

## RESEARCH ARTICLE

# Blockchain Based an Efficient and Secure Privacy Preserved Framework for Smart Cities

**ADLA PADMA**  **AND MANGAYARKARASI RAMAIAH** 

School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore, Tamil Nadu 632014, India

Corresponding author: Mangayarkarasi Ramaiah (rmangayarkarasi@vit.ac.in)

This work was supported by the Vellore Institute of Technology, Vellore, Tamil Nadu, India.

**ABSTRACT** Building smart services for smart cities has become a significant focus of the Internet of Things (IoT). These IoT devices are able to sense their surroundings and react appropriately. Smart city applications emphasize the necessity of safe data sharing across heterogeneous devices. Certain behaviors taken while sharing could aim at compromising security, privacy, and integrity. The centralized repository that is currently in place made the majority of hacks possible. The sharing of sensitive data and authentication are essential stages in guaranteeing the security of applications associated with IoT. Blockchain and IoT are two widely used technologies, with IoT focusing on data collection via various devices and blockchain enabling data integrity. This paper introduces a novel blockchain-based framework to ensure the security and integrity aspects of IoT data. The proposed SecPrivPreserve framework ensures security through various phases including initialization, registration, data protection, authentication, data access control, validation, and data sharing and download. Diverse security mechanisms such as passwords (OTP), encryption, and hashing have been deployed in various phases to strengthen security merits confidentiality, privacy, and integrity. Since the SecPrivPreserve framework is simulated in a permissioned blockchain platform the merits and tamper-proof and non-repudiation are automatically considered. Moreover, data protection uses Chebyshev polynomials and interpolation. The presented framework has experimented with Fabric SDK. The experimental results of the proposed framework are compared with the BaseLine state-of-the-frameworks, The experimental analysis reveals that the proposed SecPrivPreserve approach achieved 34 Sec improvement in terms of responsiveness 94 Sec as computational time, encryption quality as 0.87 Sec and 0.82 Sec for detection rate.


**INDEX TERMS** Authentication, blockchain, IoT, security, smart city, smart contracts.

## I. INTRODUCTION

Recently, all the nations in the world have been gearing up their services, applications, and infrastructure for the betterment of their people's life using smart technologies. In this context, the Internet of Things (IoT) is crucial for connecting physical devices to the internet using different protocols to facilitate data transfer among diverse places. In recent decades, there has been an enormous necessity for IoT-based services in various sectors such as healthcare, manufacturing, financial services, traffic monitoring, weather monitoring, and energy transfer [1]. Due to their compactness and minimal power consumption, the usage of IoT devices

is expected to reach more than \$1.4 trillion in 2027 [2]. Many countries invest a lot of money in initiatives relating to smart cities. For instance, China is engaged in more than 220 initiatives that aim to create a smart city and improve the quality of life for citizens. Associated technologies for smart cities assist urban municipalities in managing their day-to-day operations. According to IBM, the smart city has three main characteristics instrumented (sensors, actuators), interconnected (information sharing among devices), and intelligent (improve quality of citizens' life) [3].

Recent observation reveals that the smart city has substantially enhanced the quality of life and amenities of inhabitants in urban areas. According to a United Nations Population Fund report [4], more than half of the world's population lives in urban areas. The smart city has caught

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

the attention of both academia and business since it has significantly decreased the logistical problems related to acquiring services. Several cities worldwide have begun to build their own smart city strategies to improve their inhabitants' quality of life. IoT and smart environments have become synonymous. IoT technology is capable of sensing every entity in the real world, so it finds importance in healthcare, transport, traffic system, public safety, smart building, and smart agriculture. Amid many merits, due to the presence of inconsistent protocol standards, resource-constrained nature, and centralized repository IoT devices are vulnerable to security and privacy breaches [5], [6]. In a smart environment, people may face security and privacy risks due to the vulnerabilities in smart city applications. For instance, malicious attackers may fabricate data to execute their ill intent, which may jeopardize the decision-making system. In addition, these malicious attackers also make all sorts of attempts to prevent the legitimate users' service by executing denial-of-service (DoS) attacks, transmission, disrupting sensing, and control in order to degrade the quality of intelligent city services [7], [8].

Furthermore, as new devices or software are connected, the complexity level of the risks of smart city applications grows, particularly while ensuring privacy. Unfortunately, most protection methods (encryption, authentication mechanism) are insufficient to protect smart city applications against the new dynamic threats. Implementing complex procedures wouldn't be possible since the devices have less computational power. Hence, a simple framework that considers simple cryptography techniques would be an appropriate solution for IoT's heterogeneity and dynamic characteristics [9] is appreciable. Data breaches can occur during data storage, transmission, and sharing, posing significant risks to data owners and providers. Regulations are in place to protect the data source and the system from potential harm caused by target data nodes. As a result, during data transactions, it is imperative that both the source and target nodes comply with the policies and regulations of their respective areas [10].

Smart cities are built around integrating sensors and smart technologies, allowing citizens and organizations to access data through their smart devices to process and utilize data. However, the utilization of data in smart cities raises privacy concerns, including hacking sensitive data through injecting data poisoning attacks. These attacks could result in the alteration of sensitive data, which in turn leads to the disruption of communication within smart entities. IoT networks in smart cities are particularly susceptible to cyber-attacks that threaten the data integrity, confidentiality, and availability of these systems. To mitigate these risks, smart cities must implement robust security mechanisms to protect their assets against cyber-attacks (Distributed Denial of Service (DDoS), DoS, Man-in-the-Middle, ransomware). The frequency and impact of these attacks emphasize the need for adequate privacy and security measures in smart cities [11], [12]. Researchers have developed many data-securing schemes to offer privacy and security for applications meant

for smart cities. Earlier centralized cloud-based data-sharing frameworks have failed to address smart applications' data integrity and privacy issues. However, blockchain-based solutions provide greater improvement in solving privacy issues. Initially, data collected from sensors using a detection algorithm takes client data into various communities based on similarity labels. It has a specific type of control on community data with specifying detection algorithm. However, this framework has not addressed data protection while transferring the data through sensors [13].

Blockchain is one of the most promising ways to guarantee data integrity for Internet of Things applications. Because of its tamper-proofing and traceability features, it allows for transparent sharing services and decentralized data storage for smart services. The blockchain consists of individual blocks containing a block header, timestamp, transaction data, and a previous block hash. Every node has a complete copy of the transaction data to implement transparency, high overhead is placed on the system as a result. And finally, blocks are added on a time basis. In Smart city applications, multiple participants are making transactions at different instants of time and venues, and uploading data directly could place a significant load on the blockchain [14], [15]. Solutions built on the blockchain offer improved security options for smart cities. The ensured distributed environment through blockchain is appropriate for most of the smart applications [16], [17], [18]. Although blockchain is a promising application to solve smart cities' privacy and security challenges. Many IoT devices are unable to run complicated tasks due to their low power, limited data storage, and limited battery resources. Moreover, existing consensus like PoW in blockchain-based networks also demands more resources. In the mining process, nodes involved in the distributed network decision-making demand substantial computing power. Smart contracts are an additional intriguing blockchain application that implements multiple access restrictions on IoT smart applications. And also, the provenance of data is crucial to the security and privacy of smart applications [19], [20], [21], [22].

Access based access control (ABAC) provides the flexibility, and granularity needed to effectively protect the data collected and shared by smart entities, and fine-grained policies used to grant/deny various activities of smart ecosystems. However, ABAC faces challenges of complexity, overhead, and privacy before deploying the framework [23]. In the context of a smart city, diverse configuration devices and networks work together to complete the service. Since these devices are developed by different vendors ensuring compatibility while completing the task is challenging. Hence, to remove the issue raised through incompatibility open standard protocols (JSON, CoAP) have been considered as a viable solution. Through these open standard protocols, incompatibility occurred using diverse data formats addressed by the middleware layer by including mutual authentication for homogeneous devices and conditional probability authentication for heterogeneous devices [24].

In a smart ecosystem to protect user privacy and comply with regulatory frameworks like the European Union General Data Protection Regulation (EUGDPR). It follows the consent-based data processing (personal data processed with the consent of data owner), transparency and accountability (data collection, processing, and usage practices must be transparent by providing entities with clear information about how their data is being handled), integrity and accuracy (data must be accurate, complete, and up-to-date, and appropriate measures should be taken to prevent unauthorized data destruction), security and access control (security measures implemented to protect personal data from unauthorized access, use, disclosure, alteration, or destruction. And it should be restricted to access data only to authorized entities), and privacy by design (data protection measures should be embedded into the design of smart city technologies from the outset throughout the development process) [25].

Consequently, this paper presents a SecPrivPreserve framework based on the Hyperledger Fabric blockchain. The main objective of this research is to design an efficient and secure privacy-preserved framework through smart contracts.

The novel framework consists of seven different phases to successfully store and validate the client data to ensure the various security benefits privacy and integrity. The presented phases taxonomy table is shown in Table 1. Each participant identification number was produced using a random number string and hashed using SHA256, in the initialization phase to initiate the request. In the registration phase, different kinds of peers implement the enrollment task using passwords generated through the hashed random number (OTP) and current time. The data protection phase uses the AES algorithm (128-bit key length) to encrypt the client data. To increase the degree of confidentiality, interpolation, and Chebyshev polynomials are also deployed in the data protection phase. The MSP uses hashing to authenticate clients' data by using the digital signature. To facilitate data access control, hashed passwords are used to prevent data tampering. To download the data for verification purposes data sharing downloading is done through AES and Chebyshev polynomials. The major contributions of the proposed framework as follows:

- In this paper, we proposed an efficient and secure Privacy-preserving framework, namely SecPrivPreserve using Hyperledger Fabric blockchain intended to implement data protection for smart cities.
- To withstand security threats the presented framework encompasses the following steps: initialization, registration, data protection, authentication, data access control, validation, and data sharing and download.
- SecPrivPreserve framework uses encryption, and hashing passwords (OTP) and digital signature to increase the degree of security aspects.
- To prove its efficiency the tested results are compared with the well-established frameworks in the candidate domain using quantitative metrics.

## A. PAPER ORGANIZATION

The subsequent sections of the paper are structured in the following manner: Section II provides an overview of the existing literature. Section III presents an elaborate exposition of the SecPrivPreserve framework that is being proposed. Section IV presents an in-depth performance analysis and experimental findings. Finally, the conclusion provides a summary of the findings and suggests potential avenues for further research.

## II. LITERATURE REVIEW

The literature demonstrates a substantial need for integrating blockchain with various business logic due to the valuable properties of the blockchain. The most important aspect of a blockchain is that it is a peer-to-peer network with no central authority. Many researchers proposed various frameworks using blockchain technologies to enhance the security aspects of smart city applications. The immutability is the highlight of distributed ledger technology, grabbing more attention in developing robust data-sharing algorithms in smart cities. Ensuring privacy has been one of the main objectives of smart environment applications. privySharing is one of the attempts milestones for implementing privacy in smart cities. The decentralized network is segmented into various channels for the registered organizations to make efficient data sharing and security, and data are maintained through smart contracts. Experimented results reveal that the multi-channel extensively improves the latency and throughput compared to the results through the single-channel network. However, this framework still needs to address the fog node integration for device security [26].

A centralized cloud-based platform for cross-domain data sharing, incorporating blockchain technology to meet the demands of industrial requirements, is necessary to establish trust among various components of the Industrial IoT (IIoT). To mitigate the limitations of traditional cloud storage, data owners are storing their data on the blockchain. This enables the detection of any malicious actions attempted in centralized storage by reporting them to the blockchain. Implementing the Ephemeral Elliptic Curve Diffie-Hellman algorithm enhances the security features in line with industrial requirements. It is vital to strike a balance between security measures and financial considerations. However, the framework's capability to handle large-scale data needs to be further evaluated [27].

In the context of IoT, reducing third-party involvement can significantly enhance security. A blockchain-based IoT device gateway architecture presents a potential solution to this challenge. The data shared through this architecture is protected by cryptography techniques, ensuring the confidentiality and reliability of the data from remote servers. However, the robustness of the architecture against potential attacks in terms of IoT infrastructure has yet to be evaluated [28]. A blockchain-based authentication and data-sharing scheme is demonstrated to guarantee the integrity and

TABLE 1. Taxonomy table.

Phases	Encryption / Decryption	Hash	Digital signature	Security
Initialization		✓		Integrity
Registration		✓		Integrity
Data Protection	✓	✓		Confidentiality, Integrity
Authentication		✓	✓	Integrity, Authenticity
Access control		✓	✓	Integrity, Authenticity
Validation		✓	✓	Integrity, Authenticity
Data Sharing and download	✓	✓		Confidentiality, Integrity

privacy of IoT data. The presented model employs various security mechanisms at the registration, authentication, and transmission phases to detect the possible attack in the IoT context. The framework enhances its security aspects and ensures efficiency by completing the security procedures with minimal computation time [29].

The degree of confidentiality aspect will be improvised via the invention of a Proxy re-encryption scheme using blockchain. Such a platform is considered a viable alternative for managing IoT data. To maintain anonymity, the smart contract makes the data visible only to specific users. However, implementing a smart contract-based Access Control List (ACL) and auditing system for the cloud server powered by the blockchain proved ineffective in improving the security of the cloud. [30]. Recently machine learning algorithms have shown improvements in detecting anomalies while handling data from multiple IoT devices.

The optimization of trust-based smart city applications is a critical area of study and research. Authors [31] proposed a ChainComm framework for building a smart community system based on blockchain technology, to address the challenges of traditional community governance models, such as lack of transparency, centralization, and vulnerability to manipulation. This framework utilizes blockchain's inherent features to create a more secure, efficient, and trustworthy community governance system. However, it does not explicitly address the smart community implementation details concerning privacy-preservation citizens. A novel architecture that utilizes blockchain technology in combination with Federated Learning and a Trusted Execution Environment to effectively and securely preserve privacy in the context of Smart Cities. the authors introduced a proposed integrated architecture designed to enhance privacy preservation in the context of Smart Citizens. Mainly, focuses on the resolution of data privacy concerns by leveraging the combined potential of blockchain technology and edge/fog computing [32].

Ledger technology-based databases (LTBD) have emerged as a potential breakthrough, utilizing the power of distributed ledger technology to revolutionize data storage and management. These decentralized, transparent, and secure databases have tremendous potential in areas including banking, healthcare, and supply chain management.

However, LTBD suffers several challenges, including scalability concerns caused by consensus procedures and block size constraints. To fully realize the transformational potential of LTBD, energy usage, and privacy problems must be handled efficiently. Despite these problems, LTBD provides a promising route towards safe, transparent, and efficient data management systems, and continued research and development are critical to addressing limits and paving the road for mainstream implementation [33]. VeDB is a promising approach for establishing trusted relational database management that combines software-based and hardware-based trusted computing techniques to achieve high performance, strong auditability, and data integrity. Its versatility and security make it well-suited for a wide range of applications that require reliable and trustworthy data management. However, it requires high cost, complexity, and limited scalability to deploy the large-scale environment [34]. LedgerDB utilizes a tamper-proof Merkle tree to provide data integrity proofs, allowing users to verify the authenticity of data and it offers significantly higher throughput and lower latency compared to traditional blockchain systems. Additionally, it employs a two-way peg protocol with a trusted timestamp authority (TSA) to prevent malicious behavior from both users and service providers, ensuring data non-repudiation. However, this relies on trusted TSA which makes it susceptible to a single point of failure [35].

Additionally, energy consumption is a concern, as cryptographic operations and hardware requirements may increase energy demands. Moreover, privacy issues persist, as the immutability and traceability of data stored on the blockchain may raise concerns about data linkage and re-identification. despite all, the authors propose a private blockchain-based access control system (PBACS-PECIIoT) for IIoT applications to safeguard sensitive data and prevent unauthorized access. It offers a promising approach for securing IIoT-based pervasive edge computing environments. Further research is needed to address scalability, energy consumption, and privacy concerns, paving the way for its widespread adoption [36]. In another paper, the authors proposed a framework called FRUIT (eFficient and pRivacy-preserving qQuality-aware Incentive), a blockchain-based reward system for high-quality data contributions while maintaining anonymity. To safeguard data privacy and assure accurate quality



evaluation, it utilizes proxy re-encryption, matrix decomposition, and polynomial fitting. FRUIT uses smart contracts to handle incentives efficiently and the Dirichlet distribution to forecast reputation. However, FRUIT is limited in its capacity to handle possible privacy problems posed by blockchain's immutability and traceability [37].

Similarly, the authors propose a blockchain-enabled privacy-preserving framework to utilize machine learning techniques to analyze and extract meaningful insights from IoT data while preserving data privacy. It employs homomorphic encryption, secure multi-party computation, and differential privacy mechanisms to protect sensitive information during data processing and analysis. However, differential privacy introduces noise into data to protect individual privacy. This noise can degrade the accuracy of the extracted insights and make it more difficult to identify meaningful patterns [38]. Similarly in other work, the proposed scheme employs non-interactive zero-knowledge (NIZK) proofs to enable anonymous authentication of IoT devices without compromising device privacy. It ensures that devices can prove their identity to the operation center without revealing their actual identities, preventing tracking and profiling. Additionally, bilinear pairing-based cryptography establishes secure key agreements between the operation center and authenticated IoT devices. However, this should not focus on designing a revocation mechanism that can effectively remove malicious entities from the system without compromising the anonymity of legitimate users [39].

The authors [40] presented a secure Support Vector Machine (SVM) based approach along with blockchain to overcome the security issue of collecting training data from multiple data providers. A homomorphic cryptosystem paillier has been deployed to ensure the confidentiality of the sensitive data collected from various IoT devices and thus created secure building blocks.

In another work, integrating blockchain and AI enhances the security aspects of various use cases. Data sharing between IoT devices is empowered with the inclusion of blockchain along with machine learning techniques and federated learning (FL). FL-based systems prevent data leakage and privacy to a greater extent. The system exchanges the learning model in contradiction to data. However, the model still needs to improve the data utility [41]. The authors presented a blockchain-based data sharing and access control system. Here, multiple smart contracts are proposed to secure, authenticate, network management of users, and detect misbehavior of users. The penalty is formed for the detection of user misbehavior. Finally, the experimental results show that overall execution cost was highly reduced and enhanced data sharing security [42]. The traditional methods contributed to providing security to smart cities' data sharing using various blockchain-based frameworks presented in Table 2. This section accounts for the advantages of blockchain technology in enhancing the security of various services in smart environments while acknowledging its

potential shortcomings. It further emphasizes the significance of developing novel frameworks to proactively identify and mitigate emerging threats, thereby minimizing their impact.

### III. PROPOSED METHODOLOGY

#### A. BASIC TERMINOLOGY

Our model used specific imperative entities of the Hyperledger Fabric blockchain.

- **Client (C):** An authenticated client  $C_i, i \in 1, 2, \dots$ , is responsible to collect the particular records of user data from the blockchain.
- **Data Owner:** Data owner has all permissions on data like delete, update, and insert on user records.
- **Membership Service Providers (MSP):** Certificate Authorities (CA) are responsible for issuing X.509 certificates to network entities. MSP specifies which CA is permitted to participate in the blockchain network and uses this information to identify which peer nodes belong to which groups. MSP maintains the distributed ledger between organizations and associated systems that the network trusts.
- **Smart Contract (SC):** Smart contract register under MSP unit. SC is a program that transfers digital assets among specific constraints for information encapsulation and the automation of routine commercial transactions. The ledger keeps track of transactions generated by applications that have invoked smart contracts.
- **Endorsing Peers (EP):**  $EP_j, j \in 1, 2, \dots$ , is special peers to run smart contract and receive the transaction proposal for endorsement policy, these peers are responsible for either grant or denies the endorsement submitted by the respective channel.
- **Ordering Peers (OP):** OP is responsible for the inclusion of transaction blocks into the distributed ledger.
- **Committing Peers (CP):** The  $CP_k, k \in 1, 2, \dots$ , is responsible for validating and committing transactions from OP. However, a committing peer does not contain a smart contract setup. then, blocks are added to the blockchain and maintain the ledger state. s
- **Channel:** Organization peers are communicated through channels. Multiple channels are allowed within a network to interact with peer nodes.
- **Chaincode:** Smart contracts are distinct from chaincode in that they specify the transaction logic that modifies the world state of a business object. In contrast, a chaincode can be viewed as a technical container that contains multiple SC for implementation and deployment. Whenever a chaincode is deployed, applications can access all of its smart contracts.

#### B. SECURE PRIVACY-PRESERVING FRAMEWORK

Every user of smart city applications would require that the sensitive and private information gathered by different

**TABLE 2. The review of traditional blockchain models for IoT.**

Reference	Technique/Method used	Key Contributions	Limitations	Metrics
[26]	<ul style="list-style-type: none"> <li>PoC (Proof of Concept)</li> </ul>	<ul style="list-style-type: none"> <li>Blockchain-enabled privacySharing schema for secure data sharing.</li> </ul>	<ul style="list-style-type: none"> <li>The system lacks the ability to accommodate fog nodes that use mobile BTS technology.</li> </ul>	<ul style="list-style-type: none"> <li>Latency</li> <li>Throughput</li> </ul>
[27]	<ul style="list-style-type: none"> <li>Ephemeral Elliptic Curve Die-Hellman (ECDHE), and Identity-Based Signature (IBS).</li> </ul>	<ul style="list-style-type: none"> <li>A cloud-based framework along with blockchain intent for IIoT. Penalization is executed through smart contracts.</li> </ul>	<ul style="list-style-type: none"> <li>The system needs to address large-scale data for data receivers and data providers.</li> </ul>	<ul style="list-style-type: none"> <li>Transaction cost</li> <li>Time</li> </ul>
[28]	<ul style="list-style-type: none"> <li>AES, DES and Triple DES</li> </ul>	<ul style="list-style-type: none"> <li>Reliability and confidentiality are improvised with the help of symmetric cryptographic techniques</li> </ul>	<ul style="list-style-type: none"> <li>Specific layers for handling attacks are not included.</li> </ul>	<ul style="list-style-type: none"> <li>Memory</li> </ul>
[29]	<ul style="list-style-type: none"> <li>Blockchain, Encryption and decryption</li> </ul>	<ul style="list-style-type: none"> <li>Attempts to strike a balance between security and efficiency</li> </ul>	<ul style="list-style-type: none"> <li>The applicability of the framework for other domains is yet to be considered.</li> </ul>	<ul style="list-style-type: none"> <li>Computational cost</li> <li>Communication cost</li> </ul>
[30]	<ul style="list-style-type: none"> <li>Proxy Re-encryption smart contracts</li> </ul>	<ul style="list-style-type: none"> <li>Using the blockchain, a proxy re-encryption scheme is introduced. This method provides a quick, secure, and efficient platform for managing, trading, and storing sensor data.</li> </ul>	<ul style="list-style-type: none"> <li>The system couldn't cope with cloud infrastructure. security</li> </ul>	<ul style="list-style-type: none"> <li>Execution cost</li> <li>Throughput</li> <li>Delay</li> </ul>
[31]	<ul style="list-style-type: none"> <li>Support Vector Machine (SVM), blockchain</li> </ul>	<ul style="list-style-type: none"> <li>SVM-based models are trained by the encrypted data to implement the confidentiality of sensitive data.</li> </ul>	<ul style="list-style-type: none"> <li>The system yet considers its robustness while preserving privacy for multiple encrypted data sets.</li> </ul>	<ul style="list-style-type: none"> <li>Accuracy</li> <li>Precision</li> <li>Recall</li> </ul>
[41]	<ul style="list-style-type: none"> <li>Proof of training quality (PoQ)</li> </ul>	<ul style="list-style-type: none"> <li>Data privacy is improvised through blockchain with FL.</li> </ul>	<ul style="list-style-type: none"> <li>System fails to address the enhancement of the data utility.</li> </ul>	<ul style="list-style-type: none"> <li>Area under ROC curve</li> <li>Running time</li> </ul>
[42]	<ul style="list-style-type: none"> <li>Smart contracts</li> </ul>	<ul style="list-style-type: none"> <li>Presented blockchain-based access control and data sharing framework. It reduces overall execution costs.</li> </ul>	<ul style="list-style-type: none"> <li>System should demonstrate its applicability for the private network</li> </ul>	<ul style="list-style-type: none"> <li>Transaction cost</li> <li>Execution cost</li> <li>Gas</li> </ul>
[37]	<ul style="list-style-type: none"> <li>ECC</li> </ul>	<ul style="list-style-type: none"> <li>Blockchain-based access control system to protect sensitive data and prevent unauthorized access.</li> <li>By utilizing mutual authentication and session key management.</li> </ul>	<ul style="list-style-type: none"> <li>Implementation of real-time scenario for IoT access control</li> </ul>	<ul style="list-style-type: none"> <li>Communication cost</li> <li>Computation cost</li> <li>Storage cost</li> </ul>

TABLE 2. (Continued.) The review of traditional blockchain models for IoT.

[38]	<ul style="list-style-type: none"> <li>Proxy re-encryption</li> <li>Smart contracts</li> </ul>	<ul style="list-style-type: none"> <li>FRUIT incentives high-quality knowledge by combining privacy-preserving task allocation, lightweight encryption and blockchain-based reputation.</li> </ul>	<ul style="list-style-type: none"> <li>The system doesn't address the trustworthiness of the computation node</li> </ul>	<ul style="list-style-type: none"> <li>Computation cost</li> <li>Gas consumption</li> <li>communication overhead</li> </ul>
[39]	<ul style="list-style-type: none"> <li>PCA, SVM, PoW</li> </ul>	<ul style="list-style-type: none"> <li>Privacy-preserving data management for IoT using blockchain for secure and transparent data sharing, and machine learning for intelligent analysis to maintain data anonymity</li> </ul>	<ul style="list-style-type: none"> <li>Potential privacy leaks due to data aggregation and computational complexities</li> </ul>	<ul style="list-style-type: none"> <li>Computation time</li> <li>Accuracy</li> <li>Precision</li> <li>Recall</li> </ul>
[40]	<ul style="list-style-type: none"> <li>NIZK, q-SDH problem</li> </ul>	<ul style="list-style-type: none"> <li>Proposed efficient privacy preserving schema introduced NIZK authentication with cloud-aided trustworthiness and key agreement for smart cities.</li> </ul>	<ul style="list-style-type: none"> <li>This framework is not suitable for large-scale networks</li> </ul>	<ul style="list-style-type: none"> <li>Communication overhead</li> <li>Computation cost</li> </ul>

TABLE 3. Symbols description.

Symbols	Description
$x, x_1, x_2, y_1, y_2$	Constant
$a, b, c, d, r$	Random numbers
$e, f, g, m$	Service parameters
$V_1, V_2$	Verification message
$OTP_1, OTP_2$	One time password
$A_1$	Authentication message
$T_{sc}$	Session password
$S_g$	Digital sign
$S_{id}$	Smart contract ID
$S_{pwd}$	Smart contract password
$C_{id}$	Client ID
$C_{id}$	Client password
$E_{id}$	Endorsing peer ID
$E_{pwd}$	Endorsing peer password
$M_{id}$	Membership service provider ID
$M_{pwd}$	Membership service provider password
$h$	Hash function
$R$	Access Record
$T$	Timestamp
$y$	Interpolation

devices should not be disclosed to the public. To establish trust, designing a fine-tuned blockchain system intends to guarantee such confidence is always appreciable. This section elaborates on the flow of the proposed model for establishing an efficient privacy-preserving scheme. Smart city-based applications need robust authentication to prevent data leakage and implement privacy. Hence, an improvised

blockchain framework SecPrivPreserve is a great need for the current context along with the in-built security features. The presented framework considers the implementation of various phases, namely initialization, registration, data protection, authentication, data access control, validation data sharing and download. Moreover, The designed scheme also makes use of various security operators like hashing, encryption, interpolation, and OTP in the data access and control phase. The implementation of data storage and sharing is facilitated by the use of smart contracts. In addition, the blockchain prevents the destructive impacts of server hacking and the falsification of permissions.

In privacy-preserving framework built upon the components such as client, data owner, membership service providers, smart contract, endorsing peers, committing peers, and ordering peers are discussed in detail. We use Hyperledger Fabric, a private blockchain, and public blockchain is different regarding user access. In a public blockchain, any user can register and access the data, but in a private blockchain, only validated users who have the private key, which is given by the system admin can access the resources. The specific rules of any business can be enforced through smart contracts. Hyperledger Fabric blockchain network utilizes smart contracts to summarize the information and systematize certain features of business transmission. One distinguishing factor between smart contracts and chaincode is the nature of their functionalities. Specifically, a smart contract is responsible for defining the transmission logic that facilitates the modification of the state of a business object that is included inside the world state. In the Hyperledger

Fabric blockchain, every peer node serves as a committed peer, whereas endorsing peers are a subset of committing peers who possess the additional capability of executing smart contracts. Furthermore, the term MSP pertains to the recognition of certificate authorities inside the Hyperledger Fabric blockchain network and the identification of peer nodes belonging to specific organizations. Figure 1 shows the overall structure of the proposed framework in smart cities including diverse devices and entities. All the phases of the proposed SecPrivPreserve are shown in Figure 2.

### 1) INITIALIZATION PHASE

It is the first step in the proposed scheme, transaction requests are initiated through MSP. In which MSP initialization is completed by generating random numbers a,b,c,d, and hash function h. Here, the random number ranges between[0,5].

### 2) REGISTRATION PHASE

After completing the initialization phase registration takes place. The registration phase is performed in four steps: the registration between the smart contract and MSP, the client and smart contract endorsing peers and MSP, and finally MSP and blockchain. Algorithm 1 shows the process of registration and the details of each phase mentioned below.

- Step1: Registration between smart contract and MSP smart contract id  $S_{id}$  and password  $S_{pwd}$  created at the smart contract and which is saved in MSP like  $S_{id}^*$ ,  $S_{pwd}^*$  through committing peers. Then, the verification message is formulated at MSP, which is given by,

$$V_1 = (h(S_{id}^*) || d) \oplus e \quad (1)$$

Furthermore, the verification message  $V_1$  is saved through committing peers at the smart contract and MSP like  $V_1^*$ . The service parameter is saved at MSP as,

$$\sim e = v_1 \oplus (h(S_{id}^*) || d) \quad (2)$$

Here, check whether  $e = \sim e$  then the smart contract is verified with MSP.

- Step2: Registration between client and smart contract Consequently, registration is executed among clients and a smart contract for that, client user ID and password is generated at client and represented as,  $C_{id}$  and  $C_{pwd}$ . The client ID ( $C_{id}$ ) is saved in smart contract as  $C_{id}^*$  and saved in committing peers as  $C_{id}^{**}$  and accumulated in endorsing peers as  $C_{id}^{***}$  and finally it is saved in MSP as  $C_{id}^{****}$ . In addition, client password  $C_{pwd}$  is also saved in a smart contract, committing peers, endorsing peers and MSP as  $C_{pwd}^*$ ,  $C_{pwd}^{**}$ ,  $C_{pwd}^{***}$  and  $C_{pwd}^{****}$ . Meanwhile, a verification message is created at the smart contract as,

$$V_2 = (h(C_{id}^{**}) | a) \oplus f \quad (3)$$

The verification message is further saved in the client as  $V_2^*$  and it is again saved at the smart contract as  $V_2^{**}$ . The service parameter f is created and it is saved in a smart

contract expressed as,

$$\sim f = v_1^{**} \oplus (h(C_{id}^{**}) || a) \quad (4)$$

Check if  $f = \sim f$  then the client is registered with a smart contract.

- Step3: Registration between endorsing peers and MSP After that, registration is performed between endorsing peers and MSP. The endorsing peer user name  $E_{id}$  and password  $E_{pwd}$  are generated in endorsing peers and the user ID and password are saved in MSP as  $E_{id}^*$  and  $E_{pwd}^*$ . Furthermore, the endorsing peer user ID and password are further saved in blockchain as  $E_{id}^{**}$  and  $E_{pwd}^{**}$ . Meanwhile, OTP is generated at MSP, which is specified by,

$$OTP = h(E_{id}^* || a) \oplus g \quad (5)$$

Further, the OTP is saved in endorsing peer as  $OTP^*$  again it is saved in MSP as  $OTP^{**}$ . Additionally, the service parameter g is created and it is saved as  $\sim g$  at MSP, which is denoted by,

$$\sim g = OTP^{**} \oplus (h(E_{id}^*) || a) \quad (6)$$

Here, if  $g = \sim g$  then endorsing peer is registered with MSP.

- Step4: Registration between MSP and blockchain After registering the endorsing peer with MSP, then registration between MSP and blockchain is performed. The MSP ID and password are created at MSP as  $M_{id}$  and  $M_{pwd}$  as well as it is saved at blockchain as  $M_{id}^*$  and  $M_{pwd}^*$ . Then, another OTP is created at blockchain, which is illustrated as,

$$OTP_2 = h(M_{id}^* || a) \oplus m \quad (7)$$

The generated  $OTP_2$  is saved at MSP as  $OTP^*$ , and the service parameter m is saved at blockchain as,

$$\sim m = OTP_2^{**} \oplus (h(M_{id}^*) || a) \quad (8)$$

If the saved service parameter is equal to the service parameter  $m = \sim m$ , then MSP is registered with the blockchain. After the completion of registration, the client saves the record  $R^*$  in the blockchain through a smart contract in an encrypted manner.

### 3) DATA PROTECTION

Once the registration phase is finished, then data protection is carried out between MSP and blockchain. Advanced Encryption Algorithm (AES) with a key length of 128 bits is used to encrypt registered details. To strengthen the privacy aspects of interpolation, Chebyshev polynomials, and encryption steps are included. The detailed steps are furnished in the below equations and the same process has been portrayed in Algorithm 2. The encrypted message is generated at blockchain, which is the combination of records and keys and it is given by,

$$E_n = E(R, K) \quad (9)$$



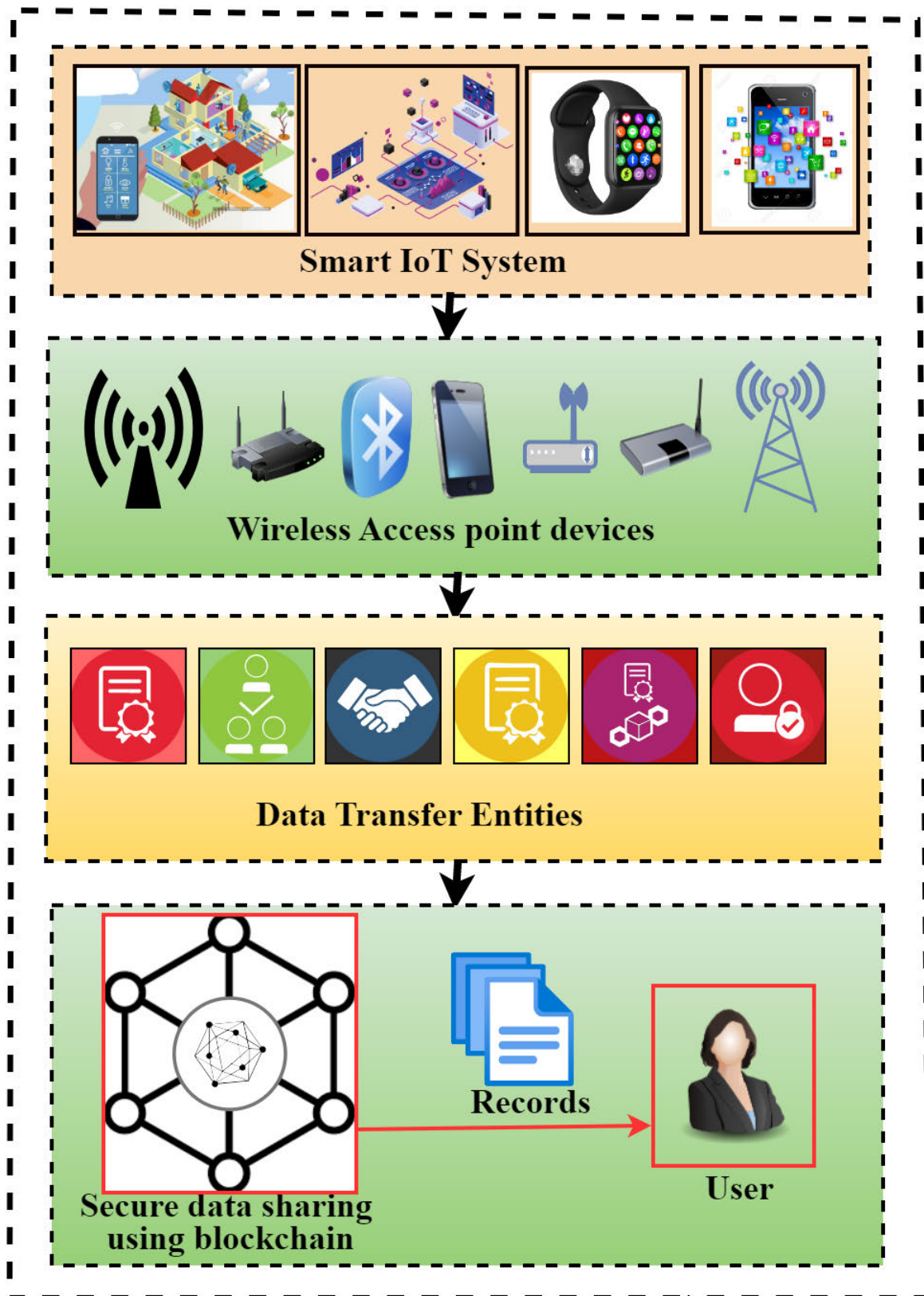


FIGURE 1. The proposed model of blockchain-based secure framework in smart city.

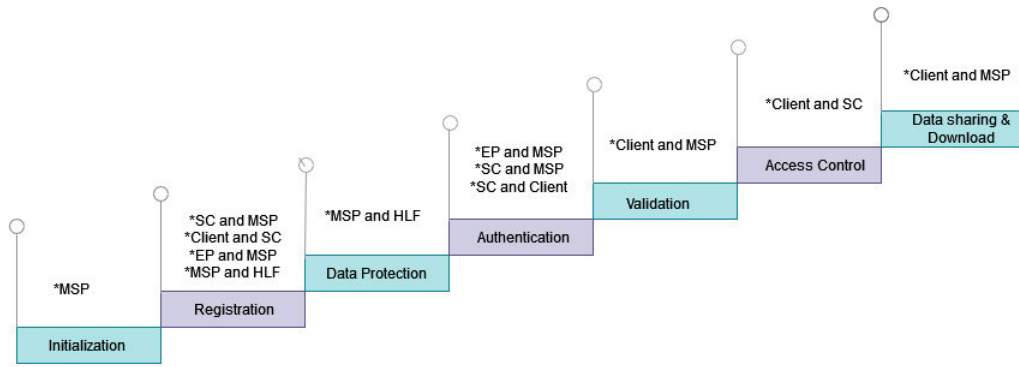


FIGURE 2. The proposed SecPrivPreserve framework phases.

Moreover, encrypted message is saved in MSP as  $E_n^*$  in which the key  $k$  is illustrated by,

$$K = y \oplus (c^*d) \tag{10}$$

where,

$$y = y_1 + \frac{(y_2 - y_1)}{x_2 - x_1} (x - x_1) \tag{11}$$

$$c = 60x^3 + 6x^2 + 6 \tag{12}$$

$$x = h(C_{id}) \text{ mod } r \tag{13}$$

Meanwhile, the key  $K$  is further saved in MSP as  $K^*$ . and the accessed record is expressed as,

$$R = D_n(E_n^*, K^*) \tag{14}$$

Here, the accessed record is obtained by decryption of the saved encryption message and saved key. The data protection process is shown in algorithm 2.

#### 4) AUTHENTICATION

The authentication phase is performed in three steps: the authentication between endorsing peers and MSP, authentication between smart contract and MSP, and finally, authentication between client and smart contract. Algorithm 3 shows the process of authentication and the details of each phase mentioned below.

- Step1: Authentication between endorsing peers and MSP In the authentication process, initially, authentication among endorsing peers and MSP is done for that, endorsing ID and password  $E_{id}, E_{pwd}$  is verified in MSP, whether it is same as saved endorsing ID and password  $E_{id}^*, E_{id}^*$ . Then, the digital signature is created at MSP, which is illustrated as,

$$S_g = (h(M_{id}) || b) \text{ mod } a \tag{15}$$

Then, the created digital sign  $S_g$  is saved in the client as  $S_g^*$  through endorsing peers and the smart contract. The authentication message is formulated at committing peers, and it is specified as,

$$A_1 = h(E_{id}) || S_g^* || T \tag{16}$$

Moreover, the authentication message is then saved at endorsing peer as  $A_1^*$ , which is given by,

$$\sim A_1 = h(E_{id}^*) || S_g || T \tag{17}$$

MSP checks the timestamp validity, if it is valid, the session continue, else terminate it and check if the authentication message is the same as the saved authentication message.

- Step2: Authentication between smart contract and MSP After that, authentication between the smart contract and MSP is executed for that, another authentication message is generated at the smart contract, which is illustrated by,

$$A_2 = h(S_{id} || S_{pwd}) \text{ mod } r || T \tag{18}$$

MSP checks the time stamp is valid, if yes, the session continues, else terminate it. The authentication message is then saved at MSP as  $A_2^*$ , which is specified as,

$$\sim A_2 = h(S_{id} || S_{pwd}) \text{ mod } r || T \tag{19}$$

If  $\sim A_2 = A_2$ , the smart contract is authenticated with MSP.

- Step3: Authentication between client and smart contract Here authentication is performed between the client and the smart contract such that other authentication message  $A_3$  is created at the client, which is illustrated by,

$$A_3 = (h(C_{id} || S_g^{**}) || a) \text{ mod } r || T \tag{20}$$

Here, the smart contract checks the timestamp validity, if yes it continues or else terminates it. Moreover, the authentication message  $A_3$  is saved in smart contract by,

$$A_3 = (h(C_{id}^* || S_g^{**}) || a) \text{ mod } r || T \tag{21}$$

If  $\sim A_3 = A_3$ , then client is authenticated with smart contract.

**Algorithm 1** Registration Between Smart Contract and MSP, Client and Smart Contract, Endorsing Peers and MSP, MSP and BC

**Input:** User records

**Output:** User record saved at blockchain or not

```

1: PROCEDURE: Smart contract and MSP
2: Smart Contract registered( $S_{id}, S_{pwd}$ )
3:  $(S_{id}, S_{pwd}) \rightarrow$  MSP, CP
4: MSP:  $V_1 \rightarrow$  SC, CP
5: if MSP: verify  $V_1^*$  then
6:    $(\sim e)$  generated: MSP
7: end if
8: if  $e = \sim e$  then
9:   SC verified: MSP
10: end if
11: end procedure
12: PROCEDURE: Client and Smart contract
13: Client( $C_{id}$  and  $C_{pwd}$ )
14: Client:  $(C_{id}, C_{pwd}) \rightarrow$  CP, EP, MSP
15: SC:  $(V_2) \rightarrow$  Client
16: if Client: verify  $V_2^*$  then
17:    $(\sim f)$  generated: SC
18: end if
19: if  $f = \sim f$  then,
20:   Client registered with SC
21: else
22:   Terminated
23: end if
24: end procedure
25: PROCEDURE: Endorsing peers and MSP
26: EP( $E_{id}$  and  $E_{pwd}$ )
27: EP:  $(E_{id}, E_{pwd}) \rightarrow$  MSP, BC
28: MSP:  $(OTP_1) \rightarrow$  EP
29: if EP validate  $(OTP_1^*)$  then
30:    $(\sim g)$  generated: MSP
31: end if
32: if  $g = \sim g$  then
33:   EP registered with MSP
34: else
35:   Terminated
36: end if
37: end procedure
38: PROCEDURE: MSP and BC
39: MSP ( $M_{id}$  and  $M_{pwd}$ )
40: BC  $\leftarrow (U_{id}$  and  $U_{pwd})$ 
41: MSP  $\leftarrow (OTP_2)$ : BC
42: if BC validate  $(OTP_2^*)$  then
43:    $(\sim m)$  generated: BC
44: end if
45: if  $m = \sim m$  then
46:   MSP registered with BC
47:   Data stored in BC(R)
48:   SC, BC  $\leftarrow R^*$ 
49: else
50:   Terminated
51: end if
52: end procedure

```

## 5) DATA ACCESS CONTROL

After the authentication process, data access control is performed among the client and MSP, here, MSP generates the session password  $T_{sc}$  at MSP, which is given by,

$$T_{sc} = h(S_g) \bmod r \quad (22)$$

**Algorithm 2** Data Protection Between MSP and BC

**Input:** Encrypted record

**Output:** Decrypted record

```

1: PROCEDURE: MSP to BC
2: MSP  $\leftarrow E_n(R, K)$ : BC
3: if MSP verify key then
4:   MSP  $\leftarrow D_n(E_n^*, K^*)$ 
5:   MSP  $\leftarrow R, K^*$ 
6: else
7:   Terminated
8: end if
9: end procedure

```

Then, validate  $T_{sc}$  at smart contract and saved it in client as  $T_{sc}^*$ , which is expressed as,

$$\sim T_{sc} = h(S_g^{**}) \bmod r \quad (23)$$

If  $T_{sc} = \sim T_{sc}$ , then, compute the term C in a smart contract and it is saved in the client as  $C^*$ ,

$$C = (h(T_{sc}^{**}) \bmod p) \quad (24)$$

The term  $\sim C$  is denoted by,

$$\sim C = (h(T_{sc}) \bmod p) \quad (25)$$

If  $C_* = \sim C$ , then AC is generated at client,

$$AC = C^* * h(C_{id} \| d \| b) \quad (26)$$

The term AC is saved in the smart contract as,

$$\sim AC = C * h(C_{id} \| d \| b) \quad (27)$$

If  $AC = \sim AC$ , then denies the access to client. Algorithm 4 refers to the process of data access control.

## 6) VALIDATION

When the citizen has to update the record, such as name change or address change client needs to request through smart contracts. Besides, MSP validates the client user name and password if it is validated, MSP updates the blockchain. At the client phase, the client user ID and password are created and it is saved in a smart contract as  $C_{id}^*$  and  $C_{pwd}^*$ . Furthermore, it is again saved in MSP as  $C_{id}^{**}$  and  $C_{pwd}^{**}$ . if  $C_{id} = C_{id}^{**}$ , then create session password, which is specified by,

$$SS_{pwd} = h(c \| S_g) \bmod r \quad (28)$$

The session password is saved in the smart contract as  $SS_{pwd}^*$  and further saved in the client as  $SS_{pwd}^{**}$ . Algorithm 5 depicts the process of the validation phase.

**Algorithm 3** Authentication Between Endorsing Peers and MSP**Input:** Client, SC and EP details**Output:** Client authenticated with SC or not

```

1: PROCEDURE: Authentication between EP and MSP
2:  $MSP \leftarrow E_{id}, E_{pwd}$ : EP
3: if ( $E_{id} = E_{id}^*$ ) and ( $E_{pwd} = E_{pwd}^*$ ) then
4:   Generate digital sign  $S_g$ 
5:   Ep, SC, C  $\leftarrow S_g$ 
6:   CP generate authentication message  $A_1$ 
7:   EP  $\leftarrow A_1^*$ : CP
8: else
9:   Terminated
10: end if
11: if ( $A_1 = \sim A_1$ ) then
12:   EP authenticated with MSP
13: else
14:   Terminated
15: end ifend procedure
16: PROCEDURE: Authentication between SC and MSP
17: SC generate authentication message  $A_2$ 
18:  $MSP \leftarrow A_2^*$ : SC
19: if SC verify ( $A_2$ , T) then
20:   if ( $A_2 = \sim A_2$ ) then
21:     SC authenticated with MSP
22:   else
23:     Terminated
24:   end if
25: end if
26: end procedure
27: PROCEDURE: Authentication between SC and Client
28: Client generate authentication message  $A_3$ 
29:  $SC \leftarrow A_3^*$ : Client
30: if SC verify ( $A_3$ , T) then
31:   if ( $A_3 = \sim A_3$ ) then
32:     Client authenticated with SC
33:   else
34:     Terminated
35:   end if
36: end if
37: end procedure

```

## 7) DATA SHARING AND DOWNLOAD

Finally, data sharing is done between the client and MSP. Here, if the client ID and password are validated at MSP, then MSP gets records from BC. Then, MSP ID and password are passed to BC if it same as the saved MSP ID and password, then the record is sent to MSP from BC. In the download process, the client ID and password are sent to the smart contract and it is verified whether it is the same as the saved client ID and password if it is valid, then the smart contract ID and password are passed to MSP. If the smart contract ID and password are similar to the saved smart contract ID

**Algorithm 4** Data Access Control Between Client and MSP**Input:** Client and MSP details**Output:** Client access the record or not

```

1: PROCEDURE: Access control between Client and MSP
2: MSP generate  $T_{sc}$ 
3: if SC validate  $T_{sc}$  then
4:   Client  $\leftarrow T_{sc}$ 
5:   if ( $T_{sc} = \sim T_{sc}$ ) then
6:     Compute C
7:     Client  $\leftarrow C^*$ : SC
8:   else
9:     Terminated
10:   end if
11:   if ( $C^* = \sim C$ ) then
12:     Client generate AC
13:     SC  $\leftarrow AC$ : Client
14:   else
15:     Terminated
16:   end if
17:   if  $AC = \sim AC$  then
18:     Client access denied
19:   else
20:     Terminated
21:   end if
22: end if
23: end procedure

```

**Algorithm 5** Validation: Client and Smart Contract**Input:** Client and SC details**Output:** Client validation and updation in BC

```

1: PROCEDURE: Validation between Client and SC
2: Client: ( $C_{id}, C_{pwd}$ )
3: SC, MSP  $\leftarrow (C_{id}, C_{pwd})$ : Client
4: if ( $C_{id} = C_{id}^*$ ) and ( $C_{pwd} = C_{pwd}^*$ ) then,
5:   Client, SC  $\leftarrow S_{pd}$ : BC
6:   Client generate  $S \rightarrow SC$ 
7: else
8:   Terminated
9: end if
10: if ( $S = \sim S$ ) then
11:   Client validated and update BC
12: else
13:   Terminated
14: end if
15: end procedure

```

and password then decrypted data is created in MSP, which is expressed as,

$$D_h = h(M_{id} \| E_n^*) \| K_y \quad (29)$$

The decrypted message is saved in smart contract as  $D_h^*$ , in which the key is expressed as,

$$K_y = h(M_{id}) \| K^* \| h(H_{pwd}) \bmod r \quad (30)$$

The key is then saved in a smart contract and if the decrypted message is the same as the saved decrypted message, then sends the record to the client in an encrypted manner. Thus, the client can download the data from MSP. Algorithm 6 illustrates the process of data sharing and downloading.

---

#### Algorithm 6 Data Sharing and Download

---

**Input:** User records

**Output:** Client downloads user record or not

```

1: PROCEDURE: Data sharing between Client and MSP
2: Client  $\rightarrow C_{id}, C_{pwd}$ 
3: SC, MSP  $\leftarrow C_{id}, C_{pwd}$ : Client
4: if  $(C_{id} = C_{id}^*)$  and  $(C_{pwd} = C_{pwd}^*)$  then
5:   MSP access R from BC
6: else
7:   Terminated
8: end if
9: MSP  $\rightarrow M_{id}, M_{pwd}$ 
10: BC verify  $M_{id}, M_{pwd}$ 
11: if  $(M_{id} = M_{id}^*)$  and  $(M_{pwd} = M_{pwd}^*)$  then
12:   MSP  $\leftarrow R^*$ : BC
13: else
14:   Terminated
15: end if
16: end procedure
17: PROCEDURE: Download between Client and MSP
18: Client  $\rightarrow C_{id}, C_{pwd}$ 
19: SC  $\leftarrow C_{id}, C_{pwd}$ : Client
20: if  $(C_{id} = C_{id}^*)$  and  $(C_{pwd} = C_{pwd}^*)$  then
21:   Valid
22: else
23:   Terminated
24: end if
25: if  $(S_{id} = S_{id}^*)$  and  $(S_{pwd} = S_{pwd}^*)$  then
26:    $(D_h, K_y)$  created
27:    $K_y \rightarrow SC$ 
28: else
29:   Terminated
30: end if
31: if  $(D_h = \sim D_h)$  then
32:   Client download R
33: else
34:   Terminated
35: end if
36: end procedure

```

---

#### C. USECASE SCENARIO

To demonstrate the presented SecPrivPreserve framework dataset from Kaggle (<https://www.kaggle.com/datasets/csafri/t2/maternal-health-risk-data/data>) has been considered [43]. The experimented dataset contains patient health parameters collected through diverse IoT sensors. From the dataset extract all the features and data converted into JSON format to

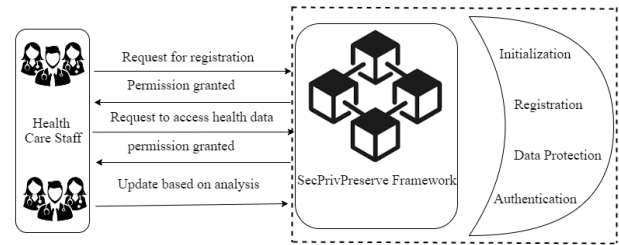


FIGURE 3. Patient health monitoring using proposed framework.

give as an input transaction. These transactions are submitted to Fabric SDK and validated through smart contracts. The architecture view of SecPrivPreserve is expressed in Figure 3. As of now, this framework ensures privacy and integrity for healthcare data solely needed by the Health care staff. The advantages of the proposed framework are to improve computing speed and security benefits, the proposed SecPrivPreserve uses multiphase security techniques with minimum cryptographic operations. Process automation is made easier by the deterministic design of smart contracts. Fast access or submission of requests is a top priority in a smart environment. To handle such situations, the framework was presented uses only those cryptographic operators that are required to do the task.

#### IV. RESULTS AND DISCUSSION

The results of the simulated SecPrivPreserve framework using blockchain and smart contracts are furnished in this section. The implementation is carried out on a system with Ubuntu 20.4 LTS OS, Intel i7 processor, 16 GB of RAM, and the Fabric SDK is used for developing the SecPrivPreserve framework. The configuration details of Fabric SDK are discussed in Table 4. To evaluate the effectiveness of the proposed model, we considered metrics such as computational time, encryption quality, detection rate, and responsiveness. The conventional blockchain-based approaches BaseLine 1 and BaseLine 2 are considered to compare the effectiveness of the proposed SecPrivPreserve framework. The BaseLine 1 model is mainly concerned with the authentication process and the protection of data, whereas the BaseLine 2 model is primarily concerned with the validation process and controlling access to data from various sources. These BaseLine models are developed based on the classical blockchain approach and the phases are explained in the proposed methodology.

The following metrics are considered for experimental evaluation with the proposed model, and these metrics are compared with BaseLine models.

**Computational Time:** It is the run time, or the entire amount of time the system needs to complete the authentication process, measured using computational cost.

**Detection Rate:** It is measured as the proportion of the total number of users who have been verified as real to all users.



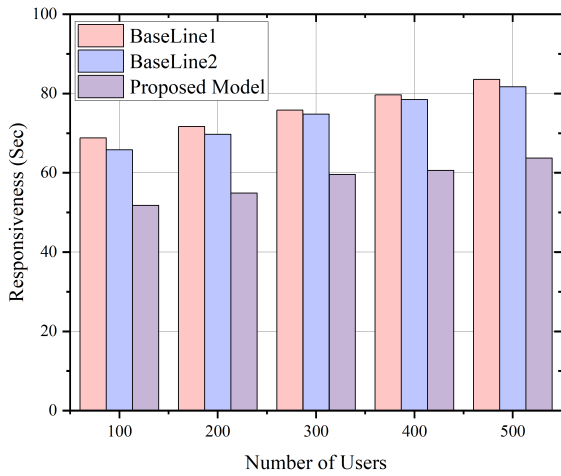


FIGURE 4. Responsiveness.

**Encryption Quality:** It is a measurement that determines the comparison between the unaltered and protected versions of the data to determine the highest encryption quality for higher performance [44].

**Responsiveness:** It is the rate at which to detect changes in a time frame when a request is accepted. Which is represented as,

$$Rp = 1 - \frac{\partial^k_{p1=1}(Z_{p1})}{Z_{p\max}} \quad (31)$$

where,  $Z_{p\max}$  denotes the maximum acceptable time used to complete the request, such that  $Z_{p\max} \geq Z_{p1}$ ,  $k$  specifies the number of requests, and  $Z_{p1}$  shows the time between the completion of  $p1^{th}$  request. In addition,  $p1$  is the function that is utilized to calculate the central tendency of data.

By varying the number of transactions, channels, organizations, endorsing peers, committing peers, ordering peers, and rounds to achieve superior throughput, scalability, performance and distribution of load across multiple channels in this environment. The experimentation was conducted in 100 sets of rounds for all users. Initially, one channel scenario considers with four ordering peers, committing peers, and endorsing peers. There were two organizations with a hundred users and a total of thousand transactions were given input to the system. For all scenarios, the configuration details are shown in Table 4.

**A. EXPERIMENTAL ANALYSIS**

The proposed SecPrivPreserve framework performance is evaluated based on responsiveness, computational time, encryption quality, and detection rate by varying numbers of users and block size. For instance, with 100 users, the SecPrivPreserve technique achieves lower responsiveness, lower computational time, higher encryption quality, and a high detection rate of 50 (Sec), 30 (Sec), 0.77 (Sec), and 0.8 (Sec) respectively. Likewise, with 200 users, the SecPrivPreserve approach gains low responsiveness, low

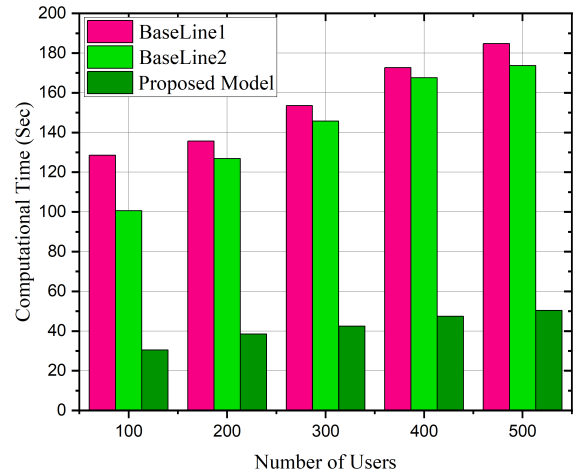


FIGURE 5. Computational time.

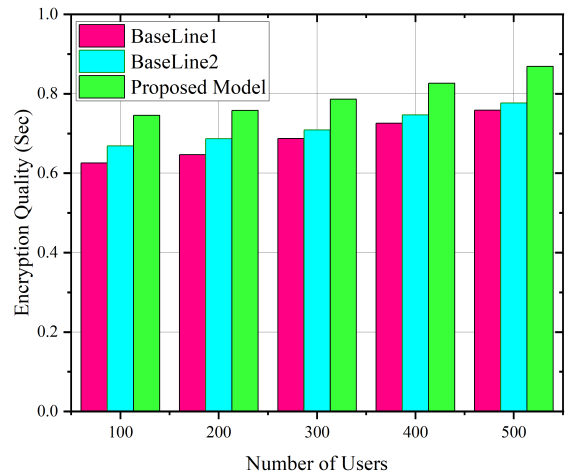


FIGURE 6. Encryption quality.

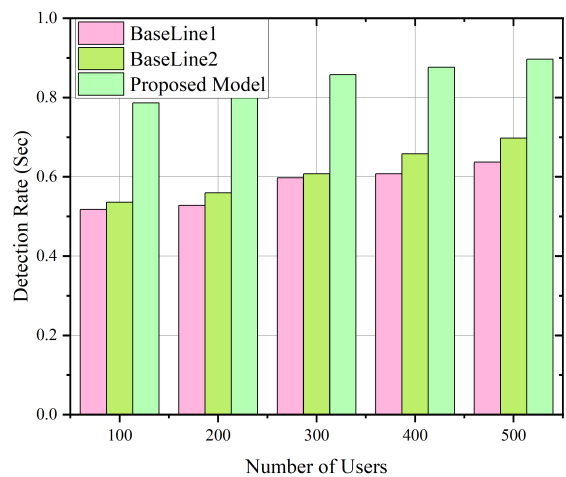


FIGURE 7. Detection rate.

computational time, high encryption quality, and a high detection rate of 55 (Sec), 39 (Sec), 0.78 (Sec), and 0.82 (Sec) respectively. Simultaneously, with 300 users

TABLE 4. Configuration parameters.

No of users	Input Tx's	Channels	Organizations	Ordering peers	Committing peers	Endorsing peers	Rounds
100	1000	1	2	4	4	4	100
200	2000	2	2	8	8	8	100
300	3000	3	2	12	12	12	100
400	4000	4	2	16	16	16	100
500	5000	5	2	20	20	20	100

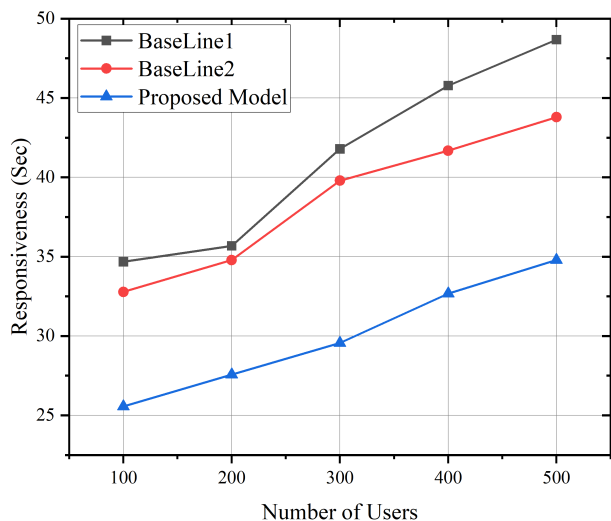


FIGURE 8. Responsiveness.

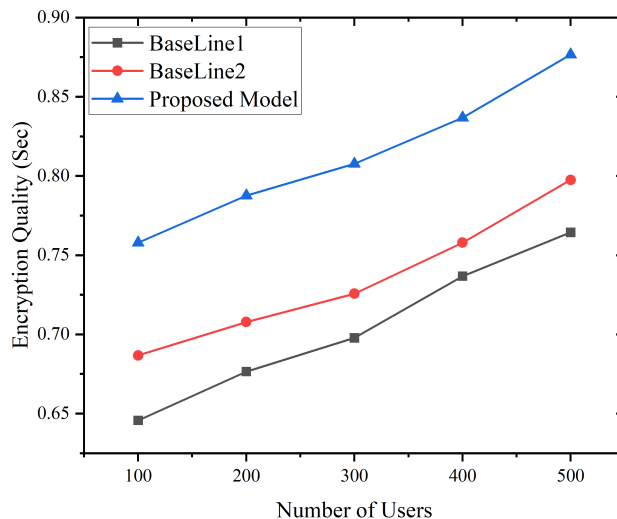


FIGURE 10. Encryption quality.

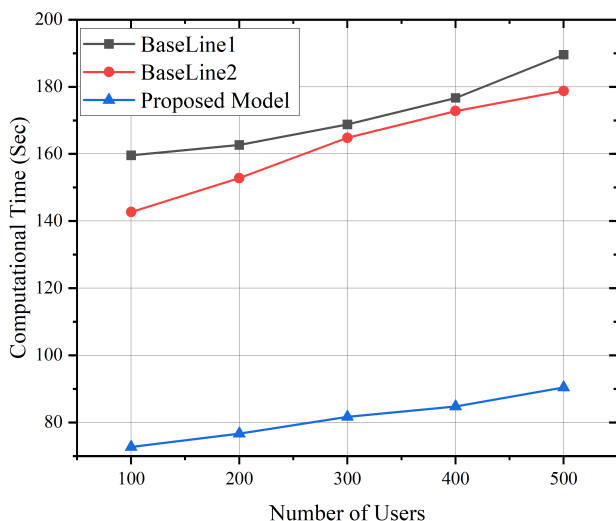


FIGURE 9. Computational time.

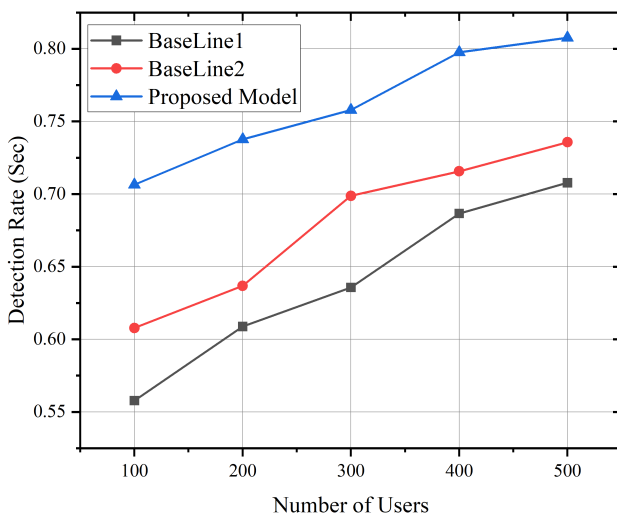


FIGURE 11. Detection rate.

the SecPrivPreserve methodology attains responsiveness, computational time, encryption quality, and detection rate of 54 (Sec), 42 (Sec), 0.79 (Sec), and 0.87 (Sec) respectively. Likewise, with 400 users, the SecPrivPreserve approach gains responsiveness, computational time, encryption quality, and detection rate of 60 (Sec), 46 (Sec), 0.82 (Sec), and 0.89 (Sec)

respectively. Finally, with 500 users, the SecPrivPreserve system reaches responsiveness, computational time, encryption quality, and detection rate of 62 (Sec), 50 (Sec), 0.88 (Sec), and 0.90 (Sec) respectively.

In evaluating the SecPrivPreserve in terms of responsiveness, computation time, detection rate, and encryption

**TABLE 5.** The analysis of performance metrics for several measures.

Performance Metrics	BaseLine1	BaseLine2	ProposedSec PrivPre-serve
block size 2MB			
Computational time	184.77	173.67	50.34
Encryption quality	0.758	0.776	0.868
Detection rate	0.636	0.697	0.896
Responsiveness	83.567	81.678	63.678
block size 4MB			
Computational time	189.56	178.77	90.45
Encryption quality	0.764	0.797	0.876
Detection rate	0.707	0.735	0.807
Responsiveness	48.678	43.789	34.789

**TABLE 6.** Comparative outcome of enhanced SecPrivPreserve approach with baseLine models.

Model	Responsiveness (Sec)	Computational Time (Sec)	Encryption Quality (Sec)	Detection Rate(Sec)
BaseLine1	48	190	0.74	0.71
BaseLine2	42	169	0.77	0.74
Proposed Model	34	94	0.87	0.82

quality, Figures 4, 5, 6, and 7 combine a graphical depiction of the tested results with the comparative results provided in Table 5. Compared to the baseline methodologies included in the analysis, the proposed method obtains less computational time when altering the number of users and using a fixed block size of 2 MB. Two crucial parameters for the candidate proposal are responsiveness and computation time are shown in Figure 4 and Figure 5. Despite a substantial increase in the number of users from 100 to 500, responsiveness and calculation time remain modest thanks to the employment of efficient cryptographic operators. In terms of the encryption quality detection rate with a fixed block size of 2MB, SecPrivPreserve outperformed Baseline1 and Baseline2 in terms of the varying number of users, resulting in Baseline1 and Baseline2 having superior outcomes are shown in Figure 6 and Figure 7.

In another experiment, the simulation was carried out with a fixed block size of 4MB. The tested results are as follows: while observing the framework performance in terms of responsiveness, it is 48.678 (Sec), 43.789 (Sec), and 34.789 (Sec) for BaseLine1 and BaseLine2. The proposed model yields better computational time compared to existing models at 189.56 (Sec), 178.77 (Sec), and 90.45 (Sec), respectively are shown in Figure 8 and Figure 9. In of perception of encryption quality, SecPrivPreserve outperformed Baseline 1 and Baseline 2 in terms of the varying number

of users. In comparison to Baseline 1, Baseline 2 results are 0.764 (Sec), 0.797 (Sec), and 0.876 (Sec) are shown in Figure 10. In terms of detection rate, our proposed model reveals better outcomes compared to traditional models as 0.707 (Sec), 0.735 (Sec), and 0.807 (Sec) are shown in Figure 11.

## B. PERFORMANCE ANALYSIS

The proposed SecPrivPreserve framework performance is examined based on assorted block sizes and user counts. We analyze computational time, encryption quality, detection rate, and responsiveness, confirming that the new model outperforms the existing BaseLine models. Initially, the default block size is set to 2MB, and the model is updated based on performance metrics and varying user counts. In this case, the results demonstrate the superiority of the proposed model. Second, we enhanced the block size to 4MB by adjusting the number of users up to 500 and evaluated the model based on evaluation criteria. In this case, the proposed model got enhanced results. Table 6 shows the enhanced results compared to BaseLine models.

In experimental investigations utilizing distinct cutting-edge methodologies, the SecPrivPreserve approach demonstrated superior performance in comparison to state-of-the-art systems. Specifically, it obtained a better level of responsiveness, measured at 34, a reduced computational time of 94, an encryption quality of 0.87, and a detection rate of 0.82.

## V. CONCLUSION

IoT smart devices are essential in the digital age. Smart devices, including smart homes, cities, and transport systems, create vast amounts of data. Data communication in smart cities requires increased security and privacy. Blockchain secures and anonymizes IoT and its applications. Smart city challenges include user security, privacy, bandwidth, anonymity, and scalability. Therefore, this study proposes a blockchain-based SecPrivPreserve system. The presented framework ensures the privacy and safety of the user's data throughout processing. In the Hyperledger Fabric blockchain, information is summarized, and specific features of business transmission are systematized based on the model. Initialization, registration, data protection, authentication, data access control, validation, data sharing and download comprise in SecPrivPreserve framework. Security features include passwords, OTP, encryption, hashing, digital signature, Chebyshev polynomials, and interpolation. Cutting-edge experiments demonstrated that SecPrivPreserve outperformed state-of-the-art systems in responsiveness, processing time, encryption quality, and detection rate. However, the experimentation was carried out through Fabric SDK, and the obtained results show that the proposed framework reduces computational time and responsiveness. Furthermore, it enhances the encryption quality and detection rate compared to conventional BaseLine models. Our future endeavors will test the suitability of the candidate

SecPrivPreserve for other smart service applications by making appropriate changes in the initialization and registration phase. The applicability of the presented work for smart grids and smart vehicular networks is yet to be considered. Another possible enhancement includes leveraging the fog/cloud computing to enhance the scalability issue along with security merits. Data-driven model's performance after the application of the presented works was analyzed.

## ACKNOWLEDGMENT

### CONFLICTS OF INTEREST:

The authors declare no conflicts of interest.

### DATA AVAILABILITY STATEMENT:

Data should be available with the corresponding author based on request.

## REFERENCES

- [1] C. Vanmathi, R. Mangayarkarasi, and R. J. Subalakshmi, "Real time weather monitoring using Internet of Things," in *Proc. Int. Conf. Emerg. Trends Inf. Technol. Eng. (ic-ETITE)*, Feb. 2020, pp. 1–6.
- [2] B. Bryant and H. Saiedian, "Key challenges in security of IoT devices and securing them with the blockchain technology," *Secur. Privacy*, vol. 5, no. 5, p. e251, Sep. 2022.
- [3] F. Al-Turjman, H. Zahmatkesh, and R. Shahroze, "An overview of security and privacy in smart cities' IoT communications," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 3, p. e3677, Mar. 2022.
- [4] The Editors of Encyclopaedia. (Dec. 9, 2023). *United Nations Population Fund. Encyclopedia Britannica*. Accessed: Jun. 6, 2023. [Online]. Available: <https://www.britannica.com/topic/United-Nations-Population-Fund>
- [5] V. Moustaka, Z. Theodosiou, A. Vakali, and A. Kounoudes, "Smart cities at risk! Privacy and security borderlines from social networking in cities," in *Proc. Companion The Web Conf. Web Conf.*, 2018, pp. 905–910.
- [6] I. Makhdoom, M. Abolhasan, J. Lipman, R. P. Liu, and W. Ni, "Anatomy of threats to the Internet of Things," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1636–1675, 2nd Quart., 2019.
- [7] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. S. Shen, "Security and privacy in smart city applications: Challenges and solutions," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 122–129, Jan. 2017.
- [8] S. Chaudhary and P. K. Mishra, "DDoS attacks in industrial IoT: A survey," *Comput. Netw.*, vol. 236, Nov. 2023, Art. no. 110015.
- [9] L. Cui, G. Xie, Y. Qu, L. Gao, and Y. Yang, "Security and privacy in smart cities: Challenges and opportunities," *IEEE Access*, vol. 6, pp. 46134–46145, 2018.
- [10] Z. Xihua and D. S. B. Goyal, "Security and privacy challenges using IoT-blockchain technology in a smart city: Critical analysis," *Int. J. Electr. Electron. Res.*, vol. 10, no. 2, pp. 190–195, Jun. 2022.
- [11] P. M. Rao and B. D. Deebak, "Security and privacy issues in smart cities/industries: Technologies, applications, and challenges," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 1, pp. 1–37, Feb. 2022.
- [12] M. Ramaiah, V. Chandrasekaran, V. Ravi, and N. Kumar, "An intrusion detection system using optimized deep neural network architecture," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 4, p. e4221, Apr. 2021.
- [13] M. Gupta, F. M. Awayseh, J. Benson, M. Alazab, F. Patwa, and R. Sandhu, "An attribute-based access control for cloud enabled industrial smart vehicles," *IEEE Trans. Ind. Informat.*, vol. 17, no. 6, pp. 4288–4297, Jun. 2021.
- [14] M. Ramaiah, V. Chithanuru, A. Padma, and V. Ravi, "A review of security vulnerabilities in Industry 4.0 application and the possible solutions using blockchain," in *Cyber Security Applications for Industry 4.0*. London, U.K.: Chapman & Hall, 2023, pp. 63–95.
- [15] C.-L. Chen, J. Yang, W.-J. Tsaur, W. Weng, C.-M. Wu, and X. Wei, "Enterprise data sharing with privacy-preserved based on hyperledger fabric blockchain in IIoT's application," *Sensors*, vol. 22, no. 3, p. 1146, 2022.
- [16] U. Khalil, O. A. Malik, and S. Hussain, "A blockchain footprint for authentication of IoT-enabled smart devices in smart cities: State-of-the-art advancements, challenges and future research directions," *IEEE Access*, vol. 10, pp. 76805–76823, 2022.
- [17] N. Deepa, Q.-V. Pham, D. C. Nguyen, S. Bhattacharya, B. Prabadevi, T. R. Gadekallu, P. K. R. Maddikunta, F. Fang, and P. N. Pathirana, "A survey on blockchain for big data: Approaches, opportunities, and future directions," *Future Gener. Comput. Syst.*, vol. 131, pp. 209–226, Jun. 2022.
- [18] C. Li, M. Dong, X. Xin, J. Li, X.-B. Chen, and K. Ota, "Efficient privacy preserving in IoMT with blockchain and lightweight secret sharing," *IEEE Internet Things J.*, vol. 10, no. 24, pp. 22051–22064, Dec. 2023.
- [19] H. Honar Pajooh, M. Rashid, F. Alam, and S. Demidenko, "Hyperledger fabric blockchain for securing the edge Internet of Things," *Sensors*, vol. 21, no. 2, p. 359, Jan. 2021.
- [20] N. K. Tyagi and M. Goyal, "Blockchain-based smart contract for issuance of country of origin certificate for Indian customs exports clearance," *Concurrency Comput., Pract. Exp.*, vol. 35, no. 16, p. e6249, Jul. 2023.
- [21] A. Padma and R. Mangayarkarasi, "Detecting security breaches on smart contracts through techniques and tools a brief review: Applications and challenges," in *Proc. Int. Conf. Inf. Manage. Eng.* Singapore: Springer, 2022, pp. 361–369.
- [22] P. Sharma, S. Namasudra, N. Chilamkurti, B.-G. Kim, and R. G. Crespo, "Blockchain-based privacy preservation for IoT-enabled healthcare system," *ACM Trans. Sensor Netw.*, vol. 19, no. 3, pp. 1–17, 2023.
- [23] M. Tahir, M. Sardaraz, S. Muhammad, and M. S. Khan, "A lightweight authentication and authorization framework for blockchain-enabled IoT network in health-informatics," *Sustainability*, vol. 12, no. 17, p. 6960, Aug. 2020.
- [24] N. B. Truong, K. Sun, G. M. Lee, and Y. Guo, "GDPR-compliant personal data management: A blockchain-based solution," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1746–1761, 2019.
- [25] J. Chi, Y. Li, J. Huang, J. Liu, Y. Jin, C. Chen, and T. Qiu, "A secure and efficient data sharing scheme based on blockchain in industrial Internet of Things," *J. Netw. Comput. Appl.*, vol. 167, Oct. 2020, Art. no. 102710.
- [26] I. Makhdoom, I. Zhou, M. Abolhasan, J. Lipman, and W. Ni, "PrivySharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities," *Comput. Secur.*, vol. 88, Jan. 2020, Art. no. 101653.
- [27] P. Singh, M. Masud, M. S. Hossain, and A. Kaur, "Cross-domain secure data sharing using blockchain for industrial IoT," *J. Parallel Distrib. Comput.*, vol. 156, pp. 176–184, Oct. 2021.
- [28] M. Šarac, N. Pavlović, N. Bacanin, F. Al-Turjman, and S. Adamović, "Increasing privacy and security by integrating a blockchain secure interface into an IoT device security gateway architecture," *Energy Rep.*, vol. 7, pp. 8075–8082, Nov. 2021.
- [29] Q. Fan, J. Chen, L. J. Deborah, and M. Luo, "A secure and efficient authentication and data sharing scheme for Internet of Things based on blockchain," *J. Syst. Archit.*, vol. 117, Aug. 2021, Art. no. 102112.
- [30] A. Manzoor, A. Braeken, S. S. Kanhere, M. Ylianttila, and M. Liyanage, "Proxy re-encryption enabled secure and anonymous IoT data sharing platform based on blockchain," *J. Netw. Comput. Appl.*, vol. 176, Feb. 2021, Art. no. 102917.
- [31] M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani, "Privacy-preserving support vector machine training over blockchain-based encrypted IoT data in smart cities," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7702–7712, Oct. 2019.
- [32] J. Hong, "Chaincomm: A framework for future communities based on blockchain," in *Proc. IEEE 8th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2022, pp. 1324–1328.
- [33] A. Simonet-Boulogne, A. Solberg, A. Sinaeepourfard, D. Roman, F. Perales, G. Ledakis, I. Plakas, and S. Sengupta, "Toward blockchain-based fog and edge computing for privacy-preserving smart cities," *Frontiers Sustain. Cities*, vol. 4, p. 136, Sep. 2022.
- [34] D. L. Fekete and A. Kiss, "A survey of ledger technology-based databases," *Future Internet*, vol. 13, no. 8, p. 197, 2021.
- [35] X. Yang, R. Zhang, C. Yue, Y. Liu, B. C. Ooi, Q. Gao, Y. Zhang, and H. Yang, "VeDB: A software and hardware enabled trusted relational database," *Proc. ACM Manage. Data*, vol. 1, no. 2, pp. 1–27, 2023.
- [36] X. Yang, Y. Zhang, S. Wang, B. Yu, F. Li, Y. Li, and W. Yan, "LedgerDB: A centralized ledger database for universal audit and verification," *Proc. VLDB Endowment*, vol. 13, no. 12, pp. 3138–3151, Aug. 2020.

- [37] S. Saha, B. Bera, A. K. Das, N. Kumar, S. H. Islam, and Y. Park, "Private blockchain envisioned access control system for securing industrial IoT-based pervasive edge computing," *IEEE Access*, vol. 11, pp. 130206–130229, 2023.
- [38] C. Zhang, M. Zhao, L. Zhu, W. Zhang, T. Wu, and J. Ni, "FRUIT: A blockchain-based efficient and privacy-preserving quality-aware incentive scheme," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 12, pp. 3343–3357, Dec. 2022.
- [39] P. M. Kumar, B. Rawal, and J. Gao, "Blockchain-enabled privacy preserving of IoT data for sustainable smart cities using machine learning," in *Proc. 14th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2022, pp. 1–6.
- [40] X. Xia, S. Ji, P. Vijayakumar, J. Shen, and J. J. P. C. Rodrigues, "An efficient anonymous authentication and key agreement scheme with privacy-preserving for smart cities," *Int. J. Distrib. Sensor Netw.*, vol. 17, no. 6, Jun. 2021, Art. no. 155014772110268.
- [41] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4177–4186, Jun. 2019.
- [42] T. Sultana, A. Almogren, M. Akbar, M. Zuair, I. Ullah, and N. Javaid, "Data sharing system integrating access control mechanism using blockchain-based smart contracts for IoT devices," *Appl. Sci.*, vol. 10, no. 2, p. 488, 2020.
- [43] M. Ahmed and M. A. Kashem, "IoT based risk level prediction model for maternal health care in the context of Bangladesh," in *Proc. 2nd Int. Conf. Sustain. Technol. Ind. 4.0 (STI)*, Dec. 2020, pp. 1–6.
- [44] S. Som, A. Kotal, A. Mitra, S. Palit, and B. B. Chaudhuri, "A chaos based partial image encryption scheme," in *Proc. 2nd Int. Conf. Bus. Inf. Manage. (ICBIM)*, Jan. 2014, pp. 58–63.



**ADLA PADMA** received the B.Tech. and M.Tech. degrees in CSE from JNTUH Telangana. She is currently a Research Scholar with the Vellore Institute of Technology, Vellore, India. Her research interests include blockchain, the IoT, and artificial intelligence.



**MANGAYARKARASI RAMAIAH** received the Ph.D. degree in information technology and engineering from the Vellore Institute of Technology (VIT University), Vellore, India, and the M.E. degree in computer science from Anna University. She is currently an Associate Professor with the School of Computer Science Engineering and Information Systems, VIT University. She has attended many national and international conferences and published articles in reputed journals.

Her research interests include computer vision, image processing, cyber security, machine learning, and artificial intelligence.

• • •