Modeling and Simulation of a Blockchain Consensus for IoT Node Data Validation

Bismark Tei Asare[1], Laurent Nana[2], Kester Quist-Aphetsi[3]
Lab-STICC, CNRS UMR 6285, F-29200, Computer Science Dept, GCTU, Ghana
Cyber Security Division, CRITAC, Directorate of Info. Assurance & Intelligence Research, CRITAC, Ghana
Université de Bretagne Occidentale, UBO, Brest, France[1]
Lab-STICC, CNRS UMR 6285, F-29200, Université de Bretagne Occidentale, UBO, Brest, France[2]
Computer Science Dept, GCTU, Cyber Security Division, CRITAC, Ghana
Directorate of Info. Assurance & Intelligence Research, CRITAC, Ghana[3]

***Abstract*—The classical blockchain developed for the Bitcoin cryptocurrency has evolved since its introduction more than a decade ago. Blockchain exists in different forms for different purposes and operational contexts. There has been a significant growth in the business use cases of blockchain which is based on the unique attributes of the distributed ledger technology. Blockchain provides peer-to-peer distribution of data in a traceable and decentralized architecture that attains data authentication using consensus protocols. Blockchain as a distributed ledger is the fusion of cryptography, peer-to-peer networking technology, distributed system technology, and consensus mechanism to assure information security and digital asset management. Consensus mechanisms are applied to the distributed ledger that operates in a peer-to-peer network where message transmission between peers is validated and stored across all active peers. Reaching an agreement to validate message transmission and maintaining the correctness of the state of data in a network for critical wireless sensor networks have become a necessary requirement for networks that span several subsystems covering a large operational area. Due to the resource constrained nature of the active actors of wireless sensor networks, any cryptographic solution to be adopted must be lightweight and efficient as well. This paper proposes a blockchain-based decentralized mechanism for authentication of node data for storage onto a distributed ledger. The coloured Petri net was used to model and simulate by detailing the critical attributes of the workings of the system that is based on cyber-physical IoT architecture.**

Keywords*—Blockchain consensus; ripple consensus algorithm; coloured petri net; cyber-physical system; IoT architecture; node data security

I. Introduction

Reaching agreements to validate the authenticity of node data and subsequent transmission and storage of such network resources for cyber-physical systems have been a challenging and interesting domain for academia and information security industry players in recent times. Distributed ledgers use consensus algorithms to reach agreement among all connected active nodes to validate message transmission in a peer-to-peer approach. Recent advancements in connectivity, artificial intelligence, machine learning approaches although have provided an advantage for the expansion of network coverage and prediction and visualization of network resource sharing for enterprises and institutions, these available passive technologies in the hands of bad actors and hackers could render sophisticated cyber-attack exploitations to networks and user accounts resulting in breaching data, corrupting data, and compromising the security of such network systems [1].

Reaching an agreement to validate messages as well as authenticate the state of a distributed ledger have been the requirement for networks whose major components are resource constrained.

Distributed consensus research has become popular since the Nakamoto Satoshi introduced the Bitcoin blockchain cryptocurrency more than a decade ago. Internet of things have provided a platform for expanding the network resources to secure a new value proposition for scaling the scope of an enterprise's network. Available reports support that the technological and cybersecurity budgets for business and institutions have increased allocations and are making efforts to include internet-of-things integration, expansion, and management [2].

Cybersecurity investments and the annual budgetary allocations across most enterprises have increased largely due to the increasing number of cyber-attack incidences on enterprise systems which have resulted in data corruption, data theft and huge revenue losses in some cases [3].

During the COVID-19 lockdown period, most businesses and institutions adopted telecommuting as a measure to regulate and manage people in observing physical distancing to avoid possible person-to-person infection. There were reported cases of cyber-attacks during this period where personal and enterprise data were breached as a result [4].

Cascading effect of cyber-attacks on heterogeneous systems for wireless sensor networks like the internet of things has a wide and costly impact due to the critical messages that the resource-constrained devices in such networks transmit [5].

The effect of cyber-attacks on heterogeneous wireless sensor networks resulting in the compromise of critical data in enterprise networks have taken an alarming trajectory due to the complexity of the interconnectedness of the components of the subsystems that make up the internet of things architecture [6].

Availability of pervasive applications and their integration in enterprise networks that have most of its component relying on internet of things architecture could make the security management of such wireless sensor systems a complex challenge to undertake.

A consensus algorithm for decentralized authentication and distributed ledgers for an IoT with heterogeneous system architecture requires a blockchain-based agreement mechanism that operates with relatively less energy, fully scalable and most importantly byzantine fault-tolerant [7].

Wireless sensor networks achieve privacy and integrity for message transmissions using either third-party trust enforcement systems that adopt a centralized entity to authenticate devices and validate messages or a decentralized mechanism for authentication of devices and validating messages. The centralized authentication mechanism is prone to several attacks including single point of failure attacks. In a distributed system that rely on decentralized authentication mechanism, agreement is reached by all active nodes on the network using consensus. Since the introduction of Bitcoin cryptocurrency, several consensus algorithms have been developed. The Bitcoin cryptocurrency uses a distributed consensus mechanism that is based on proof of work [8].

The Coloured Petri Net (CPN) is a modeling and simulation tool for modeling and simulating systems, and verifying their properties (real-time, behavioral, security properties …).

The CPN modeling, simulation, and validation of critical security properties of an efficient blockchain-based consensus mechanism that does not compromise the security requirements of a cryptographic solution and offers low latency with improved resistance to the Byzantine fault tolerance is presented by this paper.

The rest of the paper is structured in sections and represented as follows. In Section II, related work describing the state of the art for CPN in modeling and simulating security protocols for networks, node data security and critical security challenges in cyber-physical systems is presented. In Section III, the Ripple consensus algorithm is described. Section IV outlines the implementation of the consensus algorithm in establishing agreements for storing messages on the distributed ledger. Section V concludes the paper.

II. Related Work

A. Consensus Algorithm

In a stand-alone system, validating transmitted message or any transaction is vested in a dedicated centralized node. Consensus algorithms are useful in networks that do not have a dedicated node to singularly authenticate users, processes and transmitted messages or transactions. A consensus algorithm was employed by the system to agree on a single data among multiple processes and agents. To ensure that situations of some multi-agents failing to agree or be unreliable by not being available for consensus to actively reach agreement, a consensus protocol must adopt mechanisms to make them flexible and fault-tolerant [9].

In [10] a decentralized multi-agent system achieved consensus using consensus problem to control these multi-agents. More than half of all the multi-agents and processes agreed by voting on the state and integrity of a process.

B. Blockchain Consensus Algorithm

Blockchain as a cryptocurrency framework for Bitcoin has evolved since its introduction. Blockchains are uniquely categorized based on the type of consensus algorithm in use. In a Proof-of-Work (PoW) consensus, agreement on the validity of a process was achieved using the computing power challenge. The node that had more computing power achieved consensus through a completely decentralized approach. There is Proof-of-Stake (PoS) consensus that is based on financial power competition where the node that controls more than a third of all the resources within the network gets to validate processes within the blockchain. PoS operates by selecting validators to authenticate transactions within the blockchain based on the quantity of the cryptocurrency holdings forming a stake by a node. The more stake a node possesses the higher the chance of being selected to validate transactions. In a PoS, less computational power is involved since it takes shorter time to reach consensus than in PoW. In Delegated Proof-of-Stake (DPoS), consensus is reached based on election and voting process to guard malicious usage and centralization of blockchain. In DPoS, less computing power and time are involved in achieving consensus [11].

A consensus mechanism must provide a trade-off between performance, fairness, and security. There is Proof-of-Activity (PoA) that is a fusion of PoW and PoS. The PoA operated on an economic phenomenon with the assumption of "Tragedy of the Commons" which described a situation where a limited resource for several agents could be ruined in situations where there is uncontrolled use [12].

C. Ripple Consensus Algorithm

The Ripple consensus algorithm is a permissioned blockchain consensus algorithm that requires access permission for nodes in the network because it is not publicly accessible, and operates in rounds using active nodes as servers. It adopts an approach of closing an active ledger updating session once a consensus is reached to store and maintain an identical state of the ledger on all active nodes. For each round within the ripple protocol consensus algorithm (RPCA) [13];

- End users of the server forwards all new transactions to each server. These valid transactions are compiled and made public in the form of a list to constitute the "candidate set".
- All the candidate sets from several servers are merged on every server's unique node list (UNL), to authenticate these transactions.
- The transactions that do not pass the authentication minimal percentage of "Yes" votes are either discarded or included in the candidate set at the commencement of the consensus process for the next ledger. Conversely, transactions that obtain the minimal

percentage of "yes" votes are passed onto the next round of consensus.

The minimum percentage of 80% of a server's UNL is a requirement for agreeing on a transaction and that constitutes the final round of consensus. The final round of transaction closes the ledger after appending the authenticated transactions onto the ledger.

1) Composition of the ripple consensus protocol: The Ripple Consensus Protocol consists of several components: Server, Ledger, Last-closed ledger, Open-ledger, Unique node list (UNL), Proposer.

The server is an entity that runs the Ripple server software.

Ledger is an append only record of the amount of currency in each user's account and represents the ground truth of the network. The ledger grows with updating transactions using the consensus protocol.

The last-closed ledger describes the most recent state of the ledger after the consensus protocol has validated transactions and appended the validated transaction onto the ledger.

The Open ledger is used to represent the current operating status of a ledger on a node.

2) Correctness of the consensus: There is the likelihood of a validating node being compromised to form a cartel of corrupt validating nodes to comprise the byzantine-fault-tolerance integrity of the consensus. The ripple consensus protocol maintains correctness for agreements and resistive to Byzantine failures by adopting a mechanism where a transaction is approved only when 80% of the validating nodes agree using the consensus algorithm. Dishonest agreement to validate a message transmission is possible only after the number of faulty validating nodes exceed 80% of the unique node list. The consensus protocol with honest nodes in the UNL will maintain correctness if the unique node list UNL of n nodes in the network meets this condition:

$$f \leq (n-1)\,/\,5$$

Where f is the number of Byzantine failures. In situations of $(n-1)/5 + 1$ Byzantine failures, the correctness of the consensus is maintained. The consensus will only confirm a fraudulent transaction when there are $(4n+1)/5$ failures or more. The probability of occurrence p^* hereafter, points to the likelihood of growing the size of the nefarious cartel below the maximal threshold of Byzantine failures.

$$p^* = \sum_{i=0}^{\lceil(\frac{n-1}{5})\rceil} \binom{n}{i} p_c^i\,(1-p_c)^{n-1}$$

Where p_c denotes the probability of any node colluding with other nefarious cartel.

In Fig. 1, the process for the Ripple consensus is outlined.

The validation nodes are IoT sink nodes from several local IoT networks.

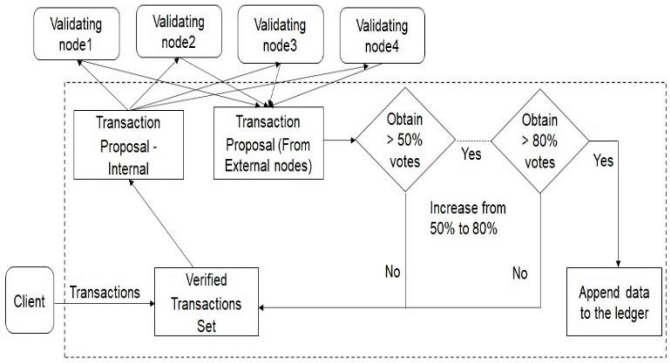

Fig. 1. Ripple consensus data flow diagram

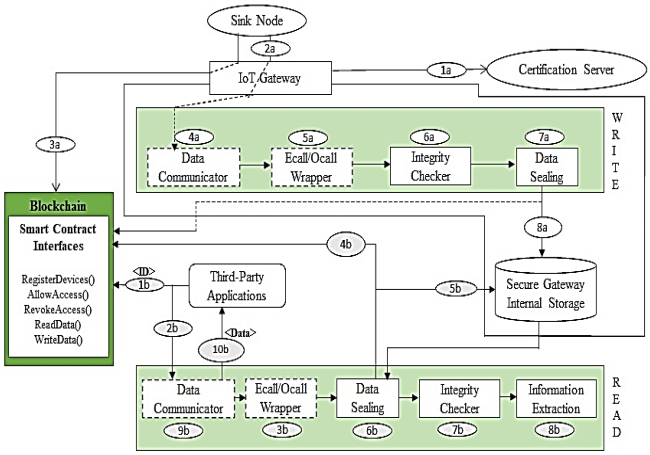

Fig. 2. Smart contract dataflow diagram

In Fig. 2, the smart contract data flow is presented. The smart contract operates between the sink node and the cloud network. The IoT gateway runs the blockchain smart contract. The smart contract ensures that users, devices, and data are verified and validated for data storage operations on the distributed ledger. Data from the sink node get stored onto the distributed ledger through the IoT gateway. The sink node registers itself on the blockchain in step 3a. The IoT gateway always verifies the state of the blockchain using the internal storage distributed ledger as input to validate the integrity of the blockchain using the cloud or remote. Connected sink nodes constitute the consensus nodes for performing user, device and data integrity checking before either writing onto the blockchain or access data from the blockchain. Data writing operations on the blockchain are done by the sink nodes, to append data onto the blockchain. They are referred to as data write operations in step 2a. The hash and encrypted data from the sink node are used in the next phase. The *writedata* function in the smart contract is used to append the hash of the sink node data onto the blockchain. The encrypted data is then written to the gateway internal memory in steps 4a-8a. The Ecall/Ocall wrapper communicates with the gateway internal memory as illustrated in the step 5a. The hash of the data from the sink node is verified by recalculating the hash-based message authentication code (HMAC) based on the encrypted and comparing the given hash with the derived hash. The Integrity Checker verifies and validates IoT data by ensuring that the given hash and the derived hash are the same,

the encrypted data is sealed and written to disk in step 7a. If the report from the Integrity Checker shows a difference in the string structure of the derived hash from the given hash, that will result in discarding the data including the hash from the sink node. Step 7a and Step 8a are used in validating the hash and proceeding to either write the encrypted data to disk or disproving the hash and discarding the data from the sink node.

Data accessing activities from the blockchain is done using the data read module. A user module first registers third-party users using the *allowAccess* method with the smart contract. The user calls the *revokeAccess* function to revoke access for a user. Step 1b outlines the interaction of the third-party user with the smart contract in obtaining the hash of the data generated by the sink node after providing the device ID of the sink node. The smart contract checks if the third-party user device ID and the address have the validation necessary to access the data after doing integrity checking for the third-party user ID and address. The hash of the sink node data is only returned from the cloud storage after the integrity checker grants the access permission to the third-party user to enable it to access the data from the IoT gateway persistent storage (IoT gateway internal memory) that represents local storage of the data. The smart contract uses the *READDATA* API as illustrated in step 4b, to confirm if the third-party user has the access permission to read the data hash identifier supplied by the third-party request. In step 5b, it illustrates how data is retrieved from the secured internal gateway storage once data access permission is granted. The data is unsealed in step 6b, and the integrity of the data is checked in step 7b, after recalculating and verifying the digital signature by comparing the given and the derived digital signatures. The sensor data stored in the gateway internal memory is read and returned by the user only after the digital signature verification is completed. Steps 9 and 10 illustrate the data flow for this operation [14].

In Table I, the pseudo-code for the smart contract is presented.

Where:

OwnerAddress: Sink node identity (SNId)

Device: Sensor

DeviceID: SensorID (SsId)

In [15] five main blockchain consensus protocols were examined using the unique properties of type (probabilistic or absolute finality), level of fault-tolerance, power consumption, scalability, and application. The five consensus protocols are: Proof-of-Work (PoW), Proof-of-Stake (PoS), Delegated Proof-of-Stake (DPoS), Practical Byzantine Fault Tolerance (PBFT) and Ripple.

The ripple consensus proved to have a good scalability, involved negligible power consumption, low fault tolerance and operated using permissioned application. In ripple consensus, the entire network will continue to function to support correct consensus even if 20% of the nodes are attacked by Byzantine generals problem [16].

TABLE I. SMART CONTRACT PSEUDOCODE

Algorithm: Smart Contract Pseudo-code

```
1: HashMap deviceRegistry(key:ownerAddress, value:List[deviceIDs])
2: HashMap deviceData(key:(ownerAddress, deviceID),
value:List[DataHash])
3: HashMap DataAccessRegistry(key:(ownerAddress,  thirdpartyAddress,
deviceID), value: bool isAllowed)
4: function REGISTERDEVICE(ownerAddress, deviceID)
5:   InsertToHashMap(key:ownerAddress, value:List[deviceIDs])
6: end function
7: function WRITEDATA(ownerAddress, deviceID, Data)
8:    if owner == ownerAddress
9:    deviceData([owner, deviceID], List.InsertData(hash(Data)))
10: end function
11: function READDATA(ownerAddress, thirdPartyAddress, deviceID)
12: if DataAccessRegistry(thirdPartyAddress) == true
13:    return deviceData[hash(ownerAddress, deviceID])
14: end function
15: function GRANTACCESS(ownerAddress, thirdPartyAddress,
deviceID)
16:    if owner == ownerAddress
17:    DataAccessRegistry[hash(ownerAddress, thirdPartyAddress,
deviceID)] = true
18: end function
19:    function REVOKEACCESS(ownerAddress, thirdPartyAddress,
deviceID)
20:    if owner == owner Address
21:    DataAccessRegistry[hash(ownerAddress, thirdPartyAddress,
deviceID)] = false
22: end function
```

D. Modeling Languages for Verification Systems

Modeling systems exist to provide opportunity for designing, developing, and implementing critical systems. Although there are several kinds of tools and platforms for modeling, simulation, and verification of systems, the coloured petri nets (CPN) is distinguishable in the following aspects: CPN offers several functions and provides a flexible manipulation of the functions in developing a model. The CPN tool has been improved and tested to support the modeling of complex systems [17].

Study [18] surveyed several modeling tools for checking, validating, and some cases improving the design requirements of systems. Notable modeling systems mentioned included the Practical Robust Implementation and Sustainability Model (PRISM), Numeric Symbolic Model Verifier (NuSMV), UPPAAL, Symbolic Analysis Laboratory (SAL), SPIN, Beryl, D-Finder.

The related works showed available research on IoT solutions that is based on blockchain. It however confirmed the absence of an implementation for a blockchain-based solution to authenticate and protect IoT data transmitted between actively connected network elements of the IoT gateway and the cloud.

Additionally, there is an implementation gap for a formal model for IoT systems that directly involved the sensor, sink node, IoT gateway, and the cloud elements in a blockchain-based IoT architecture.

In the next section, the methodology for the paper is presented.

III. METHODOLOGY

The target and design principles that formed the basis for the chosen methodology to support a blockchain-based consensus mechanism for authentication of node data for IoT systems are hinged on a security solution appropriate for an environment where the devices are resource-constrained.

The design principles for the methodology are the usage of decentralized authentication, smart contract for consensus among sink nodes, lightweight cryptographic solution, digital signature, smart contract with lightweight cryptographic function, a formal modeling tool that allows for dynamic behavior modeling, and the provision of a visual simulation tool.

The design goals on the other hand involved the elimination of a single point of failure, stronger security, extending data protection with a lightweight cryptographic solution, enforcing authentication with smart contract, the proposal of a formal model for a generic blockchain-based IoT solution, and the validation of a blockchain-based IoT solution through simulation.

A. Heterogeneous IoT Architecture

An architecture consisting of several subsystems was adopted for the implementation of the blockchain-based authentication mechanism.

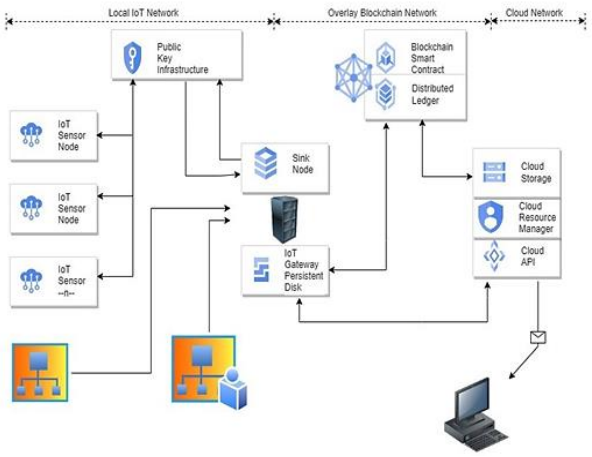

Fig. 3. An IoT architecture

In Fig. 3, an IoT architecture with components for a heterogeneous cyber-physical network is displayed. The architecture has three components involving a local IoT network which consisted of sensor end devices and sink nodes, an overlay network that employs blockchain-based distributed ledger, and a cloud network (remote storage) to receive and store the hash values of the sensor data. The local IoT networks amalgamate validated sink nodes with their validated data and transmit them through an IoT gateway to be stored on the distributed ledger [19].

The local IoT sub model is composed of elements that consist of security management, devices and sensors, internet connectivity things, Application Programming Interface (API) libraries, System Development Kits - SDK. Distributed systems that operate based on a decentralized authentication mechanism is prone to attacks such as the double spending instances where validated messages that represent independent transactions have the possibility of getting used in simultaneous transfers without considering the output of each transfer in the simultaneous transactions [20].

B. Coloured Petri Net

The coloured Petri Net (CPN) is a graphical mathematical modeling language. It is used to describe and check system properties, security requirements and synchronization characteristics for real-time distributed systems, and more generally event-driven systems. CPN comprises essential tools for analyzing boundedness, reachability, resource conflicts, deadlock as well as the structural properties of a real-time system [21].

The formalization of CPN is composed of nine tuples.

$$CPN = (P, T, A, \sum, V, C, G, E, I)$$

Where:

P = {P1, P2, …, Pm} represents a finite set of places.

T = {T1, T2, …, Tn} denotes a finite set of transitions.

A: Directed arc set

$\sum$: A finite set of colour set types

V: Denotes a finite set of variables whose type $\in \sum$.

C: It represents the colour set function from P to $\sum$.

G: Denotes the set of guard functions of transitions.

E: It represents a function that associates an arc expression to each transition.

I: denotes the function that gives the initial marking of each place.

The graphical representation of Petri net comprises of rings representing Places, rectangles denoting Transitions, arrows symbolizing Arcs.

A coloured Petri net is composed of variables, values, and expressions. CPN objects are described using colour domain that comprises variables, data values, operators, a syntax for expressions, and typing rules. An abstract colour domain consists of : Data values $\mathbb{D}$, Variables $\mathbb{V}$ and Expressions ($\mathbb{E}$) [22].

- $\mathbb{D}$ *is the set of data values*; These data values include integer values, Boolean values (True and False), and special undefined value $\perp$;
- $\mathbb{V}$ *is a set of variables*, that are represented using single letters $x, y, \ldots$, or as subscribed letters $x_1, y_k, \ldots$

- $\mathbb{E}$ *is the set of expressions*, that are composed of values, variables, and suitable operators.

Variables or values may form a basic expression. Thus, $D \cup V \subset E$. For example, let $e \in E$, the expression $vars(e)$ denotes the set of variables from V involved in e.

A *binding* is a restricted function $\beta : \mathbb{V} \to \mathbb{D}$. Let $e \in \mathbb{E}$ and β be a binding. $\beta(e)$ represents the evaluation of e under β; if the domain of β does not include $vars(e)$ then $\beta(e) \stackrel{def}{=} \perp$.

Both sets and multisets of expressions are subjected to binding evaluations.

For example, if $\beta \stackrel{def}{=} \{x \mapsto 1, y \longmapsto 2\}$, we have $\beta\,(x + y) = 3$. With $\beta \stackrel{def}{=} \{x \mapsto 1, y \mapsto 2\}$, according to the colour domain, we may have $\beta(x + y) = \perp$ (no coercion), or $\beta(x + y) =$ "12" (coercion of integer 1 to string "1"), or $\beta(x + y) = 3$ (coercion of string "2" to integer 2), or even other values as defined by the concrete colour domain.

Two expressions $e_1, e_2 \in \mathbb{E}$ are said to be equivalent which is represented as $e_1 \equiv e_2$, if for all possible binding β, the binding for both expressions are the same $\beta(e_1) = \beta(e_2)$. For example, $x + 1, 1 + x\ and\ 2 + x - 1$ are pairwaise equivalent expressions for the usual integer arithmetic.

Definition 1 (Petri nets). A Petri net is a tuple with several elements such as $(S, T, l\ )$ where:

- S *is the finite set of places*; S is also represented as P
- T, *disjoint from* S, *is the finite set of transitions*;
- l *is a labelling function such that*:

for all $s \in S, l(s) \subseteq$ $\mathbb{D}$ *is the type of* $s, i.e.,$ *the set of values that* s is allowed to carry,

for all $t \in T,\ l(t) \in \mathbb{E}$ *is the guard of* $t, i.e.,$ a condition for its execution,

for all $(x, y) \in$ $(S \times T) \cup (T \times S), l(x, y)$ *is a multiset over* $\mathbb{E}$ and defines the *arc* from x *toward* y.

Definition 2 (Markings and Sequential Semantics) Let $N \stackrel{def}{=} (S, T, l)$*be a Petri net.*

A marking M and N is a *function on S that maps each place* s to a *finite multiset over* $l(s)$ *representing the tokens held by* s.

A transition $t \in T$ is enabled at *a marking M and a binding* β, *which is denoted by* $M[t, \beta\rangle$, *iff the following conditions hold*:

- M *has enough tokens, i.e., for all* $s \in S, \beta\big(l(s, t)\big) \leq M$;
- *the guard is satisfied, i.e.,* $\beta\big(l(t)\big) =$ True
- *place types are respected, i.e., for all* $s \in S, \beta\big(l(t, s)\big)$ *is a multiset over* $l(s)$.

If $t \in T$ *is enabled at marking* M *and binding* β, *then* t *may fire and yield* a marking M'*defined for all* $s \in S$ *as* $M'(s) \stackrel{def}{=} M(s) - \beta\big(l(s, t)\big) + \beta\big(l(t, s)\big)$. *This is denoted by* $M[t, \beta\rangle\ M'$.

The marking graph *G of a Petri net marked with M is the smallest labelled graph* such that:

- M *is a node of* G
- *if* M'*is a node of* G *and* $M'[t, \beta\rangle M''$*is also a node of* G *and there is an arc in* G *from* M' *to* M'' *labelled by* (t, β).

The definition of marking graphs allows the addition of infinitely many arcs between two markings. If $M[t, \beta\rangle$, there might exist infinitely many other enabling bindings that differ from β only on variables not involved in t. Finally only firings $M[t, \beta\rangle$ such that the domain of β *is* $vars(t) \stackrel{def}{=} vars(l(t)) \cup \cup_{s \in S}(vars\big(l(s, t)\big) \cup vars\big(l(t, s)\big))$ is considered.

In the next section, the results and discussion for the proposed methodology is presented.

IV. Results and Discussion

The use of the mathematical and visualization features of CPN provided the set of mathematical foundations necessary for representing and describing the elements and security properties of the proposed solution that is based on the IoT architecture.

In this section, the simulation results for the blockchain consensus using CPN tools are presented and discussed.

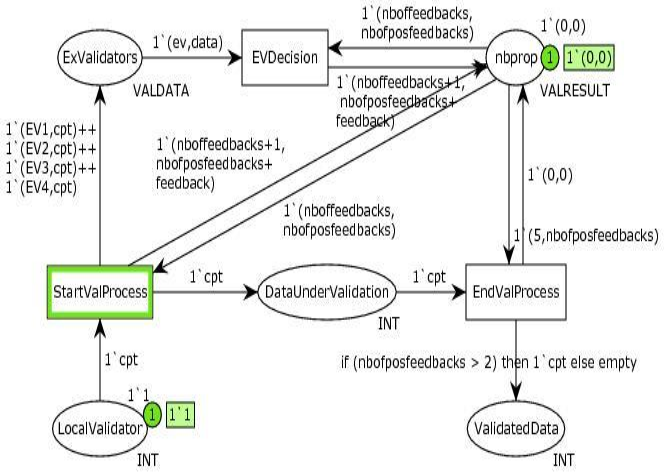

Fig. 4. CPN modeling of the blockchain consensus protocol

In Fig. 4, the CPN modeling of the blockchain consensus mechanism with validators is presented. The validators are an amalgamation of the current sink node which presents the data to be validated as well as other external sink nodes within the hierarchical IoT network. The external sink nodes form the external validators. The sink node whose data is to be validated through the consensus assumes the local validator status. The

validators (local and external) reach agreement to validate data if and only if the number of the positive feedbacks are more than half of all the total decisions from the validation voting by all the validators. Once a message does not get at least more than half of the total decisions to be positive feedbacks, that message is discarded. A session for the consensus by the validators is considered closed once the decision on a message has been made in accordance with the consensus correctness criterion of the adopted blockchain consensus. The correctness criterion of the consensus is critical to make the algorithm byzantine fault tolerant.

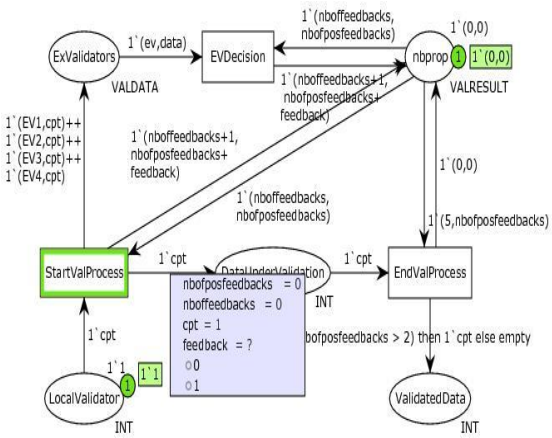

Fig. 5. Local validator feedback decision

In Fig. 5, the consensus commencement for validators is illustrated. The blockchain consensus starting with the local validator to decide by voting by on the feedback of the data under validation is presented. There are two feedback options (0, 1) to be selected by a validator. Option 1 symbolizes positive feedback whereas option 0 denotes non-positive feedback. The "start validation transition" has not been fired yet. Selecting a choice for the feedback will fire the transition. The token (1`1) on the local validator symbolizes a single node data and the specific data to be validated is 1. The update on the number of proposals "nbprop" of 1`(0, 0) shows that voting on the decision feedback on the data under validation has not started (0, 0).

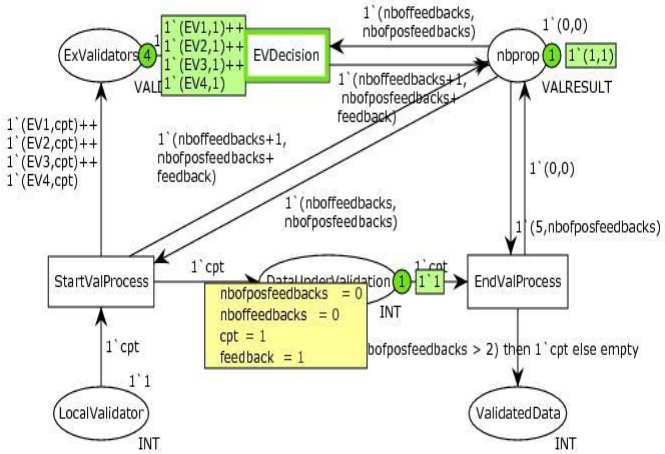

Fig. 6. Local validator feedback decision proposal update

In Fig. 6, a feedback decision of '1' on the data under validation is presented. The feedback from the local validator confirms that the "start validation transition" has been fired. The update on the number of proposals "nbprop" of 1`(1, 1) shows that voting has started on the decision feedback on the data under validation. That only 1 validator has voted on the decision feedback. That decision is a positive decision (1 – 'Number of decisions',1-'number of positive decisions').

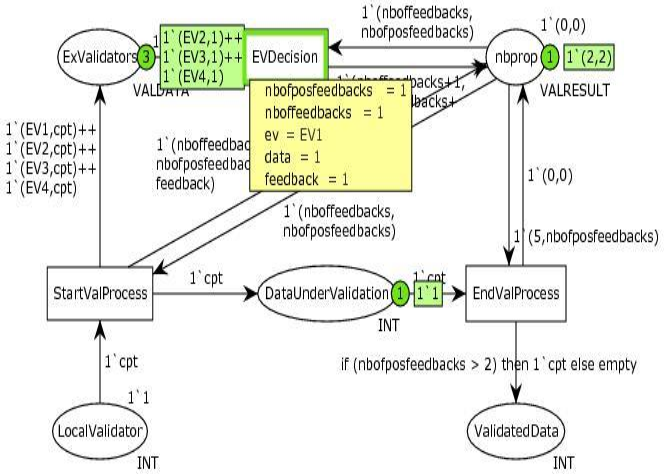

Fig. 7. External validator1 feedback decision

In Fig. 7, the feedback decision from external validator1 on the data under validation is illustrated.

The update on the nbprop place 1`(2,2) shows that there have been two voting decisions and all the decision are positive decisions.

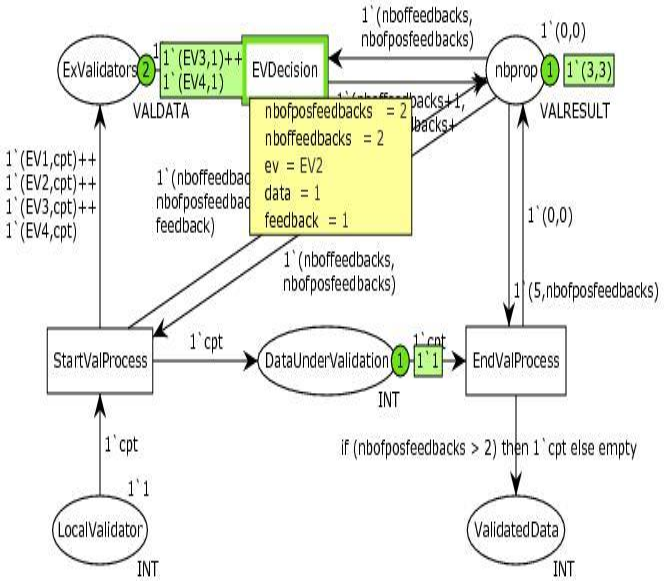

Fig. 8. External validator2 feedback decision proposal update

In Fig. 8, the feedback decision from external validator2 on the data under validation is shown.

The decision feedback voting update on the number of proposals "nbprop" – 1`(3,3) shows that there have been three

feedback decisions with all three being positive feedback decisions.

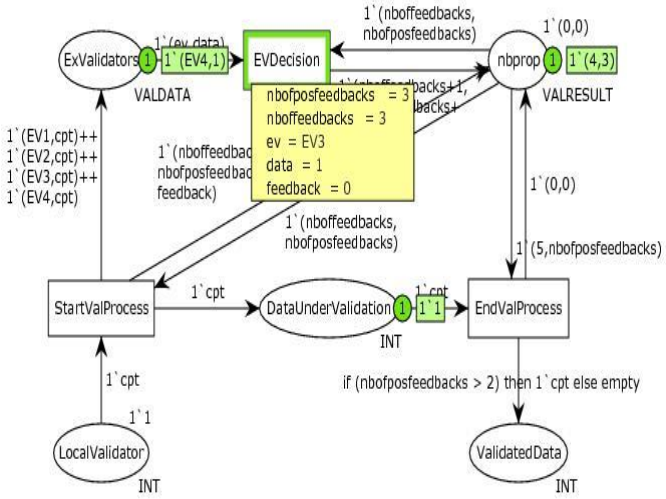

Fig. 9. External validator3 feedback decision proposal update

In Fig. 9, the feedback decision from external validator3 on the data under validation is illustrated. The local validator, external validators 1, 2, 3 have all voted on the decision and have the feedback updated and stored on the nbprop place. The token value has been updated to 1`(4,3) to show that there have been four votes (local validator, external validators 1,2,3). And that three out of the four votes are positive feedback decisions.

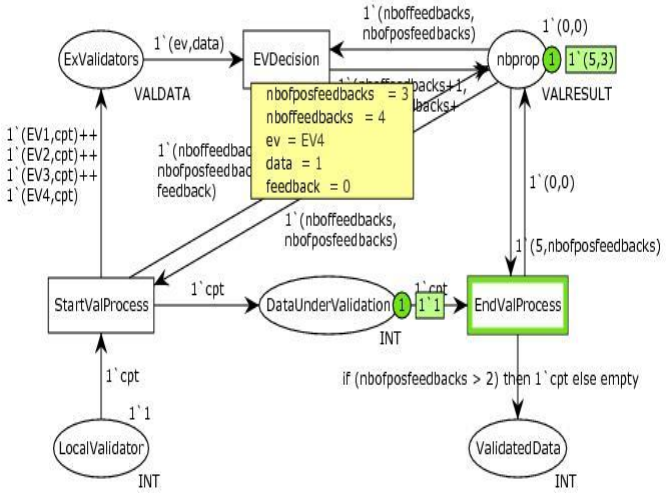

Fig. 10. External validator4 feedback decision proposal update

In Fig. 10, the feedback decision from external validator4 on the data under validation is depicted. Additionally, it provides the update as illustrated in the place for the number of proposals “nbprop” for a total of five decisions, with three positive feedback decisions. The EndValProcess transition is highlighted to show that it is the next action or step to be taken for the simulation.

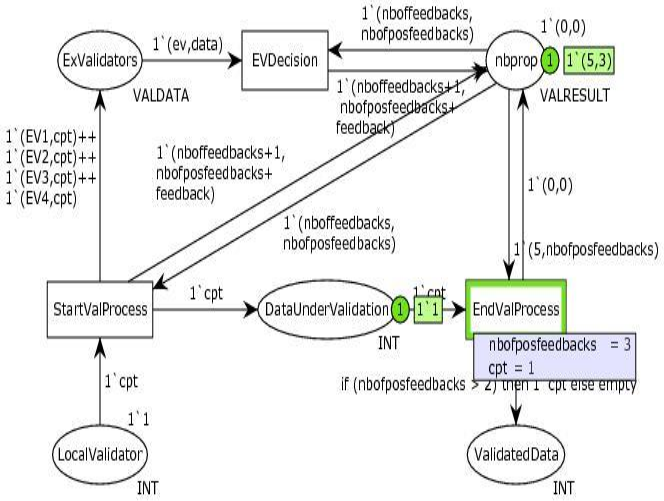

Fig. 11. Decision result after the proposals

In Fig. 11, the data flow CPN simulation on the decision feedback results at the end of the decision voting process is represented. The token value on the nbprop 1`(5,3) and the summary information on the transition confirm that there was one data identity that represented 1 data element to be validated and that there were 3 positive decision feedbacks.

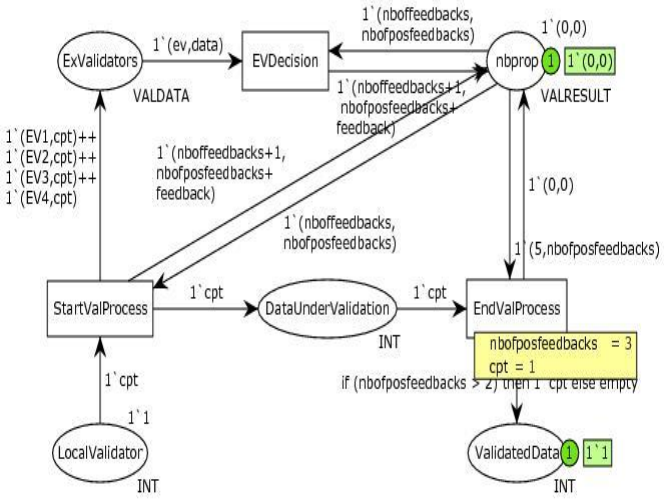

Fig. 12. Consensus decision

In Fig. 12, the decision on the data at the end of the consensus process is shown. The initial token element 1`1 on the local validator place has been moved to the place for the ValidatedData. Additionally, the consensus session is closed and the nbprop token element is reset to 1`(0,0).

The firing of the EndValProcess transition ends and session for the consensus activity. The data is then validated and the colour token 1`1 on the ValidatedData place finalizes the consensus.

TABLE II. DESCRIPTION OF CPN MODELING FEATURES FOR THE BLOCKCHAIN CONSENSUS MECHANISM

Abbreviation	Description	CPN Component
LocalValidator	It represents an internal sink node. The container for keeping a sink node data prior to a validation operation	Places
ExValidators	It denotes external sink nodes that form the external validators. These are all the other sink nodes within the hierarchical IoT network. They join the internal validator to reach an agreement on a message through a blockchain consensus.	
nbprop	A container for keeping all the decisions resulting from validators using the consensus rule to vote on a data under validation.	
ValidatedData	It represents the results after the voting decisions undertaken by all the validators have ended. When the number of positive feedbacks where at least 60% of the total decisions by the validators, the data will be moved to a new state of ValidatedData.	
DataUnderValidation	It denotes a place that specifies the current data being validated is kept. It is represented by the identity of the data which is captured as (`cpt) on the arc inscription.	
StartValProcess	It is an event that signifies the start of the consensus session. The local validator is an input to this event. It fires the data from the local validator to the external validators as well as updates the DataUnderValidation and the "nbprop" places.	Transitions
EVDecision	It is a transition label for the CPN event that fires the decision of each external validator as feedback on a data. The input of the transition is the external validator and the data to be validated based on the consensus rules. The output for this transition is the number of decision feedbacks and the number of positive decision feedbacks.	
EndValProcess	It is a transition to signify the close of a consensus session. It has DataUnderValidation, and nbprop as input. The output of this transition is the ValidatedData.	
cpt	A token for describing the identity of data under validation. The data under validation is submitted by the local validator to the validators where the consensus mechanism is applied on the data using other established rules in the consensus to vote on the data in validating it.	Variables
EVi (i = 1 .. 4)	The token identifying the external validator i.	
data	A data element representing the data under validation by the external validator.	
nboffeedbacks	It is a counter that records the decisions of voting activities by providing an update on the total number of feedback decisions	
nbofposfeedbacks	It is a counter that records the total number of positive feedback decisions. Both the number of feedbacks and the number of positive feedback decisions are stored as a token in the "nbprop" place and are updated each time an external validator decision is taken.	

In Table II, the CPN simulation components for the consensus mechanism for the system are presented. The components for the simulation consisted of CPN places, transitions, arc expressions, and initial marking of places using coloured tokens. The various components used in the simulation of the consensus mechanism were described in the table.

The use of the proposed IoT architecture is an improvement on a related work that used blockchain mechanisms for IoT data security. In [19] the blockchain solution did not indicate how the node data from the sensor was protected as well as an approach to maintain the integrity of the data communicated between the sensor and the sink node. The proposed blockchain-based IoT architecture used a centralized approach with a lightweight-cryptographic mechanism to protect the content of data between the sensor and the sink node. Additionally, the use of a non-monetary-based blockchain consensus mechanism where only the IoT gateway and other sink node clusters formed the consensus nodes is used to implement a smart contract with a lightweight cryptographic function for decentralized authentication of node data.

The use of the decentralized consensus ensured the elimination of a single point of failure situation for the IoT network and supported a distributed ledger that guaranteed the availability of validated node data on the IoT internal storage and the cloud for authorized users in the IoT system.

In the next section, the general conclusion of the work is presented.

V. CONCLUSION

Distributed systems that rely on decentralized processing for authentication and validation of processes like the blockchain system use agreements through consensus mechanisms to assure and maintain the correctness of decisions, and to guarantee stable systems. For a blockchain mechanism to be deployed in an IoT network where the devices are resource-constrained, an architecture was designed that factored in the challenges regarding memory, computational processing, and energy limitations of sensors and sink nodes. The distinctive security features in the proposed consensus mechanism enabled the core elements of the IoT architecture to reach an agreement among the sparsely fragmented network elements in the IoT architecture. The use of the IoT gateway, PKI, and cloud network in the architecture supported a security solution that provided a trade-off between performance, fairness in load balance, and security.

Additionally, the ripple consensus mechanism provided a byzantine fault-tolerant approach with good scalability to support a large network consisting of several subsystems.

The modeling and simulation of the blockchain-based authentication mechanism provided the possibility of validating the security properties of an IoT security solution that is based on a decentralized authentication approach. The CPN features including places, transitions, arcs, expressions, and initial markings of places with tokens were used to represent the entities or physical attributes of the system as well as the design decisions of the system. The design decisions and the dynamic nature expectations of the distributed ledger system were represented using places, transitions, arc expressions, and tokens.

REFERENCES

[1] M. Keerthika and D. Shanmugapriya, "Wireless Sensor Networks: Active and Passive attacks-Vulnerabilities and Countermeasures," Global Transitions Proceedings, vol. 2, no. 2, pp. 362–367, 2021.

[2] W. Fu, X. Wei, and S. Tong, "An improved blockchain consensus algorithm based on raft," Arabian Journal for Science and Engineering, vol. 46, no. 9, pp. 8137–8149, 2021.

[3] I. Lee, "Cybersecurity: Risk management framework and investment cost analysis," Business Horizons, vol. 64, no. 5, pp. 659–671, 2021.

[4] H. S. Lallie et al., "Cyber security in the age of COVID-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic," Computers & Security, vol. 105, p. 102248, 2021.

[5] L. Dong, H. Xu, X. Wei, and X. Hu, "Security correction control of stochastic cyber–physical systems subject to false data injection attacks with heterogeneous effects," ISA transactions, vol. 123, pp. 1–13, 2022.

[6] B. T. Asare, K. Quist-Aphetsi, L. Nana, and G. Simpson, "A nodal Authentication IoT Data Model for Heterogeneous Connected Sensor Nodes Within a Blockchain Network," in 2021 International Conference on Cyber Security and Internet of Things (ICSIoT), 2021, pp. 65–71.

[7] S. Zhang and J.-H. Lee, "Analysis of the main consensus protocols of blockchain," ICT express, vol. 6, no. 2, pp. 93–97, 2020.

[8] E. Zaghloul, T. Li, M. W. Mutka, and J. Ren, "Bitcoin and blockchain: Security and privacy," IEEE Internet of Things Journal, vol. 7, no. 10, pp. 10288–10313, 2020.

[9] X. Wang, H. Su, X. Wang, and G. Chen, "Fully distributed event-triggered semiglobal consensus of multi-agent systems with input saturation," IEEE Transactions on Industrial Electronics, vol. 64, no. 6, pp. 5055–5064, 2016.

[10] X. Wang, G.-P. Jiang, H. Su, and Z. Zeng, "Consensus-based distributed reduced-order observer design for LTI systems," IEEE Transactions on Cybernetics, 2020.

[11] J. Zhang and M. Wu, "Blockchain-Based Authentication with Optional Privacy Preservation for Internet of Vehicles," Mathematical Problems in Engineering, vol. 2021, 2021.

[12] D. Wang, C. Jin, H. Li, and M. Perkowski, "Proof of Activity Consensus Algorithm Based on Credit Reward Mechanism," in Web Information Systems and Applications: 17th International Conference, WISA 2020, Guangzhou, China, September 23–25, 2020, Proceedings, Berlin, Heidelberg, 2020, pp. 618–628. doi: 10.1007/978-3-030-60029-7_55.

[13] D. Schwartz, N. Youngs, and A. Britto, The ripple protocol consensus algorithm. Ripple Labs Inc.(2014). 2021.

[14] G. Ayoade, V. Karande, L. Khan, and K. Hamlen, "Decentralized IoT data management using blockchain and trusted execution environment," in 2018 IEEE International Conference on Information Reuse and Integration (IRI), 2018, pp. 15–22.

[15] S. Zhang and J.-H. Lee, "Analysis of the main consensus protocols of blockchain," ICT Express, vol. 6, no. 2, pp. 93–97, Jun. 2020, doi: 10.1016/j.icte.2019.08.001.

[16] B. Chase and E. MacBrough, "Analysis of the XRP Ledger Consensus Protocol." arXiv, Feb. 20, 2018. Accessed: Jun. 02, 2022. [Online]. Available: http://arxiv.org/abs/1802.07242

[17] H. Kaid, A. Al-Ahmari, Z. Li, and R. Davidrajuh, "Single controller-based colored Petri nets for deadlock control in automated manufacturing systems," Processes, vol. 8, no. 1, p. 21, 2019.

[18] M. Luckcuck, M. Farrell, L. A. Dennis, C. Dixon, and M. Fisher, "Formal specification and verification of autonomous robotic systems: A survey," ACM Computing Surveys (CSUR), vol. 52, no. 5, pp. 1–41, 2019.

[19] G. Wang, Z. Shi, M. Nixon, and S. Han, "ChainSplitter: Towards Blockchain-Based Industrial IoT Architecture for Supporting Hierarchical Storage," 2019 IEEE International Conference on Blockchain (Blockchain), pp. 166–175, 2020, doi: 10.1109/blockchain.2019.00030.

[20] U. W. Chohan, "The double spending problem and cryptocurrencies," Available at SSRN 3090174, 2021.

[21] W. Duo, H. Xin, and M. Xiaofeng, "Formal Analysis of Smart Contract Based on Colored Petri Nets," IEEE Intell. Syst., vol. 35, no. 3, pp. 19–30, May 2020, doi: 10.1109/MIS.2020.2977594.

[22] C. Gaucherel, C. Carpentier, I. R. Geijzendorffer, C. Noûs, and F. Pommereau, "Discrete-event models for conservation assessment of integrated ecosystems," Ecological Informatics, vol. 61, p. 101205, 2021.