IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 May 2024, accepted 31 May 2024, date of publication 7 June 2024, date of current version 18 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3411388

METHODS

Development of a Customized Blockchain Technology-Based Parameter Matching Protocol for Semiconductor Manufacturing

MYOUNG SOO CHOI[1], **JUMYUNG UM**[2], **AND SANG-SEOK LEE**[1,3], **(Senior Member, IEEE)**

[1]Graduate School of Engineering, Tottori University, Tottori 680-8552, Japan
[2]Department of Industrial and Management System Engineering, Kyung Hee University, Yongin, Gyeonggi-do 17104, Republic of Korea
[3]Advanced Mechanical and Electronic System Research Center, Faculty of Engineering, Tottori University, Tottori 680-8552, Japan

Corresponding author: Sang-Seok Lee (sslee@tottori-u.ac.jp)

ABSTRACT This paper introduces a novel solution to mitigate the vulnerability of operational parameter tampering in the semiconductor industry through a customized blockchain protocol. Faced with high costs and rapid market demands, the industry often struggles with production efficiency due to the variability in equipment operators' skills and human errors. This challenge is compounded by the necessity for continuous parameter updates and collaboration with equipment suppliers. Our proposed solution leverages the security strengths of blockchain technology, renowned for its secure, immutable record-keeping capabilities. We focus on developing a specialized blockchain-based system, the blockchain-based parameter matching (BBPM) protocol, to manage and safeguard operational parameters within semiconductor manufacturing. This system is designed to ensure secure, tamper-proof information updates, addressing the industry's need for reliable and efficient parameter management. The BBPM protocol addresses the dual risks of unauthorized parameter modifications and unintended changes by authorized users, ensuring the integrity of operational parameters crucial for quality control in semiconductor processes. This protocol represents a transformative shift in operational parameter management systems from traditional centralized models to a distributed approach. By adopting a decentralized and tamper-resistant framework, the BBPM protocol can enhance the efficiency and quality of operational parameter management while reducing the need for extensive data storage, a common challenge in centralized systems. This shift marks a critical advancement in information management and security within the semiconductor manufacturing.

INDEX TERMS Blockchain, operational parameter, parameter matching, protocol, semiconductor equipment.

I. INTRODUCTION

Currently, the semiconductor industry faces high costs and time-to-market pressure [1]. However, the production improvement is often hindered due to equipment operator's skill such as human error proofing [2], [3], personal experiences, or implicit knowledge [4]. Moreover, it is challenging to iteratively update production parameters (e.g., recipes, equipment operational parameters—hereafter referred to as 'operational parameters') required to stabilize or enhance equipment performance without unintended errors. To assist in this, many equipment suppliers have started to provide online support services through edge-cloud computing to help customers prevent accidental faults. For instance, CNC controllers are directly connected to cloud service platforms, a setup that is already applied to machining centers and industrial robots [5]. The cloud-based collaboration platform operates under the assumption that all participants either

The associate editor coordinating the review of this manuscript and approving it for publication was Mueen Uddin.

agree to share information or provide data that is less critical to their business. In the realm of semiconductor manufacturing, not only are the requirements for direct data exchanges increasing both within and beyond the factory integration space, but information security is also becoming an increasingly significant challenge in this domain [6].

Blockchain technology stands out as a solution for secure information sharing [7]. Practically, blockchain technology is acknowledged for its application in various domains, including cloud computing [8], supply chain management [9], vehicular networks [10], digital healthcare [11], [12], and financial transactions [13]. As widely recognized, secure information sharing is imperative in high-quality manufacturing industries for purposes such as quality control [14], process planning [15], and part traceability [16]. As a method of implementing blockchain technology, there are research cases that propose an edge computing platform integrating the data ledger with equipment controllers. For example, Kumar et al. proposed the blockchain-edge framework [17], [18]. Isaja et al. also showed the implementations of the edge ledger in truck quality control lines and plug-and-play modular factories [19], [20]. These widespread applications emphasize blockchain's revolutionary potential to enhance traditional systems in sectors ranging from finance to manufacturing.

Despite its broad potential, studies focused on deploying blockchain solutions for information exchange between semiconductor equipment remain notably scarce. Moreover, the challenges associated with integrating blockchain technology into the semiconductor manufacturing supply chain are anticipated to persist beyond 2025, as noted in recent forecasts [21]. This situation underscores the urgent need for focused research into the applicability and adaptability of blockchain in this critical field. Given this backdrop, this paper seeks to address the following pivotal questions:

(1) What are the technical and practical requirements for implementing a blockchain-based operational parameter matching protocol in semiconductor manufacturing?
(2) How can blockchain technology be effectively customized to decentralize the operational parameter matching system in semiconductor manufacturing?

The primary objective of this paper is to bridge the existing gap in the application of blockchain technology in the semiconductor manufacturing industry. This study is centered on the development and implementation of the blockchain-based parameter matching (BBPM) protocol. The protocol is designed to adopt blockchain technology, facilitating a transformative shift from traditional centralized systems to a distributed operational parameter matching system. By introducing this protocol, we aim to demonstrate a viable application of blockchain technology within the semiconductor manufacturing domain.

The remainder of this paper is organized as follows: Section II provides the background and rationale specific to the semiconductor manufacturing domain for the BBPM protocol. Section III addresses the technical modifications required to adapt blockchain technology specifically for the BBPM protocol. Section IV details the design and implementation of the BBPM protocol. Section V demonstrates and analyzes the practicality of the BBPM protocol. Finally, the conclusions of this study are summarized in section VI.

II. BACKGROUND AND RATIONALE

To understand the BBPM protocol in terms of the semiconductor manufacturing supply chain, it is very informative to refer to 'Factory Integration' in the International Roadmap for Devices and Systems (IRDS). In the context of the 'Factory Integration' functional area, the BBPM protocol corresponds to production equipment (PE), which includes run-to-run (R2R) control, fault detection and classification (FDC), statistical process control (SPC), fault prediction (FP), and equipment health monitoring (EHM). The interests of PE include improving wafer process quality, verifying equipment functionality, and limiting utilities and electricity consumption, among other objectives [22]. In the hierarchical layer structure, the BBPM protocol corresponds to the supervisory control and data acquisition (SCADA) level. The structure comprises four levels: equipment sensor (Level 1), SCADA (Level 2), manufacturing execution system (MES) (Level 3), and enterprise resource planning (ERP) (Level 4). Applications of PE are situated at the SCADA level [23]. Our proposal contributes to the semiconductor manufacturing supply chain by pioneering the instantiation of a blockchain application within it. A deep discussion of the semiconductor manufacturing supply chain is beyond the scope of this paper. For more details, see [24].

To detail the BBPM's application environment specific to semiconductor manufacturing, this section discusses the following four topics.

A. UNDERSTANDING THE DISTINCT ROLES OF OPERATIONAL PARAMETERS

Regarding semiconductor equipment, wafer processing recipes define various processing variables essential for achieving the desired quality at each wafer processing step. Conversely, operational parameters establish settings that not only ensure equipment safety but also optimally configure the equipment according to the specific wafer processing recipes assigned. For example, in the plasma etch process, while wafer recipes specify the RF power values, operational parameters set the RF power limits. These limits are then used to verify the RF power values during the execution of the etch process. If these RF power limits are set incorrectly, the etch process might proceed without triggering alarms, even when the RF power values exceed established safety thresholds. Such oversight could lead to severe issues, compromising both equipment safety and the quality of wafer processing.

Additionally, as further examples, scanning electron microscopes include such operational parameters as accelerating voltage, collection solid angle, beam current, and spot size [25], [26], [27]. Brightness, contrast, and secondary electron intensity are prominent parameters to calibrate the operational state of helium ion microscopes [28], [29]. Ion-beam microscopes use beam voltage and the pressure of vacuum electrons [30]. These parameters, distinguished from process recipes, significantly influence the equipment's performance.

Meanwhile, providing operational parameter settings to the equipment supplier for troubleshooting enables the supplier to conduct tests at their site, mirroring the conditions under which the semiconductor company's equipment operated. This provides a clear advantage in quickly identifying and addressing the underlying causes of issues. However, sharing operational parameter settings with equipment suppliers is challenging due to the inclusion of the semiconductor company's manufacturing know-how and process knowledge [31].

B. UNDERSTANDING OPERATIONAL PARAMETER MANIPULATION RISK

In the semiconductor manufacturing, the occurrence of tampered operational parameters can be broadly classified into two main pathways. The first scenario involves the manipulation of the operation parameter file, which serves as the ledger for operational parameters in the equipment. Some equipment allows editing of the operation parameter file using text editors or Microsoft Excel without permission checks. Careless alteration of the content of such files results in the loading of a corrupted ledger. The second scenario involves the occurrence of tampered operational parameters by authorized users. Upon completion of loading the operational parameter file depicted in Fig. 1, editing of the operational parameters stored in the equipment controller's memory can be generally performed through the equipment's graphical user interface (GUI). To edit operational parameters via the GUI, a permission verification process similar to that required for recipe editing at the superuser level is necessary. However, instances of tampered operational parameters can occur even at the hands of authorized users. This is not due to malicious intent but rather stems from situations where users with modification permissions make temporary alterations out of necessity and inadvertently fail to restore the original settings. Fig. 2 illustrates the typical scenario in which these situations occur.

C. RATIONALE FOR INTRODUCING THE BBPM PROTOCOL

In the event of wafer processing defects in the equipment, comparing the operational parameters of the affected equipment (hereafter referred to as suspected-OP) with those of equipment operating without defects (hereafter referred to as authentic-OP) facilitates the verification of their integrity. Suspected-OP denotes the operational parameters of equipment suspected to have process defects, whereas authentic-OP denotes the operational parameters of equipment confirmed to be operating without defects. However, manually comparing the two OPs without the aid of an automated comparison system is time-consuming and challenges the accuracy of verification. Therefore, semiconductor manufacturers often establish an in-house information system (hereafter referred to as the OP-system) to manage these operational parameters.

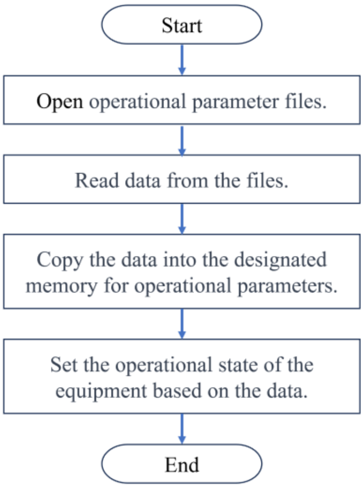

Start
Open operational parameter files.
Read data from the files.
Copy the data into the designated memory for operational parameters.
Set the operational state of the equipment based on the data.
End

FIGURE 1. Loading procedure for operational parameters.

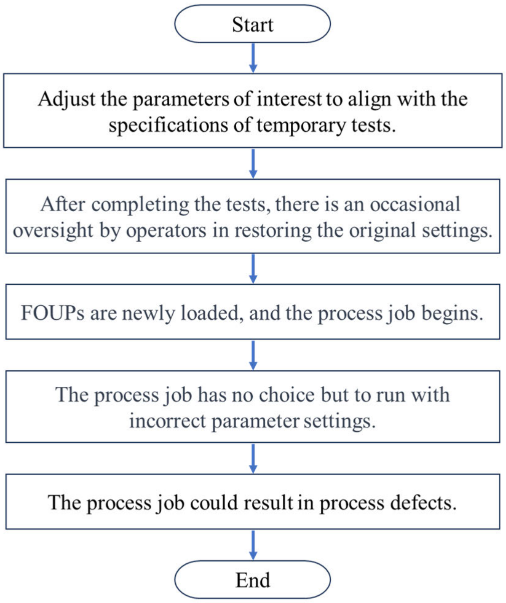

Start
Adjust the parameters of interest to align with the specifications of temporary tests.
After completing the tests, there is an occasional oversight by operators in restoring the original settings.
FOUPs are newly loaded, and the process job begins.
The process job has no choice but to run with incorrect parameter settings.
The process job could result in process defects.
End

FIGURE 2. A typical case of authorized users unintentionally altering operational parameters.

The OP-system must, at a minimum, provide functionality for the creation and maintenance of the operational parameter ledger (OP-ledger) and the detection of suspected-OP. The decision to implement the OP-ledger using a centralized or distributed ledger framework depends on the preferences and circumstances of the development entity. However, considering that semiconductor manufacturing is a complex process involving many pieces of equipment [32], choosing a centralized ledger may require a significant amount of data storage.

On the other hand, semiconductor equipment, as depicted in Fig. 1, already allocates the necessary memory space for operational parameter. Therefore, implementing the OP-ledger in an edge ledger format, distributed to the equipment controllers, can eliminate the need for additional data storage investment. Nonetheless, the implementation of the OP-system in a distributed ledger framework requires the existence of a protocol machine capable of creating, maintaining, and comparing distributed ledgers.

In the domain of semiconductor manufacturing, distributed edge computing solutions are being suggested by equipment manufacturers mainly for the purpose of advanced process control (APC). Despite these advancements, it is widely recognized within the domain that such systems are not yet meeting the needs of semiconductor manufacturers. This gap has led to a decrease in trust in the edge computing approaches of semiconductor manufacturers. Moreover, the necessity to develop innovative protocols presents a significant challenge, adding considerable overhead for the manufacturers. This backdrop motivates the introduction of the BBPM protocol in this paper.

D. DISTINCT FEATURES AS SOLUTIONS IN SEMICONDUCTOR MANUFACTURING

Blockchain is a technology that enables the creation of tamper-resistant digital ledgers in a distributed manner, thereby eliminating the need for a central authority [33], [34]. This capability is particularly crucial for the distributed OP-ledger in the BBPM network, which comprises semiconductor equipment equipped with the BBPM protocol, each piece acting as a node in the BBPM network.

As a solution for semiconductor manufacturing, the BBPM protocol treats operational parameters, rather than cryptocurrency, as its assets. Therefore, the data structure for the BBPM protocol's transactions and blocks must be modified to accommodate these assets. Additionally, the pursuit of block mining within this protocol is not driven by the aspiration to attain economic value. Instead, its primary objective is to distribute the role of a tentative supervisor among the nodes, specifically to the mining winner. The winning node takes on the responsibility of updating the BBPM's blockchain with the newly mined block. Consequently, block mining in the BBPM protocol does not need to be competitive.

III. CUSTOMIZING BLOCKCHAIN COMPONENTS

In this section, we detail the customization of key blockchain components—transactions, blocks, and the proof of work mechanism—to specifically align with the BBPM protocol's objectives. Our aim is to tailor these foundational elements to uniquely cater to the BBPM protocol's requirements, thereby effectively supporting its operational goals. The necessary customizations are driven by the following specific needs:

(1) Operational parameters management: The BBPM protocol handles operational parameters, which are used to configure semiconductor equipment, diverging from typical blockchain applications that focus on cryptocurrencies.

(2) Does not pursue rewards: Unlike typical blockchain applications, the BBPM protocol does not need to reward processing transactions or mining blocks.

(3) Avoiding excessive computing power: The BBPM protocol should prevent excessive computational demands so as not to compromise the wafer processing performance of semiconductor equipment equipped with the BBPM protocol.

A. TRANSACTION

Blockchain technology is commonly used for cryptocurrencies [35]. A cryptocurrency transaction typically consists of inputs and outputs. The inputs are usually a list of digital assets that users can transfer to other users in the blockchain network. The outputs are usually the recipient digital asset accounts, which show how much digital assets will be received [36]. As mentioned above, the BBPM protocol treats operational parameters as its digital assets. These operational parameters usually specify conditions under which semiconductor equipment is expected to operate, and they are typically represented in the form of 'tag = value.' A tag indicates the meaning of the parameter, and the value specifies the state of the parameter. For example, 'heater_temp_upper_limit = 250' indicates that the upper limit temperature of the heater is set to 250°C. This implies that if the heater temperature exceeds 250°C, the semiconductor equipment with an overheated heater may behave unexpectedly.

A BBPM transaction signifies a change in the state of a tag, and does not require the input-output concept necessary in transactions dealing with cryptocurrency. Therefore, the data structure of a BBPM transaction is designed to include a list of 'tag = value' to represent the tag to be changed and its new state.

In blockchain system, transactions must be digitally signed for authentication [37]. This is an essential feature of blockchain systems for the prevention of malicious transactions. Given the impossibility of guaranteeing complete security of the semiconductor manufacturing network from external cyber attacks, it is imperative for the BBPM protocol, functioning within this network, to implement robust measures against such threats. Should the BBPM data be compromised as a result of an external cyber attack, it could lead to significant disruptions in the wafer processing operations. Therefore, the blockchain functionality for verifying the validity and authenticity of transactions is also necessary for the BBPM protocol.

B. BLOCK

Bitcoin is the most prominent example inherently associated with blockchain technology [38]. In Bitcoin, blocks are mined to create new coins or to execute pending transactions. The miners receive two types of rewards for their efforts: (1) new coins that are generated with each new block, and (2) the transaction fees for all the transactions included in that block [39]. Simply speaking, this reward model is essential for maintaining the Bitcoin system. It motivates participants to adhere to blockchain protocols, which necessitate considerable computing power. When reconfiguration of the operational parameters of semiconductor equipment is necessary, transactions encapsulating these updates are generated and subsequently broadcast to the BBPM network.

Each node within the network captures these incoming transactions in a memory pool. The transactions accumulated in this pool are subsequently executed through blocks, in a manner analogous to the Bitcoin system. However, a notable distinction in the BBPM protocol is that the blocks are exclusively dedicated to processing transactions, as there is no coin creation involved.

Another significant difference lies in the data structure used for managing transactions. In Bitcoin, the block utilizes a Merkle tree to encapsulate the summary of all transactions. This Merkle tree structure is particularly effective in handling significant increases in transaction volumes, as the Merkle path required to validate the inclusion of a specific transaction grows at a much slower rate [40]. In contrast, the BBPM block adopts a different approach for managing transactions.

Compared to Bitcoin, the BBPM protocol is anticipated to experience a significantly low volume of transactions. This is based on transaction activity and the number of nodes within the BBPM network. The transaction activity is not greedy, because BBPM transactions are typically required for accidental examinations of BBPM nodes, yet sometimes also for planned maintenance. A single BBPM network does not encompass all equipment in a semiconductor factory, but is instead segmented according to identical models from the same supplier, such as those operating at the same wafer processing step. This segmentation results in a limited number of nodes per network. In fact, it is even feasible for a BBPM network to be comprised of as few as a single node. Such a configuration allows for extreme flexibility in adapting to the specific needs and scales of different semiconductor manufacturing environments.

Consequently, in light of the low volume of transactions, the BBPM block is uniquely designed. They are structured to contain arrays of transactions, referred to as 'List of TRs,' which differs from Bitcoin's approach of summarizing all transactions. This design results in simplifying blockchain technology, aligning well with our objective to enhance its applicability and adaptability within the semiconductor manufacturing domain.

C. PROOF OF WORK

As previously discussed, our aim for block mining in the BBPM protocol is to decentralize the supervisory role by distributing it among the nodes that win the mining process. This strategy is driven by the dynamic nature of semiconductor manufacturing, where equipment deployment layouts change to accommodate new production plans or to maximize equipment uptime.

Blockchain technology inherently operates without centralized intermediaries, such as supervisory nodes, by selecting various leaders through consensus mechanisms [41]. Among these mechanisms, the proof of work (PoW) has emerged as a foundational consensus mechanism, largely due to its successful implementation in the Bitcoin blockchain protocol. The PoW is widely acknowledged as a decentralized consensus mechanism that probabilistically selects mining winners [42], [43], [44], [45], [46], [47]. Given these attributes, the PoW has been chosen as the consensus mechanism for the BBPM.

However, Bitcoin's PoW is well-known to have issues with power consumption [48], [49], [50] and scalability [51], [52], [53]. To address these issues, researchers are actively investigating alternative solutions [54], [55], [56], [57], [58]. Nevertheless, the operating environment of the BBPM protocol, characterized by the low transaction volume features introduced in the preceding B. Block—specifically, non-greedy transaction activity and a limited number of nodes within a single BBPM network—is anticipated to significantly mitigate the likelihood of scalability issues becoming severe.

It remains critical, however, to ensure that the computational power expended in the PoW hash puzzle resolution process does not adversely affect wafer processing [59]. This necessitates the implementation of measures to prevent the use of excessive computing resources.

In the PoW process, nodes compete to solve a hash puzzle, with the winner being granted the right to create a new block on the blockchain. The essence of this task is to find a hash value that begins with a specified number of leading zero bytes [60], a requirement that is adjusted based on level of difficulty. A winning solution is characterized by a hash that is less than or equal to a target value, which is also determined by the current difficulty level. Hence, the computational power expended in the PoW process can be effectively managed by adjusting this difficulty level.

Consequently, it is imperative to calibrate the difficulty level of the BBPM protocol's PoW mechanism to an optimal value. This calibration is supposed to both prevent the overconsumption of computational resources in the hash solving tasks of BBPM nodes and enable a non-deterministic distribution of the supervisory role among them. By doing so, it ensures that the process of solving hashes does not interfere with or cause damage to the wafer processing operations within the BBPM protocol.

IV. DESIGN AND IMPLEMENTATION

The BBPM protocol is structured into two layers: the kernel as the lower layer and the application as the upper layer. The kernel layer encompasses processes such as network joining, transaction propagation, and block mining, while the application layer is responsible for the matching process. We refer to the software implemented to execute these layers as a 'protocol-machine.'

In this section, to embody the 'protocol-machine,' we detail the algorithmic sequences for the layers' processes, the protocol data units (PDUs) for handshaking between BBPM nodes, and the data structures required by the PDUs. Additionally, to evaluate our design, we implemented the simulation software, SimBBPM. This tool was developed in C++ within

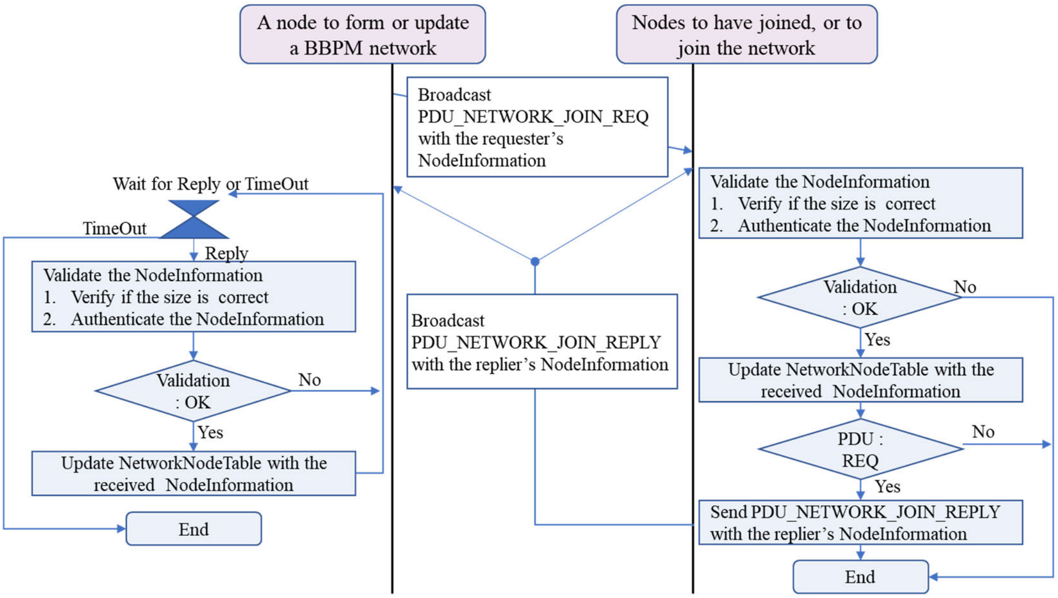

A node to form or update a BBPM network
Nodes to have joined, or to join the network
Broadcast PDU_NETWORK_JOIN_REQ with the requester's NodeInformation
Wait for Reply or TimeOut
TimeOut
Reply
Validate the NodeInformation
1. Verify if the size is correct
2. Authenticate the NodeInformation
Validation : OK
No
Yes
Update NetworkNodeTable with the received NodeInformation
End
Broadcast PDU_NETWORK_JOIN_REPLY with the replier's NodeInformation
Validate the NodeInformation
1. Verify if the size is correct
2. Authenticate the NodeInformation
Validation : OK
No
Yes
Update NetworkNodeTable with the received NodeInformation
PDU : REQ
No
Yes
Send PDU_NETWORK_JOIN_REPLY with the replier's NodeInformation
End

FIGURE 3. Description for network joining process.

a Windows 10 environment and incorporates OpenSSL to manage cryptographic hash functions.

A. JOINING THE BBPM NETWORK

In the BBPM network, the semiconductor equipment functions as a node within the BBPM network. Consequently, any equipment intending to utilize the proposed protocol is required to join the BBPM network. Broadcasting the 'PDU_NETWORK_JOIN_REQ' either facilitates the construction of a BBPM network or enables a new node to join an existing network. When nodes receive this PDU, they register the 'NodeInformation' conveyed by the PDU in their 'NetworkNodeTable.' Subsequently, they broadcast a 'PDU_NETWORK_JOIN_REPLY.' Upon receiving this reply, the recipient nodes register the 'NodeInformation' from the reply in their 'NetworkNodeTable.' As a result, either a new BBPM network is formed, or new nodes are added to the existing BBPM network.

In the BBPM network, nodes have the capability to dynamically join or leave the network, responding to their operational needs. In order to succinctly convey complex information and avoid lengthy explanations, this study employs Table and Figure. Table 1 delineates the data structure of the 'NodeInformation.' Fig. 3 visually illustrates the network joining process, including the PDU handshake and the algorithmic sequence involved.

The BBPM protocol implements the authentication of 'NodeInformation' through the following sequence of steps:

(1) Initially, the sender computes the hash value of the NodeInformation data field intended for transmission.
(2) This hash value is then encrypted using the sender's private key, establishing a secure digital signature.
(3) Subsequently, the encrypted hash value, constituting the digital signature, is transmitted to the recipients.

TABLE 1. Nodeinformation data structure.

Field		Description
Header	Size	Size of the NodeInformation data structure
	pduType	Show if pduType is Request or Reply.
	PublicKey	The sender's public key generated by OpenSSL API.
	DigitalSignature	The sender produces a hash of the data field in NodeInformation and encrypts the hash with the sender's private key. This is the encrypted hash.
Data	EqID	This is a unique number to identify the node sending the NodeInformation.

(4) Upon receipt, all recipients independently recalculate the hash value of the received data field to verify its integrity.
(5) The recipients then decrypt the received digital signature using the sender's public key.
(6) Finally, the recipients compare the recalculated hash value (from step 4) with the decrypted hash value (from step 5).

A match in this comparison confirms that the NodeInformation is authentic and unaltered, thus ensuring the integrity and trustworthiness of the information within the BBPM network.

B. GENERATING TRANSACTION AND AUTHENTICATION

The BBPM protocol features the 'EVT_CONFIG_VAL_UPDATE' event as a user interface between the protocol itself and operators of equipment equipped with the BBPM protocol. When operators reconfigure the operational parameters and wish to secure these changes in the BBPM OP-ledger, they can notify the BBPM protocol by activating 'EVT_CONFIG_VAL_UPDATE.'

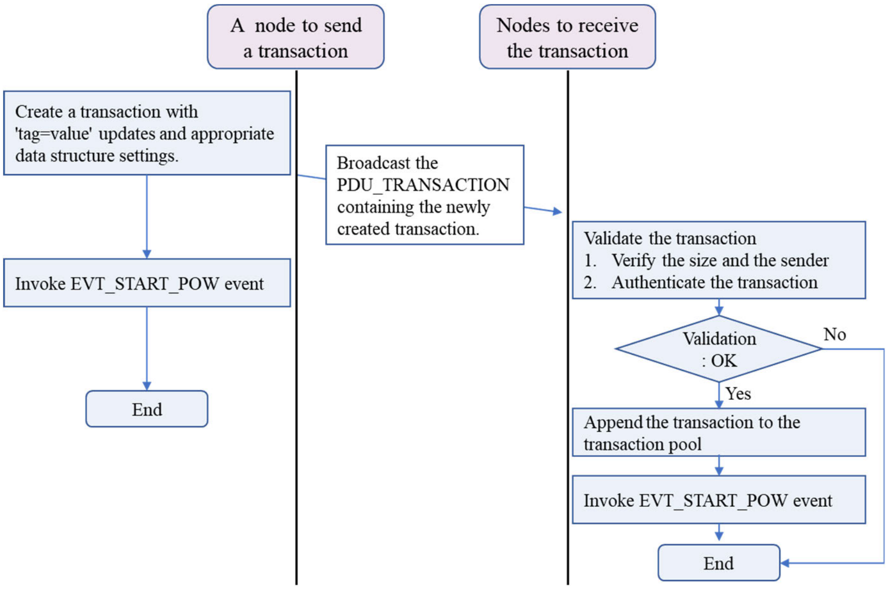

A node to send a transaction
Nodes to receive the transaction
Create a transaction with 'tag=value' updates and appropriate data structure settings.
Broadcast the PDU_TRANSACTION containing the newly created transaction.
Invoke EVT_START_POW event
End
Validate the transaction
1. Verify the size and the sender
2. Authenticate the transaction
Validation : OK
No
Yes
Append the transaction to the transaction pool
Invoke EVT_START_POW event
End

FIGURE 4. Description for transaction propagation.

TABLE 2. Transaction data structure.

Field		Description
Header	Size	Size of Transaction data structure.
	EqID	This is is a unique number to identify the node sending a Transaction.
	CfgID	This is currently reserved for future usage to align with extending functions of OPCODE.
	DigitalSignature	The sender produces the hash of the data field of the transaction and encrypts the hash with the sender's private key. This is the encrypted hash.
Data	UpdateCount	The total number of updates in the transaction.
	ListOfTag	Tags to be updated.
	ListOfValue	New values of the tags to be updated.

Activating 'EVT_CONFIG_VAL_UPDATE' prompts the BBPM protocol to generate and broadcast transactions across the BBPM network in the form of a 'PDU_TRANSACTION.' Table 2 delineates the data structure of the transaction. Fig. 4 visually elucidates the mechanism of 'PDU_TRANSACTION,' illustrating how it executes the propagation of the changes and intermediately maintains the transactions for subsequent processes.

The BBPM protocol implements transaction authentication through the following sequence of steps:

(1) Initially, the sender computes the hash value of the transaction's data field intended for transmission.
(2) This hash value is then encrypted using the sender's private key, creating a secure digital signature.
(3) The encrypted hash value, constituting the digital signature, is transmitted to the recipients.
(4) Upon receipt, all recipients independently recalculate the hash value of the received data field to verify its integrity.

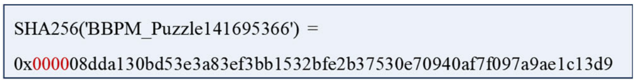

SHA256('BBPM_Puzzle141695366') =
0x00000008dda130bd53e3a83ef3bb1532bfe2b37530e70940af7f097a9ae1c13d9

FIGURE 5. An example solution for the hash puzzle.

(5) The recipients then decrypt the received digital signature using the sender's public key, which is retrieved from the 'NetworkNodeTable,' as detailed in Fig. 3.
(6) Finally, the recipients compare the recalculated hash value (from step 4) with the decrypted hash value (from step 5) to verify the authenticity of the transaction.

A transaction within the BBPM protocol is considered authenticated only if the hash value comparison, as outlined in step (6), results in a match. Additionally, the 'EVT_START_POW' event, which is depicted in Fig. 4, is discussed in the subsection C. 'Mining Process.'

C. MINING PROCESS

In the BBPM network, all nodes act as miners, ensuring decentralized operation of the BBPM protocol. As previously mentioned, this protocol utilizes the proof of work (PoW) consensus mechanism. It is crucial to calibrate the difficulty level of the PoW to an optimal value. This optimal value is especially significant to maintain a balance between effective decentralization and the prevention of excessive computing power needed to solve the hash puzzle. Such an approach makes the BBPM protocol practicable, ensuring that it does not compromise the wafer processing performance of semiconductor equipment equipped with the BBPM protocol. The protocol uses the fixed string 'BBPM_Puzzle' and a variable nonce to solve the puzzle, as illustrated in Fig. 5.

In the BBPM protocol, the mining process is initiated by the 'EVT_START_POW' event, which can be activated in different ways:

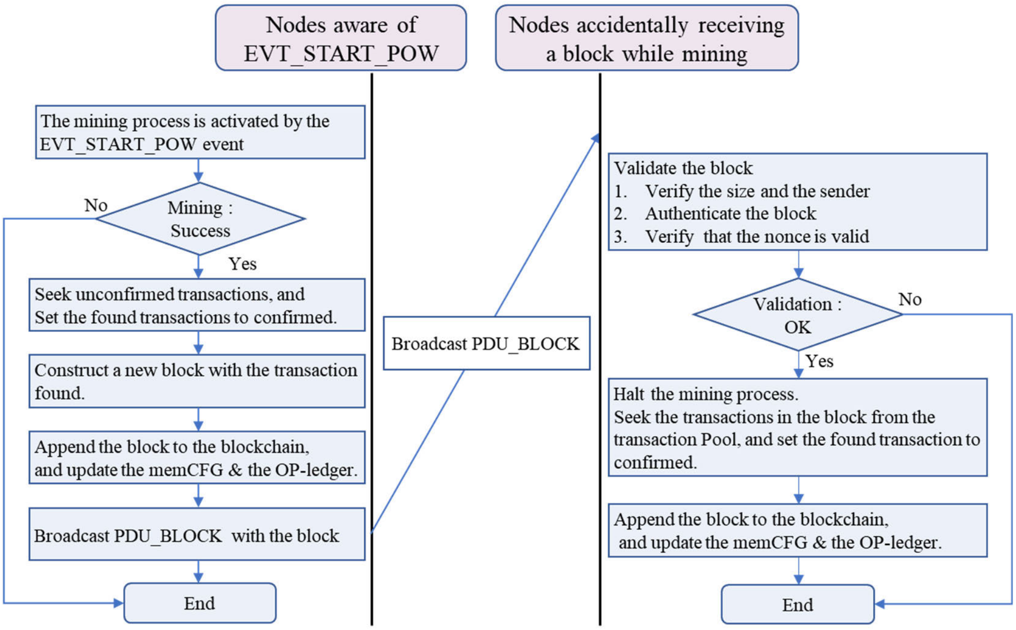

Nodes aware of EVT_START_POW
Nodes accidentally receiving a block while mining
The mining process is activated by the EVT_START_POW event
Mining : Success
No
Yes
Seek unconfirmed transactions, and Set the found transactions to confirmed.
Construct a new block with the transaction found.
Append the block to the blockchain, and update the memCFG & the OP-ledger.
Broadcast PDU_BLOCK with the block
End
Broadcast PDU_BLOCK
Validate the block
1. Verify the size and the sender
2. Authenticate the block
3. Verify that the nonce is valid
Validation : OK
No
Yes
Halt the mining process. Seek the transactions in the block from the transaction Pool, and set the found transaction to confirmed.
Append the block to the blockchain, and update the memCFG & the OP-ledger.
End

FIGURE 6. Description for mining process.

TABLE 3. Block data structure.

Field		Description
Header	Size	Size of Block data structure
	preBlockHash	SHA256 hash of the previous block's header.
	EqID	This is a unique number to identify the miner.
	timeStamp	This is the time when the miner solved the hash puzzle.
	nonce	An arbitrary number used when calculating a hash of the hash puzzle. The miner changes the nonce repeatedly.
	OPCODE	Specify the method for updating the OP-ledger. In this paper, it is set to the default value: OP_FULL_HASH.
	DigitalSignature	The miner produces a hash of the data field of the block, and encrypts the hash with the miner's private key. This is the encrypted hash.
	numberOfTransaction	This is the total number of transactions in the block.
	dataHash	SHA256 hash of the data field of the block.
Data	ListOfTransactions	All transactions in the block.

(1) Automatically activated by 'PDU_TRANSACTION.'
(2) Periodic activation via a timer-based task.
(3) Manual activation through a user interface.

In Fig. 4, the 'EVT_START_POW' event is shown as being triggered by 'PDU_TRANSACTION.' Choosing different activation methods offers practical flexibility in a real-world environment and does not alter the essence of the BBPM protocol.

Once 'EVT_START_POW' is activated, BBPM nodes begin mining a block. The node that succeeds in finding the puzzle solution creates a new block, appends the block to the BBPM blockchain, and broadcasts the block across the BBPM network. Should a node receive a block during its mining process, it validates the received block. If the block proves authentic, the node halts its mining process immediately and appends the block to the BBPM blockchain. The BBPM protocol distributes and synchronizes the BBPM blockchain across each node in the BBPM network in its realm. In subsection B 'Integral Funtionality Check,' in section V 'Practicality Evaluation,' we evaluate the synchronization. The block data structure and the mining process are detailed in Table 3 and Fig. 6, respectively.

Validating blocks includes both block authentication and nonce verification. The block authentication employs the same principles and procedures as those used in transaction authentication, detailed in the subsection B. 'Generating Transactions and Authentication.' The nonce verification conducts the following steps to confirm whether the received block came from the mining winner or not:
(1) Generate hash values using SHA256, taking a string that combines 'BBPM_Puzzle' and the nonce as an input.
(2) Verify that the first two leading bytes of the hash value are zeros.

When a new block is appended to the BBPM blockchain, 'memCFG' is updated based on the transactions contained within the block. In semiconductor equipment, operational parameters are typically stored as text files, known as configuration files. Upon initial start-up, the BBPM protocol loads all configuration files into a dedicated memory storage, referred to as 'memCFG.' Subsequently, the protocol interacts exclusively with 'memCFG' rather than directly with the configuration files. Additionally, the BBPM protocol allows equipment controllers to access 'memCFG' when configuring the operational parameters of the equipment. This design aims to maintain coherence in operational parameters between the protocol and the equipment controllers.

Additionally, each block within the BBPM blockchain is linked to a dedicated OP-ledger, containing a unique hash

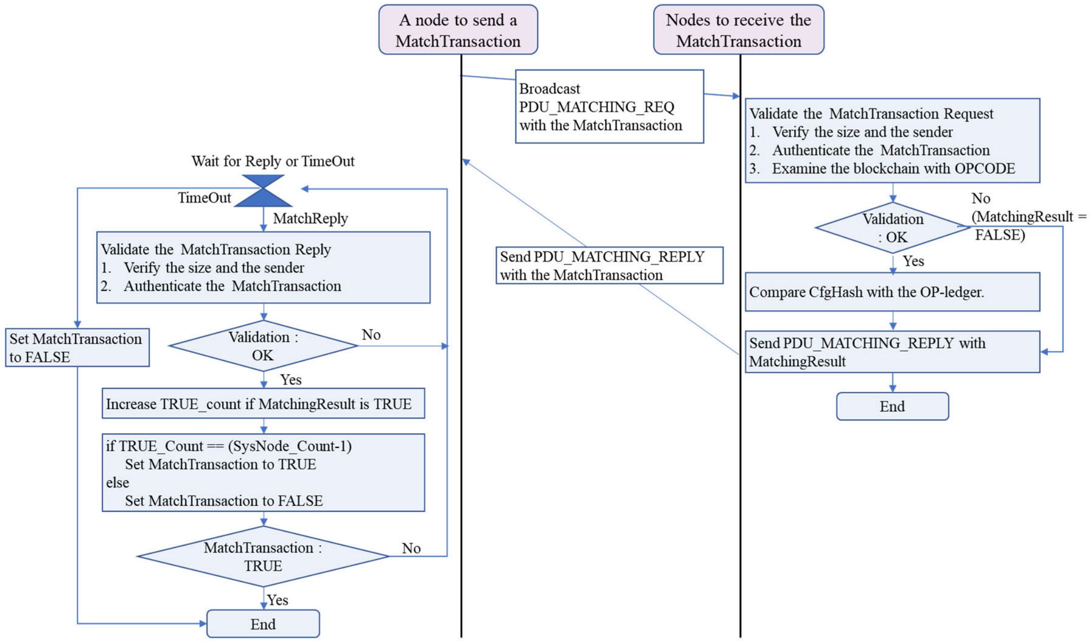

A node to send a MatchTransaction
Nodes to receive the MatchTransaction
Broadcast PDU_MATCHING_REQ with the MatchTransaction
Validate the MatchTransaction Request
1. Verify the size and the sender
2. Authenticate the MatchTransaction
3. Examine the blockchain with OPCODE
Validation : OK
No (MatchingResult = FALSE)
Yes
Compare CfgHash with the OP-ledger.
Send PDU_MATCHING_REPLY with MatchingResult
End
Send PDU_MATCHING_REPLY with the MatchTransaction
Wait for Reply or TimeOut
TimeOut
MatchReply
Validate the MatchTransaction Reply
1. Verify the size and the sender
2. Authenticate the MatchTransaction
Set MatchTransaction to FALSE
Validation : OK
No
Yes
Increase TRUE_count if MatchingResult is TRUE
if TRUE_Count == (SysNode_Count-1)
Set MatchTransaction to TRUE
else
Set MatchTransaction to FALSE
MatchTransaction : TRUE
No
Yes
End

FIGURE 7. Description for matching process.

TABLE 4. Matchtransaction data structure.

Field		Description
Header	Size	Size of MatchTransaction data structure.
	pduType	Show if pduType is MP_REQ or MP_REPLY.
	RequestEqID	A node's identifier to send MP_REQ.
	ReplyEqID	A node's identifier to send MP_REPLY.
	OPCODE	Selector to identify blocks with the OP-ledger that need to be matched. In this paper, it is set to the default value: OP_FULL_HASH.
	DigitalSignature	The sender produces the hash of the data field of the MatchTransaction and encrypts the hash with the sender's private key. This is the encrypted hash.
	MatchingResult	This is the result of ParaMatching performed according to OPCODE.
Data	CfgHash	This is the SHA256 hash calculated based on the OPCODE. The MatchRequester calculates the CfgHash using parameters of the memCFG. MatchRepliers do not generate CfgHash.

value. This hash is generated from parameters stored in 'memCFG,' guided by a specific 'OPCODE' included in that block. The 'OPCODE' is capable of not only determining which parameters from 'memCFG' to include but also specifying the necessary preprocessing algorithms before hashing. It is purposefully designed to provide a broad range of options for selecting conditions under which operational parameters are matched. In this study, we utilize an 'OPCODE' named 'OP_FULL_HASH,' which directs the BBPM protocol to calculate the OP-ledger's hash value using all available parameters in 'memCFG.' The exploration and enhancement of OPCODE options is a critical area for future research.

D. MATCHING PROCESS

Having covered the design and implementation of processes within the kernel layer of the BBPM protocol, we will now elaborate on the design and implementation of the matching process within the application layer. This matching process, supported by underlying kernel processes, performs the operational parameter matching, which is the primary application objective of the BBPM protocol.

For the matching process, the BBPM protocol utilizes two PDUs: 'PDU_MATCHING_REQ' and 'PDU_MATCHING_REPLY,' abbreviated respectively as 'MP_REQ' and 'MP_REPLY' for brevity and ease of reference throughout this paper. Additionally, the protocol employs a special transaction type, 'MatchTransaction,' which is integral to the matching process.

The data structure of 'MatchTransaction' is detailed in Table 4, providing an in-depth view of its composition. The aforementioned PDUs are encoded using the 'MatchTransaction' data structure, but there is a difference in the encoding process between 'MP_REQ' and 'MP_REPLY.' The CfgHash for 'MP_REQ' is calculated using 'memCFG,' whereas for 'MP_REPLY', this calculation is not necessary. With CfgHash, regardless of the volume of operational parameters to be verified, the parameters are encrypted into a SHA256 hash, providing the benefit of confining the data size of 'MP_REQ.' The principles and procedures for 'MatchTransaction' authentication are the same as those for transaction or block authentication, as mentioned in the corresponding former section.

Next, we will provide an in-depth exploration of the behaviors of nodes involved in the matching process. For the convenience of explanation, the node that sends 'MP_REQ' and then waits for 'MP_REPLY' will be referred to as a 'MatchRequester,' and the node that receives 'MP_REQ' and then sends 'MP_REPLY' will be referred to as a 'MatchReplier.' The 'MatchReplier' verifies whether the operational parameters sent by the 'MatchRequester' are identical to the contents of the OP-ledger in the BBPM blockchain. Upon completing this verification, the 'MatchReplier' then returns the results to the 'MatchRequester.' To carry out this task, the 'MatchReplier' must locate the correct OP-ledger within the BBPM blockchain. This ledger should include the same 'OPCODE' as the one sent by the 'MatchRequester.' To accurately locate the correct OP-ledger, the 'MatchReplier' examines the BBPM blockchain to identify blocks containing the same 'OPCODE' as that of the 'MP_REQ.'

As detailed in the subsection C. 'Mining Process,' each block in the BBPM blockchain is linked to its own OP-ledger. In instances where multiple blocks are detected, the block closest to the tip of the blockchain is selected. The 'MatchReplier' then compares MP_REQ's CfgHash with the hash value of the selected OP-ledger and then sends the 'MP_REPLY' to the 'MatchRequester' with the comparison result, referred to as 'MatchingResult.' In this process, the MP_REQ's CfgHash represents the operational parameters of the 'MatchRequester,' while the hash value of the selected OP-ledger represents those parameters as securely maintained by the BBPM blockchain.

The 'MatchRequester' tallies the number of TRUE 'MatchingResult' responses received through 'MP_REPLY.' When the count of TRUE 'MatchingResult' responses reaches a predetermined threshold, the 'MatchRequester' then determines that the 'MatchTransaction' is TRUE, indicating that there are no abnormal operational parameters. In this study, the threshold is set at 'SysNode_Count – 1,' where 'SysNode_Count' represents the total number of nodes participating in the BBPM network. In practical applications, the threshold for declaring a TRUE 'MatchTransaction' can be adjusted according to the specific usage environment and decided by the user to best fit their operational needs.

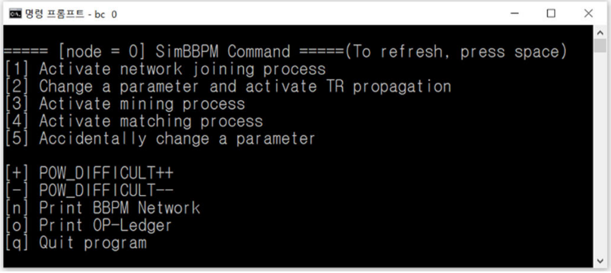
The content elucidated regarding the matching process is effectively illustrated in Fig. 7, which depicts the complex steps in a flowchart format. This visual representation simplifies the comprehension of the intricate details involved in the matching process, facilitating easier understanding.

V. PRACTICALITY EVALUATION

A. METHODOLOGY OVERVIEW

To evaluate the BBPM Protocol's practicality, this section demonstrates and analyzes the protocol's behavior using the simulation software SimBBPM, as introduced in section IV. The SimBBPM simulations have been conducted on a single computer with an Intel CPU @ 2.50 GHz, 8 GB RAM, and Windows 10.

SimBBPM was specifically designed to instantiate multiple independent console applications on a single computer. Each application emulates a BBPM node, which is a piece of semiconductor equipment equipped with the BBPM protocol-machine, the software implemented to execute the BBPM protocol. The console application features a straightforward text-based user interface, where pressing a specific number activates the corresponding process of the 'protocol-machine.' A screenshot of the console application is displayed in Fig. 8. This evaluation is configured as follows:



```
===== [node = 0] SimBBPM Command =====(To refresh, press space)
[1] Activate network joining process
[2] Change a parameter and activate TR propagation
[3] Activate mining process
[4] Activate matching process
[5] Accidentally change a parameter

[+] POW_DIFFICULT++
[-] POW_DIFFICULT--
[n] Print BBPM Network
[o] Print OP-Ledger
[q] Quit program
```

FIGURE 8. Screenshot of the console application.

- Integral functionality check: We assess the integral functionality of the BBPM protocol by sequentially running the BBPM processes as detailed in section IV and analyzing the outcomes at each step. This evaluation aims to verify the algorithmic sequences of the processes, ensure proper handshaking of the PDUs, validate the data structures of 'tag = value' pairs in transactions and 'List of TRs' in blocks, and assess the impact of the difficulty level on the PoW mechanism.
- Authentication and security: This evaluation focuses on the BBPM protocol's security measures, specifically their effectiveness in preventing unauthorized access. We simulate scenarios in which unregistered nodes attempt to tamper with the OP-ledger in the BBPM blockchain by broadcasting transactions.
- Parameter matching capability: We verify that the matching process accurately identifies 'MatchingResult' as TRUE or FALSE in the corresponding scenarios. Additionally, within the BBPM Network, we observe the impact on other nodes when a specific node transitions to the 'MatchingResult False' state, and we analyze which characteristics of the BBPM protocol contribute to these outcomes.
- Investigation of implementation performance: Although the performance challenges of the BBPM protocol are beyond the scope of this paper, we measure the trend in elapsed time of the matching process as the number of nodes increases. This measurement helps to identify how different implementations of the BBPM protocol, exemplified by SimBBPM, influence its performance. By analyzing potential causes of these results, we lay the foundation for future research on performance issues.

B. INTEGRAL FUNCTIONALITY CHECK

For this evaluation, we instantiated three BBPM nodes, each assigned a pair of (Node ID, EqID) values: (0, 100), (1, 200), and (2, 300), respectively.

1) NETWORK JOINING PROCESS

Building a BBPM network with these three nodes was achieved by pressing '1' in the SimBBPM interface. After the SimBBPM had completed building the network, we proceeded to investigate the 'NetworkNodeTable' for each of the three nodes within our simulation.

We found that each table had successfully registered the equipment IDs (EqIDs) 100, 200, and 300, indicating that all three nodes were effectively part of the BBPM network. This confirms that the network joining process, as depicted in Fig. 3, operates as envisioned in our study.

Here is a detailed account of how each node in the BBPM network identified its participants: The node with EqID 100 identified the nodes with EqIDs 200 and 300 through the NodeInformation delivered by 'PDU_NETWORK_JOIN_REPLY.' In contrast, the node with EqID 200 identified EqID 100 using 'PDU_NETWORK_JOIN_REQ' and identified EqID 300 via 'PDU_NETWORK_JOIN_REPLY.' Meanwhile, the node with EqID 300 recognized EqID 100 through 'PDU_NETWORK_JOIN_REQ' and identified EqID 200 using 'PDU_NETWORK_JOIN_REPLY.'

2) TRANSACTION PROPAGATION

Having one node change an operational parameter and broadcasting a transaction encapsulating this change to the BBPM network, accomplished by pressing '2' on the SimBBPM.

For this test, SimBBPM provides a test-purpose OP-ledger containing ten 'tag = value' pairs, ranging sequentially from 'tag(0) = default(0)' to 'tag(9) = default(9).' Node 0 created and broadcast a transaction to the BBPM network to store the new setting of tag(0) in the BBPM blockchain, simulating a scenario where Node 0's user intentionally tries to change the operational condition of their semiconductor equipment. In a similar manner, Node 1 and Node 2 each executed the same actions for their respective tags, tag(1) and tag(2).

Now, the BBPM blockchain includes three new transactions, each containing the updated settings for their respective tags. These transactions are about to be executed through the mining process. This means that the OP-ledger still maintains its default settings because the transactions have not yet been executed.

Investigating the OP-ledger after executing the mining process will allow us to discern the functional results of transaction propagation depicted in Fig. 4.

3) MINING PROCESS

Securing the changes described in the transaction propagation into the BBPM OP-ledger is achieved by initiating the mining process, accomplished by pressing '3' on the SimBBPM.

For this simulation, we opted for manual activation of the mining process to investigate the OP-ledger after the issuance of the transaction and before the execution of the mining process.

By investigating the OP-ledger, we were able to verify that it successfully transitioned from its default settings to the updated settings following the mining process. Furthermore, the most notable outcome is the synchronization of the OP-ledger across nodes in the BBPM network. These outcomes indicate that transactions with a 'tag = value' structure and blocks with 'List of TRs' structure are functioning as intended.

Consequently, this confirms that the network joining process, transaction propagation, mining process, PDU handshake control, and modifications to data structures are efficiently integrated and function in precise alignment with the comprehensive design of our study.

4) OBSERVATION OF DIFFICULTY ADJUSTMENT

This observes the behavior of the BBPM PoW based on the heuristic calibration of the difficulty level. The quantitative results of this observation are constrained by the computing power of the computer used in the simulation, yet the qualitative trends can be deemed reliable.

For computing power consumption, this study investigates the relationship between different difficulty levels and their impact on computing power of the BBPM PoW. The times observed to find a winning hash under varying conditions are summarized as follows:

- Difficulty Level 1:
  Finding a hash that starts with '0 × 00' in the first byte required a time range from 60 to 1,500 milliseconds to complete the PoW.
- Difficulty Level 2:
  For a hash starting with '0 × 0000' in the first two bytes, the time to determine a mining winner ranged from 1,000 to 3,000 milliseconds.
- Difficulty Level 3:
  When the task was to find a hash beginning with '0 × 000000' in the first three bytes, no mining winner could be identified within a time frame of up to 30 minutes, indicating a significant increase in difficulty and computational demand.

In the non-deterministic distribution of the supervisory role, it has been observed that mining winners are more evenly distributed at difficulty level 2 than at level 1. However, this does not undermine the sustainability of distributing the supervisory role in the BBPM network. Even if certain nodes predominantly win the mining competition, the departure of such nodes from the network would simply result in other nodes becoming winners.

Based on these observations, it can be stated that calibrating the difficulty level exerts a dominant influence on the time required for the PoW process, allowing the BBPM protocol to set the difficulty level to an optimal value tailored to its specific objectives.

C. AUTHENTICATION AND SECURITY

For this evaluation, we instantiated an additional node, assigning it node ID 4, and forced it to attempt tampering with the OP-ledger managed by the BBPM network, which was established for the Integral Functionality Check. This node did not join the network but broadcast a transaction containing an arbitrary 'tag = value.'

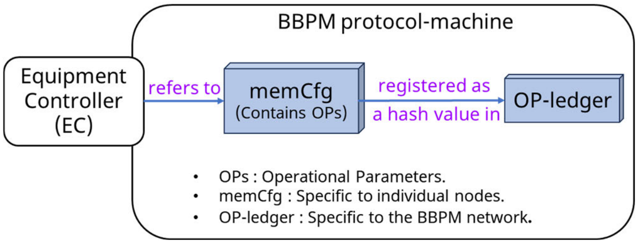

BBPM protocol-machine
Equipment Controller (EC)
refers to
memCfg (Contains OPs)
registered as a hash value in
OP-ledger
• OPs : Operational Parameters.
• memCfg : Specific to individual nodes.
• OP-ledger : Specific to the BBPM network.

FIGURE 9. Data referencing between equipment controller and BBPM protocol-machine.

By observing how the network's participating nodes responded to this unauthorized action, we confirmed that all participating nodes deemed the transaction from node 4 unacceptable because it originated from an unidentifiable node.

Consequently, this confirms that the validation mechanism embedded in the BBPM design is capable of detecting illegal access.

Furthermore, this mechanism incorporates the digital signature algorithm used in Bitcoin to leverage its proven authentication performance. However, additional research is needed to address the possibility of various illegal access scenarios that the BBPM may encounter in real-world environments.

D. PARAMETER MATCHING CAPABILITY

The BBPM network, comprising the three nodes, continues to serve in this evaluation.

1) TERMINOLOGY

From the perspective of a single node, 'A-State' is defined as the state in which the hash generated from the operational parameters stored in 'memCFG'-a dedicated memory storage designed specifically for these parameters-matches the hash stored in the BBPM OP-ledger. The node's equipment controller refers to this storage when configuring the equipment. Fig. 9 illustrates data referencing between the equipment controller and the BBPM protocol-machine. 'A-State' is recorded as TRUE in the 'MatchingResult.' Conversely, 'B-State' occurs when the two hashes do not match, and is recorded as FALSE in the 'MatchingResult.' The 'B-State' scenario arises when a node modifies operational parameters in its 'memCFG' without registering the changes in the BBPM OP-ledger.

Collectively, these two states are termed the 'Matching-State.' Importantly, the 'Matching-State' is not specific to the entire BBPM network (network-specific) but is instead specific to individual nodes (node-specific). This distinction allows individual nodes to be in different 'Matching-States' from others within the same BBPM network.

2) SIMULATION SCENARIO

For this test, the state of SimBBPM is restored to the condition following the completion of the mining process described in subsection B. 'Integral Functionality Check.' To evaluate 'A-State' matching, we initiated the mining process by pressing '4' on Node 0's console application, thereby simulating the evaluation of 'A-State.' For the 'B-State' evaluation, we simulated accidental changes to operational parameters by first pressing '5' on Node 0's console application, followed by pressing '4' to conduct the 'B-State' matching evaluation.

3) SIMULATION RESULTS

In the conducted simulation, it was demonstrated that the BBPM protocol consistently delivered the anticipated outcomes: specifically, returning the 'A-State' when evaluated under the 'A-State' scenario and the 'B-State' under the 'B-State' scenario. These results confirm the protocol's efficacy in distinguishing between the two operational scenarios.

Our simulations demonstrated that the BBPM protocol consistently returns the 'A-State' for Node 1 and Node 2, even when Node 0 inadvertently shifts to the 'B-State.' This is because Nodes 1 and 2 remain unaffected by accidental alterations to their operational parameters, thereby maintaining the 'A-State.' These findings highlight the BBPM protocol's capability to prevent a network-wide collapse when a single node transitions to the 'B-State,' by managing the 'Matching-State' on a node-specific basis.

The fact that Nodes 1 and 2 return the 'A-State' can be detailed through the following algorithmic processes:

Broadcasting MP-REQ:

(1) The matching process is initiated by pressing '4' on Node 1's console application.
(2) Node 1 computes 'CfgHash' from its 'memCFG,' which is synchronized with the BBPM OP-ledger.
(3) Subsequently, Node 1 broadcasts an 'MP_REQ' that includes the 'CfgHash.'

Receiving MP-REQ:

(4) Upon receiving the 'MP_REQ,' Nodes 0 and 2 compare the CfgHash in the 'MP_REQ' with the hash in the OP-ledger (When Node 2 sends the 'MP_REQ,' this task is performed by Nodes 0 and 1).
(5) Despite Node 0's accidental parameter alteration, the OP-ledger remains synchronized with 'memCFG' of Nodes 1 and 2. It's important to note that Node 0 compares the hash created from Node 1's or Node 2's 'memCfg,' not its own, with the hash in the OP-ledger.
(6) This synchronization results in Nodes 0 and 2 obtaining a TRUE 'MatchingResult' (When Node 2 sends the 'MP_REQ,' Nodes 0 and 1 perform this step).

Sending MP-REPLY:

(7) Following the comparison, Nodes 0 and 2 send an 'MP_REPLY,' indicating a TRUE 'MatchingResult,' back to the 'MP-REQ' sender (When Node 2 sends the 'MP_REQ,' Nodes 0 and 1 are the ones sending the 'MP_REPLY').

Receiving MP-REPLY:

(8) Consequently, Node 1 receives a TRUE 'MatchingResult' from both Nodes 0 and 2 (When Node 2 is the sender of the 'MP_REQ,' Node 2 receives a TRUE 'MatchingResult' from Nodes 0 and 1).

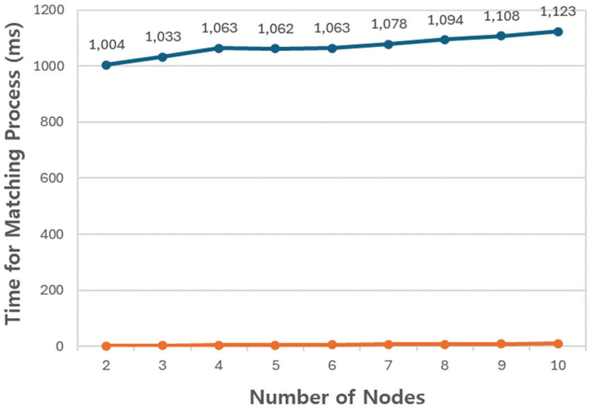


FIGURE 10. Trend of elapsed time of matching process.

Returning Matching-State:

(9) Based on the TRUE 'MatchingResult,' the BBPM protocol-machine of Node 1 returns the 'A-State' (When Node 2 sends the 'MP_REQ,' the protocol-machine of Node 2 returns the 'A-State').

In conclusion, after an in-depth evaluation and analysis of the parameter matching capability, we confirm that the matching process operates as intended by this study.

E. INVESTIGATION OF IMPLEMENTATION PERFORMANCE

As mentioned before, although the performance issues of BBPM is beyond the scope of this paper, we include an analysis of measurement data relevant to them, which should be considered for BBPM application in real-world environments. The primary concerns about BBPM's performance originate from the software designs used to implement the BBPM protocol. This analysis establishes the foundational criteria for exploring BBPM's performance in future research. Fig. 10 shows the trend of elapsed time for the matching process, as measured on the computer used to conduct our evaluation. This data shows that the elapsed time slightly increases with the addition of nodes, which could be attributed to the following potential causes:

(1) Resource contention: Each additional node application may compete for the same computing resources, such as CPU, memory, and I/O.

(2) Software overhead: The matching process might have software overhead that becomes more pronounced as the number of simulated nodes increases.

(3) Concurrency and threading: Because SimBBPM is implemented with a multi-threaded approach, the software architecture's ability to manage multiple threads could impact the latency of the matching process.

(4) TCP/IP overheads: An increase in TCP/IP connections can lead to higher traffic, potentially simulating the effects of network congestion.

Among these causes, 'resource contention' can be excluded in real-world environments as BBPM nodes operate on separate machines. 'Software overhead' and 'concurrency and threading' significantly influence BBPM's performance. Therefore, subsequent research would benefit from evaluating and improving the software architecture of the BBPM protocol for field application efficiency. 'TCP/IP overheads' are affected by the BBPM protocol and the complexities of the network environment. Consequently, it would be beneficial for future research to consider focusing on testing the protocol in conditions that closely mimic the actual deployment scenarios.

F. SUMMARY

The practicality evaluation of the BBPM protocol has yielded several key findings that affirm the effectiveness and robustness of its components and processes designed for blockchain technology adoption. We confirmed the effective construction of a BBPM network during the network joining process by verifying the successful registration of the equipment IDs in the 'NetworkNodeTable' for each participant. The examination of the OP-ledger after the mining process verified exact updates and synchronization across the BBPM network, confirming that the tailored data structures, alongside the integration of network joining, transaction propagation, and mining processes, function as intended. Additionally, an examination of the PoW's difficulty level revealed its significant impact on computing times, suggesting that calibrating the difficulty level can optimize the BBPM PoW's operational intensity to meet the protocol's requirements.

Further aspects of the protocol's practicality were assessed through authentication and security measures, as well as through parameter matching capabilities and performance implementation investigations. The network's ability to reject transactions from unrecognized nodes highlights its robust security protocols, ensuring that only authorized participants can engage. However, the wide variety of potential illegal access scenarios in the real world necessitates further research. The BBPM protocol also proved effective in managing the 'Matching-State' on a node-specific basis under prescribed conditions. This is crucial to prevent a network-wide collapse when a single node transitions to the 'B-State.' Furthermore, the 'OPCODE,' which specifies the method for matching operational parameters, requires further refinement to more effectively cover diverse matching scenarios. Lastly, the identification of architectural factors such as software overhead, concurrency, threading and network complexities guides the ongoing refinement of the BBPM protocol, aiming to enhance its efficiency and reliability for real-world applications.

These comprehensive evaluations not only underscore the BBPM protocol's readiness for practical deployment but also highlight the need for further research to effectively adapt to the diverse real-world scenarios.

VI. CONCLUSION

Despite the recognized potential of blockchain, research on its application in semiconductor equipment is notably scarce. To address this gap, this paper introduces and develops the blockchain-based parameter matching (BBPM) protocol, which is installed in semiconductor equipment. This protocol facilitates the transition from a centralized to a decentralized operational parameter system, preserving the integrity and security of operational parameters. Additionally, it autonomously adapts to changes in equipment layout without requiring a dedicated administrator for network management.

To develop the protocol, this study customizes the data structures: transactions now include a list of 'tag = value' pairs, and blocks contain arrays of these transactions, referred to as 'List of TRs.' Additionally, the difficulty level of the PoW is calibrated to avoid requiring excessive computational power. Subsequently, key algorithmic processes-the network joining process, the transaction propagation, the mining process and the matching process-are defined to realize the BBPM's task of the operational parameter matching within the realm of blockchain technology.

The practicality evaluation demonstrates that the BBPM protocol effectively tailors and integrates blockchain components and defined processes for its adoption of blockchain technology, ensuring they operate seamlessly in alignment with BBPM's objectives. Additionally, this paper proposes future research themes, including assessing BBPM's performance based on its software designs, tackling various unauthorized access scenarios in the real world, and detailing the 'OPCODE' to cover diverse matching scenarios.

Overall, this pioneering study not only establishes foundational technology but also opens new avenues for innovation in applying blockchain technology within the semiconductor manufacturing domain.

REFERENCES

[1] G. Schneider, S. Keil, and F. Lindner, "Benefits of digitalization for business processes in semiconductor manufacturing," in *Proc. 22nd IEEE Int. Conf. Ind. Technol. (ICIT)*, vol. 1, Mar. 2021, pp. 1027–1033.

[2] H. Razouk and R. Kern, "Improving the consistency of the failure mode effect analysis (FMEA) documents in semiconductor manufacturing," *Appl. Sci.*, vol. 12, no. 4, p. 1840, Feb. 2022.

[3] T. C. Eng, A. M. Sani, and P. K. Yu, "Methods to achieve zero human error in semiconductors manufacturing," in *Proc. 8th Electron. Packag. Technol. Conf.*, Dec. 2006, pp. 678–683.

[4] Y.-C. Wang, C.-F. Lee, and S.-T. Hou, "Knowledge structure affect members' technology adoption: A study of semiconductor manufacturing process," in *Proc. PICMET Portland Int. Conf. Manage. Eng. Technol.*, Aug. 2009, pp. 3252–3256.

[5] X. V. Wang, M. Givehchi, and L. Wang, "Manufacturing system on the cloud: A case study on cloud-based process planning," *Proc. CIRP*, vol. 63, pp. 39–45, Jan. 2017.

[6] (2022). *Security Needs*. Factory Integration, International Roadmap for Devices and Systems 2022 Edition, Sec. 5.7. Accessed: Apr. 22, 2024. [Online]. Available: https://irds.ieee.org/editions/2022/irds%E2%84%A2-2022-factory-integration

[7] C. Yang, L. Tan, N. Shi, B. Xu, Y. Cao, and K. Yu, "AuthPrivacyChain: A blockchain-based access control framework with privacy protection in cloud," *IEEE Access*, vol. 8, pp. 70604–70615, 2020.

[8] N. Sanghi, R. Bhatnagar, G. Kaur, and V. Jain, "BlockCloud: Blockchain with cloud computing," in *Proc. Int. Conf. Adv. Comput., Commun. Control Netw. (ICACCCN)*, Oct. 2018, pp. 430–434.

[9] S. Bose, M. Raikwar, D. Mukhopadhyay, A. Chattopadhyay, and K.-Y. Lam, "BLIC: A blockchain protocol for manufacturing and supply chain management of ICS," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jul. 2018, pp. 1326–1335.

[10] K. Fan, Q. Pan, K. Zhang, Y. Bai, S. Sun, H. Li, and Y. Yang, "A secure and verifiable data sharing scheme based on blockchain in vehicular social networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5826–5835, Jun. 2020.

[11] M. Mettler, "Blockchain technology in healthcare: The revolution starts here," in *Proc. IEEE 18th Int. Conf. e-Health Netw., Appl. Services (Healthcom)*, Sep. 2016, pp. 1–3.

[12] J. Sun, X. Yao, S. Wang, and Y. Wu, "Blockchain-based secure storage and access scheme for electronic medical records in IPFS," *IEEE Access*, vol. 8, pp. 59389–59401, 2020.

[13] A. Pal, C. K. Tiwari, and N. Haldar, "Blockchain for business management: Applications, challenges and potentials," *J. High Technol. Manage. Res.*, vol. 32, no. 2, Nov. 2021, Art. no. 100414.

[14] Z. Shahbazi and Y.-C. Byun, "Integration of blockchain, IoT and machine learning for multistage quality control and enhancing security in smart manufacturing," *Sensors*, vol. 21, no. 4, p. 1467, Feb. 2021, doi: 10.3390/s21041467.

[15] L. Herrgoß, J. Lohmer, G. Schneider, and R. Lasch, "Development and evaluation of a Blockchain concept for production planning and control in the semiconductor industry," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manag. (IEEM)*, Sep. 2020, pp. 440–444.

[16] I. Meidute-Kavaliauskiene, B. Yıldız, Ş. Çiğdem, and R. Činčikaitė, "An integrated impact of blockchain on supply chain applications," *Logistics*, vol. 5, no. 2, p. 33, Jun. 2021, doi: 10.3390/logistics5020033.

[17] T. Kumar, E. Harjula, M. Ejaz, A. Manzoor, P. Porambage, I. Ahmad, M. Liyanage, A. Braeken, and M. Ylianttila, "BlockEdge: Blockchain-edge framework for industrial IoT networks," *IEEE Access*, vol. 8, pp. 154166–154185, 2020.

[18] T. Kumar, A. Braeken, V. Ramani, I. Ahmad, E. Harjula, and M. Ylianttila, "SEC-BlockEdge: Security threats in blockchain-edge based industrial IoT networks," in *Proc. 11th Int. Workshop Resilient Netw. Design Modeling (RNDM)*, Oct. 2019, pp. 1–7.

[19] M. Isaja, J. Soldatos, and V. Gezer, "Combining edge computing and blockchains for flexibility and performance in industrial automation," in *Proc. Int. Conf. Mobile Ubiquitous Comput., Syst., Services Technol. (UBICOMM)*, 2017, pp. 159–164.

[20] P. Petrali, M. Isaja, and J. K. Soldatos, "Edge computing and distributed ledger technologies for flexible production lines: A white-appliances industry case," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 388–392, 2018.

[21] (2022). *Challenges*. Factory Integration, International Roadmap for Devices and Systems (IRDS) 2022 Edition, Sec. 4. Accessed: Apr. 22, 2024. [Online]. Available: https://irds.ieee.org/editions/2022/irds%E2%84%A2-2022-factory-integration

[22] (2022). *Potential Solutions*. Factory Integration, International Roadmap for Devices and Systems (IRDS) 2022 Edition, Sec. 6. Accessed: Apr. 22, 2024. [Online]. Available: https://irds.ieee.org/editions/2022/irds%E2%84%A2-2022-factory-integration

[23] (2022). *Digital Twin Needs*. Factory Integration, International Roadmap for Devices and Systems (IRDS) 2022 Edition, Sec. 5.8.6. Accessed: Apr. 22, 2024. [Online]. Available: https://irds.ieee.org/editions/2022/irds%E2%84%A2-2022-factory-integration

[24] (2022). *Factory Integration*. International Roadmap for Devices and Systems (IRDS) 2022 Edition. Accessed: Apr. 22, 2024. [Online]. Available: https://irds.ieee.org/editions/2022/irds%E2%84%A2-2022-factory-integration

[25] A. K. Chee, "The mechanistic determination of doping contrast from Fermi level pinned surfaces in the scanning electron microscope using energy-filtered imaging and calculated potential distributions," *Microsc. Microanalysis*, vol. 28, no. 5, pp. 1538–1549, Oct. 2022.

[26] A. K. W. Chee, "Quantitative dopant profiling by energy filtering in the scanning electron microscope," *IEEE Trans. Device Mater. Rel.*, vol. 16, no. 2, pp. 138–148, Jun. 2016.

[27] A. K. W. Chee, R. F. Broom, C. J. Humphreys, and E. G. T. Bosch, "A quantitative model for doping contrast in the scanning electron microscope using calculated potential distributions and Monte Carlo simulations," *J. Appl. Phys.*, vol. 109, no. 1, Jan. 2011, Art. no. 013109.

[28] A. K. W. Chee, "Principles of high-resolution dopant profiling in the scanning helium ion microscope, image widths, and surface band bending," *IEEE Trans. Electron Devices*, vol. 66, no. 11, pp. 4883–4887, Nov. 2019.

[29] A. K. W. Chee and S. A. Boden, "Dopant profiling based on scanning electron and helium ion microscopy," *Ultramicroscopy*, vol. 161, pp. 51–58, Feb. 2016.

[30] A. K. W. Chee, "Unravelling new principles of site-selective doping contrast in the dual-beam focused ion beam/scanning electron microscope," *Ultramicroscopy*, vol. 213, Jun. 2020, Art. no. 112947.

[31] (2022). *Introduction*. Factory Integration, International Roadmap for Devices and Systems (IRDS) 2022 Edition, Sec. 1. Accessed: Apr. 22, 2024. [Online]. Available: https://irds.ieee.org/editions/2022/irds%E2%84%A2-2022-factory-integration

[32] Z. J. Lou, H. H. T. Qian, and M. E. Liu, "Advanced process equipment matching methodology in semiconductor manufacturing," in *Proc. China Semiconductor Technol. Int. Conf.*, 2018, pp. 1–4.

[33] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," Nat. Inst. Standard Technol., Gaithersburg, MD, USA, Tech. Rep. NISTIR 8202, p. 1, Oct. 2018. Accessed: Jun. 7, 2021. [Online]. Available: https://doi.org/10.6028/NIST.IR.8202

[34] S. Sun, R. Du, S. Chen, and W. Li, "Blockchain-based IoT access control system: Towards security, lightweight, and cross-domain," *IEEE Access*, vol. 9, pp. 36868–36878, 2021.

[35] S. Srivastava, M. A. Kumar, S. K. Jha, P. Dixit, and S. Prakash, "Event-driven data alteration detection using block-chain," *Secur. Privacy*, doi: 10.1002/spy2.146.

[36] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," Nat. Inst. Standard Technol., Gaithersburg, MD, USA, Tech. Rep. NISTIR 8202, p. 10, Oct. 2018. Accessed: Jun. 7, 2021. [Online]. Available: https://doi.org/10.6028/NIST.IR.8202

[37] A. Alruwaili and D. Kruger, "Intelligent transaction techniques for blockchain platforms," in *Proc. Int. Conf. Comput., Electron. Commun. Eng. (iCCECE)*, Aug. 2019, pp. 177–182.

[38] M. Crosby, N. Nachiappan, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain technology: Beyond Bitcoin," *Appl. Innov. Rev.*, vol. 71, no. 2, pp. 6–19, Jun. 2016.

[39] A. M. Antonopoulos, *Mastering Bitcoin*, 2nd ed. Sebastopol, CA, USA: O'Reilly, 2017, p. 214.

[40] A. M. Antonopoulos, *Mastering Bitcoin*, 2nd ed. Sebastopol, CA, USA: O'Reilly, 2017, pp. 201–207.

[41] I. Bashir, *Mastering Blockchain*, 3rd ed. Birmingham, U.K.: Packt, 2020, p. 37.

[42] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328–22370, 2019.

[43] D. Bazzanella and A. Gangemi, "Bitcoin: A new proof-of-work system with reduced variance," *Financial Innov.*, vol. 9, p. 91, May 2023.

[44] W. Deng, T. Huang, and H. Wang, "A review of the key technology in a blockchain building decentralized trust platform," *Mathematics*, vol. 11, no. 1, p. 101, Dec. 2022.

[45] I. Malakhov, A. Marin, S. Rossi, and D. Smuseva, "On the use of proof-of-work in permissioned blockchains: Security and fairness," *IEEE Access*, vol. 10, pp. 1305–1316, 2022.

[46] M. Rahouti, D. Lyons, S. K. Jagatheesaperumal, and K. Xiong, "A decentralized cooperative navigation approach for visual homing networks," *IT Prof.*, vol. 25, no. 6, pp. 71–81, Nov. 2023.

[47] Y. Zhou, Y. Guan, Z. Zhang, and F. Li, "A blockchain-based access control scheme for smart grids," in *Proc. Int. Conf. Netw. Netw. Appl. (NaNA)*, Oct. 2019, pp. 368–373.

[48] M. Platt, J. Sedlmeir, D. Platt, J. Xu, P. Tasca, N. Vadgama, and J. I. Ibañez, "The energy footprint of blockchain consensus mechanisms beyond proof-of-work," in *Proc. IEEE 21st Int. Conf. Softw. in Quality, Rel. Secur. Companion (QRS-C)*, Dec. 2021, pp. 1135–1144.

[49] M. H. Miraz, P. S. Excell, and M. K. S. B. Rafiq, "Evaluation of green alternatives for blockchain proof-of-work (PoW) approach," in *Proc. Ann. Emerg. Technol. Comput. (AETiC)*, 2021, pp. 54–59.

[50] R. Zhang and W. K. Chan, "Evaluation of energy consumption in blockchains with proof of work and proof of stake," *J. Phys., Conf. Ser.*, vol. 1584, no. 1, Jul. 2020, Art. no. 012023.

[51] D. Khan, L. T. Jung, and M. A. Hashmani, "Systematic literature review of challenges in blockchain scalability," *Appl. Sci.*, vol. 11, no. 20, p. 9372, Oct. 2021.

[52] H. Alshahrani, N. Islam, D. Syed, A. Sulaiman, M. S. Al Reshan, K. Rajab, A. Shaikh, J. Shuja-Uddin, and A. Soomro, "Sustainability in blockchain: A systematic literature review on scalability and power consumption issues," *Energies*, vol. 16, no. 3, p. 1510, Feb. 2023.

[53] R. Chokkalingam, "Comparison of acclaimed consensus algorithm," Capstone Project Report, Univ. Alberta, Alberta, AB, Canada, Tech. Rep. MINT-709, 2020.

[54] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, and G. Danezis, "Consensus in the age of blockchains," in *Proc. 1st ACM Conf. Adv. Financial Technol.*, 2019, pp. 183–198.

[55] S. Fahim, S. Katibur Rahman, and S. Mahmood, "Blockchain: A comparative study of consensus algorithms PoW, PoS, PoA, PoV," *Int. J. Math. Sci. Comput.*, vol. 9, no. 3, pp. 46–57, Aug. 2023.

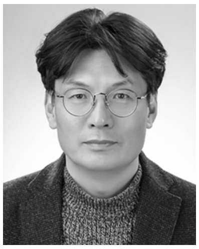
[56] S. S. Hazari and Q. H. Mahmoud, "Improving transaction speed and scalability of blockchain systems via parallel proof of work," *Future Internet*, vol. 12, no. 8, p. 125, Jul. 2020.

[57] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *IEEE Access*, vol. 8, pp. 16440–16455, 2020.

[58] H. Kohad, S. Kumar, and A. Ambhaikar, "Scalability issues of blockchain technology," *Int. J. Eng. Adv. Technol*, vol. 9, no. 3, pp. 2385–2391, 2020.

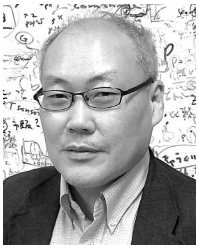
[59] (2022). *Big Data Needs*. Factory Integration, International Roadmap for Devices and Systems (IRDS) 2022 Edition, Sec. 5.8.3. Accessed: Apr. 22, 2024. [Online]. Available: https://irds.ieee.org/editions/2022/irds%E2%84%A2-2022-factory-integration

[60] H. M. A. Aljassas and S. Sasi, "Performance evaluation of proof-of-work and Collatz conjecture consensus algorithms," in *Proc. 2nd Int. Conf. Comput. Appl. Inf. Secur. (ICCAIS)*, May 2019, pp. 1–6.



MYOUNG SOO CHOI was born in Incheon, South Korea. He received the master's degree from the Graduate School, Department of Electrical and Computer Engineering, Hanyang University, South Korea, in 2005. He is currently pursuing the Ph.D. degree with Tottori University, Japan. From 2003 to 2011, he was a Principal Engineer with the Mechatronics and Manufacturing Technology Center, Samsung Electronics, South Korea. From 2012 to 2018, he was the Vice President (VP) of the Mechatronics Research and Development Center, Samsung Electronics. In 2019, he was an Invited Researcher with Tottori University. He is also currently working at SK Hynix, South Korea. His research interests include smart software and equipment for semiconductor manufacturing.

JUMYUNG UM received the Ph.D. degree from POSTECH, South Korea, in 2012. He was a Postdoctoral Researcher with the Swiss Federal Institute of Technology Lausanne (EPFL), from 2012 to 2014. He was a Research Associate with the University of Cambridge, from 2014 to 2015. Since 2016, he has been a Senior Researcher with the German Research Center for Artificial Intelligence (DFKI). Since September 2018, he has been a Professor with the Department of Industrial and Management System Engineering, Kyung Hee University, South Korea. He joined Kyung Hee University, as a Faculty Member of artificial intelligence, in 2022. His research interests include virtual reality and artificial intelligence of smart factories.



SANG-SEOK LEE (Senior Member, IEEE) was born in Busan, South Korea. He received the Ph.D. degree from the Graduate School of Information Sciences, Tohoku University, Japan, in 1998. From December 1998 to December 1999, he was a Postdoctoral Researcher with the Venture Business Laboratory, Tohoku University. From January 2000 to January 2002, he was a Postdoctoral Researcher with the Electronic Instrumentation Laboratory, Delft University of Technology, The Netherlands. From March 2002 to September 2011, he was a Researcher with the Advanced Technology Research and Development Center, Mitsubishi Electric Corporation, Japan. Since moved to Tottori University, Japan, in October 2011, he had been an Endowed Chair Professor and has been a Professor with the Faculty of Engineering. His research interests include MEMS/NEMS, microfluidics, sensor, and semiconductor technology. He is an Associate Editor of IEEE Sensors Journal, and *Micro and Nano Systems Letters*.