International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Blockchain-Based Service Oriented Privacy-Preserving Data Sharing over Distributed Data Streams in Asynchronous Environment

Suganthi P.*[1], Malini A.[2] , Felicia Lilian J.[3], Divya Vetriveeran[4]

Submitted: 06/05/2023 **Revised**: 18/07/2023 **Accepted**: 07/08/2023

Abstract: Innovative city applications use information and communication technologies to function various operations efficiently. The widespread use of the Internet of Things (IoT) can be viewed in several applications like smart cars, smart cities, e-commerce, and cyber-physical systems. The huge amount of data produced and transmitted by these systems is handled by cloud-based storage services, which are vulnerable to multiple threats risking the privacy and security features of the application. Cloud storage services employ encryption algorithms to ensure data confidentiality, but it fails to address the privacy issues. Apart from the privacy risks, in these systems, the identity of a user who shares and accesses the data is traceable, as it is required to verify user eligibility before providing access. Also, a vast amount of daily data is stored on a centralized system that processes service requests from multiple users, posing considerable risks to the system's stability during peak periods. To address these challenges faced during the data sharing process in a centralized system, Service Oriented Privacy-Preserving Data Sharing (SOPPDS) platform based on a blockchain framework is proposed. The modified Key Policy-Attribute-based Encryption (MKP-ABE) technique is applied to securely share the data between the service owners and the service consumers. It was evident from the performance evaluation of the proposed SOPPDS platform that the encryption process takes lesser time than the decryption process. Also, the cryptographic operations performed on the prime order sets exhibited increased latency and computational cost. It was observed that comparatively, cryptographic operations performed on composite order sets could overcome the issues in prime order sets. SOPPDS platform works well in preserving the users' privacy, ensuring anonymity in the data sharing process, and maintaining the confidentiality of the data shared in the system

Keywords: *Blockchain, Privacy-preserving, data sharing, smart contract, Internet of Things, Smart City*

1. Introduction

With the advent of Smart cities, services offered by IoT devices in manufacturing and healthcare have increased consistently. As per recent statistics, the number of IoT-connected devices in use is expected to grow to 35 million in 2024 [1]. This tremendous growth and employment of IoT devices contributes to improvement in people's quality and empowers the world's economy. Thus, the contribution of IoT to the improvement of the global economy is expected to be around USD 8.1 trillion in 2025[2]. Another survey predicts that the urban population around the world will go up to 5 billion by the end of 2030. Such a drastic increase in urbanization across the globe creates a high demand for the effective utilization of the available resources in the cities, thereby leading to perceptive governance and effective delivery of services. The intelligent services provided by the IoT devices [5] include smart parking, which can be used to enhance the usage of parking spaces and thus regulate the parking operations and traffic flow, and smart health, which can be used for tracking patient health in remote and responding to the emergency cases instantly, eToll which can be used to allow people to make the road toll payments electronically. Though the use of IoT devices in Smart cities offers innumerable benefits, the downside lies in the fact that these devices are vulnerable to security and privacy attacks [3]. This, in turn, poses a threat to the Smart city environment in the form of attacks on data availability, data integrity, and data privacy. In addition, there is also evidence of false injection of data, and in the case of centralized control, it may lead to a single-point failure.

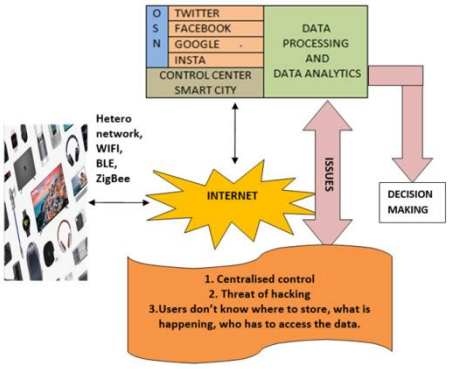

Fig. 1. Issues in Data Sharing and Processing

[1]*Department of CSBS, Thiagarajar College of Engineering, Madurai – 625015, Tamilnadu, INDIA*
ORCID ID : 0000-0002-3815-5177
[1]*Department of CSBS, Thiagarajar College of Engineering, Madurai – 625015, Tamilnadu, INDIA*
ORCID ID : 0000-0002-3324-5317
[3]*Department of CSBS, Thiagarajar College of Engineering, Madurai – 625015, Tamilnadu, INDIA*
ORCID ID : 0000-0002-1221-2460
[3] *Department of CSE, School of Engineering and Technology, CHRIST University, Bangalore – 560074, INDIA*
ORCID ID : 0000-0003-2249-7591
** Corresponding Author Email: suga.pathma@gmail.com*

From Figure 1, it can be inferred that the user data is generally collected through various smart devices such as smartphones, smart watches, smart meters, and intelligent home surveillance. All this data is then processed via social media networks such as Facebook, Google, or control centers available in the smart cities for data processing and analytics [4]. This processed data is further used to provide personalized services to the users through automated procedures, directory assistance, and targeted ads. The main problem with such data processing is that it involves centralized control, which is directly associated with many security breaches such as cyber-attacks, denial-of-service attacks, data leakage, unauthorized access, and malicious intruder attacks. There are three major user concerns linked with the use of centralized control:

1. Centralized control implies a drawback to introducing single point failure whenever there is an event of a cyber-attack and may also lead to technical malfunctioning of the system.

2. There are also chances for issues related to trust, as the system users must trust the entity involved in the data processing.

3. The users have no clue regarding the location of data storage and processing.

The details about the authorized parties to the data are unknown, and if there are any possibilities for unauthorized access is also hidden from the user. These concerns are proven realistic due to the evidence of events like the exposure of personal data of millions of users by Facebook and Google Plus.Privacy leaks in intelligent city data can be classified under two broad categories. One is the External Threats, and the other is Internal Threats. The latter category of threats is most often caused by the personnel working in an organization for several reasons, such as illegal mitigations or improper usage owing to personal interests. Attackers in the former category of threats exploit the system's vulnerabilities to explore the background to invade information during data sharing. Recently, there has been a greater interest in using blockchain technology [6] for innovative city applications. Blockchain for smart cities [7] can be a safe and reliable solution, thus improving the transparency of the data shared across several sectors in the smart city. Furthermore, any intelligent city application incorporates storing, processing, and analysis of users' data. Blockchain, an innovative and disruptive technology, is a promising solution to IoT security and privacy issues. These issues can be leveraged by the cryptographic security benefits offered by blockchain technology. The main contribution of this work is as follows:

We propose a Service Oriented Privacy-Preserving Data Sharing (SOPPDS) platform to explore the feasible blockchain approach for privacy-preserving data sharing through the construction of a practical framework, smart contract, and data sharing procedure.

1. We propose a Modified Key Policy-Attribute Based Encryption (MKP-ABE) technique to assure the confidentiality of data when it is shared between the service owner and service consumer.

2. We propose a novel approach of creating a verifiable user access list that is stored on the blockchain to manage the system users who are registered as service owners and consumers. With this approach, the user can demonstrate the right over the data shared in the system without having the necessity to reveal the identity.

The remainder of the research article is organized as follows: We first present related works in Section 2. Section 3 describes the system architecture with the system and threat models. Section 4 outlines the various components involved in the proposed architecture. Next, Section 5 details the implementation of the proposed SOPPDS approach. We analyze the privacy-preserving nature of the proposed platform in Section 6. In Section 7, we evaluate the performance of the proposed system. We conclude our work in Section 8.

2. Related Work

Researchers have been found worldwide to implement the new technologies in binding the blockchain technology in the environment of IoT. It is more convenient in the case of decentralized control, cryptographic security, immutability, and smart tolerance. In [8], the author developed a hybrid network for a smart city with Software Defined Networking (SDN) and blockchain-based network architecture for an intelligent city. The proposed network architecture addresses the issues in the smart city, such as security, high TX latency, privacy, high computational resources, and bandwidth congestion. The proposed network is divided into a distributed core network and a centralized edge network. With this, the access policies for the locally registered nodes are stored by the edge nodes. The proposed network architecture reduces the TX latency and the network bandwidth. Several works focus on user anonymity in distribution networks, such as Freenet [9], which allows the owners to encrypt the data with their names, and Free Haven [10], which does not allow an encryption mechanism to protect the data confidentiality. Users can freely access and query the networks to get the desired output data. The routing protocol is used to pass the requests to the data host and to return the data to the requesters to protect the data from the data owners and the data retriever. This system does not allow the data owners to restrict access to specific users. Hence, user revocation is also not possible in these systems. In the cloud-based

environment, Ferran et al. [11] proposed a group signature technique to solve the issues of user anonymity. The group manager finds the master key (MK) and the system public key (PK) in the proposed group signature. Each user in the group receives the private key in the context of the master key. To access the shared data, every member can develop their signature **δ** in place of the group by using its private key. The storage provider verifies the validation of the signature **δ** with the help of PK before granting access to the requested data. It allows the storage provider to verify the signature **δ** to check whether the group member creates the **δ** or not. This approach has two disadvantages. First, the group members must directly communicate with the group manager about the private keys before retrieving data. This condition is eliminated in this proposed system design. Secondly, the data access policy is much simplified and is used to verify the group signatures. The access control policy is stored in the storage service for further updation. In [12], [13], the authors make use of Ciphertext-Policy Attribute-Based Encryption (CP-ABE), which allows the ciphertext to control lists in order to protect the confidentiality of the data. Even though the data can have the drawback of preserving the privacy the data. In another attempt [14], the researchers developed the proposed work to manipulate the different privacy issues of the user. The major work in [15], [16], and [17], is to develop a blockchain based on access control solutions and no one has not worked into the user's private account. This proposed work simply solves the suitability of the data, such that the data owner may audit the data later. In the blockchain, it can access the control lists through the blockchain in which the data can access everyone so that it violates the privacy of the data. Similarly, [18], they have proposed a secure, private, and very lightweight architecture of a smart home application based on the blockchain. Many researchers developed a data-sharing framework with a blockchain-based concept for making a smart city environment [19,]. This framework is often called a "speedy chain" which is used to reduce the TX settlement time for real-world applications such as smart vehicles. It also aims at providing user privacy, data integrity, non-repudiation, and tamper resistance. other issues handled by [20], to solve the issues on computational issues, latency, and energy consumption. In order to reduce the energy consumption and computational overhead, every block is computed with Proof of Work (PoW). In [26], he proposed a technique with Artificial Intelligence (AI) and blockchain for data analytics and to store the results respectively for providing the smart contract-based environment [29].

3. System Architecture

3.1 System Model

The system model for the proposed Service Oriented Privacy-Preserving Data Sharing (SOPPDS) approach consists of Service Owner (SO), Service Consumer (SC), Service Record (SR), Blockchain (BC), Interplanetary File System (IPFS). The data sharing mechanism in a traditional system is depicted in Figure 2

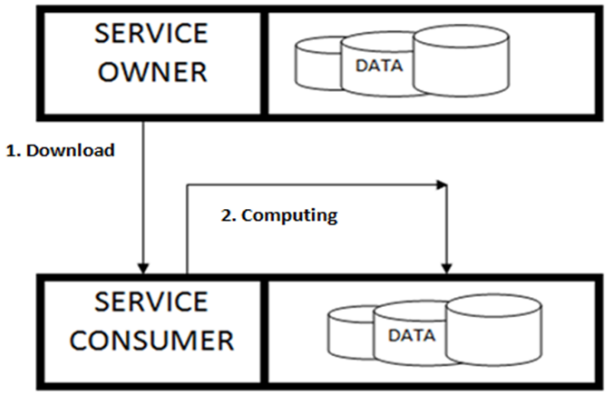

Fig. 2. Traditional Data Sharing

3.1.1 Interplanetary File System

IPFS is a peer-to-peer protocol that is widely adopted across the world to perform data uploading, downloading, and sharing operations in a rapid manner. This protocol makes use of the content-based addressing mechanism and is also known as the successful integration of the advantages of other peer-to-peer systems such as BitTorrent and Git [21]. IPFS is a powerful decentralized system that offers fast retrieval of data blocks in the blockchain. Filecoin [24], is an extension of the IPFS system which is aimed to be a global decentralized storage platform. Filecoin also amalgamates the incentive mechanism through which it allows a price for the unused disks and sells it out to others for data storing purposes [22].

3.1.2 Blockchain

Blockchain can generally be considered as a peer-to-peer network which is a distributed ledger-based database. It is mainly generated using cryptographic techniques [33]. Each and every transaction must be approved by all the participating entities in the system. The various components that constitute the blockchain system comprise peer-to-peer networks [27], smart contracts [25], consensus mechanism, distributed ledger, and asymmetric encryption [23]. These are the building blocks of blockchain technology which enables the transactions to be validated with steady flow continuously on distributed networks. The proposed architecture incorporates the Ethereum blockchain for creating smart contracts which can be used to manage the data sharing process [34].

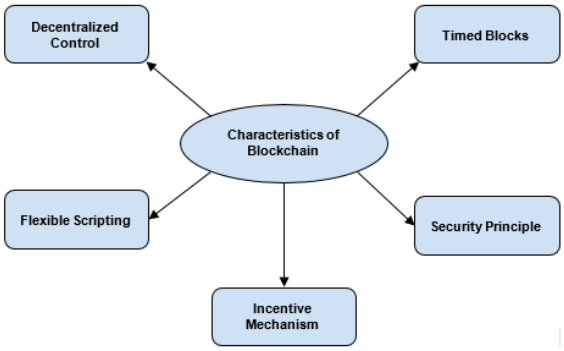

Fig. 3. Blockchain technology

Blockchain technology has major characteristics (Figure 3) such as decentralized control, timed blocks, incentive mechanism, flexible scripting, and security principle.

1. Decentralized control - Distributed system structure of the Blockchain technology is responsible to track, store, maintaining, and transmitting the blockchain data. In the blockchain, complete intuitive mathematical models [30] are employed to create the trust relations between all the nodes included in the distributed system. This characteristic is essential in order to build a distributed system that is decentralized as well as trustworthy in nature.

2. Timed blocks - Every data stored in the blockchain is associated with a block that is timestamped [31]. This feature is necessary to include the time parameter to the blockchain data thus improving its robust verifiable and traceable nature.

3. Incentive mechanism - This mechanism is needed to ensure that all the nodes that are included in the distributed system must participate in the process of verification of the data blocks which is similar to the mining process of bitcoin. In order to add a new block to the blockchain network, consensus algorithms can be used.

4. Flexible scripting - The programming structure in the blockchain is very flexible such that it allows the creation of enhanced smart contracts, different currencies, and many applications that are decentralized in nature. Consider the example of Ethereum which is a scripting language but Turing complete that can be used to build smart contracts.

5. Security principle - The cryptographic algorithm used in blockchain technology to encrypt the data is asymmetric encryption [32]. The consensus algorithm is responsible to endure external attacks on the system and prevent the data from being corrupted.

3.1.3 Service Owner

These are the users who store the data as services in the IPFS. The service owner is authorized to encrypt the data, alter the access rights and revoke access to any data. Whenever the access for a service consumer is revoked, then the service owner is responsible to create the re-encryption key and communicate it to all the storage nodes for enabling them to re-encrypt the data. In this way, the users whose access was removed will no longer be able to decrypt the data.

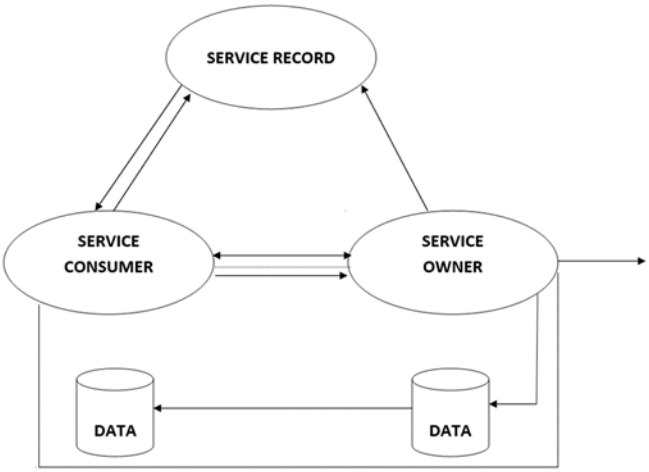

Fig. 4. System model of the proposed platform

3.1.4 Service Consumer

These are the users who request access to the data shared as services by the service owners. Once access is granted, service consumers gain authorization to the data provided by the service owners. The interconnection between Service Owner and Consumer is depicted in Figure 4.

3.1.5 Service Record

This is a directory that acts as a public blockchain platform to allow the users to participate in the network. The users in this system are free to access the ledger for reading and writing purposes. It is also possible to add nodes to the system with ease. There is a consensus that is commonly distributed among all the participating users in the network and this is the most contributing factor to the transparency of this blockchain system.

3.2 Threat Model

It is highly essential to formulate the security as well as privacy requirements that are to be satisfied by the proposed SOPPDS platform [28]. The proposed platform is built with the motive to assure the following requirements.

1. **Data Secrecy** - The proposed model maintains the secrecy of data by sharing it in encrypted form with the authorized users only, thereby prohibiting data access for anonymous users who do not have the required decryption key to decrypt the data. If in case the user access to data is removed, then the data is encrypted again with the re-encryption technique, by which the users whose access was removed will not be able to access the data further.
2. **Data Consumer Privacy** - The identity of the users

involved in the system is hidden from each other. Service Owners and Service Consumers request and provide the data anonymously without sharing any personal information with respect to their identity. Thus, the users of the system will never be able to identify who has shared the information ever.

3. **User and Data Connectivity** - This approach is devised in such a way that it is not possible to connect the activities performed by the same user in the network. It is ideally difficult to recognize the number of times a particular user has consumed the data shared in the network and also it is highly impossible to predict the number of times a service owner has executed data sharings.

4. Proposed Architecture

This section briefly describes the components involved in building the architecture of the proposed Service Oriented Privacy-Preserving Data Sharing (SOPPDS) Platform as in Figure 5. The architecture of the proposed platform is divided between two perspectives: The user Perspective and Blockchain Perspective.

4.1 User Perspective

Once the user of the system intends to share data anonymously then the user is connected with the following components of the platform to execute the task.

4.1.1 Data Sharing Source (DSS)

This data sharing source maintains a verifiable user access list $VUAlist$which consists of the information about the service consumers with whom the data uploaded by the service owner is shared. This data sharing platform is designed as an asynchronous distributed environment which does not require the service consumers to be available online when the data is shared by a service owner. The data consumers later can detect the new data shared using the $VUAlist$.

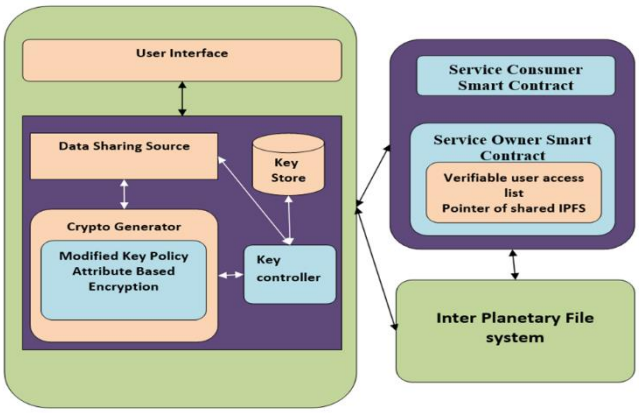

Fig. 5. Proposed SOPPDS Platform Architecture

4.1.2 Crypto Generator (CG)

This component is responsible for handling all the operations relevant to the cryptographic computations in the platform. Verifiable user access list is regularly updated by CG based on the data sharing requests to the DSS and it ensures confidentiality of data by performing encryption and decryption on it. This component is used to identify and validate the eligibility proof of each service consumers to access shared data. It is also in control of creating hidden transactions for the service owners and service consumers to hide their identity from each other.

4.1.3 Key Controller (KC)

The role of KC is to maintain an authenticated storage and retrieval of keys involved in encryption and decryption process from the Key Store.

4.1.4 Key Store (KS)

This is a database that is available on the user side to store the keys that are distributed between the service owner and the service consumer.

4.2 Blockchain Perspective

There are two types of Smart Contracts created on the platform from the blockchain perspective.

4.2.1 Service Consumer Smart Contract

This is generally a common contract for all the users of the platform to register the unique public keys which are associated with their real identity. This is used by the service owners to recognize the service consumers registered on the platform before sharing the data with them. Prior to sharing data in the platform, it is essential for all the users to exchange unique public keys over a stable out-of-band medium. In cases when the user wishes to change the existing unique public key with a new one, then both the signed unique public keys must be updated mutually in the platform.

4.2.2 Service Owner Smart Contract

The service owner created this contract to identify and locate the data shared in the platform and also update the details on the $VUAlist$. This list is maintained regularly and audited on regular intervals of time. All the requests made by the service consumers are registered in this contract on a regular basis. This aspect is essential to ensure that the transactions made on the shared data are transparent and auditable by the service owner.

4.3 Key Generation Mechanisms

The key Controller is responsible for loading and fetching keys from the Key store in the proposed platform. Four different categories are defined in this platform with the

public key(Pk) and private key(Sk) defined for each of the key categories.

Type 1 - Unique Keys (uPk,Sk)

Every usr registered in the platform is associated with a uPk and uSkregistering their identity in the system. These public keys are generally transmitted across an out-of-the-band medium in a most secure way without violating the rules for preserving privacy.

Type 2 - Contract Keys (cPk,Sk)

A user of the system is allowed to hold any number of contract keys with a restriction that using one private contract key only one smart contract should be deployed.

Type 3 - Transient Keys (tPk,Sk)

The kind of keys are short lived keys which are created for one-time purposes. These keys can be employed by the user to sign any smart contract with a restriction to contracts which needs to be signed along with the Contract Private key(cSk).

Type 4 - Latent Keys ($lPk{-}So{-}SC$,$Sk{-}So{-}SC$)

These keys are used in order to hide the identity of the users involved in the transactions from the third parties in the system.

4.4 Unique Key Generation Algorithm

Type 1, 2 and 3 keys are generated based on the $Algorithm$ 1. Though $Algorithm$ 1 describes the generation of unique keys, it can be utilized for generating the other two types of keys as well and the only difference between them depends on the usage purposes.

Algorithm 1:UniqueKeyGenerationAlgorithm

Input: ConsideracyclicGroup$R_G$inthe order$D$whichisderivedby$F$

Output: P_k,uS_k

Step 1: Select arandom numberathatbelongs toinfinitecyclicgroupZ_G

Step 2: Assign $uS_k = a$

Step 3: Assign $uP_k = aF$

Step 4: return uP_k,uS_k

preserving the identity of both the Service Owner (SO) and Service Consumer (SC). As per this platform, SO is allowed to agree with any transaction by signing it with one transient Key to hide the identity. The main difficulty lies in protecting the identity of the SC. Whenever a SC requests for a service, it is sent to SC through a random address such that it is known to SC only. It is also required that SC knows the SO from whom the service is offered. The reason for this is to make the SC aware of the private key he needs to use for decryption purposes. These issues are overcome in the proposed platform using the latent keys. Latent Transactions are also employed as an additional measure to tackle this issue. Consider a scenario where the SO wants to be involved in a latent transaction with the SC. To accomplish this, the first step is the generation of a transient key pair consisting of $tPk{-}SO$ and $tSk{-}SO$ where $tPk{-}SO = t$ and $tSk{-}SO = tF$. In the next step, SO uses the public unique key of SC $uPk{-}SC$and $tPk{-}SO$. These keys are applied to $Algorithm$ 2 in order to generate the public latent key $lPk{-}So{-}SC$. Following this, SO begins a transaction and uses the private transient key $tSk{-}SO$ to sign the transaction and this is sent to the resultant public latent key $lPk{-}So{-}SC$.

SC will flip through the blockchain network to determine if there is any latent transaction registered under it and computes the relevant private latent key $lSk{-}So{-}SC$ for the transaction. If the public transient key $tPk{-}SO$of the latent transaction is known, Sc will use its private unique key $uSk{-}SC$to determine the new latent key $nlPk{-}So{-}SC = (tPk{-}SO.uSk{-}SC)F + uPk{-}SC = H(l.ar.F)F + arF$. If $lPk{-}So{-}SC = lPk{-}So{-}S$, it can be concluded that the latent transaction is for the intended SC only. Then the private latent key $lSk{-}So{-}SC = (l.ar.F)+ar1$ is computed. In this case, it is not possible for the third parties to determine the $lSk{-}So{-}SC$without the $uSk{-}SC$, thus making it impossible to detect the identity of the service consumer in the transaction. To facilitate SC to identify the SO of the transaction, SO is required to encrypt the public unique key $uPk{-}SO$along with the public latent key $lPk{-}So{-}SC$in the transaction. Thus, SC determines the private latent key to decrypt $uPk{-}SO$. In this way, the privacy of the distributed data sharing is preserved.

Algorithm2:LatentKeyGenerationAlgorithm
Input: $S_k{=}1$, $uP_k{=}a_rF$
Output: $P_{k-So-SC}$
Step 1: Determine$x = H(l.a_r.F)$
Step2: Determine$lP_{ks} = xF + a_rF$
Step3:return$lP_{k-So-SC}$

Modified Key Policy-Attribute based Encryption (MKP-ABE). This section describes the encryption and decryption process involved in the MKP-ABE algorithm in detail.

5. Definition

As per Newton's interpolation, polynomial of degree d from a group of points which are d added to 1, $\{\boldsymbol{a_k}, \boldsymbol{P(a_k)}\}_{\boldsymbol{k=0\ldots d}}$using the equation (1),

$$\boldsymbol{P(a)} = \boldsymbol{\Sigma_{k=0}^{d} P(a_k)} \triangle_{\boldsymbol{k,N(a)}}; \quad \textbf{(1)}$$

where $\triangle_{k,N(a)} = \Pi_{x \neq y; y \in [0,d]} \frac{a - a_x}{a_x - a_y}$

Setup

Consider a security factor sf for which a group R_g is generated in cyclic order $= a.b.c$, here a, b *and* c are prime numbers. The groups R_{g-a}, R_{g-b},R_{g-c} correspond to groups for a, b *and* c respectively. Assume PF to be the function to be used over the generated groups for performing pairing.

Function PF is defined as $PF : R_g \times R_g \rightarrow R_{g-T}$. Also consider the hash function $HF: \{0,1\}^* \rightarrow R_g$. In order to produce the primary key PM_k and public key P_k , values such as $S_{1,k}$,$S_{2,k} \in R_{g-c}$ and $t_{1,k}, t_{2,k} \in R_{g-a}$ are chosen. Next, $\in Z_p^*$, $t \in R_{g-a}$ is chosen to compute PM_k and P_k using the equation (2) and (3).

$$PM_k = (a, b, c, r_{g-b}, t^{-\sigma}, \{t_{1,k}, t_{2,k}\}_{k=1}^{d}) \quad (2)$$

$$P_k = (R_{g-a}, R_{g-c}, B = R_{g-b} S_0, A = PF(R_{g-a}, t)^{\sigma}, \{T_{1,k} = t_{1,k} S_{1,k}, T_{2,k} = t_{2,k} S_{2,k}\}_{k=1}^{d}) \quad (3)$$

To perform repudiation of users, a number is chosen in random $\boldsymbol{n \in Z_p^*}$ to form a polynomial of degree d to repudiate the users in equation (4).

$$P(a) = \Sigma_{k=0}^{d} p_k a^k \; where \; P(0) = n \quad (4)$$

Again, a random number is selected from Z_p^* as $r \in Z_p^*$ to determine the primary repudiation key PR_k in equation (5), non-repudiation proof $NR_p$and public repudiation key public repudiation key PB_k in equation (6).

$$PR_k = n \quad (5)$$

$$PB_k = (s_c^y, s_c^x) \quad (6)$$

$$NR_p = s_c^{xy} \quad (7)$$

Encrypt

To encrypt the data d with a set of vectors $v = (v_1, v_2, v_3, \ldots, v_k)$, where $v_n \in Z_N^*$, random values for $\nu, \delta, \varepsilon \in Z_N^*$ and $S_{3,k}$,$S_{4,k} \in R_{g-c}$ are chosen to compute the following values.

$$B = B' = dP^{v} \hat{e}(s_c^y, s_c^x)^{v}, B_1 = s_p^v, \{B_{1,i} = H_{1,i}^{v} C^{\delta v_n} S_{3,k}, B_{2,i} = H_{2,i}^{v} C^{\varepsilon v_n} S_{4,k}\}_{k=1}^{d} \quad (8)$$

Key Generate

Consider a set of vectors$m = (m_1, m_2, m_3, \ldots, m_k)$ to produce the private key linked with each user. To accomplish this, random numbers such as $s_{1,k}$,$s_{2,k} \in Z_a^*$, here the value of k varies between 0 and d.

Also, few random values are chosen for $g_1, g_2 \in Z_b^*$, $S_5 \in R_{g-c}$ and $C_6 \in R_{g-b}$. Then the private key is generated using the equation (9).

$$S_{k,m} = \left(K = S_5 C_6 t^{-\sigma} \Pi_{k=1}^{d} t_{1.k}^{-s_{1,k}} t_{2.k}^{-s_{2,k}}\right) \quad (9)$$

The non-repudiation key(NR_k) is computed in equation (10) using a special number $sNum$which is unique to every user.

$$NR_k = sNum \quad (10)$$

Key Update

Consider the case when k users are supposed to be repudiated from the system who are linked with a list of k non-repudiation keys $N = \{NR_{k,j}\}_{j=1\ldots k}$. Suppose $k < t$, then the system will pick $t - k$ numbers in random and determines $NR_j = P(j)$. Further, $\underline{e} \in Z_p^*$. The non-repudiation update is determined in equation (11).

$$NR_U = (h^{\underline{e}}, \{k, h^{\underline{e} NR_{k,j}}\}_{k=1\ldots d}) \quad (11)$$

Re-Encrypt

The ciphertext B can be re-encrypted by modifying B'element in B into B'_{re}using a re-encryption key.

$$B'_{re} = B' \underline{e}(s_c^{-xy} H(A), s_c^r, A \underline{e}(s_c^x, s_c^y)^r) \quad (12)$$

The re-encrypted ciphertext can be represented as follows.

$$B_{re} = (B'_{re} = sP^m \hat{e}(H(A), s_c^m), B_1 = s_c^m, B_a = s_c^a, B_x = A \underline{e}(s_c^x, s_c^y)^r,$$

$$\{B_{1,i} = H_{1,i}^{v} C^{\delta v_n} S_{3,k}, B_{2,i} = H_{2,i}^{v} C^{\varepsilon v_n} S_{4,k}\}_{k=1}^{d}) \quad (13)$$

Decrypt

Decryption process depends on the nature of the ciphertext and private key is used to decrypt the ciphertext$B$.

$$\hat{d} = \frac{B' \hat{e}(B_1, D) \Pi_{k=1}^{d} \hat{e}(B_{1,k}, D_{1,k}) \hat{e}(B_{2,k}, D_{2,k})}{\hat{e}(NR_P, B_1)}$$

$$\hat{d} = d \; \hat{e}(s_a, s_b)^{(\delta g_1, \varepsilon g_2) < v, m >} \quad (14)$$

When the re-encrypted ciphertext is to be decrypted, non-repudiation proof is used.

$$D = \frac{B_a}{\hat{e}(NR_P, B_r)} = \frac{D \hat{e}(s_c^x, s_c^y)^r}{\hat{e}(s_c^{xy}, s_c^r)} \quad (15)$$

The original message can be obtained by applying the above value to the decryption using equation (16).

$$\hat{d} = \frac{B_{re}' \hat{e}(B_1, D) \Pi_{k=1}^{d} \hat{e}(B_{1,k}, D_{1,k}) \hat{e}(B_{2,k}, D_{2,k})}{\hat{e}(H(D), B_1)}$$

$$\hat{d} = d \; \hat{e}(s_a, s_b)^{(\delta g_1, \varepsilon g_2) < v, m >} \quad (16)$$

5. Soppds Platform Implementation

This section discusses the implementation of SOPPDS platform and the operations performed are described in Figure 6.

5.1 Registering in SOPPDS Platform

For any user to be a part of the proposed platform, it is mandatory to get registered as an initial step. As part of the registration process, the user must register with the public unique key smart contract which is implemented on the blockchain. This process involves the creation of a unique key pair containing the public (uPk)and private ($uSk{-}SC$) keys as per $Algorithm$ 1. After creation of the keys,$Pk$is used in the contract to complete the registration process. These unique keys describe the identity of the user and must not be shared with other users which would result in violation of privacy rules.

5.2 Distributing keys for Service sharing

The service owner employs Modified Key Policy-Attribute based Encryption (MKP-ABE) technique to perform encryption and secure the data before sharing it to the service consumers. It is the responsibility of the service owners to distribute the decryption keys as well to the service consumers to perform the decryption process on the consumer end. The private decryption keys are shared with the service consumers. These private decryption keys need to be shared to the public unique key $uP_k$of the service consumers. One major issue associated with this type of transaction is the privacy risk involved in sharing the private encryption keys directly to the public unique keys of a service consumer.

In order to tackle this issue, $Algorithm$ 3 is proposed to create a hidden transaction in order to share the private decryption keys to the service consumers. This latent transaction is sent to the blockchain and when the service consumer flips through the blockchain, the latent transactions can be viewed by the intended service consumers.

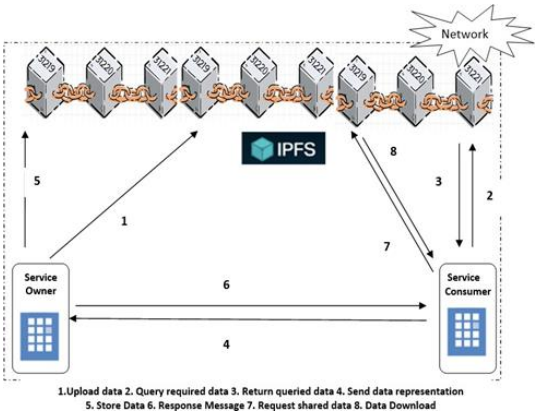

Fig. 6. SOPPDS Platform Implementation

Algorithm3: Creation of Latent Transaction Algorithm

Input: public unique key of service owner

uP_{k-SC} private encryption keydS_{k-SC}

Output: latent transaction LT_{SC}

Step1: Determine public latent key$lP_{k-So-SC}$

Step2: Perform encryption $e_1$ofuP_{k-SO} using $lP_{k-So-SC}$

Step3: Perform encryption$e_2$ofdS_{k-SC}using

$lP_{k-So-SC}$

Step 4: Create a latent transaction $LT_{SC}$with $e_1$and

$e_2$and send it along with private transient keytS_k

Step5: returnLT_{SC}

Once the latent transaction is received by the service consumer, the private latent key is computed $lS_{k-SO-SC}$ which can be used to decrypt $e_1$ande_2 to obtain the private decryption key dS_{k-SC} and public unique key uP_{k-SC}. The sedS_{k-SC},uP_{k-SC} keys are stored on the service consumer end for any future purposes. In this type of data sharing, it is highly impossible for any third parties to recognize the identity of the service owner and service consumer.

5.3 Service Sharing by SO

To upload the data into the network for sharing purposes, it is essential to create a pair of keys based on the smart contract $smP_k$and smS_k. A new smart contract can further be created on the blockchain by using smS_k. smP_krepresents the public key of the service owner and one data upload requires on smart contract to be created. Data is uploaded after encrypting it using the KP-ABE technique. The encrypted data is uploaded onto the Interplanetary File System. Upon successful upload of data, the address of the data location in IPFS is returned. The integrity of data can be verified by using the address which is a hash of the data that is in encrypted form. This address along with smS_k is updated in the smart contract to retrieve in the future. Once the data has been uploaded in the network, it should be shared with the service consumers. To facilitate this, a latent access record listLAR_{list} is used which contains the details of the service consumers registered in the system. Every element associated with the list is a combination of two keys. One is the public latent key lP_{k-SC} of a service consumer which can be generated using $Algorithm$2. The second key is an encrypted form of the public unique key of the service owner uP_{k-SO} and public latent key lP_{k-SC}. This transaction is then updated in the LAR_{list} in the corresponding smart contract and added to the blockchain. Whenever a service consumer flips through the blockchain, public latent keys can be verified to identify the new data being shared

recently along with the details about the service owner who shared it.

5.4 Service Access by SC

For a service consumer to access the data being shared by the service owner, the service consumer must validate two steps. The first step is to prove that the service consumer is registered in the latent access record list LAR_{list} of the corresponding smart contract. To accomplish this, the service consumer uses the private latent key lS_{k-SC} equivalent to the public latent key in the smart contract. In the second step, the service consumer is expected to possess a private key to decrypt the shared data.

6. Privacy Preserving Analysis for Proposed Architecture

The following section validates the proposed architecture to check if the privacy preserving data sharing requirements are satisfied by the SOPPDS platform.

6.1 Data Secrecy

To validate the data secrecy feature of the proposed Modified Key Policy-Attribute based Encryption (MKP-ABE), the setup denoted by MK is defined as follows.

Setup:

Consider a group $\boldsymbol{R_G}$ in cyclic order c generated by the series $\boldsymbol{s_c}$. Choose two numbers x and y in random that belongs to $\boldsymbol{Z_p^*}$. Construct primary repudiation key $\boldsymbol{PR_k} = \boldsymbol{y}$ Construct public repudiation key $\boldsymbol{PB_k} = (\boldsymbol{s_c^y}, \boldsymbol{s_c^x})$

Construct non repudiation proof $\boldsymbol{NR_p} = \boldsymbol{s_c^{xy}}$

Encrypt:

To encrypt any data d, m is chosen in random from Z_p^* and the equivalent ciphertext is determined using

$C_{text} = (s^m, d\hat{e}(s_c^x, s_c^y)^s)$.

Decrypt:

The encrypted data can be decrypted by using

$$d = \frac{d\hat{e}(s_c^x, s_c^y)^s}{\hat{e}(s_c^m, s_c^{xy})^s}$$

Those users who are repudiated from the transaction needs to determine $\hat{\boldsymbol{e}}(\boldsymbol{s_c^x}, \boldsymbol{s_c^y})^{\boldsymbol{s}}$ from the combination of values like $(\hat{\boldsymbol{e}}, \boldsymbol{s_c^x}, \boldsymbol{s_c^y}, \boldsymbol{s_c^m})$. These encryption and decryption schemes prove to preserve the secrecy of the data shared in the system.

6.2 Data Consumer Privacy

To protect the privacy of the service owner, everyone data is shared by a owner to the system a new smart contract is implemented with a set of key pairs $\boldsymbol{smP_k}$ and $\boldsymbol{smS_k}$. When a service consumer is to be added to the latent access record list $\boldsymbol{LAR_{list}}$, SO generates a public latent key $\boldsymbol{lP_{k-SO-SC}}$ to SC.

Thus $\boldsymbol{LAR_{list}}$ will contain all the public latent keys of the service consumers registered in the system and this list is then consecutively stored on the blockchain. It is provable that $\boldsymbol{lP_{k-SO-SC}}$ cannot be used to recognize the identity of either the SO or SC. According to Algorithm 2, $\boldsymbol{lP_{k-SO-SC}}$ is determined based on transient key pair $\boldsymbol{tP_{k-SO}} = \boldsymbol{t}$ and $\boldsymbol{tS_{k-SO}} = \boldsymbol{tF}$ and public unique key $\boldsymbol{uP_k} = \boldsymbol{a_r F}$ of SC such that $\boldsymbol{lP_{k-So-SC}} = \boldsymbol{H(l.a_r.F)F} + \boldsymbol{a_r I}$. Thus it is clear that if the identity of the service consumer needs to be identified through the $\boldsymbol{LAR_{list}}$ using $\boldsymbol{lP_{k-So-SC}}$, then it is essential to iterate through all the public unique keys $\boldsymbol{uP_k}$ associated with the $\boldsymbol{LAR_{list}}$, which is unfeasible and thus it is proved that the privacy of the service consumers are ensured by the proposed platform.

6.3User and Data Connectivity

It is highly impossible to link the activities carried out by a specific user in the proposed platform. Every data sharing activity in the system is performed by deploying a new smart contract using new key pair $\boldsymbol{smP_k}$ and $\boldsymbol{smS_k}$. These keys are volatile keys which can be used one time only. Similar to this, every data sharing activity is also linked with a set of latent keys $\boldsymbol{lP_{k-So-SC}}$ and $\boldsymbol{lS_{k-So-SC}}$. For every consumer these keys are also generated atrandom for every data sharing activity. Thus, it is obvious that for every data sharing activity of a service consumer an equivalent number of latent key pairs are generated which makes the service consumers untraceable across the system.

7. Performance Evaluation

The performance of the proposed system is discussed extensively in this section based on the cost incurred for deploying smart contracts in the blockchain and executing complex computations in the system. Ethereum blockchain is used for the implementation of the proposed SOPPDS platform. In this type of blockchain, miners are the key players who hold the responsibility to validate and insert transactions into the blockchain. Therefore, it is required to pay the miners to execute all the operations in the blockchain. The amount of work required to perform the computations is measured using the metric known as gas. To begin a transaction in this blockchain it is essential to set values for two factors such as gas bound and gas charge. Gas bound refers to the maximum boundary for the gas that can be spent by the sender. Gas charge refers to the price associated with the gas. Gas charge depends on ETH which is the crypto currency for this blockchain on which SOPPDS is deployed.

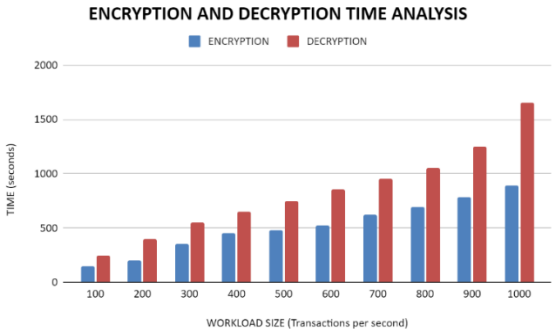

Fig. 7. Comparison of Encryption and Decryption Time

The Smart contracts such as Service consumer smart contract and Service owner smart contract are implemented using Solidity. The cryptographic operations involved in the proposed system requires two important operations such as Encrypt and Decrypt. Figure 7 depicts the comparison of encrypt and decrypt processes based on computational time and it is evident that the decrypt process consumes more time, thus assuring the efficiency of the proposed technique to perform privacy aware secure transactions.

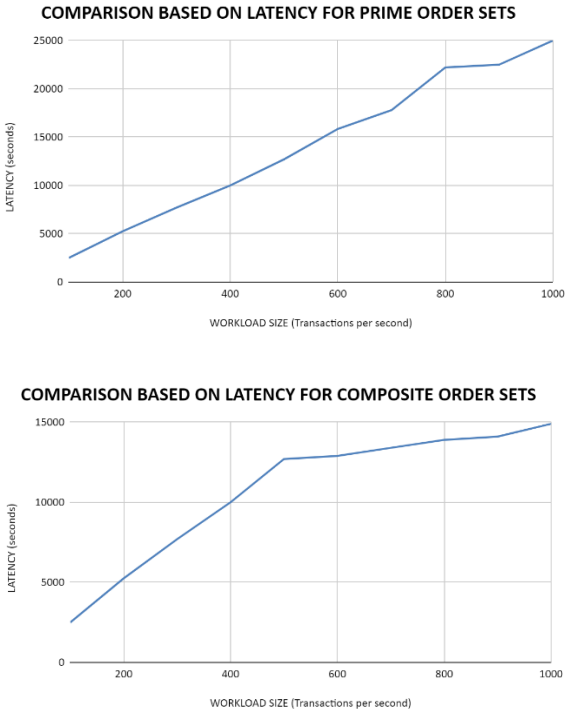

Fig. 8. Comparison of Latency for Order Sets

Further, the latency and cost incurred to execute the complex computations on the user side is analyzed. To demonstrate the efficiency of the proposed platform in performing these complex operations, it is implemented on a computer with an i7 processor having RAM capacity of 8 GB. The performance exhibited by the platform for both the operations is measured based on the cryptographic operations performed by varying the workload size. One possible solution to reduce the latency and cost for execution is to introduce composite order sets instead of prime order sets. There was a commendable improvement in the performance after applying composite order sets for the complex computations. It can be observed from Figure 8 and 9 that the composite order sets exhibit reduced latency and computational cost.

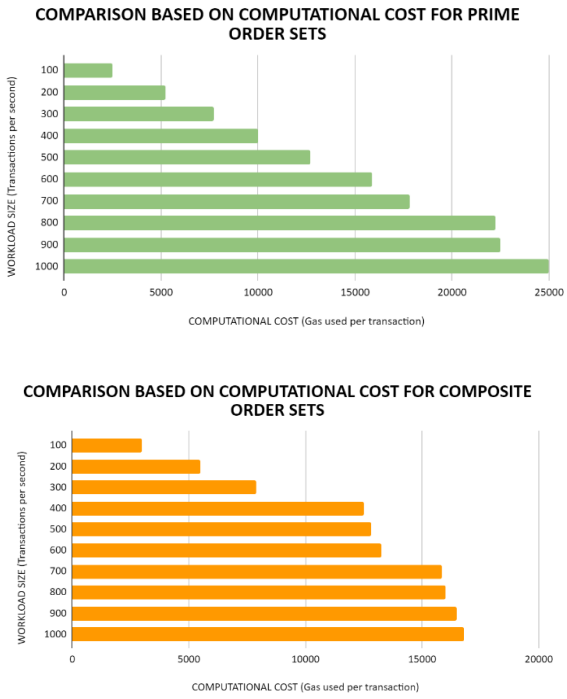

Fig. 9. Comparison of Computational Cost for Order Sets

8. Conclusion

Data generated from electronic gadgets are vulnerable to security and privacy issues to a more significant extent. Users have full authority over the data to manage and share it without compromising privacy. The transparent nature of Blockchain, along with decentralization features, can help to tackle the security and privacy threats related to data sharing. This paper proposes a Service Oriented Privacy-Preserving Data Sharing platform based on Blockchain. This platform ensures the confidentiality of the shared data while preserving the privacy of the activities performed by the users in the system. The users who own and share the data as well as who consume the data, can be monitored through the details available in the Verifiable user access list, which is saved on the Blockchain. The proposed platform serves as a powerful solution for the data sharing process with less burden on the user to ensure privacy. On the downside, though composite order sets reduce the latency and computational cost of cryptographic operations, there is still scope to find an approach to perform those operations with less complexity. In the future, we aim to extend this work by including more optimal parameters to this model to enable a balance between privacy-preserving as well as the availability of data.

Conflicts of interest

Authors do not have any conflict of interest based on financial or other personal considerations

References

[1] Apthorpe, N., Reisman, D., Feamster, N., "A smart home is no castle: privacy vulnerabilities of encrypted iot traffic," *arXiv,* 1705.06805, pp.1–6, 2017.

[2] Biswas, K., Muthukkumarasamy, V., "Securing Smart Cities using Blockchain Technology," *In: Proceedings of the 14th International Conference on Smart City High Performance Computing and Communications*, IEEE, pp. 1392–1393, 2016.

[3] M. V. Ahluwalia, A. Gangopadhyay, Z. Chen and Y. Yesha, "Target-Based, Privacy Preserving, and Incremental Association Rule Mining," in *IEEE Transactions on Services Computing*, vol. 10, no. 4, pp. 633- 645, 2017.

[4] Shen, J., Zhou, T., Chen, X., Li, J. and Susilo, W., "Anonymous and traceable group data sharing in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol.*13*,no.4, pp.912-925, 2017.

[5] Ouaddah, A. AbouElkalam, and A. AitOuahman, "Fairaccess: a new blockchain-based access control framework for the internet of things," *Security and Communication Networks*, pp. 5943–5964, 2016.

[6] J. Liu, X. Li, L. Ye, H. Zhang, X. Du, and M. Guizani, "Bpds: A blockchain based privacy-preserving data sharing for electronic medical records," in *GLOBECOM. IEEE*, pp. 1–6, 2018.

[7] S. Wang, Y. Zhang, and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," *IEEE Access*, vol. 6, pp. 38 437–38 450, 2018.

[8] Ahmad, I., Kumar, T., Liyanage, M., Okwuibe, J., Ylianttila, M., Gurtov, A.,"Overview of 5G security challenges and solutions", *Commun. Standards Mag.* Vol.2 (1), pp.36–43, 2018.

[9] Dorri, A., Kanhere, S.S., Jurdak, R., "Mof-bc: a memory optimized and flexible blockchain for large scale networks", *Future Generation Computational. System*, vol.92, pp.357–373,2019.

[10] Faber, B., Michelet, G.C., Weidmann, N., Mukkamala, R.R., Vatrapu, R.,"BPDIMS: a blockchain-based personal data and identity management system", *In: Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS)*. *IEEE*, pp. 6855–6864, 2019.

[11] Ferrag, M.A., Maglaras, L., Argyriou, A., Kosmanos, D., Janicke, H.,"Security for 4G and 5G cellular networks: a survey of existing authentication and privacy-preserving schemes", *Journal of Network and Computer Applications*, vol.101, pp.55–82, 2019.

[12] Gordon, W.J., Catalini, C., "Blockchain technology for healthcare: facilitating the transition to patient-driven interoperability*", Computational and Structural Biotechnology*, vol.16, pp. 224–230, 2018.

[13] Haidar, F., Kaiser, A., Lonc, B.,"On the Performance Evaluation of Vehicular PKI Protocol for V2X Communications Security", *In: Proceedings of the 86th Vehicular Technology Conference (VTC-Fall). IEEE*, pp. 1–5, 2017.

[14] Huh, S., Cho, S., Kim, S.,"ManagingIoT Devices using Blockchain Platform", *In: Proceedings of the 19th International Conference on Advanced Communication Technology (ICACT). IEEE*, pp. 464–467.

[15] Makhdoom, I., Abolhasan, M., Abbas, H., Ni, W., "Blockchain'S adoption in IoT: the challenges, and a way forward", *Journal of Network and Computer Applications, vol.*125, pp. 251–279, 2019.

[16] Makhdoom, I., Abolhasan, M., Lipman, J., Liu, R.P., Ni, W., "Anatomy of threats to the internet of things", *IEEE Communications Surveys & Tutorials*, vol.21 (2), pp.1636–1675, 2018a.

[17] Makhdoom, I., Abolhasan, M., Ni, W., "Blockchain for IoT: The Challenges and a Way Forward", *In: Proceedings of the 15th International Joint Conference on e-Business and Telecommunications*, SECRYPT. INSTICC. SciTePress , vol. 2., pp. 428–439, 2018b.

[18] Marchesi, M., Marchesi, L., Tonelli, R., "An agile software engineering method to design blockchain applications", *In: Proceedings of the 14th Central and Eastern European Software Engineering Conference Russia*. ACM, pp. 1–8, 2018.

[19] Michelin, R.A., Dorri, A., Steger, M., Lunardi, R.C., Kanhere, S.S., Jurdak, R., Zorzo, A.F., "SpeedyChain: A Framework for Decoupling Data from Blockchain for Smart Cities", *In:Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services.* ACM, pp. 145–154, 2018.

[20] Kaaniche, N., Laurent, M., "A blockchain-based data usage auditing architecture with enhanced privacy and availability", *In: Proceedings of the 16th International Symposium on Network Computing and Applications (NCA). IEEE*, pp. 1–5, 2017.

[21] Kountché, D.A., Bonnin, J.-M., Labiod, H., "The Problem of Privacy in Cooperative Intelligent Transportation Systems (C-ITS)", in *Proceedings of the Computer Communications Workshops (INFOCOM WKSHPS). IEEE*, pp. 482–486, 2017.

[22] Krishnan, K.N., Jenu, R., Joseph, T., Silpa, M., "Blockchain Based Security Framework for IoT

Implementations" in *Proceedings of the International CET Conference on Control, Communication, and Computing (IC4). IEEE*, pp. 425–429, 2018.

[23] Mach, P., Becvar, Z., "Mobile edge computing: a survey on architecture and computation offloading", *IEEE Communications Surveys & Tutorials*, vol.19 (3), pp.1628–1656, 2018.Moustaka, V., Theodosiou, Z., Vakali, A., Kounoudes, A., "Smart cities at risk: privacy and security borderlines from social networking in cities", *Athena* 357, pp.905–910.

[24] Qian, Y., Liu, Z., Yang, J., Wang, Q., "A Method of Exchanging Data in Smart City by Blockchain", in *Proceedings of the 16th International Conference on Smart City. IEEE*, pp. 1344–1349.

[25] Rahman, M.A., Rashid, M.M., Hossain, M.S., Hassanain, E.,Alhamid, M.F., Guizani, M., "Blockchain and IoT-based cognitive edge framework for sharing economy services in a smart city*", IEEE Access* vol.7, 18611–18621, 2019.

[26] J. Hua, A. Tang, Y. Fang, Z. Shen and S. Zhong, "Privacy-Preserving Utility Verification of the Data Published by Non-Interactive Differentially Private Mechanisms," in *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 10,pp. 2298-2311, 2016.

[27] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "Medshare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14 757–14 767, 2017.

[28] B. A. Tama, B. J. Kweka, Y. Park, and K. H. Rhee, "A critical review of blockchain and its current applications," *in Proceedings of 2017 International Conference on Electrical Engineering and Computer Science (ICECOS)*, pp. 109–113, 2017.

[29] Rabah, "Challenges & opportunities for blockchain powered healthcare systems: A review," *Mara Research Journal of Medicine and Health Sciences*, vol. 1, no. 1, pp. 45–52, 2017.

[30] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control," *Journal of Medical Systems*, vol. 40, no. 10, p. 218, 2016.

[31] Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *Proceedings of International Conference on Open and Big Data (OBD)*, pp. 25–30, 2016.

[32] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11 676–11 686, 2018.

[33] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in Proceedings of 2017 IEEE International Congress on Big Data, pp. 557–564, 2017.

[34] Huamaní, E. L. ., Leon-Ayala, R. ., Alva-Mantari, A. ., & Meneses-Claudio, B. . (2023). Prototype of a Mobile Application for the Detection of Car Accidents on the Roads of Peru . International Journal on Recent and Innovation Trends in Computing and Communication, 11(3), 37–42. https://doi.org/10.17762/ijritcc.v11i3.6198

[35] María, K., Järvinen, M., Dijk, A. van, Huber, K., & Weber, S. Machine Learning Approaches for Curriculum Design in Engineering Education. Kuwait Journal of Machine Learning, 1(1). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/111