Received 4 August 2023; accepted 18 August 2023. Date of publication 21 August 2023; date of current version 13 September 2023.

*Digital Object Identifier 10.1109/OJCOMS.2023.3307337*

# A Scalable Blockchain Framework for Secure Transactions in IoT-Based Dynamic Applications

**SULTAN BASUDAN**

Faculty of Computer Science and Information Technology, Jazan University, Jazan 45142, Saudi Arabia

CORRESPONDING AUTHOR: S. BASUDAN (e-mail: sbasudan@jazanu.edu.sa)

**ABSTRACT** The essential characteristics of blockchain, such as immutability and auditability, are leading to its increasing incorporation into society. The Internet of Things (IoT) aims to create a decentralized framework for IoT and enhance security. The processing of transactions for different IoT applications poses challenges for the convergence of IoT and blockchain due to varying performance requirements. In addition, it is important to note that the dispersed IoT system's membership may change when an IoT device adds or departs from it. As the system becomes dynamic, it presents new difficulties for managing devices. Accordingly, a dynamic application block generation (DABG) scheme for blockchain-enabled IoT with dynamic device management and conditional traceability is proposed. To begin, it is necessary to construct a framework for an IoT system based on a consortium blockchain. This framework should consist of structures for dynamic application transactions and blocks, as well as a consensus mechanism. Miners are presented in distinct ways to adaptively process urgent and routine application transactions. A protocol based on group signature, known as DABG, is proposed for this framework. It is possible to achieve nonframeability, traceability, and anonymity by the use of the group signature. The suggested method may quickly validate transactions, manage devices in a dynamic manner, enable conditional traceability while maintaining data security, and keep users' privacy intact. This is made possible by merging time-bound keys in the group signatures and node in the blockchain. Extensive testing has shown that the strategy in question has the potential to achieve very high levels of effectiveness.

**INDEX TERMS** Blockchain, Internet of Things (IoT), security and privacy, scalability.

## I. INTRODUCTION

THE INTERNET of Things (IoT) links the real world to the virtual one, creating a digital duplicate of reality. In this way, IoT can revolutionize business and culture by exploring cutting-edge uses in fields like smart cities, connected homes, and high-tech medical care [1], [2], [3], [4]. Most of the available IoT devices are based on a centralized model, which can result in bottlenecks and slow communication. However, the success of edge-based IoT depends on the trustworthiness of the fog or edge nodes in the network. Fortunately, the Internet of Things can capitalize on the new prospects provided by emerging blockchain technology. A blockchain is essentially a list of records that are interconnected and stored across multiple nodes. In recent years, academia and industry have shown great interest in the transparency, anonymity, autonomy, and immutability of this technology [5]. Integrating blockchain technology into the Internet of Things has been recommended by a number of recent studies [5], [6] for a variety of uses, including in healthcare, the Internet of Vehicles (IoV), smart cities, and the smart grid. Privacy, security, identity management, data management, and trust management are all crucial aspects of the Internet of Things, and research shows that this convergence can help the creation of new blocks that is an integral part of any blockchain infrastructure. Data in Internet of Things systems that employ blockchain technology is typically arranged in the form of transactions and then sealed off in a block [5], [6]. The Block is then sent out to all of the nodes in the blockchain network to be validated. Current systems typically process blocks at a constant or average rate. Particularly in time-critical applications like medical crises and transportation networks, this infrastructure is not flexible enough to accommodate fluctuating transaction volumes.

Furthermore, most IoT devices, including smart meters and sensors, have restricted their resources, such as low battery power, inadequate computational power, and limited storage space [6]. Consensus mechanisms on blockchains often require a considerable amount of time and energy, as is the case with the PoW used in BTC [7]. Furthermore, due to the fact that blockchain is essentially a decentralized ledger, the amount of data it produces increases as time goes on. Therefore, it is not feasible to completely execute a blockchain and retain all its data on every IoT device. Moreover, the Internet of Things (IoT) often encounters unstable network connections caused by the depletion of node energy or fluctuations in experienced in their wireless networks. It may be impossible to use a blockchain for this purpose, as it is primarily designed for applications with stable organizational connections. Recent works [8], [9], [10], [11], [12], [13] have addressed the aforementioned concerns. Kumar and Das [8] created 5 separate blockchain nodes for IoV, classified blockchain data into 5 distinct groups, and built 5 distinct blockchains to accommodate 5 various use cases and data kinds. I also did some theoretical modeling of transportation networks. The framework and network architecture were the key points of interest, rather than the traffic conditions or the dependability of the channels. The latency experienced by all parties involved in a blockchain transaction was investigated by Alaslani et al. [13]. They investigated how a mathematical algorithm to determine how long various configurations of work, such as replica machines and hop counts, would take in a Byzantine-based blockchain. The paper argued that the it seemed impossible for the BFT based blockchain technology that was to handle the massive amount to data that were produced by different IoT applications. Previous studies [9], [10], [11], [12] have indicated the importance of addressing resource limitation issues in blockchain-based IoT systems. As indicated in references [6], [10], [14], edge or fog computing seems like a promising solution for dealing with the constraints imposed by resource-constrained IoT devices. The overarching goal imposed on these projects aimed to streamline IoT device data processing and storage by moving their responsibilities to the cloud or the network's edge. However, these devices failed to directly transmit the transaction to the network, which resulted to unacceptable levels of latency in the handling of transactions for time-critical software. Scalable or adaptable blockchain architectures could be presented as an additional possible solution [15], [16] to ensure dynamic device management and boost transaction processing efficiency. The studies proposed either a highly efficient structure for IoT blockchains or a revolutionary lightweight consensus mechanism for managing IoT devices with minimal overhead.

Given its potential, blockchain adoption in IoT confronts the following challenges, even as the work indicated above is heuristic and stimulates IoT applications.

- The scalability of the blockchain is determined by comparing the number of IoT nodes to the transaction throughput per second [17]. In a typical blockchain system, all nodes are responsible for broadcasting and verifying each Block, which results in a low throughput. According to reference [8], it is well-known that the throughput of several blockchain systems, including Bitcoin and Ethereum, is low. It is essential for blockchain to be able to process numerous transactions within the shortest duration. This inconsistency limits the usefulness of blockchain-based (IoT) technologies.
- The tension between anonymity and being able to track something. The originators and recipients of blockchain transactions are node accounts. Individuals' identities behind these accounts are completely anonymous. Protecting the privacy of those involved in a transaction requires investigating options including encryption, anonymization, mixing, smart contracts, and differential privacy [18]. Intended users' identities may need to be traced in a distributed network without a trusted hub if, for example, malevolent users are discovered. This is because personal information and account information cannot be linked. It is difficult to strike a balance between ensuring there is privacy and transparency in the vast IoT infrastructure.
- Mobile Device Management. Mobility is a prominent feature of IoT devices, adding complexity to device management processes like access control, authentication, revocation, and tracing. These missions can be taken care of by a reliable center in a conventionally centralized framework. When moving to a blockchain, these operations cannot be carried out by a trusted center.

Based on the above analysis, a consortium blockchain framework for IoT is proposed with elastic block generation and dynamic device management. In particular, the transactions into urgent and ordinary ones are classified according to their time-sensitive ranks in applications. Dynamic application block generation (DABG) scheme is proposed, which employs a new defined group signature technique. The group signature is used to achieve anonymity, traceability, and nonframeability. For instance, in order to produce user identity and achieve anonymity, the group signature has been exploited which that each user has a group of public and private keys. Therefore, each user generates a transaction and will use its group private key to generate a signature, thus the transaction will not be referred to the sender. Within the blockchain, participating organizations function as group managers, responsible for overseeing their members, which are IoT devices. Meanwhile, it is the responsibility of group managers to identify any malicious participants involved in disputations. The system's devices' natural expiration can be checked by using the timestamps in blocks. The system also has batch verification enabled for fast transaction verification. To the best of my knowledge, this is the first work to propose an exchange handling plan adjusting with the exchange rates and its time-delicate positions as indicated by the applications of blockchain-based IoT frameworks. The following is a summary of the contributions.

1) Elastic block generation for the proposed framework is utilized in building a collaborative blockchain for an IoT system. The new structure for blocks and transactions is designed to cater to urgent and routine transactions, respectively. Currently, a novel DABG consensus mechanism is being utilized to process both miner **M** and miner **E**.
2) A certificateless group signature with a time-limited key and batch verification is developed. Features such as natural expiration, batch verification, traceability, and revocation are incorporated into the proposed calculation for the bunch signature. Timestamps in blockchain and renunciation, combined with the time-bound key supplied in the gathering signature, facilitate the management of devices that have a common expiration and premature termination. Devices and the server node (*SerN*) both generate the secret key, which ensures nonrepudiation.
3) The proposed group signature serves as the basis for the DABG protocol. The urgent blocks and ordinary blocks are created compared to critical exchanges and normal exchanges. It has been demonstrated that the proposed protocol can safely and effectively carry out conditional traceability, batch verification, and dynamic device management. It is important to note that the node account is connected to both the public key of the group and the public key of the node. This linkage helps to prevent the framing of arguments.

This article is structured as follows. Section II presents the related work while Section III illustrates the basic concepts of the proposed authentication protocol. Section IV describes the system model and architecture of the proposed authentication protocol. The proposed protocol based on group signature, known as DABG, is proposed for this framework in Section V. A security analysis is reported in Section VI, while the performance of the proposed protocol is evaluated in Section VII. Section VIII summarizes the work.

## II. RELATED WORK

There have been proposals in recent years from both the academic and business communities to integrate blockchain technology into the Internet of Things. Researchers have showed a great deal of enthusiasm for investigating blockchain-based IoT systems' many potentials use in fields as diverse as manufacturing, medicine, the Internet of Things, urban planning, and power distribution. Consensus mechanisms, security, privacy, incentives, system architectures, data sharing, and computational cooperation are some of the primary areas of study. In-depth and wide-ranging surveys of blockchain's IoT applications have been published elsewhere [8], [9], [10], [11], [12], [13], [17], [19]. Readers are encouraged to consult these citations for a fuller picture of the research landscape.

The main concerns of the proposed work, the administration of devices and the processing of transactions in an IoT blockchain system, have been mostly overcome by recent research. Bagchi et al. [20] introduced an aggregate signature for Internet of Drones (IoD) applications using the blockchain technology in order to enable a party to bundle a set of signatures together into a single short cryptographic signature. Moreover, Sivaselvan et al. [21] proposed scalable and secure access control scheme for IoT-blockchain-based without having the resource-constrained IoT devices to be part of the blockchain network. Further, they possess substantial amount of blockchain data as the root-of-trust. A comprehensive analysis of various security mechanisms applied in the IoT and blockchain technologies are introduced in [22]. They discussed various applications and their respective services in terms of security aspects. Furthermore, Vangala et al. [23] investigated authentication issues and proposed a blockchain- authenticated key agreement scheme for mobile vehicles-assisted precision agricultural Internet of Things (IoT) networks. They used elliptic curve operations on an active hybrid blockchain over mobile farming vehicles to fulfill a lightweight property. In [19], they showed proof of concept for transaction processing and edge computing's role in supporting data storage management. Using certificate cryptography and blockchain technology, a user-friendly authentication method is created for widespread IoT data exchange. According to [8], a Sybil proof BFT agreement was established that can attain constant agreement that would help address the real-time IoT applications. The computational requirements of the consensus process were not considered for IoT devices that have limited resources. Existing blockchain systems cannot handle the volume of transactions generated by IoT devices. To address this issue, Biswas et al. designed a local peer channel that manages new transactions entering the global blockchain while enabling local and global peer confirmation of all transactions. By addressing the problem of block storage in memory, this approach improved the scalability of IoT business transactions. Dorri et al. [24] distinguished between "local" and "overlay" transactions. Further, the transactions that took place locally were designed by the local servers. There were also intersection transactions that were handled and authenticated by the overlaying nodes that were used. As a result, they recommended that they use algorithm that is time-based consensus and uses decentralized and takes a trustworthy approach to reduce the number of transactions that must be validated per block over time. This method aims to minimize resource utilization. This work was able to reduce transaction processing time and bandwidth for IoT. However, it ran the risk of compromising security. Zhang et al. [25] proposed a decentralized structure for IoT that is more flexible, interoperable, and can manage distributed records. This article aims to develop a scalable framework for IoT devices in IoT networks. Through the use of leaders to validate blocks, a current study [15] is looking to develop a unique, accessible public blockchain with a two chain structure to accommodate fog computing and IoT service computing. Additionally, for QoS requirements and security, Qiu and their colleagues introduced a permissioned blockchain with multiple consensus protocols [12]. Gao et al. [26] utilized blockchain for secure transactions in the smart grid, and Zhaofeng et al. [10] proposed a belief information management scheme for edge processing with data encryption and a smart contract for conditional access to protected transactions and decryption queries. Ultimately, this works improved scalability of an IoT blockchain with detailed and time-sensitive transactions. Nonetheless, the prerequisites of these processes still need to be analyzed to run efficiently on IoT devices with limited resources. An important aspect of the Internet of Things (IoT) is the ability to scale the number of devices that can be connected. According to a prediction in 2022, there would be 28.5 billion devices connected to the Internet. Blockchain is a type of ledger that is decentralized and treats devices similar to accounts. It is important to analyze devices in IoT systems that use blockchain technology. Recent studies suggest that user authentication and access management could help address this issue [27], [28], [29], [30], [31]. Since IoT devices are extensive and dynamic, creating an effective centralized authentication system is impossible. To solve this issue, Hammi et al. [32] designed virtual zones that are secure, and where objects can freely interact for device identification and authentication. To establish trust between different domains, blockchains were introduced in [12], [33]. Cryptography primitives like identity-based signatures and authentication were explored to attain unidentified verification. Lin and colleagues developed an efficient method for authenticating IoT devices based on blockchain technology, even with limited resources. Reference [34] proposed several attempts to resolve these issues. Please note the cross-space confirmation.

A suggestion was put forth to incorporate blockchain, attribute signatures, and message authentication code to enable effective mutual authentication with detailed access control to terminals and gateways in Industry 4. installations [35]. In their subsequent work, the authors [35] presented the concept of group signatures, which serve the purpose of anonymous authentication of group members, ensuring dependable auditing of users' access histories, and providing effective authentication of home gateways for blockchain-based smart homes. Furthermore, Lin and colleagues discovered a method to enhance the computational abilities of IoT devices that have limited resources. The first permissioned blockchain-based secure computation outsourcing of bilinear pairings was proposed by [34]. These innovative efforts provide a practical solution for managing the actions of IoT devices that have limited resources. Access management is another crucial aspect of managing devices.

An access management system that is decentralized to be used in blockchain to maintain access control data has been proposed by Novo [36]. This system aims to address the challenge of managing how they can reach various constrained devices that would bring scalability. These devices are tied to a network by using a management hub, which does not transmit transactions directly to the blockchain. To solve this problem, Pal et al. [37] introduced a decentralized delegation model for IoT, which is identity less and asynchronous and based on block chain. The researchers used properties to enable the system to assign access rights for IoT in a fine-grained and flexible manner. Additionally, an intelligent agreement has been considered as a possible tool that would enhance the management of the device and access monitoring. Zhang et al. [38] recommended a structure for access control in distributed and dependable IoT systems using smart contracts. The framework consisted of several contracts for entrance control, one agreement for adjudication, and one contract for registration. The initial smart contract utilized a method for evaluating misbehavior and implemented control of access for a specific subject-object pairing. The registered contract has handled the two former systems.

In conclusion, the current research on device management is based on heuristics. They offered insightful and profound perspectives on managing IoT devices through blockchain technology. However, they ignored member regulation, such as user revocation, registration, and tracing, among others, and instead concentrated primarily on IoT device behavior regulation. Blockchain-based IoT device management faces additional difficulties since IoT devices can dynamically enter or exit the system as there is no centralized organization to carry out this role.

### A. GROUP SIGNATURES

Group signatures allow a vocalist to sign a message anonymously representing a group. Since its proposal by Chaum and Heyst [39] security safeguarding has received significant attention and seen a large number of applications. The research conducted by Boneh et al. [40] on short gathering marks was thoroughly researched for various purposes such as renouncement, group confirmation, and discernibility. Effective repudiation is essential for group members to acknowledge dynamic gadget management in a group. The previous works made an effort to perform verification and revocation checks to some extent. If a member's access has been revoked, they will not be able to provide a valid signature for the initial type that is required to pass the verification [41], [42], [43]. The signer was required to provide proof that the group's device was still active. As a result, the computational overhead of this signing operation was much more than that of the second kind. The second kind has a higher computational cost than the first type because verifiers must subject the system to a reversal check by examining the revocation list. This is because any person that signs can create a valid signature. An appealing variant of revocable group signature that featured time-bound keys contained both types of revocation [41], [42], [43], [44]. For each marking key, this type of gathering mark establishes an expiration date. If the key expired, also known as a typical lapse, the endorsers were unable to generate a meaningful mark. In the interim, a portion may be belatedly repudiated by being refused by denial check prior to the termination time in the disavowal list. This sort of group signature has the advantage of having a smaller revocation list than the second type because natural expiration regularly succeeds.

Another important aspect of group signature is batch verification, which increases verification efficiency by simultaneously verifying multiple signatures [45], [46]. They given a few essential perceptions to decide while matching conditions can be bunch checked and how to create a proficient bunch verifier, trailed by [47]. They used the properties of bilinear pairing operations to carry out batch verification. Group signatures with time-bound keys and batch verification are suggested as a result of the prior art in [44]. To provide conditional tracing, rapid verification, and dynamic device management, the group signature that is then proposed is implemented in both transactions and blocks.

## III. PRELIMINARIES

### A. BILINEAR PAIRING

*Definition 1 (Bilinear Pairing):* Let $G_1$, $G_2$, and $G_T$ be groups with prime order $p$. The asymmetric setting is used (Type-3 curves), i.e., $G_1 \neq G_2$, and no efficient isomorphism between $G_1$ and $G_2$. A mapping $\hat{e} : G_1 \times G_2 \longrightarrow G_T$ is called an admissible bilinear pairing if it satisfies the following properties.

1) Bilinear: For all $V \in G_1, Q \in G_2$, and $a, b \in Z_p^*$, have $e(V^a, Q^b) = e(V, Q)^{ab}$.
2) Nondegenerate: There exists $Q \in G_2$ such that $e(V, Q) \neq 1_{G_T}$.
3) Computable: Map $\hat{e}$ is efficiently computable. Let $D = (G_1, G_2, G_T, e, g_1, g_2)$, where $g_1$ and $g_2$ are the generators of $G_1$ and $G_2$, respectively.

### B. COMPLEXITY ASSUMPTIONS

*Discrete Logarithm (DL) Assumption:* On input $D$ and $X \in G_1$, there is no probabilistic polynomial time algorithm that outputs a value $x \in Z_p^*$ such that $g_1^x = X$ with nonnegligible probability. Note that DL assumption holds in $G_2$ and $G_T$ as well.

*Decisional Diffie–Hellman Assumption on $G_1$ (DDH1):* Let $x, y \in Z_p^*$. On input $D$ and $(g_1^x, g_1^y, Z \in G_1)$, there is no probabilistic polynomial time algorithm that can decide whether $Z = g_1^{xy}$ with nonnegligible probability.

*q-SDH Assumption:* Let $x, c \in Z_p^*$. On input $D$ and $(g_1^x, g_1^{x^2}, \ldots, g_1^{x^q}, g_2^x)$, there is no probabilistic polynomial time algorithm that outputs $(c, g_1^{1/(x+c)})$ with nonnegligible probability.

*DLIN Assumption:* Let $a, b, c, d \in Z_p^*$. On input $D$ and $(g_1^c, g_2^c, g_1^d, g_2^d, g_2^a, g_2^{bc}, Z)$, there is no probabilistic polynomial time algorithm that can decide $Z = g_1^{d(a+b)}$ with nonnegligible probability.

## IV. PROBLEM STATEMENT

The basic framework and paradigm of an IoT consortium blockchain are presented here. Threat model and design objectives will then be covered.

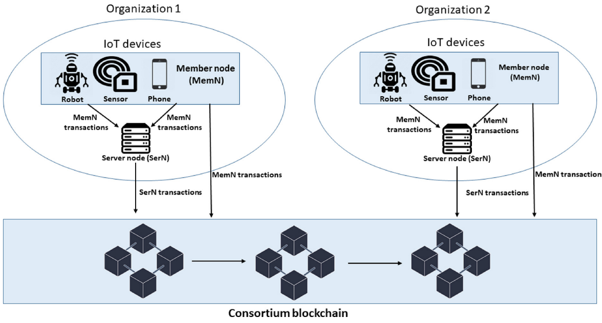

FIGURE 1. System model.

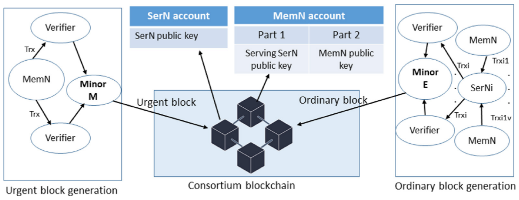

FIGURE 2. Application-oriented blocks and node accounts.

### A. SYSTEM MODEL

Fig. 1 depicts the planned blockchain architecture for the IoT consortium, which consists of two distinct types of nodes. A member node (*MemN*) and a server node *SerN*. A device that joins the *SerN* and the blockchain network is called a *MemN*. Each piece of hardware has a unique hardware identifier. A *SerN* is in charge of it. Every *SerN* is responsible for configuring the system and controlling the serving memories. The *SerN* allows for local storage of the source data created by its serving devices. Because they store a copy of every block added to the blockchain, *SerNs* can be considered full nodes as well. Metadata, such as an overview of the original material, keywords, and so on, are updated in the blockchain. Any *SerN* can alternate between mining and verifying in its turn. Transactions made and received in the blockchain by (*MemNs*) and *SerNs* are respectively referred to as *MemN* and *SerN* transactions.

### B. BLOCKS AND TRANSACTIONS

The proposed IoT consortium blockchain's application-oriented blocks are being designed. All the more exactly, a few applications may create various new exchanges in a brief time frame (e.g., specialists in clinical organizations might create some new clinical information during work hours. The transactions being discussed in this text are routine and do not require urgent attention. As seen in the accompanying Fig. 2, these transactions are grouped together into a *SerN* transaction for efficient verification and storage. All transactions go through the *SerN* routing system. The payments shown in Fig. 2 are then sent to verifiers. These payments are combined by diggers into a common block. However, in some cases, immediate access to data is necessary, such as when a doctor seeks for health record history of their patient in a rush. When in such a situations, the device requesting the data, such as an intelligent terminal used by a doctor, transmits *MemN* to the verifiers directly and with urgency. Transactions like this are merged into an urgent block by miners during a predetermined time interval. A block typically consists of four parts: the block title, payload, contributor's signature, and timestamp. The block header contains the identity, size, and hash value of the former block.

In payload, there is a sign for exchange types, that can be a pressing or a common block. There are n transactions in each Block, $Tr_{x1}, Tr_{x2}, \ldots, Tr_{xn}$ For an earnest block, every exchange comprises of exchange objective, exchange sources, exchange items, and the mark of the exchange generator. Each *SerN* transaction in a typical block is made up of *v MemN* transactions and the *SerN*'s signature. The structure of each *MemN* transaction is identical to that of the urgent block transaction. The Block includes a tamper-proof Merkle hash tree root. The block generator's signature is referred to as the contributor signature. A timestamp is added to indicate the generation time when a block is created.

### C. CONSENSUS MECHANISM

Diggers and additional blocks are chosen using an agreement instrument. Just *SerNs* has enlisted excavators. To handle time-sensitive and non-time-sensitive transactions, miners for both the urgent Block **M** and the regular Block **E** should be positioned beneath the blocks. Every round, two miners are randomly selected from the available *SerNs*. Others (SerNs) verify new transactions and relay the information to relevant miners. A new transaction is included in a block by the miner when it receives valid findings from over two-thirds of all verifiers, as required by PBFT. The block is added to the distributed ledger periodically, with the gap's size determined by the requirement. A terrible Block is often shorter than a regular Block. Urgent transactions are given priority over standard transactions when being processed by verifiers.

It is important to understand that in public-key infrastructures, the node's account functions as the public key and is used to verify the validity of signatures in blockchain verifiers. To enhance security measures such as restrictive following and dynamic gadget management, constructing a *MemN* (memory network) account is proposed using both the *SerN* (server network) and *MemN* public keys. Specifically, a *MemN* account consists of two components: 1) the *SerN* public key and 2) the *MemN* public key, as illustrated in Fig. 2. The group public key for a *SerN* is the key associated with the node account.

### D. THREAT MODELS

The proposed consortium blockchain-based IoT framework participants are expected to act in a trustworthy manner for successful smart contracts and conventions. They could be motivated by the interest of others, however, there is always a risk of a manager colluding with one member to falsely represent another individual for malicious or financial gains. Additionally, the possibility for a revoked member to misuse revoked keys to conduct transactions on the blockchain still stands.

### E. DESIGN GOALS

To combat the aforementioned threats, the subsequent objectives are prioritized.

- Data Security and Privacy Preservation. Many IoT data are personally identifiable, making it imperative to use the proposed method to guarantee data security and prevent any tampering with the original data. The most common tools for accomplishing these aims are encryption and marking. Identity reveals users' personal details while simultaneously protecting their data. The secrecy of the senders' and receivers' identities throughout a transaction is therefore paramount.
- Restrictive Discernibility. From one point of view, any observers of exchanges cannot discover the personality data for the devices without permission. Disputations, on the other hand, can be tracked back to malicious devices.
- Dynamic Device Management. Consensus should be used so that additional businesses can join the blockchain. In addition to individuals, corporate members can register for blockchain access. However, it is important to restrict the ability to submit blockchain transactions for revoked members of organizations.
- Conspiracy Obstruction. The leader may collaborate with one colleague to mask another in the above conditional traceability objective. As a result, the collusion resistance component should be included in the proposed strategy. A device with a blockchain account, for example, can send a nonrepudiation transaction like using the assistance of its team. However, if it did not initiate a transaction, it cannot claim it through collaboration with its group management.
- Effective Transaction Check. Transaction verification is performed by the majority of nodes in a consensus mechanism. In contrast, some applications may produce a flood of simultaneous financial transactions. Therefore, the signature algorithm needs to be carefully crafted to increase the efficiency of the verification.

## V. DYNAMIC APPLICATION BLOCK GENERATION (DABG) PROTOCOL

DABG is described in this section. This is followed by a detailed analysis of the protocol.

### A. AN OVERVIEW

The device can send transactions to the blockchain once it has registered as a *SerN* and become a *MemN* of the blockchain. Keep in mind that the proposed framework distinguishes between two types of transactions. 1) a routine transaction; and 2) an emergency transaction. Finding a signature for the transaction will require,

**Algorithm 1** Group Signature $Sign_{group}(SPK, SK_u, M, t_{sigexp})$

```
Input: The group public key $SPK$, the signers secret key $SK_u$, message $M$ and signature expiration time $t_{sigexp}$.
Output: The group signature $sig$.
If $t_{sigexp} > t_{exp}$, return $\perp$ and break. Otherwise, compute $\{t_{sigexpj}\}_{j\in[1,l]} \longleftarrow 0 - ENC(t_{sigexp})$. Find an integer m which satisfies $t_{sigexpm} = t_{expm}$.
Randomly choose $\varrho \in Z_p^*$ and compute $Tr_1 = g_1^{\varrho}, Tr_2 = P_m r_1^{\varrho}$.
Randomly choose $\iota \in Z{*}_p$ and compute $Tr_3 = (F_1 g_1^f)^{\iota}, Tr_4 = g_1^{\iota}, Tr_5 = g_2^{\iota}, Tr_6 = g_2^{f\varrho}, \mu = \varrho x m$.
Randomly choose $w_{\varrho}, w_{\iota}, w_x, w_f, w_k, w_k, w_{\mu} \in Z_p^*$ and computing the following:
$W_1 = Tr_1^{w_x} g_1^{-w_{\mu}}, W_2 = Tr_4^{w_f+w_k}, W_3 = A_2^{(f+k)\iota}, W_4 = A_1^{w_{\mu}} A_2^{w_f+w_h} A_3^{t_{signexp} w_{\varrho}} \hat{e}(Tr_2, g_2)^{-w_x}$.
Compute $c = H(Tr_1, Tr_2, Tr_3, Tr_4, Tr_5, Tr_6, W_2, W_2, W_3, W_4, t_{signexp}, m, M)$, $s_{\varrho} = w_{\varrho}+c_{\varrho}, s_{\iota} = w_{\iota}+c_{\iota}, s_x = w_x+c_x, s_f = w_f + c_f, s_k = r_k + c_k, s_{\mu} = w_{\mu} + c_{\mu}, s_h = w_h + c_h$.
Output $sig = (Tr_1, Tr_2, Tr_3, Tr_4, Tr_5, Tr_6, c, s_{\varrho}, s_{\iota}, s_x, s_f, s_k, s_{\mu}, s_h, W_4, t_{signexp}, m, M)$.
```

**Algorithm 2** Verification Check $Verify_{group}(sig, SPK, \Delta, RL, t_{sigexpc})$

```
Input: The group signature $sig$, the $(SerN)$'s group public key $SPK$, the $(MemN)$'s public key $\Delta$, revocation list $RL$ and the current time $t_{sigexpc}$.
Output: Valid/Invalid.
If $t_{sigexp} < t_{sigexpc}$, return Invalid and break. Otherwise, compute $\{t_{sigexpj}\}_{j\in[1,l]} \longleftarrow 0 - ENC(t_{sigexp})$. Find an integer m which satisfies $t_{sigexpm} = t_{expm}$.
Revocation check. For each $F_2 \in RL$, check whether $\hat{e}(Tr_1, F_2)\hat{e}(g_1, Tr_6) = \hat{e}(Tr_1, \Delta)$. If the equation holds, return Invalid and break.
Compute $W_1' = Tr_1^{s_x} g_1^{-s_{\mu}}, W_2' = Tr_4^{s_f+s_k} Tr_3^{-c}, W_3' = \hat{e}(Tr_4, \Delta)$.
Check whether $c = H(Tr_1, Tr_2, Tr_3, Tr_4, Tr_5, Tr_6, W_1', W_2', W_3', W_4, t_{sigexp}, m, M)$. If it doesnt hold, return Invalid and break.
Check whether $W_4 = A_1^{s_{\mu}} A_2^{s_f+s_h} A_3^{t_{sigexpm} s_{\varrho}} \hat{e}(Tr_2, g_2)^{-s_x} \hat{e}(Tr_2, r_2)^{-t_{sigexpm} c}$. If it holds, output Valid. Otherwise, output Invalid.
```

the *MemN* first employs group signature Algorithm 1. The transaction with the signature is then transmitted to the verification center. The verifiers examine the signature and submit the results to **M**, according to Algorithm 2. **M** goes further to include the transaction under an urgent block after he received the valid results. At a predetermined interval, the Block is uploaded to the blockchain.

For regular exchanges, the *MemNs* sign their exchanges with Algorithm 1 and send them to their serving *SerN* with marks. The *SerN* batch merges the confirmed $n$ transactions into a *SerN* transaction using Algorithm 3. The (*SerN* then signs the exchange and transmits it to the verifiers along with the mark. If more than two-thirds of the verifiers return Substantial, the **M** merges the *SerN* trades into a standard block.

### B. PROTOCOL DESCRIPTION

The four stages of the DABG procedure are as follows: 1) system installation; 2) registration; 3) generating blocks; and 4) dispute treatment

1) System installation. Each $(SerN)$ calls $Gen(\iota)$ to produce the serving organization's system parameters. This step accomplishes the following:
   - Given the system parameter $\iota, G_1, G_2$, and $G_T$ are prime order $p$ groups, with $g_1$ and $g_2$ functioning as their respective generators. $\hat{e} : G_1 \times G_2 \longrightarrow G_T$ denotes a bilinear guide.
   - The $(SerN)$ randomly chooses $\beta \in Z_p^*$ and computes $r_1 = g_1^{\beta}$ and $r_2 = g_2^{\beta}$.
   - $(SerN)$ chooses $h_1 \in G_1$ and two hash functions, $H_0 : \{0, 1\}^* \times G1 \times G_2 \longrightarrow Z_p^*, H_1 : G_1 \times G_1 \times G_1 \times G_1 \longrightarrow Z_p^*$. Moreover, the $(SerN)$ chooses a digital signature function $SIG_{sk}(.)$ and the corresponding verifying function $VER_{pk}(.)$, where (sk,pk) are the public key and the secret key of the signer.
   - The system parameter is $Sys_{PARA} = (g_1, g_2, h_1, G_1, G_2, G_T, \hat{e}, r_1, r_2, H_0, H1Sig, VER)$. The public key of the $(SerN)$ is $SPK = (r_1, r_2)$, which is also the group public key. The secret key of the $(SerN)$ is $\beta$.
2) Registration.
   A device registers to the *SerN* with its identity and performs $Regi(ID_u, Sys_{PARA})$; if it wishes to join an organization. Without loss of over-simplification, a gadget with character $ID_u$ registers to an association by connecting with the serving *SerN* as follows.
   - $F_1 = g_1^f$ is calculated after the device selects $f \in Z_p^*$ at random; $F_1 = g_1^f, F_2 = g_2^f$, and $F' = h_1^f$. To demonstrate the information on $f$, the gadget haphazardly picks $y_f \in Z_p^*$ and registers $W = g_1^{y_f}$

**Algorithm 3** Batch Verification $Verify_{Batch}$ $(tr_{c1}, tr_{c2}, ..., tr_{cv}, sig_{t1}, sig_{t2}, ..., sig_{tv}, \Delta_1, \Delta_2, ..., \Delta_v, RL, t_{sigexpc})$

**Input:** The group signatures of the $v$ transactions $(tr_{c1}, tr_{c2}, ..., tr_{cv}, sig_{t1}, sig_{t2}, ..., sig_{tv}$, the $(SerN)$'s group public key $SPK$, the revocation list $RL$, the nodes' public key $\Delta_1, \Delta_2, ..., \Delta_v$ and the current time $t_{sigexpc}$.

**Output:** Valid/Invalid.

```
1: If $t_{sigexpi} > t_{sigexpc}$, compute $\{t_{sigexpij}\}_{j \in [1,l]} \longleftarrow 0 - ENC(t_{sigexpi})$ For each $i \in [1, v]$ and find the $t_{sigexpimi}$.
2: Revocation check. For each $F_{2i} \in RL$, check whether $\hat{e}(Tr_{1i}, F_{2i})\hat{e}(g_1, Tr_{6i}) = \hat{e}(Tr_{1i}, \Delta_i)$. If the equation holds, return Invalid and break.
3: Compute $W'_{1i} = Tr_1^{s_{x_i}} g_{1i}^{-s_{\mu i}}$, $W'_{2i} = Tr_{4i}^{s_{fi}+s_{ki}} Tr_{3i}^{-c}$, $W'_{3i} = \hat{e}(Tr_{4i}, \Delta_i)$.
4: Check whether $c = H(Tr_{1i}, Tr_{2i}, Tr_{3i}, Tr_{4i}, Tr_{5i}, Tr_{6i}, W'_{1i}, W'_{2i}, W'_{3i}, W_{4i}, t_{sigexpi}, m_i, Tr_{ci})$.
5: If all the equations hold, randomly choose $\Upsilon_1, \Upsilon_2, ..., \Upsilon_n \in Z_p^*$ and check whether $\Pi_{i=1}^{v} W_{4i}^{\Upsilon_i} = A_1^{\Sigma_{i=1}^{v} s_{\mu_i}\Upsilon_i} A_2^{\Sigma_{i=1}^{v} s_{f_i}\Upsilon_i + s_{h_i}\Upsilon_i} A_3^{\Sigma_{i=1}^{v} t_{sigexpimi} s_{\varrho_i}\Upsilon_i} e(\Pi_{i=1}^{v} Tr_{2i}^{\hat{}-s_{xi}\Upsilon_i}, g_2) e(\Pi_{i=1}^{v} Tr_{2i}^{t-\hat{s}igexpimic_i\Upsilon_i}, r_2)$.
6: If the equation holds, output Valid. Otherwise, output Invalid.
```

and $W' = h_1^{y_f}, c_f = H_1(F_1, W, F', W'), s_f = y_f + c_y y$. It delivers $(F_1, F_2, F', s_f, c_f)$ that ties its identity to the $SerNs$.

- The $SerNs$ consider the character's credibility in addition to $c_f$ and $\hat{e}(F_1, g_2)$, with $c_f = H_1(F_1, h_1^{s_f}/F'^{c_f}, F', g_1^{s_f}/F_1^{c_f})$, and $\hat{e}(F_1, g_2) = \hat{e}(g_1, F_2)$. Then it computes $\{t_{exp}j\}_{j \in [1,l]} \longleftarrow 1 - ENC(t_{exp})$ and sets the expiration time of the device to $t_{exp}$, $h = H_0(ID_u, F_1, F_2)$.
- $P_j = (F_1 g_1^h)^{1/(a_j + t_{exp}\beta)}$, where $a_j + t_{exp}\beta \neq 0$, is selected by the $SerN$ after selecting $a_j \in z_p^*$. The $(SerNs)$ sends $(t_{exp}, \{a_j, P_j\}_{j \in [1,l]})$ to the device through a secure channel. Furthermore, for tracing and revocation, the $SerN$ stores the identity of each registered device in conjunction with the necessary secret key $(t_{exp}, \{a_j, P_j\}_{j \in [1,l]}, F_1, F_2)$.
- The device computes $\{t_{exp}\}_{j \in [1,l]} \longleftarrow 1 - ENC(t_{exp})$ for the received $t_{exp}$. For each $j \in \{1, 2, \ldots, l\}$, the device checks $\hat{e}(P_j, r_2^{t_{exp}} g_2^{a_j}) = \hat{e}(g_1, g_2)^{f+h}$, where $h = H_0(ID_u, F_1, F_2)$. If the prerequisites are met, the device recognizes the keys.

In addition, the device performs the following operations in order to join the blockchain as a $MemN$. It computes $\alpha = f + k$, $\Delta = g_2^{\alpha}$ after randomly selecting $k \in Z_p^*$. The device also sets its account address $AAD = (r_1, r_2, \Delta)$ and its public key $PK_u = \Delta$, becoming a blockchain member $MemN$. The $MemN$ keeps its secret key $SK_u = (f, \alpha, t_{exp}, \{a_j, P_j\}_{j \in [1,l]})$. Quite, to work on the effectiveness, the $MemN$ pre registers the accompanying qualities and stores them for later signature:

- $A_1 = \hat{e}(r_1, g_2)$.
- $A_2 = \hat{e}(g_1, g_2)$.
- $A_3 = \hat{e}(r_1, r_2)$.
- $\{t_{exp}\}_{j \in [1,l]} \longleftarrow 1 - ENC(t_{exp})$.

For the revocation members, the $SerN$ communicates $F_2$ to the blockchain. All the $SerNs$ keep a rundown of $RL = \{F_2\}$ for the revocation members.

3) Block generation.

Keep in mind that application-oriented blocks are built adaptively for different applications. These roadblocks might be both urgent and routine. The two methods for generating transaction generator signatures are covered in this stage, as the transaction generator's signature is required for generation. Expect a $MemN$ to need to establish a dire exchange, denoted by $Tr_c$, in the dire Block. The $MemN$ will generate a group signature for this transaction using Algorithm 1 and the signature expiration time $t_{sigexp}$. The $(MemN)$'s signature on the transaction is denoted as

$$sig_t = \big(tr_1, tr_2, tr_3, tr_4, tr_5, tr_{6i}, c, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_h, W, t_{t_{sigexp}}, m, Tr_c\big) \quad (1)$$

Following that, the $MemN$ broadcasts the transaction $Tr_c$, as well as the signature $sig_t$ and transaction kind sent to the verifiers. They then employ Algorithm 2 to validate an urgent transaction. The algorithm's output is given to miner **M**. He sends this operation into the urgent block but it is only after they get valid results from about two thirds of other verifiers dealing with it. It is then included to the blockchain at a preset time.

In contrast, assuming that $nMemNs$) transmits an regular transactions to their serving $SerN$ in the ordinary Block is arrogant. The $SerN$ does batch verification as follows:

$Verify_{Batch}$ $(tr_{c1}, tr_{c2}, \ldots, tr_{cv}, sig_{t1}, sig_{t2}, \ldots, sig_{tv})$ according to Algorithm 3, where

$$sig_i = \big(tr_{1i}, tr_{2i}, tr_{3i}, tr_{4i}, tr_{5i}, tr_{6i}, c_i, \alpha_{1i}, \alpha_{2i}, \alpha_{3i}, \alpha_{4i}, \alpha_{5i}, \alpha_{6i}, \alpha_{hi}, W_i, t_{sigexpi}, m_i, Tr_{ci}\big), i \in [1, v] \quad (2)$$

If the calculation returns Substantial, the $SerN$ integrates the $v$ exchanges into a $SerN$ exchange and signs on the exchange by performing $sig_\alpha = Sign_{SK_M}(Tr_{Xi1}, Tr_{Xi2}, \ldots, Tr_{Xiv})$. After that, the $SerN$ transaction is broadcasted, together with the transaction type and signature $sig_\alpha$. They employ the $VER_{SPK}(sig_\alpha)$ algorithm to validate the *SerN* public key. The algorithm's output is delivered to miner **E**, who includes the transaction into an ordinary block after getting Valid from many other verifiers.

4) Dispute treatment.
For the situation that questions happen, the *SerN* of a disputable *MemN* is liable for following the genuine character furthermore, send a guarantee to others *SerNs* for judgment. Based on its signature *sig*, the *SerN* follows the *MemN* as follows:
$P_m^* = Tr_2/Tr_1^\beta$.
The *SerN* can figure out the personality of the gadget with the thing $P_m^*$. It also removes the comparing $F_1^*, ID^*$. The formula for the promise (*PROM*) is $PROM = (P_m^*, F_1^*, F_2^*, ID^*, g_2^{a_m}, t_{expm})$ and handed over to the judge. In the wake of getting the commitment (*PROM*), the judger consolidates it to the mark *sig* and the *MemN*'s account (including *SPK* and $\Delta^*$) for judgment by achieving the accompanying confirmation:

$$\begin{aligned}\hat{e}(Tr_2, g_2) =& \\ &= \hat{e}(Tr_2, g_2) = \hat{e}(Tr_1, r_2)\hat{e}\left(P_m^*, g_2\right) \\ &= \hat{e}\left(P_m^*, g_2^{a_m}\right)\hat{e}\left(P_m^*, r_2\right)^{t_m}\hat{e}(Tr_4, \Delta) \\ &= \hat{e}\left(F_1^* g_1^h, g_2\right)\end{aligned} \tag{3}$$

The *MemN* that has the identity $ID^*$ is regarded the spiteful node if both equations hold.

### C. CORRECTNESS PROOF

If the signer is removed, the equation $\hat{e}(Tr_1, F_2)\hat{e}(g_1, Tr_6) = \hat{e}(Tr_1, \Delta)$ holds since

$$\begin{aligned}\hat{e}(Tr_1, F_2)\hat{e}(g_1, Tr_6) &= \hat{e}\left(g_1^\varrho, g_2^f\right)\hat{e}\left(g_1, g_2^{k\varrho}\right) \\ &= \hat{e}(g_1, g_2)^{(f+k)\varrho} \\ &= \hat{e}(Tr_1, \Delta)\end{aligned} \tag{4}$$

The verifier checks to see if $H$'s output matches $c$. Since the signature also includes the other inputs, the values of $W_1'$, $W_2'$, and $W_3'$ are shown to match $W_1$, $W_2$, and $W_3$.

$$\begin{aligned}W_1' =& \\ &= Tr_1^{\alpha_x} g_1^{-\alpha_\mu}, \\ &= g_1^{\varrho(w_x+c_x)} g_1^{-w_\mu+c_\mu} \\ &= Tr_1^{w_\varrho} g_1^{-w_\mu} \\ &= W_1\end{aligned} \tag{5}$$

$$\begin{aligned}W_2' =& \\ &= Tr_4^{\alpha_f+\alpha_k} Tr_3^{-c}, \\ &= \left(g_1^\iota\right)^{\left(w_f+c_f+w_k+c_k\right)} g_1^{-c_\iota(f+k)} \\ &= g_1^{\iota\left(w_f+w_k\right)} \\ &= W_2\end{aligned} \tag{6}$$

$$\begin{aligned}W_3' =& \\ &= \hat{e}(Tr_4, \Delta) \\ &= \hat{e}\left(g_1^\iota, g_2^{f+k}\right) \\ &= \hat{e}(g_1, g_2)^{(f+k)\iota} \\ &= W_3\end{aligned} \tag{7}$$

For more information regarding Security Definition and Security Proof, please refer to [44] and [25] in order to avoid rewriting the wheel. They demonstrated that the verifier and signer both calculated the same values for $W_4$. In addition, [44] provides definitions of nonframeability, traceability, and anonymity. Reference [44] refers to the definitions, security experiments, and proofs.

*Definition 2 (Anonymity)*: Given a gathering mark, if no other polynomial-time opponent can which among the sets is the real generator.

*Definition 3 (Traceability):* A genuine signature can be tracked, and no adversary can fake it in polynomial time.

*Definition 4 (Nonframeability):* Even if the group manager is tainted, no polynomial-time adversary can counterfeit a signature that is valid and that can be traced back to belong to a real member.

*Theorem 1:* The suggested group signature is anonymous in the random oracle under the DDH1 and DLIN assumptions.

*Theorem 2:* Both hypotheses may be traced back to the random oracle knowing the secret key (KOSK) assumption and the proposed group signature.

*Theorem 3:* Under the DL assumption, is assumed that the suggested group signature can attain nonframeability in the random oracle.

## VI. SECURITY ANALYSIS

In this part, whether and how the suggested method accomplishes the said design goals is examined.

- *Ensuring the security of data and protecting privacy:* According to the suggested layout, the (*SerN*)'s local storage should encrypt the original data and keep it there as ciphertext. The blockchain's metadata is useful for checking the accuracy of stored information. Information stored on a blockchain cannot be changed because of its immutable nature. Additionally, data integrity is safeguarded by the blocks' and transactions' digital signatures. Theorem 1 states that third-party observers of a transaction are unable to learn the true identity of the parties involved. Because the identities of the nodes involved in a transaction are not tied to the accounts of those nodes, the transactions can be conducted anonymously. Because of this, both parties to a transaction can remain anonymous.
- *Constructing Conditional Traceability:* The proposed group signature can be traceable based on Theorem 2. The serving *SerN* of the disputable *MemN* assumes responsibility for tracing the real identity, as presented in the dispute treatment phase. As a result, the *SerN*

can use its secret key $\beta$ to discover the *MemN*'s secret $P_m^*$. Likewise, the *SerN* can find out the comparing personality $ID^*$ and $F_1^*$ in its recordtable. Eminently, some other hubs or elements cannot find the genuine personality because they lack the secret key $\beta$ and the record table. The proposed scheme can resist collusion, so the claimed identity $ID^*$ is trustworthy.

- *Obtaining Resistance to Conditional Collusion:* The transaction is initially nonrepudiable after its signature is validated. Thus, to ensure that there would be no conflicts, the (*SerN*) must send the promise $PROM = (P_m^*, F_1^*, ID^*)$ to the judge in disagreements, as described in the dispute treatment phase. The promise's components $P_m^*$ and $F_1^*$ were included in the signature, (see Algorithm 1). It was as: $\hat{e}(Tr_2, g_2) = \hat{e}(Tr_1, r_2)\hat{e}(P_m^*, g_2)$, which includes the object $Tr_2$ and the stated $P_m^*$. Because $Tr_2$ is tied to $P_m^*$, it is no longer valid if *SerN* conspires with the group manager and gives another item, $P'_m$. Furthermore, it is linked to the elements $(Tr_2, Tr_4)$ of the asserted signature $(P_m^*, F_1^*, ID^*)$ and the *MemN*'s account. Furthermore, spurious $(P_m^*, F_1^*, ID^*)$ do not satisfy $\hat{e}(P_m^*, g_2^{a_m})\hat{e}(P_m^*, r_2)^{t_{expm}}\hat{e}(Tr_4, \Delta) = \hat{e}(F_1^* g_1^h, g_2)$. The two requirements show that this exchange was sent by the hub account $\Delta^*$ with legitimate personality $ID^*$. Notably, the *SerN* and *MemN* generate the items $P_m^*$ and $F^*$ separately. This certificate architecture ensures group nonrepudiation and avoids the issue of key escrow.

  According to Theorem 3, the proposed group signature is nonframeable, which means that a *MemN* cannot claim a transaction that is not its own. Without a doubt, a *MemN* with account $(SPK, \Delta')$ and personality $ID'$ is expected to guarantee a legal exchange produced by account $\Delta^*$ [with authentic personality $ID^*$ and secret $(P_m^*, F_1^*)$, in collaboration with the *SerN* assistance. $PROM = (P_m^*, F_1^*, ID')$ may make a pledge to the judger from the *SerN*. Because $Tr_2$ is connected to $P_m^*, e(Tr2; g2) = e(Tr1; r2)e(P_m; g2)$ holds in this situation. $\hat{e}(P_m^*, g_2^{a_m})\hat{e}(P_m^*, r_2)^{t_{expm}}$, on the other hand, The equation $\hat{e}(Tr_4, \Delta) = \hat{e}(F_1^* g_1^h, g_2)$ does not hold because the judge will use $ID'$ rather than the true $ID^*$. As a result, collusive behavior is observed.
- *Setting up Dynamic Device Management:* Time-bound the key is familiar with acknowledging normal termination in the suggested bunch signature calculation. To be more exact, the key expiration time of the registering *MemN Ser* is set to $t_{exp}$, while the signature expiration time of the *MemN* is set to $t_{sigexp}$ $(t_{sigexp} \leq t_{exp})$. If a suspicious *MemN* signs the exchange using the terminated key: it becomes $(t_{sigexp} > t_{exp})$, it won't be able to discover a whole integer $m$ that fulfills $t_{sigexpm} = t_{expm}$. Despite the fact that Algorithm 1 randomly chooses $t_{sigexpm'} \in \{t_{\Delta j}\}_{j \in [1,l]}$ for the signature transaction, $g_1^{f+k} \neq P^{x+\beta} t_{sigexpm}$. As a result, stage five of Algorithm 2 does not pass. If it happens, the mark usually disappears after the text. It controls the gadget automatically, requiring little user interaction and computing work. In the case if it happens an *SerN* forced to prematurely stop a *MemN*, the *SerN* can show a reversal token $F_2$ to the blockchain. These are retained by all *SerNs*. As a result, Algorithm 2 examines step 2 for revoked *MemNs*.
- *Improving Transaction Verification Efficiency:* To further develop exchange confirmation productivity, the serving *SerN* group confirms $v$ exchanges, which to a great extent eases verifiers from unreasonable confirmations for continuous exchanges in a customary block. The transaction can be handled promptly in an urgent block, but the system's overhead may rise. As a result, time efficiency and computational burden must be balanced.

## VII. PERFORMANCE EVALUATION

In this part, the proposed conspire is carried out and the exhibition is assessed. The platform and the parameter settings are first shown. The overheads associated with storage and communication are then examined. Later, the Miracle library and the C++ programming language were used to implement the proposed cryptography primitives. At last, DABG is worked on the Ethereum stage to assess its presentation.

The computing efficiency is measured using the elliptic curve implementation of cryptography primitives. A 455-b BLS bend of implanting with a degree 12 is chosen to attain a security of 128-b. the setting is similar to the one by Emura et al. [46]. The sizes of the analogous elements in $Z_p^*, G_1, G_2$, and $G_T$ are 39, 58, 115, and 456, respectively. The application environment consists of a machine running Microsoft Windows 11 and an Intel Core i7-7700 CPU which is running at 3.60 GHz. Furthermore, an Ethereum-backed local private test chain is being constructed on the macOS platform. The robustness compiler is solc@ 0.4.25, and the test system for savvy contracts is mocha@6.2.0. The Nodejs Web3js module is used to collaborate with brilliant agreements on a blockchain to assess the time cost of sending exchanges. Due to space limits, the complexities of organizational interaction are neglected. Because there are no analogous works in consortium blockchain that meet the same security objectives as ours, it is presumed that group signatures [46] and Fang and Feng [44] are legitimate. Emura et al. [46] are implemented in DABG and compared to this work results.

### A. STORAGE OVERHEAD

The suggested framework has two key elements: *SerN* and *MemN* storage overhead is investigated as a result. The storage overhead of every Block is taken into account since each of their influences the storage of nodes. $|G_1|$, $|G_2|$, and $|G_T|$; Elements in groups $G_1, G_2$, and $G_T$ have sizes represented by the symbols $|G_1|$, $|G_2|$, and $|G_T|$, respectively. In $Z_p^*$, the element size is $|Q|$. The *SerN* stores the system parameters

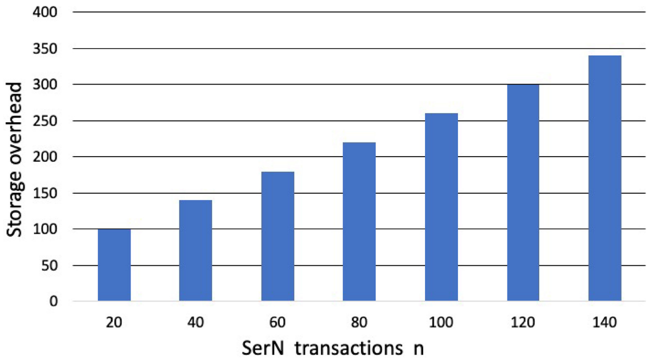

FIGURE 3. Urgent block storage overhead.

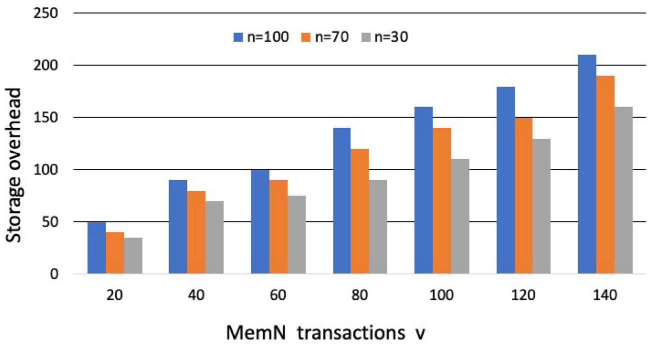

FIGURE 4. Ordinary block storage overhead.

$Sys_{PARA}$ and secret key $\beta$, whereas *MemN* registers its partial secret and real identities $(t_{exp}, \{a_j, P_j\}_{j\in[1,l]\}}, F_1, F_2, ID)$ and the *RL* revocation list.

In $Sys_{PARA}$, the components $(g_1, g_2, h_1, r_1, r_2)$ require the *SerN* $3|G_1| + 2|G_2|$ capacity. Each *MemN* that registers has the idea $(t_{exp}, \{a_j, P_j\}_{j\in[1,l]\}}, F_1, F_2, ID)$. The size is $(|Q| + |G_1|)l + |G_1| + |G_2| + 18)$. The element $F_2$ of the revocation list *RL* consumes $|G_2|$ overhead. Assume the system has two revoked users and one registered user. As a result, the absolute stockpile in *SerN* above is $(l|G_1| + l|Q| + |G_1| + |G_2| + 15)n_1 + |G_2|n_2 + 2|G_1| + |G_2| + |Q|$.

The *MemN* must record its private and public keys $(SK_u, PK_u)$, the previously computed value $(\{t_{expj}\}_{j\in[1,l]}, A_1, A_2, A_3)$, and the elements $(g_1, g_2, h_1, r_1, r_2)$ in $Sys_{PARA}$. For $SK_u = (f, \Delta, t_{exp}, \{a_j, P_j\}_{j\in[1,l]})$ and $PK_u = \Delta$, they take up $l|G_1| + |G_2| + (l+2)|Q| + 9$ storage overhead. The precomputed $(A_1, A_2, A_3)$ are $G_T$ components. Their storage overhead will be thus $3|G_T|$. Each $\{t_{expj}\}_{j\in[1,l]}$ has a length of $l$ bits. Each of the collection comprises of one element. As a result, the overall length of $\{t_{expj}\}_{j\in[1,l]} = (l \times l)/8$ bytes. The all-out stockpiling in a *MemN* with the components $(g_1, g_2, h_1, r_1, r_2)$ in PARA is $(l+2)|G_1| + 2|G_2| + 2|G_T| + (l+2)|Q| + l^2/8 + 9$. For each Block, the lengths of the block ID is thirty two, the block size is fur, the hash value is thirty two and the Merkle tree root is thirty two B.

The contributor's signature and timestamp are both 9 B, and the type of the transaction type will be 1 B. Because the exchange components are distinct, the exchange duration of the earnest Block and the regular Block differs. Each essential trade in dire Block consists of the hub account *AAD* found in the exchange source and objective, the exchange content $Tr_c$, and exchange signature $sig_t$. *AAD* is identical to $AAD = (r_1, r_2, \Delta)$ and takes up $|G_1| + 2|G_2|$ of the aforementioned stockpiling. The length of Content $Tr_c$ is considered to be $|M_u|$. The length of the exchange signature $sig_t$ is $6|G_1| + 6|G_2| + |G_T| + 8|Q| + |M_u| + 10$. As a result, the storage overhead for each urgent transaction is $9|G_1| + 3|G_2| + |G_T| + 8|Q| + |M_u| + 10$. A serious block consumes $(6|G_1| + 6|G_2| + 8|Q| + |M_u| + 10)n + 119$ capacity above. In the urgent block, the storage cost for *SerN* is affected by $n$ transactions size as shown in fig. 3.

An ordinary block's $n$ (*SerN*) transactions are made up of the (*SerN*)'s signature and $v$ *MemN* transactions. *MemN* transactions are the same length as urgent transactions, but they have different content. Let $|M_o|$ indicate the dimension of the *MemN* content transaction. Each *MemN* operation is completed in $5|G_1|+4|G_2|+|G_T|+5|Q|+|M_o|+8$. Each *SerN* transaction is $(5|G_1| + 4|G_2| + |G_T| + 5|Q| + |M_u| + 8)v + 9$ in size and bears the signature of (*SerN*) $(9B)$. The total stockpile of a standard block above is $(5|G_1| + 4|G_2| + |G_T| + 5|Q| + |M_o| + 8)vn + 9n + 115$. In the ordinary block, its noticed that the storage cost for *MemN* is affected by $n$ and $v$ transactions size, thus storage cost for the ordinary block at *MemN* will be obviously increased as illustrated in Fig. 4.

If the gathering mark is not met, In DABG, are employed, and it is assumed that the *MemN* also precomputes and saves one $1 - ENC(t_{exp})$ to reduce computational overhead (the original study did not). *SerN* and *MemN* capacity is $(|Q| + |G_1|+8)n1+(l|Q|+8)n2+2G_1+2G_2+|Q|$ and $(l+2)|G_1|+ 2|G_2|+(l+3)|G_T|+l|Q|+l2/8+9$, respectively. Tables 1, 2 shows that the Fang and Feng [44] storage overhead in men has a litter that is smaller than the suggested design. This is due to the fact that the suggested group public and user secret keys for the scheme are slightly shorter compare to the ones found in Fang and Feng [44]. If Emura et al. [46]'s group signature is employed in DABG, the storage overheads for the *SerN* and *MemN* are $(l|G_1|+2l|Q|+9)n1+|G_1|n2+ 7|G_1|+2|G_2|+3|Q|$ and $(l+7)|G_1|+2|G_2|+6|G_T|+2l|Q|+ l2/8 + 9$, respectively. Because it's collection is longer, the public and client secret keys, the capacity above is greater than the intended conspire. 1, 2 contrasts them. Remember the node account comprise of the group public key and the user public key.

### B. COMMUNICATION OVERHEAD

The registration and block generation phases account for the majority of the communication overhead. The sender's communication overhead is taken into account. At the enrollment stage, the client sends $(F_1, F_2, F', s_f, c_f)$ for registration, which introduces $2|G_1| + |G_2| + 2|Q| + 9$ communication overhead with its identity to the (*SerN*). The responses of *SerN* using $(t_{exp}, \{a_j, P_j\}_{j\in[1,l]})$ the secret key,

TABLE 1. Storage overhead.

| **Schemes** | $SerN$ | $MemN$ |
|---|---|---|
| Emura et al. [46] | $(l+2)\lvert G_1\rvert+2\lvert G_2\rvert+(l+3)\lvert G_T\rvert+l\lvert Q\rvert+l^2/8+9$ | $(l\lvert Q\rvert+l\lvert G_1\rvert+9)n_1+(l\lvert Q\rvert+9)n_2+2G_1+2G_2+\lvert Q\rvert$ |
| Fang et al. [44] | $(l+7)\lvert G_1\rvert+2\lvert G_2\rvert+6\lvert G_T\rvert+2l\lvert Q\rvert+l^2/8+9$ | $(l\lvert G_1\rvert+2l\lvert Q\rvert+9)n_1+\lvert G_1\rvert n_2+7\lvert G_1\rvert+2\lvert G_2\rvert+3\lvert Q\rvert$ |
| The proposed | $(l\lvert G_1\rvert+l\lvert Q\rvert+\lvert G_1\rvert+\lvert G_2\rvert+15)n_1+\lvert G_2\rvert n_2+2\lvert G_1\rvert+\lvert G_2\rvert+\lvert Q\rvert$ | $(l+2)\lvert G_1\rvert+2\lvert G_2\rvert+2\lvert G_T\rvert+(l+2)\lvert Q\rvert+l^2/8+9$ |

TABLE 2. Storage overhead.

| **Schemes** | Urgent block | Ordinary block |
|---|---|---|
| Emura et al. [46] | $(5\lvert G_1\rvert+2\lvert G_2\rvert+\lvert G_T\rvert+4\lvert Q\rvert+\lvert M_u\rvert+10)vn+9n+119$ | $(5\lvert G_1\rvert+2\lvert G_2\rvert+\lvert G_T\rvert+4\lvert Q\rvert+\lvert M_u\rvert+10)n+119$ |
| Fang et al. [44] | $(7\lvert G_1\rvert+3\lvert G_2\rvert+11\lvert Q\rvert+\lvert M_o\rvert)vn+9n+119$ | $(7\lvert G_1\rvert+3\lvert G_2\rvert+11\lvert Q\rvert+\lvert M_u\rvert)n+119$ |
| The proposed | $(5\lvert G_1\rvert+4\lvert G_2\rvert+\lvert G_T\rvert+5\lvert Q\rvert+\lvert M_u\rvert+8\rvert)n+115$ | $(5\lvert G_1\rvert+4\lvert G_2\rvert+\lvert G_T\rvert+5\lvert Q\rvert+\lvert M_o\rvert+8\rvert)vn+9n+115$ |

TABLE 3. Communication overhead in registration phase.

| **Schemes** | $SerN$ | $MemN$ |
|---|---|---|
| Emura et al. [46] | $(2\lvert Q\rvert+\lvert G_1\rvert)l+9$ | $2\lvert G_1\rvert+2\lvert Q\rvert+9$ |
| Fang et al. [44] | $(\lvert Q\rvert+\lvert G_1\rvert)l+9$ | $\lvert C\rvert$ |
| The proposed | $(\lvert Q\rvert+\lvert G_1\rvert)l+9$ | $2\lvert G_1\rvert+\lvert G_2\rvert+2\lvert Q\rvert+9$ |

causing $(|Q| + |G_1|)l + 9$ correspondence cost. A critical exchange, a *MemN* is expected to communicate the exchange source, objective, content, and exchange signature. An urgent transaction's communication overhead for a *MemN* is $5|G_1|+4|G_2|+8|Q|+|M_u|+10$. Since an urgent block contains $n$ urgent transactions, broadcasting an urgent block incurs $(5|G_1|+4|G_2|+8|Q|+|M_u|+10)n+119$ communication overhead for a miner **M**. Each of the $n$ *SerN* transactions in a typical block is made up of $v$ *MemN* transactions. An ordinary transaction is sent to the *SerN* by a *MemN* at the cost of $5|G_1| + 4|G_2| + 8|Q| + |M_u| + 10$ communication overhead. The *SerN* epitomizes $v$ substantial *MemN* exchanges as an *SerN* exchange, which takes up $(5|G_1|+4|G_2|+8|Q|+|M_u|+10)v+9$ correspondence above. An ordinary block with a communication cost of $(5|G_1| + 4|G_2| + 8|Q| + |M_u| + 10)vn + 9n + 115$. must be broadcast by miner **E**.

Similarly, if Fang and Feng [44] group signatures and Emura et al. [46] are employed in DABG, and Tables 3, 4, 5, 6 compare the communication overhead to the suggested design. The join/step in Fang and Feng [44] and Emura et al. [46] serves similar purpose as step registration in the proposed work. In Fang and Feng [44], the element $|C|$ refers to ciphertext length provided by the client for enlistment demand. The differences in the correspondence above are due to the length of the gathering marks, as previously discussed. Unlike Fang and Feng [44], the suggested method has a somewhat larger communication overhead, which allows it to attain nonframeability at the expense of other communication expenses. While Emura et al. [46] work has the same security features as the suggested effort, the proposed approach has less overhead Emura et al. [46].

## C. COMPUTATIONAL COST

The suggested group signature is used to construct DABG, which determines the computing overhead. To estimate the time cost that the steps used in the proposed convention will take, calculations framework formation, enrollment, GroupSign, GroupVerify, and Batch Verify are carried out in C++ using the Miracle library. Fang and Feng [44] and Emura et al. [46] also evaluated in terms of computing expenses to the suggested method employing group signature algorithms. Table 7 shows the time expenses of the algorithms.

The plan used in Fang and Feng [44] for the public key during framework formation was the one scalar multiplication, resulting in the least amount of computation required. It takes the greatest time to produce system parameters since it involves three scalar multiplications. In the proposed method, a *MemN* must connect with the *SerN* for certificateless-based authentication in order to register. Furthermore, to improve signature creation efficiency, the *MemN* precomputes the three parings $A_1$, $A_2$, and $A_3$. As a result, the registration time for the proposed scheme is 290 milliseconds. Emura et al. [46]. [from the Table 7] and the suggested algorithm were found to have equivalent time and cost. This is because they have nearly identical measures of matching tasks and scalar duplication activities, which overwhelm the computation above. Nonetheless, Emura et al. [46] plan does not apply directly for the batch verification. Besides, Fang and Feng [44] presented the lowest time cost for the majority calculations, but it lacks the qualities of recognizability and certificate framework.

## D. THE IMPLEMENTATION OF DABG ETHEREUM

The implementation of DABG Ethereum: The length of transactions is first considered since it affects the time necessary to publish a transaction or block in a blockchain. To determine the transaction length, the sizes of items found in $G_1$, $G_2$, $G_T$, and $Z_p^*$, are used as indicated in Tables 3, 4, 5, 6. If these numbers are replaced for the suggested system, $|G_1| = 58B$, $|G_2| = 115B$, $|G_T| = 456B$, and $|Q| = 39B$; urgent transaction length, or ordinary transaction, and a *SerN* transaction is $1816 + |M_u|$, $1816 + |M_o|$, and $(1816 + |M_o|)v + 9B$. It is critical to remember that the lengths of the transaction content $|M_u|$ and $|M_o|$ may differ based on the application. Without losing generality, the urgent transaction is assumed to be data generating and the normal one is a keyword search transaction. The design

TABLE 4. Communication overhead in urgent block generation phase.

| **Schemes** | $MemN$ | Miner $\mathbf{M}$ |
|---|---|---|
| Emura et al. [46] | $7\|G_1\|+6\|G_2\|+11\|Q\|+\|M_u\|$ | $(7\|G_1\|+6\|G_2\|+11\|Q\|+\|M_u\|)n+119$ |
| Fang et al. [44] | $5\|G_1\|+2\|G_2\|+\|G_T\|+4\|Q\|+\|M_u\|+10$ | $(5\|G_1\|+2\|G_2\|+\|G_T\|+4\|Q\|+\|M_u\|+10)n+119$ |
| The proposed | $5\|G_1\|+4\|G_2\|+\|G_T\|+8\|Q\|+\|M_u\|+10$ | $(5\|G_1\|+4\|G_2\|+\|G_T\|+8\|Q\|+\|M_u\|+10)n+119$ |

TABLE 5. Communication overhead in urgent ordinary generation phase.

| **Schemes** | $MemN$ | $SerN$ |
|---|---|---|
| Emura et al. [46] | $7\|G_1\|+6\|G_2\|+11\|Q\|+\|M_o\|$ | $(7\|G_1\|+6\|G_2\|+11\|Q\|+\|M_o\|)n+119$ |
| Fang et al. [44] | $5\|G_1\|+2\|G_2\|+\|G_T\|+4\|Q\|+\|M_u\|+10$ | $(5\|G_1\|+2\|G_2\|+\|G_T\|+4\|Q\|+\|M_u\|+10)v+9$ |
| The proposed | $5\|G_1\|+4\|G_2\|+\|G_T\|+8\|Q\|+\|M_o\|+10$ | $(5\|G_1\|+4\|G_2\|+\|G_T\|+8\|Q\|+\|M_o\|+10)v+9$ |

TABLE 6. Communication overhead in urgent ordinary generation phase.

| **Schemes** | Miner $\mathbf{E}$ |
|---|---|
| Emura et al. [46] | $(7\|G_1\|+6\|G_2\|+11\|Q\|+\|M_o\|)vn+9n+119$ |
| Fang et al. [44] | $(5\|G_1\|+2\|G_2\|+\|G_T\|+4\|Q\|+\|M_u\|+10)vn+9n+119$ |
| The proposed | $(5\|G_1\|+4\|G_2\|+\|G_T\|+8\|Q\|+\|M_o\|+10)vn+9n+115$ |

TABLE 7. Computational cost in ms.

| **Algorithms** | Emura et al. [46] | Fang et al. [44] | The Proposed |
|---|---|---|---|
| System installation | 224 | 107 | 120 |
| Registration | 98 | 144 | 290 |
| GroupSign | 259 | 61 | 190 |
| GroupVerify | 217 | 77 | 200 |
| BatchVerify(n=10) | – | 805 | 2100 |
| BatchVerify(n=20) | – | 1547 | 3250 |

is intended to be used as a guideline during a keyword search as well as metadata age in the consortium blockchain. Thus, the values of $|M_u|$ and $|M_o|$ are approximately 150 and $900B$, respectively. Assume that $v = 10$ and $v = 20$ *MemN* transactions are combined separately using a *SerN* transaction. These transactions are published on the implemented local Ethereum network by padding the data in the defined transaction format with the corresponding length. A *SerN* transaction can have a length of 27169 or $54329B$. The gas utilized is recorded for the purpose of reporting the relevant transactions, where $1wei = 10^{-18}ether$. Fig. 5 and Fig. 6 shows the results, and it can be observed that the cost of time and gas used to implement a transaction slowly grows as the length also increases. This is due to the fact that publishing a transaction requires signing, transferring data, and confirming the transaction, all of which take more time and gas as the package grows in size.

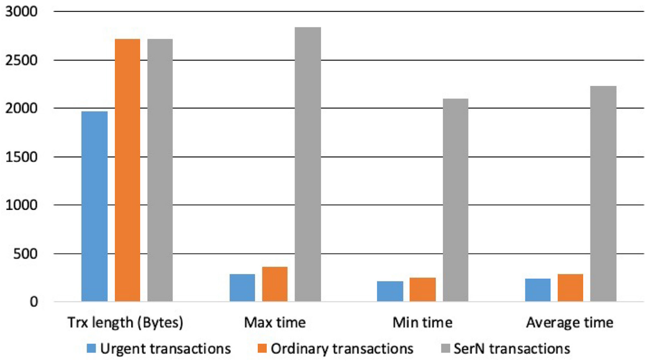

FIGURE 5. Time cost in ms for Publishing transactions.

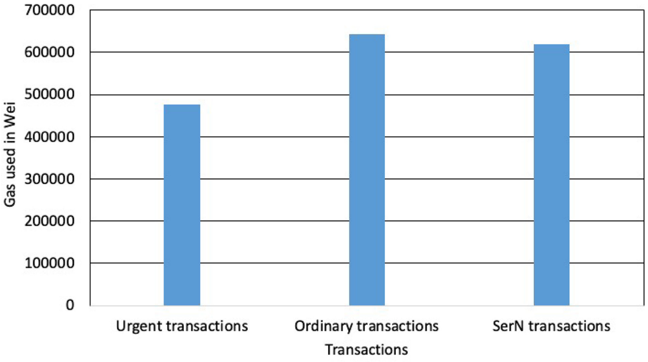

FIGURE 6. GAS used in Wei for Publishing transactions.

### E. COMPARISONS OF SECURITY PROPERTIES

This part shows detail analysis in terms of cryptographic techniques used and the comparisons of security properties of the existing studies [44], [46] including the proposed scheme. Table 8 compares the advantages and limitations for the cryptographic techniques used. Table 8 shows that the proposed scheme is based on certificateless cryptography that is more secure. References [44], [46] used ID-based cryptography that has three different stages; encryption, signature and decryption. Thus, the computations in this system will be increased. On the hand, certificateless cryptography can perform those three stages in one stage called signcryption.

Table 9 compares the proposed scheme security properties with other schemes [44], [46]. The results show that the proposed scheme meet all the security properties while [44], [46] meet some of them.

TABLE 8. Comparative analysis.

| **Schemes** | Cryptographic techniques | Advantages | Limitations |
|---|---|---|---|
| Emura et al. [46] | ID-based cryptography, encryption, signature and decryption | Public key can be an individual | key escrow problem |
| Fang et al. [44] | ID-based cryptography, encryption, signature and decryption | Public key can be an individual | key escrow problem |
| The proposed | Certificateless cryptography, signcryption | Prevent the key escrow problem | Bilinear pairing and Elliptic curve |

TABLE 9. Comparison of security properties.

| **properties** | Data security and privacy | Traceability | Revocation | Anonymity | Avoiding key escrow |
|---|---|---|---|---|---|
| Emura et al. [46] | √ | √ | × | √ | × |
| Fang et al. [44] | √ | × | × | √ | × |
| The proposed | √ | √ | √ | √ | √ |

## VIII. CONCLUSION AND FUTURE WORK

In this paper, a novel transaction processing scheme for IoT consortium blockchain adaptively with IoT applications is proposed. As a result, two distinct types of miners Miner **M** for processing urgent blocks and Miner **E** for processing normal blocks have been introduced. On this basis, a DABG protocol is being devised. Traceability, non-frameability and anonymity can be achieved by designing a certificateless group signature with time-bound keys and batch verification. The approach was developed to achieve dynamic device management, efficient transaction verification, conditional traceability, data security, and privacy preservation. As part of future research work, federated learning will be integrated to facilitate IoT-based applications to achieve the machine learning model. The plan is to deploy dropout-resilient protocol for privacy-preserving federated learning.

## REFERENCES

[1] P. M. Rao and B. Deebak, "Security and privacy issues in smart cities/industries: Technologies, applications, and challenges," *J. Ambient Intell. Human. Comput.*, vol. 14, pp. 10517–10553, Aug. 2023.

[2] A. M. A. Alamer, S. A. M. Basudan, and P. C. Hung, "A privacy-preserving scheme to support the detection of multiple similar request-real-time services in IoT application systems," *Exp. Syst. Appl.*, vol. 214, Mar. 2023, Art. no. 119005.

[3] S. Basudan, "A puncturable attribute-based data sharing scheme for the Internet of Medical Robotic Things," *Library Hi Tech*, vol. 40, no. 4, pp. 1064–1080, 2022.

[4] A. Alamer and S. Basudan, "An efficient truthfulness privacy-preserving tendering framework for vehicular fog computing," *Eng. Appl. Artif. Intell.*, vol. 91, May 2020, Art. no. 103583.

[5] Y. Yu et al., "LRCoin: Leakage-resilient cryptocurrency based on bitcoin for data trading in IoT," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4702–4710, Jun. 2019.

[6] H. Xue, D. Chen, N. Zhang, H.-N. Dai, and K. Yu, "Integration of blockchain and edge computing in Internet of Things: A survey," *Future Gener. Comput. Syst.*, vol. 144, pp. 307–326, Jul. 2023.

[7] A. Ahl, M. Goto, M. Yarime, K. Tanaka, and D. Sagawa, "Challenges and opportunities of blockchain energy applications: Interrelatedness among technological, economic, social, environmental, and institutional dimensions," *Renew. Sustain. Energy Rev.*, vol. 166, Sep. 2022, Art. no. 112623.

[8] A. Kumar and D. Das, "SioVChain: Efficient and secure blockchain based Internet of Vehicles (IoV)," in *Proc. 23rd Int. Conf. Distrib. Comput. Netw.*, 2022, pp. 284–289.

[9] M. J. Baucas, S. A. Gadsden, and P. Spachos, "IoT-based smart home device monitor using private blockchain technology and localization," *IEEE Netw. Lett.*, vol. 3, no. 2, pp. 52–55, Jun. 2021.

[10] M. Zhaofeng, W. Xiaochang, D. K. Jain, H. Khan, G. Hongmin, and W. Zhen, "A blockchain-based trusted data management scheme in edge computing," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 2013–2021, Mar. 2020.

[11] S. Biswas, K. Sharif, F. Li, S. Maharjan, S. P. Mohanty, and Y. Wang, "PoBT: A lightweight consensus algorithm for scalable IoT business blockchain," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2343–2355, Mar. 2020.

[12] J. Huang, L. Kong, G. Chen, M.-Y. Wu, X. Liu, and P. Zeng, "Towards secure Industrial IoT: Blockchain system with credit-based consensus mechanism," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3680–3689, Jun. 2019.

[13] M. Alaslani, F. Nawab, and B. Shihada, "Blockchain in IoT systems: End-to-end delay evaluation," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8332–8344, Oct. 2019.

[14] S. Basudan, X. Lin, and K. Sankaranarayanan, "A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 772–782, Jun. 2017.

[15] K. Lei, M. Du, J. Huang, and T. Jin, "Groupchain: Towards a scalable public blockchain in fog computing of IoT services computing," *IEEE Trans. Services Comput.*, vol. 13, no. 2, pp. 252–262, Mar./Apr. 2020.

[16] J. Wu, M. Dong, K. Ota, J. Li, and W. Yang, "Application-aware consensus management for software-defined intelligent blockchain in IoT," *IEEE Netw.*, vol. 34, no. 1, pp. 69–75, Jan./Feb. 2020.

[17] T. T. Anh Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling blockchain: A data processing view of blockchain systems," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1366–1385, Jul. 2018.

[18] B. K. Mohanta, D. Jena, U. Satapathy, and S. Patnaik, "Survey on IoT security: Challenges and solution using machine learning, artificial intelligence and blockchain technology," *Internet Things*, vol. 11, Sep. 2020, Art. no. 100227.

[19] M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, and H. Janicke, "Blockchain technologies for the Internet of Things: Research issues and challenges," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2188–2204, Apr. 2019.

[20] P. Bagchi et al., "Public blockchain-envisioned security scheme using post quantum lattice-based aggregate signature for Internet of Drones applications," *IEEE Trans. Veh. Technol.*, vol. 72, no. 8, pp. 10393–10408, Aug. 2023.

[21] N. Sivaselvan, V. Bhat, M. Rajarajan, and A. K. Das, "A new scalable and secure access control scheme using blockchain technology for IoT," *IEEE Trans. Netw. Service Manag.*, early access, Feb. 17, 2023, doi: 10.1109/TNSM.2023.3246120.

[22] P. M. Rao, S. Jangirala, S. Pedada, A. K. Das, and Y. Park, "Blockchain integration for IoT-enabled V2X communications: A comprehensive survey, security issues and challenges," *IEEE Access*, vol. 11, pp. 54476–54494, 2023.

[23] A. Vangala, A. K. Das, A. Mitra, S. K. Das, and Y. Park, "Blockchain-enabled authenticated key agreement scheme for mobile vehicles-assisted precision agricultural IoT networks," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 904–919, 2022.

[24] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "LSB: A lightweight scalable blockchain for IoT security and anonymity," *J. Parallel Distrib. Comput.*, vol. 134, pp. 180–197, Dec. 2019.

[25] A. Zhang, P. Zhang, H. Wang, and X. Lin, "Application-oriented block generation for consortium blockchain-based IoT systems with dynamic device management," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 7874–7888, May 2021.
[26] F. Gao, L. Zhu, M. Shen, K. Sharif, Z. Wan, and K. Ren, "A blockchain-based privacy-preserving payment mechanism for vehicle-to-grid networks," *IEEE Netw.*, vol. 32, no. 6, pp. 184–192, Nov./Dec. 2018.
[27] S. Pal, T. Rabehaja, M. Hitchens, V. Varadharajan, and A. Hill, "On the design of a flexible delegation model for the Internet of Things using blockchain," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3521–3530, May 2020.
[28] S. Latif, Z. Idrees, J. Ahmad, L. Zheng, and Z. Zou, "A blockchain-based architecture for secure and trustworthy operations in the Industrial Internet of Things," *J. Ind. Inf. Integr.*, vol. 21, Mar. 2021, Art. no. 100190.
[29] C. Lin, D. He, N. Kumar, X. Huang, P. Vijayakumar, and K.-K. R. Choo, "Homechain: A blockchain-based secure mutual authentication system for smart homes," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 818–829, Feb. 2020.
[30] B. Bera, D. Chattaraj, and A. K. Das, "Designing secure blockchain-based access control scheme in IoT-enabled Internet of Drones deployment," *Comput. Commun.*, vol. 153, pp. 229–249, Mar. 2020.
[31] S. Pal, S. Mukhopadhyay, and N. Suryadevara, "Development and progress in sensors and technologies for human emotion recognition," *Sensors*, vol. 21, no. 16, p. 5554, 2021.
[32] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, "Bubbles of trust: A decentralized blockchain-based authentication system for IoT," *Comput. Security*, vol. 78, pp. 126–142, Sep. 2018.
[33] Y. Yao, X. Chang, J. Mišić, V. B. Mišić, and L. Li, "BLA: Blockchain-assisted lightweight anonymous authentication for distributed vehicular fog services," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3775–3784, Apr. 2019.
[34] C. Lin, D. He, X. Huang, X. Xie, and K.-K. R. Choo, "Blockchain-based system for secure outsourcing of bilinear pairings," *Inf. Sci.*, vol. 527, pp. 590–601, Jul. 2020.
[35] C. Lin, D. He, X. Huang, K.-K. R. Choo, and A. V. Vasilakos, "BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0," *J. Netw. Comput. Appl.*, vol. 116, pp. 42–52, Aug. 2018.
[36] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1184–1195, Apr. 2018.
[37] S. Pal, T. Rabehaja, A. Hill, M. Hitchens, and V. Varadharajan, "On the integration of blockchain to the Internet of Things for enabling access right delegation," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2630–2639, Apr. 2020.
[38] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1594–1605, Apr. 2019.
[39] D. Chaum and E. V. Heyst, "Group signatures," *Advances in Cryptology EUROCRYPT91*. Berlin, Germany: Springer, 1991, pp. 8–11.
[40] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Proc. Workshop Theory Appl. Cryptograph. Techn.*, 2004, pp. 41–55.
[41] X. Yue, M. Xi, B. Chen, M. Gao, Y. He, and J. Xu, "A revocable group signatures scheme to provide privacy-preserving authentications," *Mobile Netw. Appl.*, vol. 26, no. 10, pp. 1412–1429, 2021.
[42] S. Basudan, X. Lin, and K. Sankaranarayanan, "An efficient compromised node revocation scheme in fog-assisted vehicular crowdsensing," in *Proc. IEEE Global Commun. Conf.*, 2017, pp. 1–6.
[43] N. Attrapadung, K. Emura, G. Hanaoka, and Y. Sakai, "A revocable group signature scheme from identity-based revocation techniques: Achieving constant-size revocation list," in *Proc. Appl. Cryptograph. Netw. Security 12th Int. Conf.*, 2014, pp. 419–437.
[44] J. Fang and T. Feng, "Group signature with time-bound keys and unforgeability of expiry time for smart cities," *EURASIP J. Wireless Commun. Netw.*, vol. 2021, no. 1, p. 113, 2021.
[45] L. Malina, J. Hajny, and V. Zeman, "Light-weight group signatures with time-bound membership," *Security Commun. Netw.*, vol. 9, no. 7, pp. 599–612, 2016.
[46] K. Emura, T. Hayashi, and A. Ishida, "Group signatures with time-bound keys revisited: A new model, an efficient construction, and its implementation," *IEEE Trans. Depend. Secure Comput.*, vol. 17, no. 2, pp. 292–305, Mar./Apr. 2017.
[47] A. Zhang and X. Lin, "Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain," *J. Med. Syst.*, vol. 42, no. 8, p. 140, 2018.

**SULTAN BASUDAN** received the Ph.D. degree in computer science from the University of Ontario Institute of Technology, Canada. He is an Associate Professor with the Faculty of Computer Science and Information Technology, Jazan University, Saudi Arabia. His research interests include computer and network security, privacy protection, applied cryptography, and software security.