Blockchain-Enabled Smart Parking

Shah Dhruvil
Institute of Technology, Nirma University, Ahmedabad, India
19bce248@nirmauni.ac.in

Shah Smit
Institute of Technology, Nirma University, Ahmedabad, India
19bce259@nirmauni.ac.in

Vachhani Jurin
Institute of Technology, Nirma University, Ahmedabad, India
19bce286@nirmauni.ac.in

Abstract

We present an integrated smart parking system in this paper. The proposed integrated smart parking system brings together multiple parking service providers under a single umbrella.In a smart city, a unified platform aims to provide commuters with one-stop parking information services. With such widespread use comes a high risk of this data being hacked and/or misused. Such attacks are common in applications that rely on a centralised server. Decentralized networks, such as the block chain, can compensate for the shortcomings of centralised servers.A consortium blockchain, in particular, is made up of parking lots to ensure the parking system's security, transparency, and availability. Then, in order to protect the drivers' location privacy, we utilise activating to conceal the drivers' locations. The blockchain validators return parking offers that are available within the cloaked area. Finally, the driver chooses the best offer and makes a direct reservation with the parking lot.

Index terms— Blockchain, Smart Parking, Privacy, Digital Signature, Ethereum,

1 Introduction

With the increased number of vehicles in recent years, finding a vacant parking space has become a major issue for drivers, particularly in congested cities. According to the report, more than 1.3 million drivers in Shanghai city struggle every day to find available parking spaces. This issue can waste drivers' time, cause traffic jams, and increase fuel consumption. In Cairo, Egypt, for example, drivers waste more than 20 minutes looking for available parking spaces. Furthermore, 30 percent of traffic jams are attributed to a lack of available parking spaces. Furthermore, in Los Angeles, the amount of gasoline consumed in finding vacant parking spaces is approximately 47,000 gallons per year, resulting in 728 tonnes of CO_2 emissions per year.Smart parking systems have emerged to help people find available parking spaces as wireless communications and Internet of Things (IoT) devices have advanced. IoT device is installed in each parking space to detect availability and send status information to a service provider. Drivers can check available parking spots and make online reservations through the service provider.

The most popular mechanism proposed by researchers to overcome the risk of single-point failure that can affect the entire system of the centralised system is blockchain. The reason for this is that blockchain technology is known as a decentralised model that also ensures data integrity.A centralised system is one in which a single controller has central authority over all of the system's components. This power can be wielded directly or indirectly through the use of hierarchy. This exhibits relatively complex behaviour as a result of this central controller's authority over the other components in the system. A decentralised system, on the other hand, is one in which each component is equally responsible for contributing to the system's overall behaviour. It accomplishes this by relying on local information rather than following the orders of any central authority. We propose a blockchain-based smart parking management system in this paper. To begin, a consortium blockchain created by various parking lots is introduced in order to securely store parking availability, rates, and available services. Each parking lot submits its parking offers to the blockchain network, which records them in a distributed shared ledger. The driver then sends a transaction to the blockchain network to retrieve parking offers in a cloaked cell in order to maintain his or her privacy. The driver then chooses the best offer based on his preferences, such as parking location, price, services, and so on. Finally, in order to maintain privacy, the driver pays for parking services with currency accepted.

2 Related Work

There are numerous parking solutions on the market. However, the majority of them are ineffective in providing individualised smart parking services. The use of Internet of Things devices to control and monitor the overall parking system is rapidly expanding. Abhirup Khanna and Rishi Anand proposed an architecture[2] for exchanging parking data that makes use of IoT devices and the TCP/IP protocol. Furthermore, the application was hosted on a centralised server, which is always vulnerable to a single point of failure.In [3], Rachapol Lookmuang et al. proposed another smart parking model aimed at reducing traffic congestion in the parking area. The researchers combined IoT devices with utilizing computer vision techniques to locate parked vehicles via smartphone application. Ajay Zajam and Surekha Dholay conducted research to find a convenient and nearby parking spot [4]. The researchers created an algorithm based on real-time traffic data to determine the best route between the user and a nearby parking location. BlueParking, an IoT-based parking reservation system, was implemented to find the best routing path for their destinations. Their traffic estimator service can automatically represent the status of various roads by analysing the congestion of location nodes.Bin Liu et al. proposed a blockchain-based framework [1] to ensure transparency and data integrity due to the lack of frameworks that can integrate data with public auditability without a trusted third party. This framework solves and improves the problem of dynamic data integrity verification in a fully decentralised environment. However, the parking centres are not linked together in a unified model to provide users with personalised parking information.

Some of the proposed schemes are incompatible with our smart parking scenario, in which parking owners own private idle parking spaces. Some of the existing schemes discussed above employ a centralised structure that is vulnerable to malicious central node misbehaviour.According to our review of existing research, the primary goal of current research on smart parking is to provide users with a parking location from a single parking service provider.Compared to previous works, we propose an integrated innovative parking system with the primary goal of connecting all parking service providers under a unified platform. While most existing solutions deal with the smart parking problem in a centralized manner that involves a trusted third party and cannot provide enough clarity, our proposed system produces transparency and is mostly inviolable due to the decentralized infrastructure.

3 Methodologies

The majority of people today are unable to find a parking lot for their vehicles. There are several applications on the market that can assist in resolving this issue.
A few of the available services are:

1. Parker
2. Parking Mate
3. ParkMe
4. SpotHero

However, the majority of these applications are built on centralised servers, which have their own set of drawbacks:

- Because there is only one central server, if a node loses connectivity, the entire system fails.
- The entire system would go downhill suddenly.
- Because the entire application is backed up by a single server, there is a high risk of various security attacks.

- There is little chance of data backup, and if the system fails, the data is permanently lost.

These issues can be addressed by developing applications on decentralised servers.

4 Design Principles

Some fundamental requirements must be met for smart parking to be fair, reliable, and privacy-preserving. The detailed design goals that correspond to these requirements are listed below.

1. Privacy Protection: The privacy of the driver and the parking owner should be protected. Identity privacy (i.e., the identity of the driver and the identity of the parking owner), location privacy (i.e., the queried location for drivers and the exact parking spot location for parking owners), time privacy (i.e., the available time for each parking spot and the parking time for each driver), and data privacy (i.e., sensitive information stored on blockchain) are the various types of privacy. Neither honest-but-curious RSUs nor malicious adversaries should be able to infringe on any type of privacy.

2. Integrity and Reliability: The proposed scheme should guarantee data integrity during transmission and storage. The malicious data modification should be detected by the receiver during data transmission. Furthermore, the decentralised network cannot be compromised, and adversaries cannot tamper with the stored data.

3. Authentication and Traceability: The drivers and parking owners should be authenticated to prevent unauthorized user from entering the system and impersonating legitimate users to interact with others. When some dispute occurs, TA needs to trace the identity of related users and realize the responsibility confirmation.

4. Fairness: The proposed scheme's fairness is comprised of two components: fair payment and reward and punishment. To be more specific, in order to achieve fairness:

    - The proposed scheme should ensure that both drivers and parking owners are paid fairly. Each driver will receive the correct matching results as long as they send parking requests and pay for them. As a reward, each parking owner will receive appropriate payments from the driver as long as a driver occupies their parking spot.

    - The honest worker participating in the decentralised network (i.e., the blockchain miner) should be rewarded, while misbehaving or malicious users and workers should be punished. The fines can be used to compensate drivers and parking lot owners who have suffered losses.

5. Efficiency: The VANET's limited bandwidth and computational resources should be considered, and communication and computation costs between drivers and RSUs should be kept low in order to achieve fast response time and efficient smart parking services.

5 Brief about the architecture

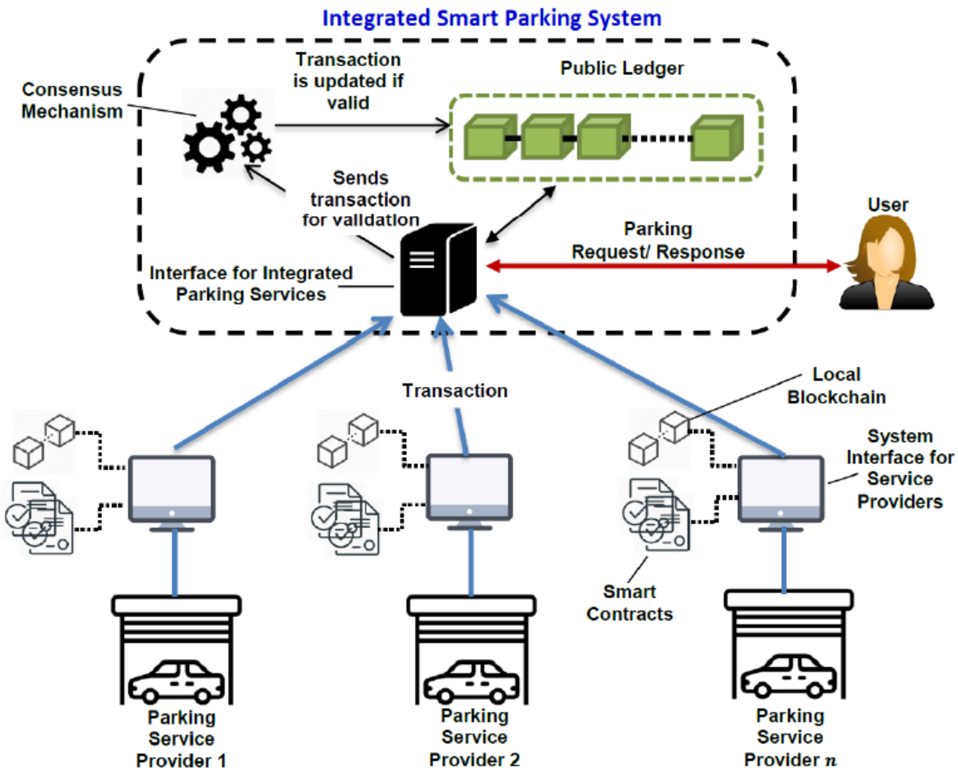

Figure 1: Overall Architecture of the system.

5.1 Stakeholders

The most important part that plays a major role in any blockchain network are its stakeholders. We will primarily consider three stakeholders in this regard, namely :-

1. Consumer

   They are the customers who are in search of a service, i.e., parking space in a parking lot. Whenever a client arrives, information about vacant and occupied positions is displayed by means of any software mesaure along with different parameters like price, size, and location. Moreover, they are the ones to invoke a transaction once they are satisfied with one of the given options.

2. Provider

   We can consider these components as a service provider. They are responsible for taking note of all the activities pertaining to user action. In correspondence to that, it shows the current availability or screenshot of the total parking system. Whenever an individual's parking status changes from occupied to vacant or vice-versa, it invokes a transaction, or we can request for it. Smart Contract are implemented by provider according to their requirement of pricing and terms and condition.

3. Blockchain network

   Distributed and decentralised system that contains nodes representing parking spaces. Each of them contains a public ledger. They are responsible for taking transactions from the pool proposed by poll and then checking for pre-defined conditions to fulfill. They can be in terms of authenticity, price, or availability.

5.2 Overview of the model

This section provides an overview of the integrated auto parking system built on blockchain that has been presented. Our suggested solution has three participants: a parking service provider, a blockchain network, and a user. The parking service provider updates parking space and offer (i.e., fee) information in the integrated system and provides parking as a service. A public ledger is present on the blockchain network, which only updates it with valid transactions. The transactions are verified through the use of a consensus process. The person who makes a parking request is known as the parking user. Each participant has a unique application interface for communicating with the integrated smart parking system. Our suggested blockchain-based integrated smart parking system is shown in Figure 1.

Assume that the city offers a variety of smart car parking options from several car parking service providers. Assume each smart parking lot is managed by a single service provider for the sake of simplicity. A blockchain-based integrated smart parking system is connected to every parking space. Typically, there is a local copy of the ledger in every parking lot (i.e local block). In the system, there can be two different kinds of transactions. first, the parking sensor's data output. Assume that each parking space in a smart parking lot is outfitted with an IoT device (such a parking sensor) that can generate transactional data on the availability of parking spaces. A smart contract is used by each provider of parking services to create the transaction. The relevant IoT device creates a transaction when a parking space changes from "empty" to "occupied." Similar to this, when the parking spot is changed from "occupied" to "empty," the IoT device generates a transaction. The local block is where the transaction is first sent. The transaction is sent for verification on the blockchain network by the local block. Secondly, information about parking costs. Assume that the cost of parking is determined by the amount of time. To set parking rates, they build smart contracts. The parking price smart contract is moved to the blockchain network. A transaction is produced each time a parking price is adjusted dynamically based on the time. The blockchain network receives the transaction for verification. Next, the blockchain network uses a consensus process to verify the transaction. The public ledger stores the transaction if it is legitimate. As a result, all local blocks are modified.

6 Layered Architecture of the model

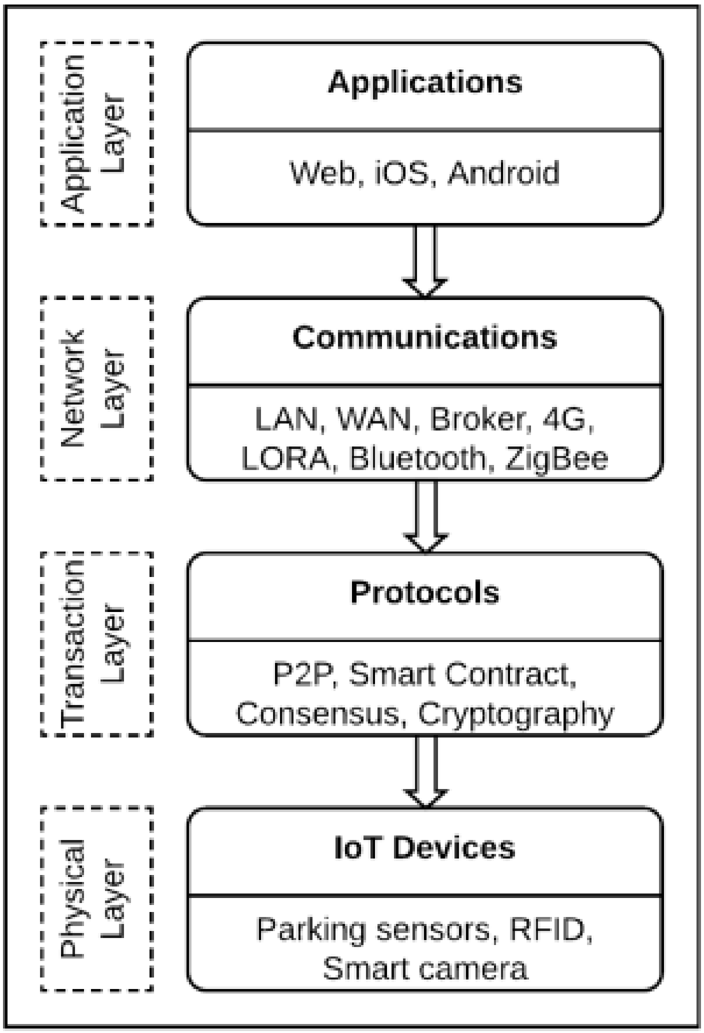

Figure 2: Layer architecture for Smart Parking System

We outline the layered architecture of our integrated smart parking system that is based on blockchain technology in this part. The integrated smart parking systems built on blockchain are typically characterised and standardised by the proposed layered architecture. Major system components are presented in the architecture as well. Application layer, network layer, transaction layer, and physical layer are the four levels that make up our architecture. Figure 2 provides a representation of the layered design.

6.1 Application layer

The top layer of the architecture stack, known as the application layer, is what allows users to interact with the system. Users can look up and reserve their preferred parking spots using an app for their smartphone (such as one for Android or iOS) or website. In a similar manner, parking service providers can feed the integrated system information about parking (such as offers and space availability). A user connects to the blockchain network from the application layer and uses an application to submit requests to the integrated parking system. Depending on the user's preferences and available spots, the integrated system is in charge of suggesting an appropriate parking area. This layer provides the finished service to the end users since customers directly engage with the integrated system.

6.2 Network Layer

The network layer makes sure that users, integrated systems, and various parking facilities can communicate with one another. This layer will be used to convey user and parking centre data to the integrated system. This layer will include LAN and WAN communication technologies that users, parking service providers, and IoT devices connected to the parking system will employ (e.g. parking sensors and security cameras). Distributed public ledger and content services are smoothly delivered to stakeholders' doorsteps as part of the standard offering thanks to the network layer. It combines existing GSM technologies like 4G and 5G with a variety of wireless communication technologies (including Lora, Bluetooth, Wi-Fi, and others). The scalability is also guaranteed by this layer. The network layer makes sure that users, integrated systems, and various parking facilities can communicate with one another. Information from As an illustration, this enables the dynamic addition and removal of stakeholders from the integrated system. The system's physical layer security is also ensured by the network layer.

6.3 Transaction Layer

The network's node-to-node transactions are handled by the transaction layer. Additionally, the complete blockchain network's consensus mechanisms will be provided through this. By utilising smart contracts and consensus mechanisms, users and various parking facilities would be able to securely share data. Through this layer, the parking centre will also update the general ledger. This layer interacts with the primary blockchain network via the integrated system's interface. This layer also verifies fresh transactions. Additionally, the transaction layer secures data transmission without the need for a reliable third party and maintains transaction transparency. We can eliminate system bottlenecks and central sources of failure thanks to P2P-based distributed design. In addition, users' data exchanges will be treated as transactions on the blockchain and verified via smart contracts. Users' information would continue to be distributed over time in this way using the cryptographic system of the blockchain.

6.4 Physical Layer

The physical layer is made up of several IoT device kinds. A p2p network protocol is used to connect each of these devices to a single shared network. The primary component of this layer are various kinds of sensors and actuators. Along with various WSN devices, there will also be some embedded technologies like the Raspberry Pi and Arduino. The transaction layer will be used to send data from IoT devices to the parking centre server. The public ledger will then be updated after the peer-to-peer network connects to the parking garage servers. Additionally, this layer allows for the accountability and traceability of sensor and actuator data over a peer-to-peer network. Since dependability is the most important component of our suggested system, data may be safely and securely transported from the IoT devices by utilising the blockchain, secure-immutable storage. A specific parking place availability will be identified from the physical layer using the IoT device sensors. The user will be confirmed through the use of encryption, and the public ledger will be updated with the latest information on available parking. The smart contract will process the transaction layer's cryptographic verification technique. The user can submit a request from the application layer to reserve a parking place, and the request will be handled by the network layer. Parking provider will connect with the transaction layer through the network layer to handle the user request. Finally, each parking provider will update the distributed ledger using the consensus mechanism protocol from the transaction layer.

7 Work-Flow of the Proposed model

We demonstrate how our suggested approach would contribute to an integrated smart parking system in this section.

Our system's users will be able to communicate with the integrated parking system through an application, which may be a website or a smartphone app. The programme will be used by users to find the parking spots they want. Since blockchain and peer-to-peer technology will connect all of the city's parking service providers, the application will automatically recommend the closest parking facility for the customer to choose. The user will next submit a request from his or her search to the parking facility server. The parking garage would reply with a thank you and establish a link. The suggested system architecture's search and request processing mechanism is described in depth in Algorithm 1 of the code. Algorithm 1 demonstrates that a request for a reservation to a specific parking service provider starts a local search for parking availability. A booking is made and the public ledger is updated if a parking space is available. If a local parking space is not available, a search of the public ledger is done to discover

Algorithm 1 Search and Request Processing

INPUT: Search for parking space, Select the desired parking center from the suggested list
OUTPUT: Reserve a parking space requested by the user, Update the public ledger

```
foreach reservation request do
    Look for available space in own server
    if space available in own server then
        Reserve for user
        Update public ledger
    else
        Search in the public ledger        ▷ Alternate parking
                                                  ▷ provider
        if Space available in public ledger then
            Reserve request with the hash
        else
            No parking Space available
        end if
    end if
end for
```

Figure 3: Algorithm

alternatives. The user is alerted via an unavailability notification if there are no substitutes in the public ledger. Our system will function as a data owner with the parking centres acting in that capacity. Each participating user and parking facility should start a blockchain while establishing a handshaking connection. The user will build two-way handshaking and then develop a smart contract with a digital signature. All transactions on the entire chain can be transparently audited because the parking centre will communicate with one another through the smart contract. The blockchain data on a node will be synchronised once the blockchain service has begun.

The hashing technique will encrypt the data after it has been validated, making it nearly

Algorithm 2 Parking Provider Operations in Public Ledger

INPUT: v : - Number of Vacant or Occupied spaces of a parking provider.
OUTPUT: $\mathcal{V}$: - a set $\{v_1, v_2, ...v_n\}$ of all available spaces in a smart city.

foreach Parking Provider P_i **do**
  $v_i \leftarrow Vacant\ Spaces$
  if *vehicle in* **then**
    $v_i \leftarrow increment$
  else if *vehicle out* **then**
    $v_i \leftarrow decrement$
  end if
  if *Authenticate with Smart Contract* **then**
    if P_i in $P2P$ Network **then**
      $\mathcal{V} \leftarrow \mathcal{V} \cup v_i$ ▷ Update Public Ledger
      ▷ Data Transaction
    end if
  end if
end for

Figure 4: Algorithm

hard for an unauthorised attacker or hacker to attack the network. The parking garage has a duplicate of the public ledger, which it may use to quickly decrypt the parking provider request because the public key is saved in the ledger. The provider will be counted as a legitimate network user if the protected hash and the decrypted hash match. By a legitimate service provider, the updated vacant spaces (vi) will be synced with the public ledger (Pi). The transaction will be stopped if this doesn't happen. The public ledger will then be updated using all of the parking centres' available slots. When a vehicle releases or reserves parking spots from the parking centre, the total number of available spaces (V = "v1, v2,...vn") will be increased or decreased.

Each parking centre will have a copy of the updated public ledger since every parking centre in the city will be using the peer-to-peer network and is aware of every p2p function verification mechanism. The ledger will be searched by a parking facility to find any open parking spaces, and it will then alert the user. The user will acknowledge the parking centre after receiving the notification. The parking provider will enter a transaction as per its own policies in the public ledger using that acknowledgement. As soon as the transaction is complete, it will be posted to the public ledger and modified in accordance with Algorithm 2.

If there isn't a free spot in the preferred parking facility, the parking facility will check the public ledger for a spot at a nearby parking facility. The parking centre will let the nearest parking centre know if a free spot is discovered in the public ledger. The parking centre will send a reservation request to that parking centre after receiving user confirmation. After that, the backup parking facility will react and record a transaction according to its own internal procedures. Because of this, the user will benefit from an integrated parking system. As a result, it is possible to connect various parking garages without requiring them to disclose their own business policies or to have faith in a third party.

8 Implementation of Smart Contract

We have implemented a fundamental smart contract for the given domain. We have kept an office or a commercial building as our goal. There are mainly two types of people requiring a parking slot. One is the owner (the regular visitor) the other is the guest. Similarly, there are two types of parking one is Special which is only available for the owner and the other one is available for the guest. There are some charges assigned by the owner for the parking system. There are functions which helps us to check whether the parking are vacant for the requirements by the user. There are mainly two stakeholders:

1. Person (Consumer)
    - isGuest : Tells information about whether the particular person is daily visitor or guest.
    - isAuthenticated : A very essential part to see whether he/she has a valid existential or are bind with some inappropriate activities which could further lead some problem to owner.
2. Parking (Supplier)
    - isParked : Occupied/Vacant
    - price : amount of parking slot
    - isSpecial : Already reserved for daily customer or for the owners or pre-booked.

Similary we have three functions namely:

1. addPerson : Called whenever a new person arrives at entry gate
2. addParking : Pre-defined at the time of construction.
3. getVacantPlace : According to current snapshot return index of vacant places
4. allocateParking : getVacantPlace check for authenticity and allocate parking if all the requirements fulfilled.

```
contract SmartParking{
    struct Parking{
        bool isParked;
        uint price;
        bool isSpecial;
    }
    struct Person{
        bool isGuest;
        bool isAuthenticated;
    }

    Parking[] public parking;
    Person[] public person;

    function addPerson(bool isGuest,bool isAuthenticated) public {
        person.push(Person(isGuest,isAuthenticated));
    }

    function addParking(bool isParked,uint price,bool isSpecial) public {
        parking.push(Parking(isParked,price,isSpecial));
    }
```

Figure 5: Implementation in solidity

```
    function getVacantPlace(bool isGuest) public view returns(uint){
            for(uint i=0;i<parking.length;i++){
                    if(!parking[i].isParked ){
                        if(isGuest && !parking[i].isSpecial){
                            return i;
                        }else if(!isGuest && parking[i].isSpecial){
                            return i;
                        }
                    }
            }
            return 1000;
    }

    function allocateParking(bool isGuest, bool isAuthenticated) public returns(bool){
        require(isAuthenticated);
        uint slotIndex=getVacantPlace(isGuest);
        if(slotIndex==1000){
            return false;
        }
        parking[slotIndex].isParked=true;
    }
}
```

Figure 6: Implementation in solidity

References

[1] S. Chen X. Xu B. Liu, X. L. Yu and L. Zhu. Blockchain based data integrity service framework for iot data. *Conference on. IEEE*, 2017.

[2] A. Khanna and R. Anand. Iot based smart parking system,. *Conference on. IEEE*, 2016.

[3] K. Nambut R. Lookmuang and S. Usanavasin. Smart parking using iot technology. *Conference on. IEEE*, 2018.

[4] A. Zajam and S. Dholay. Detecting efficient parking space using smart parking. *Conference on. IEEE*, 2018.