aware and fault-tolerant store that keeps track of records in the form of transactions. Blocks are formed by bundling together a number of transactions and each block is linked to its predecessor by a hash. A hash is a fixed-length numeric value that relates to the previous block data. In addition, each block has a timestamp indicating when it was created, a signature proving its correctness and integrity, and a random number (nonce) for cryptographic operations. The signature and nonce allow blocks to be immutable even if they are publicly accessible. The blockchain's network consists of many distributed nodes that maintain the database in a peer-to-peer network. Nodes have access to the blocks; however, they cannot change them [2].

The blockchain technology allows nodes to communicate without a trusted broker or a trusted third party. When a node wants to interact with another, it sends its interaction in the form of a transaction. Many such transactions are collected to form a block. A block is verified by everyone and is added to the chain if it is valid. Otherwise, it is dropped, and the transactions will be recorded in another block. Both transactions and blocks are signed; hence, they cannot be changed or denied in the future.

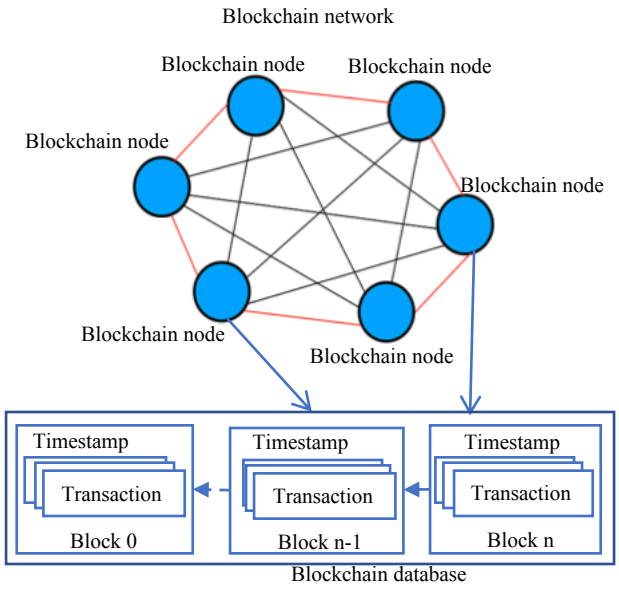

Fig. 1. Blockchain architecture

This novel architecture of the blockchain technology yields many appealing characteristics, including distributed management, decentralized consensus, trustless partners, provable security, immutability, and non-repudiation guarantees. Each term is briefly described here. The management is distributed as the blockchain database is maintained by many "blockchain nodes" and no party has full control over the system. The consensus is decentralized as there is no centralized authority and decisions can only be made by majority agreement. The trustless partners feature is added as the trust is imposed by a majority rather than by a single controller. Blockchains use sophisticated cryptographic techniques, resulting in guaranteed security by signature schemes and possible encryption schemes. They are immutable as no one can change, delete, or tamper transactions. Finally, they provide non-repudiation guarantees due to transactions and blocks being signed using elegant signature schemes.

B. Related Work

The use of the blockchains in multi-trust domain systems, or the distributed architectures in risk assessment applications, have been investigated by several researchers. The most remarkable research related to this paper is blockchain-based IDS presented in [5] and [6]. In [5], Meng et al. discuss the applicability of, and challenges in, using the blockchains for intrusion detection. In [6], Golomb et al. explore an intrusion detection framework that utilizes blockchain to gather information from Internet of Things (IoT) devices and update local intrusion detection models. The work proposed in this paper provides an improvement over other works by achieving the consensus within the blockchain architecture. Furthermore, the proposed paradigm suits many other decision-making applications.

Blockchain-based voting systems constitute another research area that relates to this work. Such systems are typically multi-trust domains, as voters, candidates, and vote casters do not necessarily trust each other. Blockchain technology can help to track the votes and possibly to cast them at the end of the

process. McCorry et al., [7], discuss the implementation of their proposed approach in the Ethereum blockchain platform and evaluate the system by the time and cost analysis. The work proposed in this paper is similar to the voting systems, as in a way the agents "vote" on the decisions made. However, it differs from the current blockchain-based voting systems in proposing a prompt and probabilistic consensus within the blockchain, making it possible to extend the concept to other decision-making applications.

Other researchers have proposed distributed architectures for management and decision making in distributed environments. For example, Han et al., [8], introduce a peer-to-peer recommendation system where decisions are made without any centralized controller. While their method does not require a centralized controller in the absence of any protection against it, false information could propagate in the hierarchy. The blockchain-based approach advances these works in providing resilient security features and inheriting blockchain characteristics discussed in the previous subsection.

The decision-making problem is an active research area in multi-agent systems. Several approaches have been proposed to achieve a consensus in such systems. For example, Li et al. [9], propose a group decision-making algorithm for heterogeneous agents, i.e., agents with different decisions. Similar to our approach, their approach achieves a consensus decision even if individual decisions are different. However, such approaches assume that agents are trustworthy and update their decisions to achieve consensus. Thus, many iterations may be involved to achieve consensus. In our approach, using a blockchain, agents can have an easy global view of the system, making it possible to achieve a consensus decision in one iteration. In addition, trust among agents is not mandatory, as the blockchains are trustless partner systems.

III. The Proposed Probabilistic Blockchain

This section describes the proposed probabilistic approach for blockchain-based decision making in multi-trust domain applications. We first explain the need to extend blockchain concepts to probabilistic blockchains. Then, we discuss the proposed metrics and the design to meet the requirements of our targeted applications. Finally, we discuss the mining technique used and present a workflow to illustrating how the proposed blockchain will work.

Saito and Yamada [10] describe how blockchain can be considered a probabilistic state machine. However, their discussion is about modeling current blockchains. It is not related to our work here on the utilization of blockchain technology to achieve a consensus decision even if different participants have differing decisions.

During our discussion, we assume three types of nodes: *miners, blockchain nodes,* and *blockchain users*. Miners are nodes that construct blocks and who compete to be the first to form a new block. Blockchain nodes store the chain, validate new blocks, and add these blocks to the chain. Blockchain users are nodes that use the blockchain. In our case, we classify the users that supply decisions as "agents" and those that inquire about the consensus ("inquirers"). Note that these are simply functions and a physical node can implement more than one function. For example, a miner may also be a blockchain node. An inquirer may also be an agent.

A. The Need to Extend Blockchains to Probabilistic Blockchains

To meet the requirements of blockchain-based decision making in multi-trust domains, the technology needs to be extended to reflect local and global decisions precisely. The objectives of the current blockchain mechanisms are to verify transactions and blocks and check for simple local decisions, such as whether a transaction is in the database or not. In other words, processing of transaction data, when the blocks are created, is not offered by the current blockchain implementations. Moreover, the blockchain conclusions are a deterministic 'yes' or 'no' without assurance or confidence guarantees. However, for most risk assessment applications, the inquirer needs to know the accuracy and confidence level of the returned results. For example, in IDS, the network administrators need to know how precise the returned value is, how many agents participated in the prediction, and so on.

To extend blockchains for decision making, we extend blockchains' transactions and blocks structure to include more precise information that reflects the probabilities and the level of uncertainty of the information enclosed. Additionally, the consensus to achieve decisions is introduced.

B. Proposed Metrics for Probabilistic Blockchains

We extend the traditional blockchain to a probabilistic blockchain, reflecting the probability and confidence of agents that participated in a particular consensus decision. The probabilistic result is produced instantaneously when the block is created rather than when requested. Further, uncertainty measures can be added to reflect the results' variability.

Transactions, blocks, and consensus should be modified to reflect extensions to the traditional blockchains mentioned above. Transactions submitted by agents, i.e., blockchain users that make decisions, have a decision variable reflecting the agent's local decision about a certain event. This variable can be probabilistic, where an agent returns a probabilistic decision value between [0,1], or deterministic, where an agent returns 0 or 1 with full confidence. Different agents may inspect the same event and may return differing local decisions. Then, miners form a block summarizing the decisions from multiple transactions and create a consensus decision about the inspected event. For example, in an intrusion detection application, different agents decide if a flow with certain features is malicious or not. Mining nodes would summarize agents' decisions for each flow. This summary represents an interpretation of the system about that flow. Hence, the summary can be considered as the consensus value of the system and can be formulated as follows:

$$Summary(event_i) = G(P_j(event_i))$$

Where $event_i$ is the inspected event, j is a participating agent, $P_j(event_i)$ is an individual probabilistic decision made by an agent and G is a function that summarizes all the decisions made by agents for the targeted event. We allow G to be any appropriate function with the following two properties:

1. *G should not be easy to manipulate such that it changes the consensus interpretation.* Specifically, manipulating a few decisions by some malicious agents should not affect G in a way that changes the consensus interpretation. Examples of a bad G would be taking the minimum or the maximum of the decisions involved. These examples are easy to manipulate as a malicious node can use a 0 or 1 decision to manipulate the consensus value. However, taking the mean or the mode would be a sufficient G since it is hard to change the interpretation of the consensus when more than half of the nodes are honest nodes.
2. *G should be easy to calculate.* Since the number of transactions is large and the process of mining is already complicated, achieving the consensus value should be simple, computationally efficient, and time constrained. A complicated recursive function could delay the process of block creation even more which is not desirable.

We allow any function that satisfies these two conditions to summarize the consensus and be used as *G*, the consensus function. It can be as simple as taking the mean or the mode, or as complex as a moving average, a weighted average, or the result of a sophisticated machine learning algorithm. A more elegant summary can also be taken from social science literature, e.g., the social choice theory [11].

A consensus function does not have to return a single number. It can return a vector or even multidimensional matrix. For example, one may summarize with a vector with mean, standard deviation, confidence interval, skewness, kurtosis, and higher order moments and other probability measures as its elements [12]. It can also include the number of agents that participated in achieving consensus. As a matter of fact, the consensus function is application dependent which calls for flexibility in choosing a function appropriate for the application.

In this way, all transactions, blocks, and consensus returned by the blockchain can be probabilistic. Further, a summary of the transactions can be presented in each block for fast recovery of any consensus decision required.

C. Block Design in Probabilistic Blockchains

To maintain consistency, we propose an architecture for the probabilistic blockchain which is similar to the traditional blockchains where blocks contain transactions and are linked in a chain. Fig. 2 shows a simplified architecture of the probabilistic blockchain for the intrusion detection use case. It may be noted that the hash and the timestamp are inherited from traditional blockchains. Each transaction has a variable, $P_j(event_i)$, that represents the agent *j's* decision for a specific flow *i* being malicious or not. The agent identity, *AgentID*, (e.g., the public key of the agent) is included in every transaction. The features that led the agent to his/her decision are added to each transaction. Sample features of intrusion detection application include network delay, packet rate, and packet sizes. Several such transactions are present in a block. Miners will summarize the decisions for each event included in the block by the consensus function, or the G function discussed earlier.

The summary of each block contains the transactions in that block. The blocks are of variable length. Also, multiple events can be predicted in the same block. Therefore, a block can include the consensus of variable length made for several events. That is, a block may have a summary of one or many events depending on the transactions available at the time.

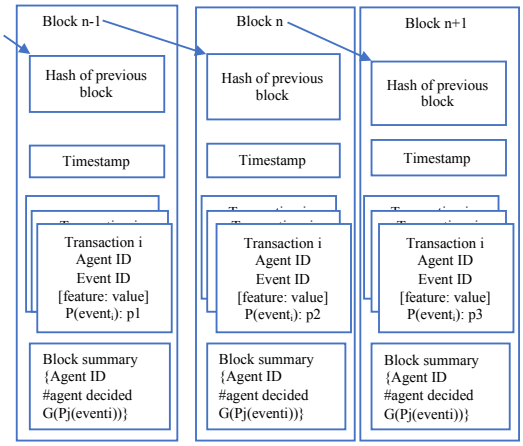

Fig. 2. The proposed transactions and blocks architecture

D. Challenges in Achieving Probabilistic Consensus

One of the challenges in achieving a group consensus is that the transactions for the same event may arrive at different miners at different times. This results in these transactions being distributed over multiple blocks. Therefore, updating the consensus value and reaching a finalized consensus are challenging.

To resolve the first challenge, i.e., updating the consensus value, it is helpful if the consensus function is such that it can be incrementally computed. That is, given the summary of one set of transactions and summary of another set of transactions the summary of the combined set can be computed directly from the two summaries. Mean is an example of such a summary, but there are many other functions. For functions not satisfying this property, the miner will need to collect all the transactions related to the event from previous blocks and adds the new transactions to compute the summary. If the transactions for an event are spread over multiple blocks, then the block may have a pointer to the previous block with that event and an indication whether the summary includes all past transactions for that event. This will make consensus inquiries fast since the users will not have to traverse the blockchain. However, the search for the last block that included the target event may require fast search algorithms.

The second challenge, reaching a finalized consensus, is harder to resolve as agents could be sending their transactions for a specific event at different times. We provide two alternatives to resolve this. In the first alternative, we assume that the decisions are final after a certain number of blocks have been constructed. This could be imperfect if the difference in time for the transactions of the same event is too long. The other alternative is to give the latest consensus value which can be updated as the system progresses in time. The specific decision among the alternatives is application dependent, and different

decisions can be made by the application developer to resolve this issue.

E. Mining Technique Used

Mining techniques have minimal effect on our blockchain-based decision making. The primary requirement is to be able to summarize the results without controlling the system. Thus, any mining technique that is justifiable can be used. Several mining techniques have been proposed in the literature; readers may refer to [13] for a comparative analysis of different mining techniques. In this work, we follow the proof of work (PoW) technique as it is the most famous and widely adopted by the blockchain implementations [14]. A detailed discussion of PoW is out of the scope of this paper. However, readers can refer to [14] for further information.

We must emphasize that the use of any particular mining technique is not central to our work. It is neither required or recommended. We use it only as an example.

F. Probabilistic Blockchain Workflow

The probabilistic blockchain works just like the traditional blockchains. However, there are some differences that need to be taken care off. Namely, how the system would work given that the workflow is different for decision-making applications. Thus, in this subsection, we provide an example workflow illustrating how the system should work.

The workflow, as illustrated in Fig. 3, is composed of 4 stages: transactions' collection, block proposal, block approval, and block commitment.

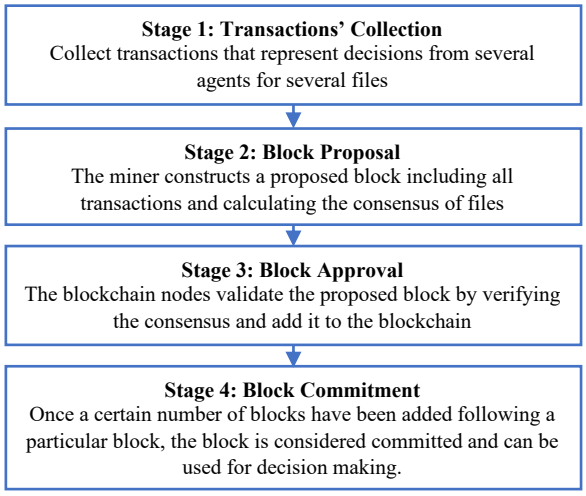

Fig. 3. The workflow for the proposed probabilistic blockchains

In *the first stage*, *Transaction Collection,* an event is initiated by an inquirer node that requires a global consensus on a particular event, e.g., a specific intrusion prediction. In response, many agents will broadcast their decisions in the form of transactions to the blockchain network. Miners will collect these decisions/transactions to form the block in the next stage.

In *the second stage*, *Block Proposal,* the miners construct the block following the architecture discussed in Section III (C). Block construction in most blockchain's implementations is time triggered, and so we assume it is time triggered, for example, a block is formed every 2 minutes. The miner should check recent past blocks to verify if the event is present in them. This ensures that all decisions made for a certain event are included in the consensus value. This block is submitted to the blockchain.

In *the third stage*, *Block Approval,* blockchain nodes in the network validate the proposed block to make sure that no invalid transaction is included and that the consensus value is correct. If the block is approved, it will be added to the local chain. However, it is not yet committed.

At the block approval stage, many valid blocks may be added to the chain. This happens when multiple received blocks point to the same previous block. Most mining nodes will be following a particular branch causing it to be longer than others. After some time, short branches get pruned since most mining nodes did not follow them. This step is inherited from traditional blockchains, e.g., Bitcoin blockchain, and this is what we refer to as *Block Commitment* stage.

By committing the block to the blockchain, the inquirers can check the summary, i.e., the consensus made for the requested event. Further, the consensus can be accessed by any other blockchain users for future use or consultancy. Future events that are submitted to the blockchain should have a unique ID to distinguish them from the previously submitted events.

IV. Security Analysis Of The Proposed Approach

The proposed approach does not add to the security of traditional blockchains, however, it extends their application domain to decision making and risk assessment applications. Compared to traditional decision-making solutions, our proposed probabilistic blockchain-based approach provides several security benefits which are discussed in this section. These include resiliency to malicious agents, resiliency to malicious miners, Distributed Denial of Service (DDoS) protection, and fraud mitigation.

A. Resiliency to Malicious or Bad Agents

Our blockchain-based solution is resilient against malicious agents that try to take control over the application. In a centralized solution, compromising the trusted centralized agent leads to the decision process failure. For example, in stock market predictions, if one centralized agent is providing the predictions, then controlling that agent will control the decision. Similarly, in a semi-distributed management architecture, compromising the centralized controller or broker will break the whole system. In contrast, with an appropriate consensus function that satisfies the discussed properties, a blockchain-based design is hard to break and is resilient against malicious agents. In the following, we sketch the proof of this resiliency.

Analysis 1: With a function that satisfies the required consensus function properties, the probabilistic blockchain is resilient against malicious agents that try to manipulate the consensus such that the interpretation is changed. The

constraint is that the number of malicious agents should be less than 50% of the total number of agents.

Note that *the manipulation of the consensus value by a small amount cannot be prevented, however, the consensus value cannot drastically change* such that the decision is flipped. The analysis is based on the choice of the consensus function that satisfies the consensus function properties discussed earlier. We sketch the proof of the analysis with the simplest consensus function, the arithmetic mean.

Consider a Consensus function that is simply the mean (first moment) of the individual decisions made by agents. That is:

$$Summary(event_i) = \frac{1}{m} \sum P_j\ (event_i)$$

Event i was inspected by an *agent j* which broadcasts a transaction T_j as follows:

$$Tj: \{ agentID: j, eventID: i, features: [], P(event_i): 1\}$$

Here, *m* is the number of transactions included in the summary. Assuming that we have *m* agents forecasting the same event, transactions $\{T_1, \ldots, T_m\}$ will be sent to the blockchain. Let's assume that *node l* is the miner at this point.

As one example, consider the case in which all agents state that the event will happen with certainty and send a deterministic 1 as their decision. The *miner l* will construct the block with $event_i$ consensus as follows:

$$\{EventID: i, \#agents: m, mean:1, stdv: 0\}$$

This indicates that the blockchain concluded that $event_i$ is happening with certainty.

Now, assuming that we have *n* malicious agents, where $n<m/2$, *e.g.*, $n=0.2m$. This means that 20% of the agents are malicious and send transactions with flipped decisions. That is, if *agent g* is malicious, the following transaction T_g will be sent:

$$T_g: \{agentID: g, eventID: i, features: [], P(event_i): 0\}$$

In the above transaction, we assume that $P(event_i)$ is manipulated to the fullest possible extent, to have the most effect on the consensus value.

Now, *miner l* will have *m* agents participating in the consensus: 80% of them are giving correct decisions "1" while 20% are giving wrong decisions "0". Consequently, the block summary:

$$\{EventID: i, \#agents: m, mean:0.8, stdv: non\text{-}zero\ value\}$$

The *stdv* (standard deviation) here is dependent on how many agents are involved, but it is known to be a non-zero value.

This indicates that the blockchain concluded that $event_i$ will happen with 80% probability instead of 100% in the no-adversary case. This is still a high probability value. Thus, the interpretation may not have changed if the application allows this level of uncertainty.

Assuming that the system has 1000 agents participating in the decision, this 20% will map to 200 faulty or malicious agents. It is difficult to compromise these many agents, compared to compromising the one party that is done in traditional centralized systems.

Actually, by taking the first moment as the consensus function, the probabilistic consensus effect has a linear relationship with the number of malicious nodes and the adverse impact is felt after 50% or more nodes become malicious. A more sophisticated consensus function should be more resilient to malicious nodes trying to flip the consensus interpretation.

B. *Resilience to Malicious Miners*

Mining techniques are resilient against malicious miners that try to control the decision process or manipulate decisions. This is true as each generated block is first validated for correct consensus. In addition, even if a malicious block is verified that block will be different from other computed blocks. In this case, malicious blocks will not be followed, thus, pruned from the blockchain. This feature is inherited from the traditional blockchains.

C. *DDoS Protection*

Our blockchain-based solution can protect decision systems from Distributed Denial of Service (DDoS) attacks, which are considered the most threatening attacks on the Internet [25]. In these attacks, a set of attackers targets the availability of the system by sending too many requests to be processed. In a centralized or semi-distributed architecture, this can be done by targeting the centralized controller. However, in a blockchain distributed design, the DDoS attack is made harder as there is no single point of vulnerability. DDoS attacks can cause the loss of billions of dollars in applications such as stock market predictions or other financial applications. Put another way, the protection provided by our approach can guarantee the availability of the system and prevent losses worth billions of dollars.

D. *Fraud Mitigation*

Our blockchain architecture can prevent fraudulent attacks; hence, help applications in keeping the integrity and the correctness of current and prior information. A fraudulent attack tries to manipulate the individual decisions, that is, break the integrity of the decisions and prevent the detection of such infringement. This is done by compromising and manipulating the data storage. Blockchains offer fraud mitigation guarantees since blocks and transactions in the chain are signed and replicated in many nodes. This leads to immutability in blockchain-based solutions, which makes it extremely difficult to alter or manipulate the decisions.

V. Experimental Setup And Evaluation

This section reports the experiment settings and evaluation of the proposed approach applied to IDS. It first discusses the choice of consensus function. Then, it presents the blockchain

setup, the dataset and the evaluation cases, the evaluation metric and the experimental results of two evaluation cases.

A. The Consensus Function

IDS predict if a network flow is malicious or normal and, in some case, the percentage or certainty of such prediction. The agents will individually make their predictions which will be shared over the blockchain as was discussed in Subsection III (C). For simplicity, we assume agents would make binary decisions as malicious flow ("1") or normal flow ("0".)

We use the first moment as a simplified consensus function. Formally, given many returned decisions from different agents and considering that we have two possible choices per flow, e.g., malicious or normal, blocks will have a consensus function of the results as follows:

$$G = P\ (Flow_i\ is\ malicious) = \frac{1}{m}\sum P_j\ (Flow_i\ is\ malicious)$$

Here m is the number of participating agents and $flow_i$ is the inspected flow and the j^{th} agent decides whether $flow_i$ is malicious or not by its prediction function P_j. P_j can be a result of a machine learning algorithm, a rule-based output or a signature-based prediction. Note that P_j in this specific case is '0' or '1' but generally can be any probabilistic value or vector. Also, this is a simplified case which involves two decision options only, but can be extended to any number of choices and applied to other applications.

In some applications of high risk, the summary function may be changed so that the consensus G is one if the mean is 0.001 or higher. That is, even if 0.1% of the agents believe that the flow is malicious, the flow is denied.

B. Experiment Setup

To build probabilistic blockchains, we simulated 1000 agents, one inquiring node, five blockchain node, and five miners to form a blockchain network. Agents make decisions that are submitted to the blockchain as transactions. As discussed, a positive decision ("1") indicates that the flow has been classified as malicious (an attack is detected) while a negative decision ("0") implies normal traffic. Then, the miners calculate the probabilistic consensus and form the blocks as was illustrated in Fig. 3. A mapping of the probabilistic consensus to malicious or normal flow is needed to evaluate the system. To do so, we use a simple mapping which states that a flow is malicious if more than 50% of the agents say so. That is, if the probabilistic consensus is more than 0.5, then the flow is declared malicious. Otherwise, the flow is normal.

We have considered a simplified example here. However, a more sophisticated decision-making process may be used in actual deployments. This may involve taking the standard deviation or even the second or third moment into consideration when making decisions. Also, the value of threshold (taken as 0.5 here) is application and attack type dependent. As an example, real-time applications may not tolerate attacks that affect their availability. They may accept some false alarms, but they need a very low DoS attack detection misses. Hence, the threshold, as indicated earlier could be as low as 1% or 0.1% in this case.

C. Accuracy as an Evaluation Metric

We will consider accuracy as the only evaluation metric used. Other metrics such as false alarm rate and attack not detected rate can be considered. However, our objective here is to show the feasibility of the method and the accuracy is sufficient to show that.

Accuracy is the most frequently used metric for evaluation. It measures the degree of correctness of the predicted values to the overall number of samples. That is,

$$Accuracy\ \% = \frac{Correct\ Predictions}{Overall\ Samples} \times 100\%$$

A correct prediction is achieved if the algorithm prediction matches the ground truth when compared offline. Here, both the numerator and the dominator represent the number of samples. The numerator is the number of samples that are correctly predicted while dominator is the overall number of samples.

D. Experimental Results

We have used the UNSW-NB15 publicly available dataset, made possible by the University of New South Wales (UNSW) in 2015 [15], to predict attacks at local agents. The dataset is composed of 9 types of attacks in addition to standard flows. The details of the dataset are out of the scope of this paper but can be read at [15] and [16].

To testify probabilistic blockchain performance, two evaluation cases are used. In the first evaluation, a few machine learning models are trained with different algorithms and the same dataset. Namely, we use Logistic Regression (LR) [17], Random Forest (RF) [18], and Decision Tree (DT) [19] to build three learning models that detect only DoS attacks from the targeted dataset. The models are distributed randomly among the 1000 agents, who use them to make decisions about the received flows. The probabilistic consensus values, as discussed in Subsections V (A) and V (B), is used to decide if there is an attack. Then, the consensus along with the decisions made by the machine learning models are used to evaluate the proposed approach.

The results of the above evaluation are shown in Fig. 4 where the three machine learning models along with probabilistic blockchain are compared. A higher accuracy indicates a better prediction. As can be seen, the proposed probabilistic blockchain (PB in Fig. 4) approach shows a performance similar to the best used machine learning model. The RF model, the DT model, and the PB have a high accuracy while the LR model has a relatively bad performance. PB has the highest accuracy compared to other models which shows the feasibility of the proposed approach. This also shows the resilience against bad behaving agents, i.e., agents with the LR model, as long as they are less than 50% of the agents.

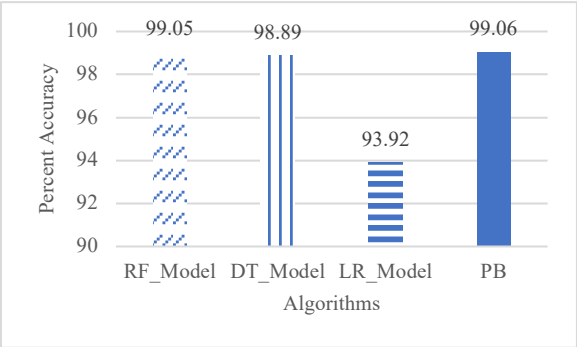

Fig. 4: Accuracy results of the first experimental setup

For the second evaluation, the training dataset is varied, and the same algorithm is used to build the learning models. Namely, we build models that use RF algorithm to detect DoS and Reconnaissance attacks. In this way, the training data differ for both models, while the test data will be the same. Similar to the first experiment, the two models have been distributed among the 1000 agents who predict the same testing sample flow and the blockchain achieves a consensus about that flow. The probabilistic consensus values, as discussed in Subsections V(A) and V(B), is used to decide if there is an attack. The consensus along with the decisions made by the machine learning models are used to evaluate the proposed approach.

The results of the second experiment are shown in Fig. 5 where the DoS model, the reconnaissance model, and the PB model are compared. As can be seen, the PB model performs the same as the DoS model which does better than Reconnaissance (Recon in Fig. 5) model. This shows that probabilistic blockchain will have a similar result to the best performing model even if only two models were involved.

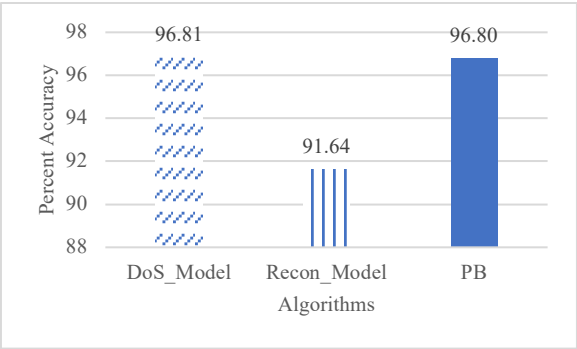

Fig. 5: Accuracy results of the second experimental setup

These evaluations demonstrate the feasibility and performance of the proposed probabilistic blockchains with simplified IDS evaluations. Results also show that the probabilistic consensus model has a performance that is comparable to the best performing machine-learning models used by agents. It should be noted that with the consensus function used here, the proposed algorithm would fail to give correct results if more than half of the agents give wrong predictions. However, even with this drawback, the results are still good, considering the added security and distribution features provided by the proposed approach.

VI. Application Of Probabilistic Blockchains

In addition to the IDS, there are several risk assessments and decision-making applications that can benefit from the proposed probabilistic blockchain paradigm. For example, the stock market prediction is a multi-trust domain application where agents compete with each other. In this application, agents predict whether the stock would rise or decline and the value by which the stock will change. Different agents give their decisions (predictions), and the blockchain achieves its consensus about the stock. The consensus function, for example, could be the weighted average, where agents are weighted based on their past performance.

Blockchain-based approaches can help build better recommendation systems for any asset, such as hotels or products. Different agents would give their decisions about how good/bad a particular asset is and the blockchain would achieve a consensus about that asset. A weighted average function, where agents are weighted based on their past predictions and reputation, would be suitable as a consensus function used in this application.

In addition to decision making, probabilistic blockchains can act as systematic feedback for reinforcement learning applications. Reinforcement learning is a type of machine learning that builds models by taking action and receiving feedback from the system. Probabilistic blockchains can be used to provide this feedback and update the learned models accordingly. If the decision made by the agent is extremely different from the consensus, the model should be updated. If the decisions match, the model is updated to reflect this new knowledge and the reward function is applied.

VII. Conclusion

Blockchain technology provides a secure, consensus-based distributed platform with a large number of potential applications. However, extensions are required to make them suitable for different applications. In this paper, we introduced probabilistic blockchains, an extended blockchain paradigm, designed for decision-making and risk assessment applications in multi-trust domains. First, we gave a brief background of the blockchain concepts, including its architecture, properties, and mining techniques. Following that, we discussed the need for probabilistic blockchains and the metric that can be used to evaluate it. We also discussed the block architecture, mining technique, and the proposed workflow. The proposed approach has been shown to be more secure than centralized approaches because of the inherent resilience to malicious nodes and miners. It additionally provides DDoS protection and fraud mitigation. An experimental evaluation involving blockchain-based IDS was presented to show the feasibility of the proposed approach along with its limitations. The proposed work is still evolving, and, when fully developed, it is expected to be suitable for many risk assessment applications, especially those in multi-trust domains. Some future work ideas include extending a popular public blockchain with the proposed concept and building an online blockchain-based IDS. Further,

testifying the concept for other financial and network applications is also planned as future work.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009.

[2] M. Pilkington, Blockchain technology: principles and applications, Research handbook on digital transformations, 2016.

[3] Wikipedia, "Consensus Decision Making," [Online]. Available: https://en.wikipedia.org/wiki/Consensus_decision-making. [Accessed 21 Augest 2018].

[4] F. Samreen, G. S. Blair and M. Rowe, "Adaptive decision making in multi-cloud management," in *the 2nd International Workshop on Cross-Cloud Systems*, Bordeaux, France, 2014, pp. 1-6.

[5] W. Meng, E. W. Tischhauser, Q. Wang, Y. Wang and J. Han, "When Intrusion Detection Meets Blockchain Technology: A Review," *IEEE Access,* vol. 6, pp. 10179-10188, 2018.

[6] T. Golomb, Y. Mirsky and Y. Elovici, "CIoTA: Collaborative IoT Anomaly Detection via Blockchain," in *Network and Distributed Systems Security Symposium (NDSS)*, San Diego, CA, 2018.

[7] P. McCorry, C. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in *International Conference on Financial Cryptography and Data Security*, Malta, 2017, pp. 357-375.

[8] P. Han, B. Xie, F. Yang and R. Shen, "A Scalable P2P Recommender System Based on Distributed Collaborative Filtering," *Expert Systems with Applications,* vol. 27, no. 2, pp. 203-210, 2004.

[9] G. Li, G. Kou, and Y. Peng, "A Group Decision Making Model for Integrating Heterogeneous Information," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 48, no. 6, pp. 982-992, 2018.

[10] K. Yamada and H. Saito, "What's So Different about Blockchain? — Blockchain is a Probabilistic State Machine," in *IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, Nara, 2016, pp. 168-175.

[11] Wikipedia, "Social Choice Theory," [Online]. Available: https://en.wikipedia.org/wiki/Social_choice_theory. [Accessed 21 Augest 2018].

[12] R. Jain, The Art of Computer Systems Performance Analysis, Wiley-Interscience, 1991.

[13] L. M. Bach, B. Mihaljevic and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO),* Opatija, 2018, pp. 1545-1550.

[14] Bitcoinwiki, "Proof of work", [online] Available: https://en.bitcoin.it/wiki/Proof_of_work, (accessed September 21, 2018).

[15] M. Nour and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network dataset)," in *IEEE Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia, 2015, pp. 1-6.

[16] M. Nour and J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information Security Journal: A Global Perspective,* vol. 25, no. 3, pp. 18-31, 2016.

[17] A. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naïve Bayes," *Advances in neural information processing systems,* vol. 14, pp. 841-849, 2002.

[18] J. Zhang, M. Zulkernine and A. Haque, "Random-Forests-Based Network Intrusion Detection Systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 38, no. 4, pp. 649-659, 2008.

[19] N. B. Amor, S. Benferhat and Z. Elouedi, "Naive Bayes vs. Decision Trees in Intrusion Detection Systems," in *ACM Symposium on Applied Computing*, Nicosia, Cyprus, 2004, pp.420-424.