EasyChair Preprint

№ 395

CarParker: A Blockchain-Based PrivacyPreserving and Accident-Proof-Preserving Private Parking Space Sharing System

Youshui Lu, Pengrui Yao, Xinpei Dong and Yong Qi

August 3, 2018

CarParker: A Blockchain-Based Privacy-Preserving and Accident-Proof-Preserving Private Parking Space Sharing System

Youshui Lu, PengRui Yao, XinPei Dong, Yong Qi, *Member, IEEE*

***Abstract*—As sharing economy boomed recently, many applications or systems in this area have entered our lives. In this paper, we proposed CarParker, a novel blockchain-based parking space sharing system, by using which users who own vacated parking space can publish and share, while users who are looking for a parking lot could reserve online via our system. Incentives will be provided to the sharing parties, and the parking space user will be charged based on the rates and parking time. Our proposed service could relief the parking pressure in the downtown to some extent. However, this kind of service does also bring in some security and privacy issues. Since CarParker is blockchain-based and provides data provenance, each complete order and transaction between the user and the sharer along with images proof taken after parked and taken before leaving will be recorded in a tamper-proof manner. In the meanwhile, user ID verification mechanism results in various risks to user's privacy as malicious activities on these records cause severe damage to the users' reputation, finances, and so on. By implementing access control mechanism to those personal data, no third party including the data storage provider could access or utilise those data.**

***Index Terms*—parking space sharing, sharing economy, blockchain, Intel SGX, Hyperledger Fabric, smart contracts, privacy-preserving, proof-preserving,**

I. Introduction

With the development of the economy, the improvement of people's quality of life, and the popularisation of private cars, people's demand for parking spaces is increasing. The supply of parking spaces cannot keep pace with the rapid increase in car ownership. The parking spaces in commercial areas and residential areas have different demand periods. Generally speaking, in the daytime on weekdays, the parking spaces in the central business district are lacking, while the parking spaces in the surrounding residential areas are relatively abundant. However, at night or on weekends, it has an inverse situation. Inspired by this phenomenon, the proposal and implementation of the staggered rush hour parking plan are gradually accepted by the public and governments.

Parking space sharing is becoming a popular parking alternative in cities across the world. Such location-based services respond and satisfy users' real-time parking requests by recommending a list of designated car parks based on the users' preferences and providing a booking function to lock a parking lot in advance. They will become an effective approach to improve city traffic conditions, relief city parking pressure and decrease parking space vacancies thus reducing energy consumptions while preserving the benefits of individuals.

At present, by utilising advanced technology, the intelligent parking management system has achieved the management automation of the parking lot, which dramatically improves the reliability of the management system while enhancing the security of the vehicle and the operational effectiveness. The automated management system does also reduce human involvement, which saves many labour costs and also minimises the errors and mistakes caused by the workforce. Therefore, the economic efficiency and the usage rate of the parking space are correspondingly improved. However, the technologies used in most of those intelligent parking management systems still have much room to improve at the moment.

Parking spaces sharing is playing a vital role in optimising urban traffic during peak hours. With sharing economy boomed in recent years, many shared parking spaces concepts have been proposed. As a result, couples of sharing platform have incorporated. Although their service or product has brought convenience and intelligence to the users to some extent, they still have not addressed the following two difficulties during implementation.

Security concern raised by nearby residents

In daytime on weekdays, the parking spaces to share are from the nearby residential area, and it does bring some security concerns as many outsiders will have the opportunities to enter the private spaces. Also, it brings in trust crisis between a parker and a sharer.

High management costs and risks

Parking space sharing service increases the traffic volume in the car park, and there is no doubt that the probability of car accident happened in the carpark will increase. So it would be challenging to find the parties involved if there were an accident. Besides it also increases the management costs and risks as the property management team would increase more staffs and security to deal with the increased traffic volume.

To this end, we are motivated to propose a new parking space sharing system, named CarParker, which provides a parking

This work is part of the "CarParker" project participated in "The 4th China College Students' "Internet+" Innovation and Entrepreneurship Competition"

Youshui Lu is with the Electronics and Communications Department, Xi'an Jiaotong University (e-mail: yolu6176@uni.sydney.edu.au).

space sharing service from users. Specifically, given a parking space request, CarParker searches the optimal lists of nearby carparks with compatible personal preferences and saved destination, recommends a list of car parks which have a vacant parking lot only, provide navigation and online reservation functions.

CarParker significantly differs from existing parking space sharing systems by providing new sharing services with two major unique features.

(1) CarParker allows parking space sharing by individual users. Individual sharers could publish their available parking lots by setting starting and ending time while the parkers could search and reserve parking spaces via our application in real-time.

(2) Blockchain-based accident-proof preserving mechanism allows the user to retrieve the proofs of before and after. If there is a scratch or crash on their car during the parking period, the proofs will help clarify the responsible party for the accident.

To the best of our knowledge, this could be the first work to provide real-time parking space sharing service between personal users with blockchain-based accident-proof preserving mechanism.

In general, the main contributions of this work can be summarised as follows.

(1) We propose a novel real-time parking space sharing system named CarParker for more than cooperate users but personal users, by using which sharers who own vacated parking space can publish and share, while parkers who are looking for a parking lot could reserve online via our system upon their preferences.

(2) A blockchain-based accident-proof preserving mechanism is proposed to help clarify the responsible party if there is an accident in carpark during the parking time. Also by implementing the smart contracts in our work, it would build trust between untrusted parties.

(3) A users' privacy information protection mechanism is developed to allow the personal identification information(ID) such as national ID number only can be accessible by the user himself. Also, any other parties including the service provider do not have such authority.

The rest of this paper is organised as follows. Section II reviews the related work. Section III elaborates the preliminaries which mainly introduces the blockchain technology and enclaves and Intel SGX. Section IV presents the system of CarParker in several aspects. The discussions and challenges are presented in Section V. Finally, we conclude this paper in Section VI.

II. Related Work

Some companies and communities have addressed the issues of parking spaces recommendation and reservation. For instance, Parkopedia[1], which is the world's leading parking service provider so far, used by millions of drivers and organisations. It offers parking recommendation and reservation services online. Although it connects many car parks, it is only limited to the public car parks but not offers sharing services between personal users.

Besides, recently, the proliferation of mobile devices in conjunction with the advancement in their communication, computing, storage and multi-modal sensing capabilities has given rise to a new sensing paradigm, the so-called Mobile Crowd Sensing (MCS)[2]. Several smart parking applications exploiting the MCS paradigm[9],[21]-[24], have recently been proposed. Most of the solutions are based on a combination of sensor deployment and information collected from users who post parking availability information. In [3], the authors introduce an application which is built upon a map view that shows the availability status of nearby car parks, which by default follows the user's location. The application collects data with explicit user participation or monitors conspicuous facts that indicate the nearby car parks availability. In [4], the authors present an application which prompts users to post information about the surrounding parking availability. They grouped the parking area into lot zones to track user‘s movement. On the one hand, it uses the accelerometer sensor to obtain the motion of the user, on the other hand, it uses the GPS to locate so that the system can determine in which zone the user is and whether the user is entering or leaving the parking area. To reduce the bias caused by crowdsourcing data manipulation, expert data is also collected by an authority.

Furthermore, in [5] the authors have developed an application which collects parking availability information from drivers, the information includes their destination, their current location, real-time driving speed and the surrounding parking spaces availability when they are passing by. When a driver arriving at his destination, the application will search for the potential parking vacancies and list the results on a parking availability map. The devices can also collect geo-tagged sensor data automatically without drivers' intervention when the car is moving. Finally, the application will reward the participants based on their contribution level. In [6] the authors propose an application that detects available parking spots in cities using smartphones combined with ultrasonic sensing devices installed on vehicles.

Therefore, this paper proposes the sharing between personal users to be another user preference used finding an appropriate vacant slot, additionally. This seems to be the third significant component, which can be identified and easy to remember because everyone always knows where he or she wants to go or visit.

III. Preliminaries

A. **Blockchain Network**

The blockchain is a distributed database which contains an ordered list of records linked together through chains, on blocks. Blocks can be defined as individual components that contain information relating to a particular transaction. An example of such information can be a log on a single event (requestor needing data from the system). A blockchain network maintains a continuous growing list of records which are immutable. Due to this reason, many systems built on the blockchain technology achieve a secured distribution of assets among untrusted clients[7].

A Distributed Ledger

At the heart of a blockchain network is a distributed ledger which records all the transactions that take place on the network.

A blockchain ledger is often described as decentralised because it is replicated across many network participants, each of whom collaborates in its maintenance. We will see that decentralisation and collaboration are potent attributes that mirror the way businesses exchange goods and services in the real world[13].

In addition to being decentralised and collaborative, the information recorded to a blockchain is append-only, using cryptographic techniques that guarantee that once a transaction has been added to the ledger, it cannot be modified. This property of "immutability" makes it simple to determine the provenance of information because participants can be sure information has not been modified after the fact.

Smart Contracts

To support the consistent update of information — and to enable a whole host of ledger functions (transacting, querying, etc.) — a blockchain network uses smart contracts to provide controlled access to the ledger.

Smart contracts are not only a core mechanism for encapsulating information and keeping it simple across the network, but they can also be written to allow participants to execute particular aspects of transactions automatically.

A smart contract can, for instance, be written to stipulate the cost of shipping an item where the shipping charge changes depending on how quickly the item arrives. With the terms agreed to by both parties and written to the ledger, the appropriate funds change hands automatically when the item is received.[13]

Consensus

The process of keeping the ledger transactions synchronized across the network — to ensure that ledgers update only when transactions are approved by the appropriate participants and that when ledgers do update, they update with the same transactions in the same order — is called consensus[13].

B. Enclaves and Intel SGX

A hardware enclave provides developers with the abstraction of a secure portion of the processor that can verifiably run a trusted code base (TCB) and protect its limited memory from a malicious or compromised OS [14]. The hardware handles the process of entering and exiting an enclave and hiding the activity of the enclave while non-enclave code runs. Enclave code invariably requires access to OS resources, so developers specify an interface between the enclave and the OS[28]. In SGX, the platform we use in our implementation, the functions made available by this interface are called OCALLs and ECALLs. OCALLs are made from inside the enclave to the OS, usually for procedures requiring resources managed by the OS, such as file access. ECALLs allow code outside the TCB to call the enclave to execute trusted code. An enclave proves that it runs an untampered version of the desired code through a mechanism named attestation. Attestation involves an enclave providing a hash of its initial state which a client compares with the expected value of the hash and rejects if there is any evidence of a corrupted or altered program[28]. The most significant feature of enclaves for our purposes concerns the protection of memory. An enclave gives developers a small Enclave Page Cache (EPC), a memory region hidden from the OS and cleared whenever execution enters or exits an enclave. In this memory, the trusted code can keep secrets from a malicious OS who otherwise controls the machine. SGX provides approximately 128MB of EPC. Beyond using EPC, the code in the enclave has to call the OS to access other memory pages.

IV. CarParker System

In this section, CarParker, our proposed parking space sharing system is presented in a detailed manner. CarParker is an "Internet+social service" system based on Client-Server(C/S) and Browser-Server(B/S) architect, but it also applies blockchain and access control technology. It is committed to fundamentally solving the needs of users who are seeking parking spaces and so on. It designed to provide parking space demanders with real-time, convenient and concessional parking spaces. Meanwhile, the architect of the system has ensured a certain level of security, reliability and privacy.

A. System Design

CarParker follows the general architectural design of a combination of C/S and B/S system, comprising the following main entities: a) the parker, b) the sharer, c)the car park management system, and 4) the platform. Thus, in CarParker, the actors involved in the life cycle of a sharing are:

Client-Parker(CL-P): parkers are those who are looking for a parking lot. They could search by putting a destination or choosing nearby available parking slots which are posted by the sharer. Also, the parker could reserve a slot online via our system upon their preferences.

Client-Sharer (CL-S): sharers are who own vacated parking spaces. They could post and share their parking lot and choose the available sharing time slot. All the information submitted by the sharer will be uploaded to the server.

Client-Carpark Management System (CL-CMS): it is a carpark management system that provides services for both nonCarParker user and Carpark user. The system includes vehicle plate recognition module, timekeeping module, customer information system module and payment module. The system will obtain orders information which comprises the parker's vehicle plate number and the available parking period of particular parking space posted by the sharer from the server.

Server-Platform(SE-P): the platform which is the server is the primary communication link between CL-P, CL-S and CL-CMS. It mainly processes the requests including registration, reservation, parking spaces information, etc. SGX will encrypt the information submitted by the user before being stored in databases. The user himself and SGX can only access this kind of information. Meanwhile, the server allows the parking spaces information collected from CL-CMS syncronized with CL-S and Cl-P.

At this point, it should be noted that an individual user could be a sharer and parker at the same time. Thus, the system allows

an individual user to post sharing information and to reserve shared parking information simultaneously.

A parking lot sharing process's sequence diagram

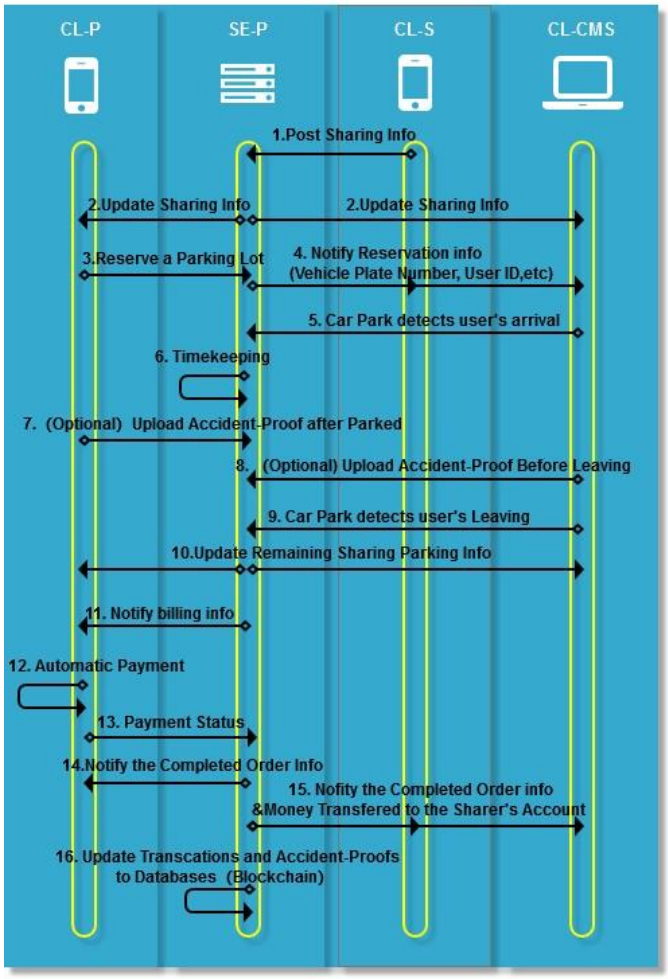

Figure 1. Sequence diagram of a parking space sharing order.

The sequence diagram indicates a successful sharing transaction between a sharer (CL-S) and a parker (CL-P). 1. The process first started by a sharer (CL-S) posting his available parking space information via mobile application. 2. After the platform(SE-P) received the sharing information from the sharer, it updates the sharing information to the carpark end and the parker. 3. Once a parker reserved a parking lot from the mobile application, the message will updated to the platform. 4. And then the reservation information will be synchronised to the sharer's end and the carpark end. 5. While the parker arrived the reserved carpark, the carpark end system will notify the platform. 6. And the platform will do timekeeping once receiving the message saying that the parker has entered the carpark. 7. When parked, the sharer may choose to use the accident-proof preserving function by taking photos of his parked car from predetermined directions set by the application. 8. Also before the sharing left the carpark, he may also have the option to preserve the accident-proof, however, the prerequisite is that he has already chosen to take photos in Step 7. And then 9. The carpark system will detect the car's left and update the information to the platform. 10. Once the car has left the carpark, the platform will update the remaining sharing parking lot information in order to serve new orders. Meanwhile, 11. The platform will send the bill to the parker. Usually, 12. The preset express payment method will automatically do the payment once it received the billing information. And then, 13. It will update the payment status to the platform. 14. Once the platform received the paid message, it would send the completed order information to client end including the parker, 15. the sharer and the carpark system. Also, the money will be automatically transferred to the sharer's account. 16. For the parker who has used the accident-proof preserving function in this completed transaction, the transaction and the accident-proof will be updated to the databases, and then be updated on the blockchain once validated.

The detailed steps will be discussed in the later part of this paper.

System Architect

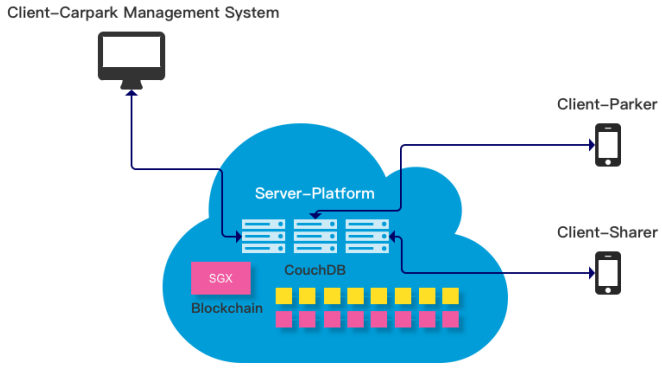

Figure 2. System Architect

The architect of the system includes the client end which comprises mobile applications for CL-P and CL-S and web browser for the carpark, as well as the server end which mainly refers to the platform. The detailed of each component will be discussed in the following paragraphs.

B. Mobile Application(CL-P)

The mobile application, as the critical interface for the parker, has functions including user registration/login, search parking spaces, recommend and list parking space information, reservation, navigation, online payment, proof-preserving and so on. Only featured functions will be discussed.

Account Information

To provide a sharing service with high accountability and responsibility, user's identification is required to be verified by the authorised party. As a mobile application user, to get verified, he needs to upload his national ID card images of both sides, as well as a photo that shows the user holding the ID. Moreover, it is essential to keep the user's information with minimal risk to data privacy during the verification process. To achieve a system with high confidentiality, SGX will establish a Private Key(PK) for each user ID by a specific key establishment algorithm when registered. The further discussion on Intel SGX will be mentioned in the later part of this paper.

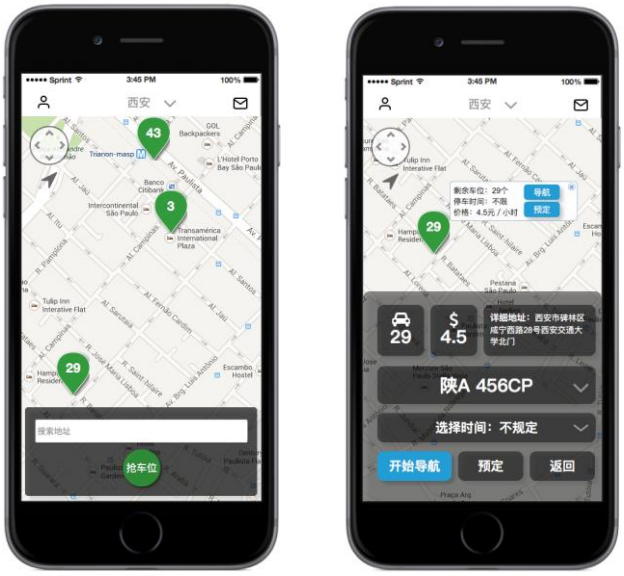

Figure 3. Mobile Application (CL-P) Prototype, the left image show the default interface once a user login, the right image show the reservation form.

Search and Reservation

The application will by default list the nearby available parking spaces based on your location. The user may also set a destination by typing in the search box, and the recommended parking spaces information will be listed on a ballon on a base map once submitting. The listed parking space information includes the location, hourly rates, available time slot, etc.

The application does also allow the user to make a reservation for a particular parking space by providing the vehicle plate number and the expected parking time which should not exceed the parking space's available time posted by the sharer. Once reserved, the reserved parking space information will be updated automatically.

Navigation module

Navigation module is built on Baidu maps API. The user could use this feature to navigate to the parking space which he has chosen.

Proof-Preserving

The proof-preserving function, which is built based on blockchain technology, enables the user to retrieve the proofs of before and after. If there were a scratch or crash on their car during the parking period, the proofs would help clarify the responsible party for the accident. This feature is only available for the user who has upload photos at before and after. Photo submission is optional for a registered user. And the photos to be uploaded must be taken via a camera in real-time, they could not be uploaded by choosing from local photo libraries. By doing so, the system could guarantee that the photo is not from historical data and the photo with the exact timestamp is precise. When a user has his car parked in the chosen parking lot, he has an option to upload the photos of the parked car from at least four viewpoints. Moreover, before he is leaving, he also has an option to upload photos. All the image data with timestamps would automatically be stored on blocks in a tamper-proof manner.

Figure 4. Mobile Application (CL-P) Prototype, the image shows the proof-preserving feature which is to ask the user to upload proofs by taking pictures.

C. Mobile Application(CL-S)

The CL-P and CL-S users could login into the same mobile application, but they can choose to be a sharer or a parker at each time when login.

The CL-S user could post and share their registered parking spaces at the platform.

Carpark space verification

To ensure the sharing parking space information provided by the sharer is real. A relevant parking space ownership document should be uploaded to the platform before verified. Besides, the user is also asked to provide the detailed address of his parking spaces.

Post and Share

Once the parking space verified, the sharer could be able to post and share his parking space at the platform. When posting the sharing information, the sharer is asked to set the available time slot before submitting. Once submitted, the information will automatically update to the server.

Figure 5. Mobile Application (CL-S) Prototype, the image shows the post and share page when a user logs in as a sharer.

D. Web Application(CL-CMS)

This web application is mainly serving for the carpark management user. Its features include but not limited to the following ones.

Car Management Module

This module is generally to visualise the daily operations of the car park. It does also comprise the daily transactions and car-in and car-out.

History Report

Thit feature is to help the manager to have a better understanding of how good the carpark operates based on the history statistics report exported from the application.

Figure. 6. Web Application (CL-CMS) Prototype, the image shows the car management system on the carpark end.

Vehicle Plate Recognition Module

This module allows the application to automatically open and close the gate by recognising the vehicle plate number without manual processes. It is very time-saving during peak hours. Also, when the system updated a reservation, with this built-in module, the system will automatically know whom the CarParker registered users are, and keep records of what time they entered and exited the carpark.

Membership Management

The membership management is generally managing the permanent users' information. The permanent users refer to those who paid the fees monthly, quarterly or annually.

E. **Bankend-CloudServer(SL-P)**

To have a better performance and higher confidentially of the system, we designed a powerful backend cloud server based on blockchain technology, Intel SGX extension, etc.

Data Storage

We choose CouchDB as our distributed databases. The reasons are as follows, 1) our backend server architect is a type of distributed storage, 2) it is a database that completely embraces the web,3) it can also enhance the security for compliance and data protection in the blockchain.

When the server gets the data from the clients, the data will be formatted and stored with JSON documents. Meanwhile, the clients could access documents via HTTP. CouchDB also supports full data rich queries capability, such as non-key queries against the whole blockchain data.

CouchDB works well with modern web and mobile apps. CarParker can distribute the data, efficiently using CouchDB's incremental replication. CouchDB supports master-master setups with automatic conflict detection. It comes with a suite of features, such as on-the-fly document transformation and real-time change notifications, that make web development much more comfortable. It even comes with an easy to use web administration console, served directly out of CouchDB[27]. CarParker pays more attention to distributed scaling. CouchDB is highly available and partition tolerant and is also eventually consistent. Moreover, CarParker cares a lot about user's data. CouchDB has a fault-tolerant storage engine that puts the safety of the users' data first[17].

CouchDB can also enhance the security for compliance and data protection in the blockchain. Because it can implement field-level security through the filtering and masking of individual attributes within a transaction, and only authorising the read-only permission if needed[16].

Blockchain Application

We implement Hyperledger Fabric which is a permission platform enables confidentiality through its channel architecture for our application. The Fabric network participants can establish a "channel" between the subset of participants that should be granted visibility to a particular set of transactions. Also, only those nodes that participate in a channel have access to the data transacted and the smart contract, preserving both the privacy and confidentiality[13].

In our work, we employ smart contracts to record 1) transactions completed by a sharing parking space order between CL-S and CL-P, 2)accident-proof uploaded by CL-P. These enable CL-P to completely gain assurance on data provenance since the entire lifeline of the uploaded proof would be kept in a reliable environment.

We choose to implement the blockchain technology in the accident-proof preserving module, for one reason, compared to the traditional methods, it could record the information in a tamper-proof manner to ensure the proofs' creditability. For instance, by using the traditional way of storing data, a malicious user could be able to tamper the data, thus making the system trustless and unreliable.

As mentioned earlier, we design to store different data separately in a parent block and a side block on the chain because when a user requires accessing the records on the block, it would be much straight-forward and computing cost-saving. For a parent block, it stores the transactions which comprise two parts, the timestamps and the data, respectively. The timestamps are made up of time entered the carpark, time the order started, time the order completed, time left the carpark, time the payment occurred. While the data includes user ID, vehicle plate number, carpark ID, parking lot ID, order ID and fees. In addition, a side block stores the information such as the photo taken time, photo files, user ID and order ID.

Transactions and proofs are processed, indexed and broadcast into a blockchain network. In our work, they are reported and stored in a smart contract permissioned database and indexed based on the ***UserID*** of each client user and ***OrderID*** of each shared parking order. We initialise sets of actions that can apply to any form of data stored to the blocks and retrieved from the blocks. The primary action sets are: ***upload***, ***retrieve***, ***paid***. These sets of actions, when performed on the data, would trigger the smart contracts to send a report based on the rules established for that particular data. Trigger actions are conveyed in smart contract scripts by a function

getAction.

The data is categorized into two types which are; plaintext and encrypted data. The plaintext comprises ***UserID***, ***OrderID***. While the encrypted data includes ***OrderStartTime***, ***OrderCompleteTime***, ***CarEnterTime***, ***CarLeftTime***, ***VehiclePlateNum***, ***CarparkID***, ***ParkingLotNum***, ***Fees***, ***PhotoTakenTime*** and ***Photo***.

Algorithm1: Smart Contracts

Require: Initialization of parameters:
getOrderStartTime, getOrderCompleteTime, getCarEnterTime,getCarLeftTime,getUserID, getVehiclePlateNum, getCarparkID, getParkingLotNum, getOrderID, getFees, getPhotoTakenTime, getPhoto, getSharedKey, getAction,storeTx, storePhoto, retrieveTx, retrievePhoto, report;

LOGIC:
if
func (getAction)==paid
then
func (getOrderStartTime, getOrderCompleteTime, getCarEnterTime, getCarLeftTime, getUserID, getVehiclePlateNum, getCarparkID, getParkingLotNum, getOrderID, getFees)←
OrderStartTime, OrderCompleteTime, CarEnterTime, CarLeftTime,UserID, VehiclePlateNum, CarparkID, ParkingLotNum, OrderID, Fees
func(getSharedKey)
EnTxInfo=encrypt (OrderStartTime, OrderCompleteTime, CarEnterTime, CarLeftTime, VehiclePlateNum, CarparkID, ParkingLotNum, Fees)
report(UserID, OrderID, EnTxInfo)
storeTx(UserID, OrderID, EnTxInfo)
end if
if
func (getAction)==upload
then
func (getUserID, getOrderID, getPhotoTakenTime, getPhoto)←
UserID, OrderID, PhotoTakenTime, Photo
func(getSharedKey)
EnPhotoInfo=encrypt (PhotoTakenTime, Photo)
report(UserID, OrderID, EnPhotoInfo)
storePhoto(UserID, OrderID, EnPhotoInfo)
end if
if
func (getAction)==retrieve
then
func(getSharedKey)
EnTxInfo=retrieveTx(UserID, OrderID)
EnPhotoInfo=retrievePhoto(UserID, OrderID)
DecryptData=decrypt(EnTxInfo, EnPhotoInfo)
end if

The function ***getSharedKey*** is to extract an encryption key to encrypt data that would be reported to the smart contract database. An action on encrypting is called to finalise this process. The action of sending the data to processing and consensus nodes is instantiated by the reported statement in the smart contract script. ***storeTx*** function is to store transactions onto the parent blocks while the ***storePhoto*** function is for side blocks.

The blockchain technology used in CarParker system will be based on Practical Byzantine Fault Tolerance(PBFT) protocol. It is a classical distributed consensus in dealing with forking problems, and it is also very suitable for a permissioned blockchain network. In addition, the participant nodes in the network will be made of the trusted nodes either by the operating authority or other third party organisation with trust-proof, and it enables the reliability and security be guaranteed by the business-related parties. Moreover, the consensus delay is acceptable which could meet the requirements of real-time processing.

Privacy-Preserving manganism via Intel SGX

A users' privacy information protection mechanism is developed upon Intel SGX, which allows the personal privacy data only can be accessed by the user. We will discuss in details in the following paragraphs.

In our system, two parties, CarParker User(CU) and Blockchain Participation Node(BPN) need SGX to establish shared keys or certificates..

A node is able to participate in the blockchain network, via the means of a digital identity issued for it by an authority trusted by the system. In the most common case, digital identities (or simply identities) have the form of cryptographically validated digital certificates that comply with X.509 standard and are issued by a Certificate Authority (CA)[18].

When a CU first registers, it must obtain its own shared keys to ensure the information transmission is at a relatively secured environment. In our system, we use Intel SGX to generate shared keys for the CU.

When a BPN first registers, it must obtain its own shared keys and certificates for identification verification for future transactions.

And the certificates need to be issued by a trusted CA. Because that the trusted hardware Intel SGX has the characteristics that it cannot be tampered and monitored, we use Intel SGX to act as a trusted CA to issue certificates for the BPN.

The procedure of keys and certificates generation are as follows,

1). The server obtains a pair of public and private keys through Intel SGX, it uses the public key to apply for a certificate to a trusted Root CA, and generates an Endorsement Certificate(EC).

2). When a BPN registers to participate, the EC is sent to the client for authentication when the SSL connection is establishing, the process is to ensure the authenticity of the server.

3). The BPN generates a pair of public and private keys based on the ECC algorithm and sends the public key gA to the server.

4). After receiving the BPN's public key, the server signs the

BPN's public key by using its own private key in order to generate the Node Certificate(NC). Meanwhile, the server generates a pair of public and private keys based on the ECC algorithm in the first space and then calculates the shared key K using the BPN's public key. Finally, the private key is used to sign gA ,gB and NC, and send it to the BPN together with gB and NC.

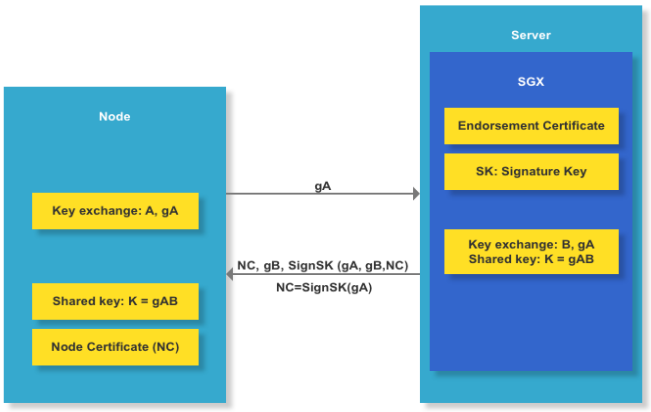

Figure. 7. Keys and certificates generation procedure between node and server.

5). After receiving the signature from the server, the BPN verifies the data using the public key of the server, and the shared key K is calculated by using the server gB once the signature is verified.

As for the CU, it is only required the shared key. So the procedure of the keys generation is much more straightforward. Compared to the BPN's procedure, it does not require the steps for generating NCs."

If a user does not want to be an actor of our system any more, he needs to request to cancel his account via the system. Once verified, the system will destroy his shared keys or certificates automatically.

The system takes advantage of Intel SGX being a digital identity trusted issuer. Moreover, this access control mechanism which encrypts the privacy information before stored in the server will guarantee the confidentiality of the users' data in the storage or even have been obtained by the third parties.

V. Discussion and Challenges

Although the intention of our shared parking system is between personal users, we still need to cooperate with the carpark management companies who are in charge of the carparks in residential areas.

This is because that when a CarParker user reserved a parking lot in a particular residential area, he needs to have access to that private carpark gate before entering. Only with the carpark management system is integrated into our system, a CarParker registered user could automatically be recognised and get access to the reserved carpark.

Besides, some people may think that our application would bring more security issues to the residents thus affecting the neighbours using our product.

Furthermore, regarding the performance of the blockchain network, a relatively high concurrency is needed for our work. Although Hyperledger Fabric, which we choose for our blockchain network, has a concurrency control where transactions are executed in parallel by the endorsers to increase throughput, it does also have challenges when setting the Block size scaling as well as the Endorser scaling.

VI. Conclusion

Due to the parking pressure issue in the downtown, many companies have proposals about the platform of sharing nearby parking spaces online. However, those parking spaces are from nearby public carparks. Those platforms could only increase the utilisation rate of those public carparks, and it would not increase the total number of the available parking spaces.

In this paper, we proposed CarParker, a novel blockchain-based parking space sharing system, by using which users who own vacated parking space can post and share, while users who are seeking for a parking lot could reserve online via our application. Our proposed service could mitigate the parking pressure in the city during peak hours to some extent, and it does increase the total number of the sharing parking spaces. Meanwhile, CarParker implements blockchain technology and Intel SGX to address the security and privacy issues brought by the sharing services.

For our future work, we will integrate the public carparks as well as the on-road parking spaces into our system. We believe that only by adding all the parking spaces from different sources, the system would become more convenient and powerful for the user. In addition, we will explore more on the blockchain consensus protocols as well as to experiment more on improving the performance of the permissioned blockchain network.

References

[1] *Parkopedia*, [online] Available: https://en.parkopedia.com.
[2] R. Ganti, F. Ye, H. Lei, "Mobile crowdsensing: current state and future challenges", IEEE Communications Magazine, vol. 49, no. 11, pp. 32-39, 2011.
[3] J. Kopecký, J. Domingue, "ParkJamJAM: Crowdsourcing Parking Availability Information with Linked Data" in Extented Semantic Web Conference, Berlin, Heidelberg: Springer, pp. 381-386, 2012.
[4] J. Villalobos, B. Kifle, D. Riley, JUQ Torrero, "Crowdsourcing Automobile Parking Availability Sensing Using Mobile Phones", UWM Undergraduate Research Symposium, pp. 1-7, 2015.
[5] X. Chen, N. Liu, "Smart Parking by Mobile Crowdsensing", Internation Journal of Smart Home, vol. 10, no. 2, pp. 219-234, 2016.
[6] Suhas Mathur et al., "Parknet: drive-by sensing or road-side parking statistics", Proc. of the 8th International Conference on Mobile Systems Applications and Services, pp. 123-136, 2010.
[7] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, M. Guizani, "MeDShare: Trust-less medical data sharing among cloud service providers via blockchain", *IEEE Access*, vol. 5, pp. 14757-14767, 2017.
[8] Y. Lyu, V. C. S. Lee, C. Y. Chow, J. K. Y. Ng, Y. Li and J. Zeng, "R-Sharing: Rendezvous for Personalized Taxi Sharing," in IEEE Access, vol. 6, pp. 5023-5036, 2018.
[9] K. Banti, M. Louta and G. Karetsos, "ParkCar: A smart roadside parking application exploiting the mobile

crowdsensing paradigm," *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*, Larnaca, 2017, pp. 1-6.
[10] M. Turkanović, M. Hölbl, K. Košič, M. Heričko and A. Kamišalić, "EduCTX: A Blockchain-Based Higher Education Credit Platform," in *IEEE Access*, vol. 6, pp. 5112-5127, 2018.
[11] J. Gu, B. Sun, X. Du, J. Wang, Y. Zhuang and Z. Wang, "Consortium Blockchain-Based Malware Detection in Mobile Devices," in IEEE Access, vol. 6, pp. 12118-12128, 2018.
[12] Brandenburger, Marcus; Cachin, Christian; Kapitza, Rüdiger; Sorniotti, Alessandro, "Blockchain and Trusted Computing: Problems, Pitfalls, and a Solution for Hyperledger Fabric," in arXiv, arXiv:1805.08541,2018
[13] Introduction — Hyperledger-fabricdocs Master Documentation. (n.d.). Retrieved from http://hyperledger-fabric.readthedocs.io/en/release-1.1/blockchain.html
[14] COSTAN, V., AND DEVADAS, S. Intel SGX explained. IACR Cryptology ePrint Archive 2016 (2016), 86.
[15] F. Villanueva, D. Villa, M. Santofimia, J. Barba, JC Lopez, "Crowdsensing smart city parking monitoring", *2nd IEE World Forum on Internet of Things (WF-IoT)*, pp. 751-756, 2015.
[16] Building Your First Network — Hyperledger-fabricdocs ... (n.d.). Retrieved from http://hyperledger-fabric.readthedocs.io/en/release-1.1/build_network.html
[17] A Database for a Web, retrieved from https://svn.apache.org/repos/asf/couchdb/site/index.html
[18] Identity — Hyperledger-fabricdocs Master Documentation. (n.d.). Retrieved from http://hyperledger-fabric.readthedocs.io/en/release-1.1/identity/identity.html
[19] R. Giffinger, "Smart City Concepts: Chances and Risks of Energy Efficient Urban Development" in International Conference on Smart Cities and Green ICT Systems, Portugal:Springer, pp. 3-16, 2015.
[20] S. Nawaz, C. Efstratiou, C. Mascolo, "ParkSense: A smartphone based sensing system for on-street parking", *Proc. of the 19th Annual International Conference on Mobile Computing & Networking*, pp. 75-86, 2013
[21] V. Coric, M. Gruteser, "Crowdsensing Maps of On-street Parking Spaces", *Proc. of the 9th Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 115-122, 2013.
[22] X. Chen, N. Liu, "Smart Parking by Mobile Crowdsensing", *Internation Journal of Smart Home*, vol. 10, no. 2, pp. 219-234, 2016.
[23] R. Salpietro, L. Bedogni, M. Di Felice, L. Bononi, "Park Here! a smart parking system based on smartphones' embedded sensors and short range Communication Technologies", *Proc. of the 2015 IEEE 2nd World Forum on Internet of Things*, pp. 18-23, 2015.
[24] J. Kopecký, J. Domingue, "ParkJamJAM: Crowdsourcing Parking Availability Information with Linked Data" in Extented Semantic Web Conference, Berlin, Heidelberg:Springer, pp. 381-386, 2012.
[25] A. N. Khan, M. L. M. Kiah, M. Ali, S. A. Madani, A. U. R. Khan, S. Shamshirband, "BSS: Block-based sharing scheme for secure data storage services in mobile cloud environment", *J. Supercomput.*, vol. 70, no. 2, pp. 946-976, Nov. 2014.
[26] J. Huang, Q. Duan, S. Guo, Y. Yan, S. Yu, "Converged network-cloud service composition with end-to-end performance guarantee", *IEEE Trans. Cloud Comput.*.
[27] 1. Introduction — Apache Couchdb 2.1 Documentation. (n.d.). Retrieved from http://docs.couchdb.org/en/2.1.1/intro/index.html
[28] Eskandarian S, Zaharia M. "An Oblivious General-Purpose SQL Database for the Cloud" in arXiv, arXiv:1710.00458. 2017.
[29] Mattias Scherer, "Performance and Scalability of Blockchain Networks and Smart Contracts" Spring 2017

Youshui Lu received the B.Arts. Degree in digital arts from The Australian National University, Australia, in 2012, and the M.Sc. degree in urban and regional planning with the University of Sydney, Australia, in 2015, He is currently pursuing the PhD degree in computer science and technology in Xi'an Jiaotong University, China. His current research interests include blockchain technology, urban computing, traffic management systems, machine learning, and privacy and big data security.

Pengrui Yao, received the B.S. degree in Software Engineering from Taiyuan University of Technology, China, in 2016, and he is currently pursuing the M.S. degree in Software Engineering in Xi'an Jiaotong University, China.His current research interests include Information Security, Trusted Hardware and Cloud Computing.

Xinpei Dong received the B.BM degree in information management and information system from The Xi'an Shiyou University, Xi 'an, in 2012, He is currently pursuing the Master degree in computer science and technology in Xi'an Jiaotong University, China.His current research interests include blockchain technology, machine learning, and big data and cloud computing.

YONG QI received the PhD degree from Xi'an Jiaotong University, China. He is currently a Full-time Professor at Xi'an Jiaotong University. His research interests include operating systems, distributed systems, and cloud computing.